

**Numerical Algorithms for Precise and Efficient Orbit
Propagation and Positioning**

by

Ben K. Bradley

B.A.E.M., Aerospace Engineering and Mechanics, University of
Minnesota Twin Cities, 2009

M.S., Aerospace Engineering Sciences, University of Colorado Boulder,
2011

A thesis submitted to the
Faculty of the Graduate School of the
University of Colorado in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
Department of Aerospace Engineering Sciences
2015

This thesis entitled:
Numerical Algorithms for Precise and Efficient Orbit Propagation and Positioning
written by Ben K. Bradley
has been approved for the Department of Aerospace Engineering Sciences

Penina Axelrad

George H. Born

Date _____

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Bradley, Ben K. (Ph.D., Aerospace Engineering Sciences)

Numerical Algorithms for Precise and Efficient Orbit Propagation and Positioning

Thesis directed by Dr. Penina Axelrad

Motivated by the growing space catalog and the demands for precise orbit determination with shorter latency for science and reconnaissance missions, this research improves the computational performance of orbit propagation through more efficient and precise numerical integration and frame transformation implementations. Propagation of satellite orbits is required for astrodynamics applications including mission design, orbit determination in support of operations and payload data analysis, and conjunction assessment. Each of these applications has somewhat different requirements in terms of accuracy, precision, latency, and computational load. This dissertation develops procedures to achieve various levels of accuracy while minimizing computational cost for diverse orbit determination applications. This is done by addressing two aspects of orbit determination: (1) numerical integration used for orbit propagation and (2) precise frame transformations necessary for force model evaluation and station coordinate rotations.

This dissertation describes a recently developed 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. It requires significantly fewer force function evaluations than explicit Runge-Kutta schemes and approaches the efficiency of the 8th-order Gauss-Jackson multistep method.

Converting between the Geocentric Celestial Reference System (GCRS) and International Terrestrial Reference System (ITRS) is necessary for many applications in astrodynamics, such as orbit propagation, orbit determination, and analyzing geoscience data from satellite missions. This dissertation provides simplifications to the Celestial Intermediate Origin (CIO) transformation scheme and Earth orientation parameter (EOP) storage for use in positioning and orbit propaga-

tion, yielding savings in computation time and memory. Orbit propagation and position transformation simulations are analyzed to generate a complete set of recommendations for performing the ITRS/GCRS transformation for a wide range of needs, encompassing real-time on-board satellite operations and precise post-processing applications. In addition, a complete derivation of the ITRS/GCRS frame transformation time-derivative is detailed for use in velocity transformations between the GCRS and ITRS and is applied to orbit propagation in the rotating ITRS.

EOP interpolation methods and ocean tide corrections are shown to impact the ITRS/GCRS transformation accuracy at the level of 5 cm and 20 cm on the surface of the Earth and at the Global Positioning System (GPS) altitude, respectively. The precession–nutaton and EOP simplifications yield maximum propagation errors of approximately 2 cm and 1 m after 15 minutes and 6 hours in low-Earth orbit (LEO), respectively, while reducing computation time and memory usage. Finally, for orbit propagation in the ITRS, a simplified scheme is demonstrated that yields propagation errors under 5 cm after 15 minutes in LEO. This approach is beneficial for orbit determination based on GPS measurements.

We conclude with a summary of recommendations on EOP usage and bias–precession–nutaton implementations for achieving a wide range of transformation and propagation accuracies at several altitudes. This comprehensive set of recommendations allows satellite operators, astrodynamicists, and scientists to make informed decisions when choosing the best implementation for their application, balancing accuracy and computational complexity.

Dedication

To my family, which has supported me from the beginning
and to my friends, who made this journey an unforgettable experience.

*“Science is much more than a body of knowledge. It is a way of thinking.
This is central to its success. Science invites us to let the facts in, even when
they don’t conform to our preconceptions. It counsels us to carry alternative
hypotheses in our heads and see which ones best match the facts.”*

— Carl Sagan

“Don’t Panic.”

— Douglas Adams

Acknowledgements

A special thanks goes to my advisor, Dr. Penny Axelrad, for her unwavering support and encouragement throughout my graduate studies. She is one of the kindest and most intelligent people I have ever known. This dissertation would not have been possible without her guidance through my ever-changing research interests. Thank you. I want to thank Dr. Brandon Jones for his wisdom and ever-evolving useful conversations over the years. I have learned much from you, and your patience in answering an unending stream of questions has not gone unnoticed. I am grateful for Dr. George Born being a second advisor to me for several years as I pursued research in space situational awareness and orbital debris. You have been a constant source of inspiration and laughter throughout the years of group meetings. I would also like to thank the other members of my committee, Dr. Webster Cash and Dr. Daniel Scheeres.

This research was made possible with partial Government support under and awarded by DoD, Air Force Office of Scientific Research, National Defense Science and Engineering Graduate (NDSEG) Fellowship, 32 CFR 168a. I am grateful for the opportunity the NDSEG program has given me to explore and contribute to numerous areas of research.

Last, but certainly not least, I want to thank my friends in Colorado, many of whom have already completed their doctorates and begun new adventures all across the country. While research may be the most important aspect of a dissertation, completing a Ph.D. would not be possible without a significant non-academic support system. I count myself lucky to be surrounded by the adventurous, intelligent friends who have so enriched my life over the past six years. I will always cherish my memories of graduate school. Thank you all for being who you are.

Contents

Chapter

1	Introduction and Overview	1
1.1	Introduction	1
1.2	Research Contributions	2
1.2.1	Characterization of BLC-IRK Integration Performance	3
1.2.2	Comprehensive Look at the ITRS/GCRS Transformation	4
1.2.3	Characterization of ITRS/GCRS Simplifications on Position Transformations	5
1.2.4	Impact of ITRS/GCRS Simplifications on Orbit Propagation	5
1.2.5	Performance of Orbit Propagation in the ITRS	6
1.3	Overview of Dissertation	7
2	Mathematical Background	8
2.1	Introduction to Orbit Propagation	8
2.1.1	Runge-Kutta Methods	9
2.2	Introduction to the ITRS and GCRS	11
2.2.1	ITRS/GCRS Transformation Procedure	11
2.2.2	Earth Orientation Parameters (EOPs)	15
3	A Bandlimited Implicit Runge-Kutta Integration Scheme	18
3.1	Abstract	18
3.2	Introduction	19

3.3	Mathematical Overview	22
3.3.1	Runge-Kutta Methods	23
3.3.2	Collocation IRK	24
3.3.3	New Scheme: BLC-IRK	25
3.4	Implementation and Analysis of BLC-IRK	28
3.4.1	Case Study Description	32
3.4.2	Intervals (step size)	35
3.4.3	Nodes	36
3.4.4	Symplectic Property	38
3.5	Performance Comparison	40
3.5.1	Orbit Propagation	40
3.5.2	Dense Output	47
3.6	Conclusions and Future Work	51
3.7	Appendix	53
3.7.1	BLC-IRK Data Files	53
3.8	Acknowledgements	54
4	ITRS/GCRS Implementation for Coordinate Transformations	55
4.1	Abstract	55
4.2	Introduction	56
4.3	Mathematical Overview	57
4.3.1	ITRS/GCRS Transformation	58
4.3.2	Earth Orientation Parameters	61
4.3.3	EOP Ocean Tide Corrections	62
4.3.4	Regularization of ΔUT1 and LOD	64
4.4	Position Transformation Analysis	65
4.4.1	Ignoring EOPs	66

4.4.2	EOP Interpolation and Ocean Tide Corrections	68
4.4.3	Interpolation of UT1-UTC Over Leap Seconds	71
4.4.4	Precession–Nutation Simplifications	73
4.4.5	Julian Date Handling	76
4.5	Summary of Recommendations	77
4.6	Conclusion	79
4.7	Acknowledgements	81
4.8	Appendix	81
4.8.1	Transformation Procedure and EOPs	81
4.8.2	X, Y, s Behavior	82
4.8.3	X, Y, s Interpolation Accuracy	84
4.8.4	Fundamental Arguments	86
4.8.5	4-Term X, Y Series Approximation	86
4.8.6	6-Term X, Y Series Approximation	87
4.8.7	15-Term X, Y Series Approximation	89
5	ITRS/GCRS Impact on Orbit Propagation	90
5.1	Abstract	90
5.2	Introduction	91
5.3	ITRS/GCRS Transformation	93
5.4	EOP and CIO Simplifications	95
5.4.1	X, Y Series Approximation	96
5.4.2	X, Y, s Interpolation	97
5.4.3	X, Y, s Polynomial Fitting	97
5.4.4	Speed and Memory Comparison: X, Y, s	99
5.4.5	EOP Polynomial Fitting	101
5.5	Impact on Orbit Propagation in the GCRS	104

5.6	Orbit Propagation in the ITRS	107
5.6.1	Velocity Transformation	108
5.6.2	ITRS Orbit Propagation Analysis and Results	115
5.7	Conclusion	120
5.8	Acknowledgements	121
5.9	Appendix	121
5.9.1	ITRS/GCRS Transformation Derivatives	121
5.9.2	Partial Derivatives for ITRS Orbit Determination	123
6	Conclusions	132
	Bibliography	134
	Appendix	

Tables

Table

3.1	Initial osculating Keplerian orbital elements and perigee altitude of each orbit investigated	33
3.2	Number of nodes per interval and the corresponding bandlimits	36
3.3	Performance summary of integration methods over three orbit periods	46
4.1	Summary of position transformation analysis setup	65
4.2	Maximum 3D position errors and angular errors of each reduced EOP set	67
4.3	Maximum 3D position differences and angular errors resulting from interpolating EOPs with different schemes	68
4.4	Percentage of the days EOP interpolation errors are greater than the IERS reported EOP uncertainties	71
4.5	Maximum 3D position differences and angular errors resulting from not performing regularization on ΔUT1 and LOD	71
4.6	Relationship between UTC and UT1 during the addition of a leap second	72
4.7	Maximum 3D position and angular differences resulting from bias–precession–nutation simplifications	75
4.8	Several common examples of ways that Julian dates can be stored	76
4.9	Maximum 3D position differences in the ITRS/GCRS frame transformation due to storing Julian date as a single value	77

4.10	Recommendations for EOP usage and precession–nutation implementations to achieve specific levels of position transformation accuracy	79
4.11	Linear approximations of the Delaunay variables	86
4.12	Series for the 4-term X, Y approximation	86
4.13	Series for the 6-term X, Y approximation	88
4.14	Series for the 15-term X, Y approximation	89
5.1	Maximum and RMS errors of X , Y , and s for various polynomial fit windows and degrees	98
5.2	Normalized time and memory required to compute CIO parameters X , Y , and s . .	100
5.3	Maximum and RMS errors of x_p , y_p , and ΔUT1 for various polynomial fit windows and degrees	101
5.4	Orbital elements and propagation settings	105
5.5	Maximum orbit propagation errors for each orbit type and ITRS/GCRS transformation method	106
5.6	Maximum velocity transformation errors (magnitude) due to simplifications of Earth’s rotation vector	114
5.7	Maximum ITRS orbit propagation position errors for each orbit type and Earth rotation vector computation method	119

Figures

Figure

2.1	Flow chart for several ITRS/GCRS frame transformation procedures	12
2.2	Behavior of daily tabulated EOP values	16
3.1	Comparison of node accumulation for exponential and polynomial-based quadratures	28
3.2	Example of nodes and intervals	29
3.3	Illustration of interpolation strategy	34
3.4	RMS values of position errors for propagations of the LEO, GEO, and MOL orbits using a range of number of intervals per orbit	35
3.5	RMS values of position errors for the LEO orbit using a range of number of nodes per interval	37
3.6	Change in Jacobi Constant during a 10-year GEO propagation	39
3.7	Comparison of RMS position errors over a 3-orbit GEO and LEO	42
3.8	Comparison of low- and high-fidelity force models used in this study over 1 orbit period for GEO and LEO	44
3.9	Comparison of RMS position errors over a 3-orbit Molniya propagation	45
3.10	BLC-IRK collocation interpolation error for a GEO propagation	49
3.11	Interpolation errors of BLC-IRK and DOPRI 8(7) trajectories using a 5th-order Lagrange scheme	50
4.1	Ocean tide corrections for applicable EOPs on 12 January 2013	64

4.2	Position errors resulting from the exclusion of EOPs and the use of GMST alone . .	67
4.3	Comparison of EOP uncertainties published by the IERS, EOP interpolation errors, and EOP ocean tide corrections	70
4.4	Demonstration of incorrectly interpolated (left) and correctly interpolated (right) UT1-UTC data series	73
4.5	Flow chart for several ITRS/GCRS frame transformation procedures	81
4.6	Behavior of daily tabulated EOP values for 90 days from 1 January 2013	82
4.7	Actual values of X and Y between 1990 to 2050 generated using the Full Theory . .	83
4.8	Actual values of CIO-locator s between 1990 to 2050 generated using the Full Theory	83
4.9	Error in CIO parameters X, Y, s from 11th-order Lagrange interpolation	84
4.10	Error in CIO parameters X, Y, s from 9th-order Lagrange interpolation	85
4.11	Error in CIO parameters X, Y, s from 7th-order Lagrange interpolation	85
4.12	Error in CIO parameters X and Y from the 4-term series approximation	87
4.13	Error in CIO parameters X and Y from the 6-term series approximation	88
4.14	Error in CIO parameters X and Y from the 15-term series approximation	89
5.1	Polynomial fit errors for X and Y using a 30-day quadratic fit	99
5.2	Polynomial fit errors for x_p and y_p using a 15-day quadratic fit	102
5.3	Polynomial fit errors for ΔUT1 using a 15-day linear and quadratic fit	103
5.4	Comparison of the time derivative of individual CIO and EOP parameters over time	111
5.5	Magnitudes of each component of Earth's total rotation vector from the year 2000 to 2050	113
5.6	Comparison of force models at different altitudes	117
5.7	Behavior of Euler/Coriolis/centripetal acceleration for an 800-km altitude LEO orbit	117

Chapter 1

Introduction and Overview

1.1 Introduction

The space age began with the launch of Sputnik 1 in late 1957. Just 17 years later, the number of tracked objects in Earth orbit was already up to 7,000. Today, we continue to rely heavily on satellites for navigation, communication, weather monitoring, geodesy, and reconnaissance, and the public space catalog now consists of approximately 17,000 objects in Earth orbit.¹ This growing population is comprised of spent rocket bodies, defunct satellites, active payloads, and debris from explosions and other incidents, posing a serious threat to our use of space (Johnson et al, 2008). Maintaining the growing catalog, performing conjunction assessment for collision prevention, and operating on-going satellite missions all require orbit propagation and orbit determination encompassing a wide range of accuracy and computational requirements.

This research is motivated by the need to improve the computational performance of existing algorithms as the number of objects to be tracked orbiting Earth is expected to increase significantly in the near future. Planned improvements in sensing capabilities are anticipated to increase the space catalog to hundreds of thousands. In addition, demands for precise orbit determination with shorter latency for science and reconnaissance missions require more efficient precise frame rotations. The computational load of orbit determination is primarily determined by orbit propagation and the computation of expected measurements. Within orbit propagation, the evaluation of force models (e.g., gravity field) and precise frame transformations can be large computation time sinks for precise

¹ Based on NASA Orbital Debris Quarterly News, Vol. 19, Issue 3, July 2015.

applications, driving the need to reduce the number of times these calculations are performed. The computation of expected measurements (e.g., pseudorange) typically requires precise frame transformations of either the satellite being tracked or the observation sites. The computational complexity of this transformation depends on the application and should be adjusted such that the precision is met while minimizing memory and computation time.

This dissertation develops procedures to achieve various levels of accuracy while minimizing computational cost for diverse orbit determination applications. This is done by addressing two aspects of orbit determination: (1) numerical integration used for orbit propagation and (2) precise frame transformations necessary for force model evaluation and station coordinate rotations. A new implicit Runge-Kutta numerical integration scheme, dubbed Bandlimited Collocation Implicit Runge-Kutta (BLC-IRK), is presented for orbit propagation and compared against commonly-used schemes in astrodynamics. The formulation of the new scheme allows longer time steps with a smaller number of force model evaluations. On the frame transformation side, rotating between the Geocentric Celestial Reference System (GCRS) (i.e., Earth-centered inertial ECI) and the International Terrestrial Reference System (ITRS) (i.e., Earth-centered, Earth-fixed ECEF) is a frequent computation that can be expensive for precise applications. The transformation includes precession, nutation, polar motion, and the Earth's proper rotation. This research presents a detailed look at this transformation, characterizing the effect of each transformation component and specifying simplifications that can be made to save computation time and memory. Orbit propagation and position transformation simulations are analyzed to generate a complete set of recommendations for performing the ITRS/GCRS transformation for a wide range of needs, encompassing real-time on-board satellite operations and precise post-processing applications.

1.2 Research Contributions

This research develops and highlights new algorithms and algorithm simplifications used in orbit propagation and frame transformations in an effort to maximize accuracy while minimizing computational load. This dissertation presents and characterizes a new numerical integration

scheme for orbit propagation, provides a detailed analysis of the ITRS/GCRS transformation, and develops appropriate ITRS/GCRS simplifications for coordinate transformations necessary in positioning and orbit propagation. The research impacts several areas of astrodynamics: (1) long-term, precise orbit propagation for catalog maintenance and conjunction assessment, (2) post-processing precise orbit determination, (3) on-board orbit determination and propagation, and (4) ground-based telescope/radar pointing and spacecraft attitude transformations. The key contributions are described in the following subsections.

1.2.1 Characterization of BLC-IRK Integration Performance

This research (Bradley et al, 2014) provides a mathematical overview of the new BLC-IRK numerical integration scheme, developed by Beylkin and Sandberg (2014), and applies it to orbit propagation to compare its efficiency with other more commonly used techniques in astrodynamics. The new scheme, BLC-IRK is an Implicit Runge-Kutta (IRK) collocation scheme which uses generalized Gaussian quadratures for bandlimited exponentials rather than the classical quadratures for orthogonal polynomials (Beylkin and Sandberg, 2014). The formulation allows the use of a large number of nodes within each time interval without producing an excessive node concentration near the endpoints of the interval.

Recently, IRK methods have received a lot of attention for use in orbit and uncertainty propagation, due to the fact that these methods can be parallelized and have improved stability properties when compared to the traditional methods (Barrio et al, 1999; Jones and Anderson, 2012; Jones, 2012; Bradley et al, 2012; Bai, 2010; Bai and Junkins, 2011a; Aristoff and Poore, 2012; Aristoff et al, 2012; Herman et al, 2013; Aristoff et al, 2014). Collocation methods have also been used for boundary value problems in trajectory design and optimization (Herman and Conway, 1996, 1998; Betts and Erb, 2003; Ozimek et al, 2009, 2010; Grebow et al, 2010, 2011; Bai and Junkins, 2011b, 2012). Another method for parallelized evaluation of the force model, dubbed Modified Chebyshev-Picard Iteration (MCPI), uses the Gauss-Lobatto-Chebyshev nodes in an algorithm similar to collocation (Bai, 2010; Bai and Junkins, 2011a).

This research demonstrates that BLC-IRK is faster at propagating orbits than explicit methods, such as Dormand and Prince and Runge-Kutta-Fehlberg schemes, and approaches the efficiency of the 8th-order Gauss-Jackson multistep method. The improved efficiency is achieved through the superior node spacing near interval boundaries and the use of a low-fidelity force model for most of the iterations, thus minimizing the number of high-fidelity force model evaluations. The improved efficiency for long-term propagations and the built-in collocation makes this method beneficial for space catalog propagation and conjunction assessment.

1.2.2 Comprehensive Look at the ITRS/GCRS Transformation

This dissertation provides a detailed and comprehensive procedure for implementing the ITRS/GCRS transformation and using Earth orientation parameters (EOPs) in the framework of astrodynamics. The numerical models and precise position solutions needed for orbit propagation, determination, and analysis of science data require accurate frame transformations, particularly between the GCRS (i.e., Earth-centered inertial ECI) and the ITRS (i.e., Earth-centered, Earth-fixed ECEF). The procedures and models needed to transform between these reference systems are maintained by the International Earth Rotation Service (IERS) and International Astronomical Union (IAU) (Petit and Luzum, 2010; Wallace and Capitaine, 2006). This transformation includes precession, nutation, polar motion, Earth’s proper rotation, and the use of EOPs. An overview of the IAU-supported transformation methods and several options for implementing the Celestial Intermediate Origin (CIO) scheme are given (Bradley et al, 2011, 2015; SOFA, 2014).

Most presentations of ITRS/GCRS models focus on a particular aspect of the transformation, such as bias–precession–nutation or ocean tide effects. Even the IERS conventions, which contain each piece of information, lack a clear procedure. This work presents an in-depth analysis of the transformation and provides a complete and straightforward procedure, where each aspect of the transformation is framed and analyzed from the perspective of astrodynamics and positioning. Included are the procedures and impacts of EOP ocean tide corrections, ΔUT1 regularization, and the less commonly addressed topic of Julian date handling.

1.2.3 Characterization of ITRS/GCRS Simplifications on Position Transformations

The implementation of the ITRS/GCRS transformation and the manner in which the EOPs are used have a notable impact on station coordinates and satellite positions. Coordinates and positions such as these are required for astrodynamics applications including orbit determination in support of operations, payload data analysis, and space catalog maintenance. This research investigates the impact of algorithm simplifications and EOP usage on precise frame transformations.

Due to the large computation time associated with implementing the full theory of the transformation, several simplifications to the ITRS/GCRS transformation, namely bias-precession-nutation are presented. Simplifications proposed by Capitaine and Wallace (2008) and Coppola et al (2009) are described and included in the analysis. New recommendations such as ignoring EOPs, trigonometric simplifications, and a new Celestial Intermediate Pole (CIP) computation are introduced to generate additional options which balance accuracy and complexity. The analysis contributes a summary of recommendations on EOP usage and bias-precession-nutation implementations for achieving a wide range of transformation accuracies at several altitudes. This comprehensive set of recommendations allows satellite operators, astrodynamicists, and scientists to make informed decisions when choosing the best transformation implementation for their application, balancing accuracy and computational complexity.

1.2.4 Impact of ITRS/GCRS Simplifications on Orbit Propagation

This research extends the mathematical overview and proposed simplifications to the ITRS/GCRS transformation described in the previous section to the application of orbit propagation. Specifically, this portion of the dissertation contributes two things: (1) an analysis of the impact of the ITRS/GCRS transformation on the evaluation of Earth's gravity field during orbit propagation and (2) characterization of the impact that including the CIP location into traditional atmospheric drag calculations has on orbit propagation at low altitudes.

The computation of the acceleration due to Earth’s gravity field (e.g., through spherical harmonics) must be performed multiple times per numerical integration step when propagating orbits using special perturbation theory. This work generates recommendations for the minimum ITRS/GCRS transformation complexity needed to achieve specific orbit propagation accuracies in different orbit regimes. It is shown that the commonly-used simplification of this transformation is not suited for even moderately accurate propagations; however, the most precise implementation is also not warranted. A significant savings in computation time and memory, while yielding negligible orbit propagation inaccuracies, can be made by the use of a polynomial and trigonometric series approximation of the CIP with a small number of terms.

1.2.5 Performance of Orbit Propagation in the ITRS

Orbit propagation is commonly performed in an inertial reference system (e.g., the GCRS when orbiting Earth) in order to avoid the additional terms necessary when working with a rotating system (e.g., the ITRS) and because most forces acting on a satellite are conveniently expressed in an inertial frame (Vallado, 2013; Montenbruck and Gill, 2000). However, there are situations when propagating an orbit in a rotating frame may be beneficial. Such is the case for satellites utilizing GPS measurements to perform precise orbit determination on-board, because the position of each GPS satellite is given in the ITRS. Propagating an orbit in the ITRS requires that each perturbing acceleration be expressed in the ITRS and that additional centripetal, Coriolis, and Euler accelerations be included. Computing the additional acceleration terms requires the rotation vector of the ITRS. Montenbruck and Ramos-Bosch (2008) describe a simple approach for computing this rotation vector, but the full expression is complex and requires time-derivatives of polar motion, CIP coordinates, and the time-derivative of the rotation vector itself. One derivation of the rotation vector is given by Capitaine et al (2006) for use in long-term propagation of the Earth’s orientation, but we require a different form for use in ITRS orbit propagation.

My work provides a detailed derivation of the rotation vector necessary for ITRS propagation. Each time-varying component of precession–nutation, polar motion, and proper motion of Earth

is analyzed. Orbit propagation simulations with a variety of rotation vector simplifications are used to generate recommendations which balance accuracy and computational complexity. The resulting recommendations give satellite operators and scientists a concrete set of options to decide if ITRS propagation is right for their application, and if so, how best to implement it. This research demonstrates that ITRS propagation is viable for both real-time on-board orbit propagation and long-term orbit propagation.

1.3 Overview of Dissertation

The dissertation begins with a mathematical overview of numerical integration and the ITRS/GCRS frame transformation. First, explicit Runge-Kutta (ERK) methods and IRK methods are introduced. It is followed by an overview of the ITRS/GCRS procedure and EOPs to put the following research into context. Some of this material is duplicated in the following chapters as part of the journal papers. Chapter 3 contains the article on the orbit propagation application of the new numerical integration technique, BLC-IRK, which has already been published in *Celestial Mechanics and Dynamical Astronomy* (Bradley et al, 2014). In chapter 4, the manuscript on ITRS/GCRS simplifications and EOPs for use in coordinate transformations is presented. This paper has been accepted and is in press at *Advances in Space Research*. Chapter 5 contains the manuscript on the impact of the ITRS/GCRS transformation on orbit propagation as well as the detailed development of propagation in the ITRS. The dissertation concludes with a summary of the findings of the body of work and ideas for future work.

Chapter 2

Mathematical Background

This chapter provides a mathematical overview of numerical integration used for orbit propagation, the ITRS/GCRS transformation, and EOP usage to provide the necessary framework for the other chapters.

2.1 Introduction to Orbit Propagation

This section details the mathematical techniques of explicit Runge-Kutta (ERK) and implicit Runge-Kutta (IRK) methods to put the new BLC-IRK scheme into context. Numerical integration techniques are used for propagating orbits via special perturbation theory. Specifically, we focus on Cowell's formulation which utilizes the precise computation of the satellite's acceleration at various points in time and space to predict its position and velocity in the future (Vallado, 2013). We consider the initial value problem (IVP) for an ordinary differential equation (ODE)

$$\mathbf{y}' = \mathbf{f}(t, \mathbf{y}), \quad \mathbf{y}(0) = \mathbf{y}_0, \quad t \geq 0. \quad (2.1)$$

For the case of orbit propagation using Cowell's formulation, \mathbf{y} , is typically a 6-element state vector containing the Cartesian position, \vec{r} , and velocity, $\dot{\vec{r}}$, of a satellite where $\mathbf{y} = \begin{bmatrix} \vec{r} & \dot{\vec{r}} \end{bmatrix}^T$. The derivative is then a set of two first-order ODEs consisting of the velocity and acceleration, $\mathbf{y}' = \begin{bmatrix} \dot{\vec{r}} & \ddot{\vec{r}} \end{bmatrix}^T$. The acceleration of the satellite is based on the current time, position and velocity, and physical properties of the satellite and includes contributions from the central body's gravity field, third-body gravitational effects (e.g., from the Sun and Moon), atmospheric drag, and solar

radiation pressure (SRP). The solution \mathbf{y} at some time h can then be written as a Picard integral

$$\mathbf{y}(h) = \mathbf{y}_0 + \int_0^h \mathbf{f}(s, \mathbf{y}(s)) \, ds. \quad (2.2)$$

Runge-Kutta methods are based on using quadratures for discretization of the integral in Eq. 2.2.

2.1.1 Runge-Kutta Methods

While ERK are commonly used in astrodynamics problems, the use of IRK methods is still infrequent. Runge-Kutta methods use M stages (nodes) within a time interval to solve Eq. 2.2 above. The basic form of Runge-Kutta methods uses quadratures to integrate from time $t = 0$ to time $t = h$ as

$$\mathbf{y}(h) = \mathbf{y}_0 + h \sum_{j=1}^M w_j \mathbf{f}(h\tau_j, \mathbf{y}(h\tau_j)), \quad \tau \in [0, 1] \quad (2.3)$$

with weights $\{w_j\}_{j=1}^M$ and nodes $\{\tau_j\}_{j=1}^M$. Using $\boldsymbol{\xi}_i$ to denote values of the solution at the nodes $\mathbf{y}(h\tau_j)$, we have

$$\boldsymbol{\xi}_i = \mathbf{y}_0 + h \sum_{j=1}^M S_{ij} \mathbf{f}(h\tau_j, \boldsymbol{\xi}_j), \quad (2.4)$$

where S is the integration matrix (Iserles, 2009), and find $\mathbf{y}(h)$ as

$$\mathbf{y}(h) = \mathbf{y}_0 + h \sum_{j=1}^M w_j \mathbf{f}(h\tau_j, \boldsymbol{\xi}_j). \quad (2.5)$$

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

$$\begin{array}{c|c} \boldsymbol{\tau} & S \\ \hline & \mathbf{w}^T \end{array} \quad (2.6)$$

which expands to

$$\begin{array}{c|ccc} \tau_1 & S_{1,1} & \cdots & S_{1,M} \\ \tau_2 & S_{2,1} & \cdots & S_{2,M} \\ \vdots & \vdots & & \vdots \\ \tau_M & S_{M,1} & \cdots & S_{M,M} \\ \hline & w_1 & \cdots & w_M \end{array} \quad (2.7)$$

We use τ , w , and S , for representing nodes, weights, and the integration matrix and note that the variables c , b , and A have also been used for this purpose in the literature (Montenbruck and Gill, 2000).

In ERK methods, the integration matrix is lower triangular, $S_{ij} = 0$ for $j \geq i$, and, consequently, such methods are explicit. In IRK methods, the set of nonlinear equations in Eq. 2.4 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 (2002, 1993), and Hairer and Wanner (1996).

The ERK methods used for this research are those commonly used in today's orbit propagation software and simulations. The following is a list of those considered and a short description of each method. All are variable-step embedded ERK methods. A comparison of each integration scheme is contained within the study of the BLC-IRK scheme in Chapter 3. Montenbruck (1992) and Montenbruck and Gill (2000) also provide a comprehensive description and comparison between integration schemes used in astrodynamics.

- 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 (Fehlberg, 1968). The software package Satellite Tool Kit, by Analytical Graphics Inc., uses this as the default integrator (other options are available as well). The implementation of RKF 7(8) used in this study is not using local extrapolation (i.e., the 7th-order result is used as the solution).
- Dormand & Prince 8(7) (DOPRI 8(7)13 or RK 8(7)13): similar to the 13-stage RKF 7(8), but uses an 8th-order method for the solution and a 7th-order method for step size control (Prince and Dormand, 1981).
- 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 (Dormand

and Prince, 1980). This integration scheme is available in MATLAB where it is known as ode45 (Shampine and Reichelt, 1997). 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.

2.2 Introduction to the ITRS and GCRS

This section gives an overview of the ITRS/GCRS transformation and EOP behavior and usage. A summary of each transformation procedure discussed in this body of work is shown in Fig. 2.1 (Petit and Luzum, 2010; SOFA, 2014).

2.2.1 ITRS/GCRS Transformation Procedure

Over the years, several recommendations have been published by the IERS on the precise transformation between the ITRS and GCRS reference systems (McCarthy, 1992, 1996; McCarthy and Petit, 2004; Petit and Luzum, 2010). Effective January 1, 2009 the IERS recommends the use of the new IAU models for nutation and precession to perform this ITRS/GCRS transformation (Petit and Luzum, 2010; Mathews et al, 2002; Buffett et al, 2002; Herring et al, 2002; Hilton et al, 2006; Wallace and Capitaine, 2006; Capitaine and Wallace, 2008; Fukushima, 2003). The new model, as a whole, is called IAU 2000A/2006. This includes the IAU 2000A_{R06} nutation theory and the IAU 2006 precession theory. The inclusion of the “R06” subscript indicates that we refer to the 2000A nutation theory that has been updated to be compliant with the 2006 precession model. Several methods for implementing this new model have been developed over the past few years, but are all within the uncertainty of the IAU model itself (Capitaine and Wallace, 2008; Hilton et al, 2006; Vallado, 2013; Coppola et al, 2009). The methods can be separated into two categories, the classical, equinox-based method and the modern CIO-based approach. Figure 2.1 depicts the procedure of the CIO-based approach and the Fukushima-Williams equinox-based approach (Vallado, 2013; SOFA, 2014).

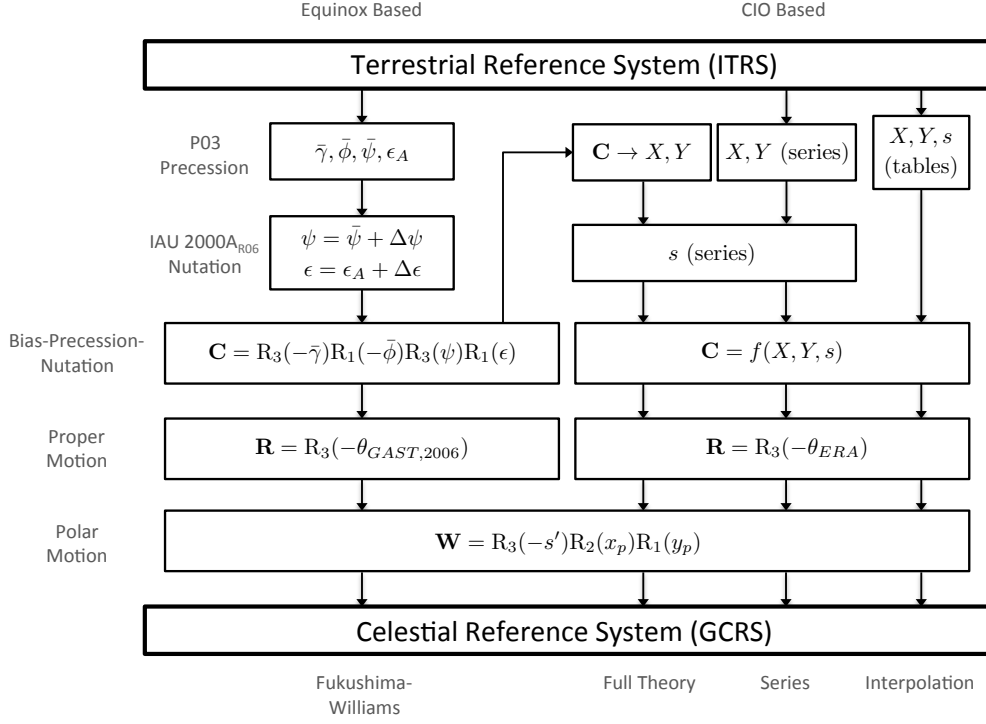


Figure 2.1: Flow chart for several ITRS/GCRS frame transformation procedures. The layout is based on Vallado (2013).

The CIO-based method is built on the direction cosine components of the CIP in the GCRS (X and Y) and the CIO locator (s). This method is preferred when the goal is the transformation between the terrestrial and celestial frames, because the ecliptic and equinox are not needed. (Wallace and Capitaine, 2006). The benefits of the CIO-based method include the following:

- (1) The bias-precession-nutation matrix is formed directly, thus eliminating potential errors when multiplying the separate pieces together as is done with the equinox-based approaches.
- (2) The IERS will publish the dX and dY celestial pole offset values far into the future, while publication of the $\delta\Delta\psi_{2000A}$ and $\delta\Delta\epsilon_{2000A}$ corrections are likely to be discontinued. Additionally, the IERS currently only publishes $\delta\Delta\psi_{1980}$ and $\delta\Delta\epsilon_{1980}$ values which correspond to the older IAU 76/FK5 theory. The $\delta\Delta\psi_{2000A}$ and $\delta\Delta\epsilon_{2000A}$ values required when using the equinox-based approach must be computed from the published dX and dY values (Kaplan, 2005).

- (3) X , Y , and s can be computed a priori using the summation series or Fukushima-Williams angles, stored in a data table, and interpolated as needed Coppola et al (2009). Other methods would require interpolating four values instead of only three.
- (4) The X and Y values are the direction cosine components of the CIP in the GCRS making the CIO-based approach simple to implement and enabling the retrieval of the Earth's instantaneous rotation axis at any point in the time (Wallace and Capitaine, 2006).

The CIO-based equations for transforming a position vector, \vec{r} , and velocity vector, \vec{v} , from the ITRS to the GCRS are given by

$$\mathbf{C} = \begin{bmatrix} 1 - aX^2 & -aXY & X \\ -aXY & 1 - aY^2 & Y \\ -X & -Y & 1 - a(X^2 + Y^2) \end{bmatrix} \mathbf{R}_3(s), \quad (2.8)$$

$$\mathbf{R} = \mathbf{R}_3(-\theta_{ERA}), \quad (2.9)$$

$$\mathbf{W} = \mathbf{R}_3(-s')\mathbf{R}_2(x_p)\mathbf{R}_1(y_p), \quad (2.10)$$

$$\vec{r}_{\text{GCRS}} = \mathbf{C} \mathbf{R} \mathbf{W} \vec{r}_{\text{ITRS}} = \mathbf{T} \vec{r}_{\text{ITRS}}, \quad (2.11)$$

$$\vec{v}_{\text{GCRS}} = \mathbf{C} \mathbf{R} \left\{ \mathbf{W} \vec{v}_{\text{ITRS}} + \vec{\omega}_{\oplus} \times \mathbf{W} \vec{r}_{\text{ITRS}} \right\}, \quad (2.12)$$

where

$$a = 1/(1 + \sqrt{1 - X^2 - Y^2}) \approx 1/(1 + Z), \quad (2.13)$$

$$\vec{\omega}_{\oplus} = \begin{bmatrix} 0 & 0 & \omega_{\oplus} \end{bmatrix}^T, \quad (2.14)$$

and the instantaneous rotation rate of the Earth is¹

$$\omega_{\oplus} = 7.2921151467064 \times 10^{-5} \left\{ 1 - \frac{\text{LOD}}{86400} \right\} \text{ rad/s}. \quad (2.15)$$

¹ This definition is based on Aoki et al (1982), which defines the rotational period of the Earth as 86164.09890369732 seconds of UT1. An overview of the computation of rotation rate and definitions of UT1 and LOD can be found here:
<http://hpiers.obspm.fr/eop-pc/earthor/ut1lod/UT1.html>

The largest computational burden for this transformation is the computation of X, Y, s . Several methods for computing these parameters are available as shown in Fig. 2.1. The series method consists of around 4,000 coefficients to compute X and Y directly and the parameter $s + XY/2$ (Petit and Luzum, 2010; Capitaine and Wallace, 2008). Each series contains a polynomial and trigonometric portion (Capitaine and Wallace, 2008)

$$q = q_0 + q_1 T_{\text{TT}} + q_2 T_{\text{TT}}^2 + q_3 T_{\text{TT}}^3 + q_4 T_{\text{TT}}^4 + q_5 T_{\text{TT}}^5 + \sum_i \sum_{j=0}^4 [(a_{s,j})_i \sin(\text{ARG}) + (a_{c,j})_i \cos(\text{ARG})] T_{\text{TT}}^j, \quad (2.16)$$

where q stands for X , Y , or $s + XY/2$ and ARG stands for combinations of the luni-solar and planetary parts of the fundamental arguments. The time parameter T_{TT} is Julian centuries of Terrestrial Time (TT) past J2000.² The large number of coefficients and trigonometric evaluations make the series evaluation burdensome, both on computation time and memory. However, the series can be truncated to speed up computation time at the loss of accuracy, as shown by Wallace and Capitaine (2006) and Capitaine and Wallace (2008). These papers compare a few of these truncated series to the full accuracy models, demonstrating the ability of these approximations and highlighting the accuracy ranges they are useful for.

A second method is the “Full Theory”, which uses the Fukushima-Williams angles for precession ($\bar{\gamma}, \bar{\phi}, \bar{\psi}, \epsilon_A$) and the IAU 2000A_{R06} nutation theory ($\Delta\psi, \Delta\epsilon$) to compute X and Y Wallace and Capitaine (2006). The CIO locator s must still be computed using the series for the quantity $s + XY/2$. This method has a high computational load as well, due to the series for $s + XY/2$ and the several thousand terms for nutation. More details on the IAU models and implementation processes can be found in Petit and Luzum (2010); Wallace and Capitaine (2006); Vallado (2013).

The third option for computing X , Y , and s is that of tabulation and interpolation. The “Full Theory” or series methods may be used to generate X , Y , and s values at 0hr each day and stored in a data table. The user can then interpolate the values to any desired time. This procedure maintains nearly the full accuracy of the “Full Theory” and series methods while greatly reducing

² Formally, the time argument is Barycentric Dynamical Time (TDB), but TT is used in practice because the resulting difference is below the precision of the IAU models themselves.

computation time. The method is developed and analyzed in detail in Coppola et al (2009) and used in this paper as one of the default recommended procedures.

2.2.2 Earth Orientation Parameters (EOPs)

EOPs are used to correct for errors in the IAU models, such as free core nutation (FCN) in the case of dX and dY , and to account for behavior that is unmodeled by the IAU models, such as polar motion. EOPs consist of the following:³

- Pole Coordinates (x_p, y_p): coordinates of the Celestial Intermediate Pole (CIP) with respect to the IERS Reference Pole (IRP) in the International Terrestrial Reference System (ITRS). The IRP is the location of the agreed upon terrestrial pole while the CIP is the instantaneous rotation axis of the Earth.
- Celestial Pole Offsets (dX, dY): observed corrections to the conventional celestial pole. The conventional celestial pole position is defined by the IAU Precession and Nutation models.
- UT1 Time Difference (ΔUT1): the offset of Universal Time (UT1) from Universal Coordinated Time (UTC), where $\Delta\text{UT1} = \text{UT1} - \text{UTC}$.
- Length of Day (LOD): time difference between the observed duration of a mean solar day and 86,400 SI seconds. LOD is the time derivative of ΔUT1 and has units of seconds per day. Omitting ‘per day’ is common when discussing LOD and is left out in this paper.
- Atomic Time Offset (ΔAT): time difference between International Atomic Time (TAI) and Universal Coordinated Time (UTC), where $\Delta\text{AT} = \text{TAI} - \text{UTC}$ (i.e., leap seconds).

Although the Atomic Time offset (leap seconds) is not an EOP, it is included in this list because it is commonly incorporated into tabulated EOP data files. EOPs are published by the IERS once per day in their Bulletin A, effective at 0h UTC of that day. Figure 2.2 illustrates the

³ The term Earth Rotation Parameters (ERP) is also a common acronym in literature. The ERPs are a subset of EOPs consisting of only the pole coordinates, UT1 time difference, and length of day.

behavior of the daily tabulated EOPs to aid our discussion of EOPs and their interpolation. The length-of-day and celestial pole offsets contain the most dramatic variation over time, while the pole coordinates and time offset parameters exhibit much smoother variations. These daily EOP values must be interpolated for maximum accuracy at other times during the day before they are incorporated into the ITRS/GCRS transformation. In literature to date, however, recommended interpolation methods to achieve specific accuracies are not mentioned. Additionally, the published values of the pole coordinates, UT1 time difference, and length-of-day are “tide-free”, meaning that users requiring high-precision need to apply additional corrections for diurnal and semi-diurnal ocean tides and libration effects after the daily values have been interpolated (Petit and Luzum, 2010). This fact is fairly unpublicized within astrodynamics literature. Nonetheless, Vallado (2013), Montenbruck and Gill (2000), Vallado and Kelso (2005) and Vallado and Kelso (2013) describe the general use of EOPs clearly. Additionally, Stamatakos et al (2008) demonstrate the effect of EOPs on satellite pointing accuracy requirements.

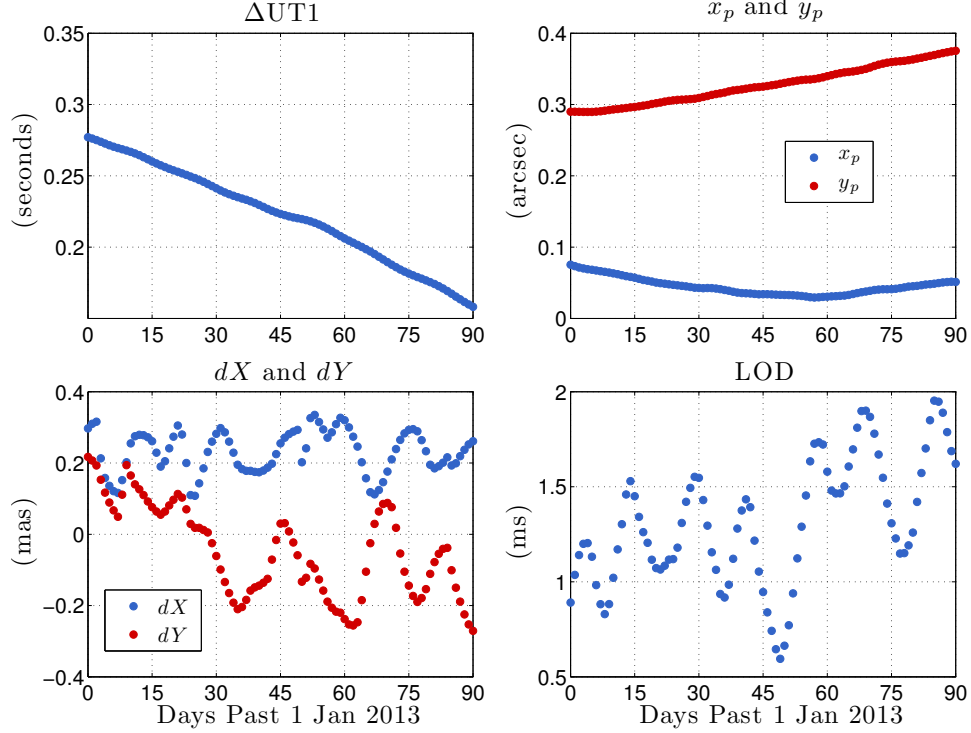


Figure 2.2: Behavior of daily tabulated EOP values for 90 days from 1 January 2013.

The list below identifies which EOP contributes to each of the components needed for the ITRS/GCRS transformation.

- $X = X + dX, Y = Y + dY \rightarrow \mathbf{C}$
- $UT1 = UTC + \Delta UT1 \rightarrow \mathbf{R}$
- $(x_p, y_p) \rightarrow \mathbf{W}$
- $LOD \rightarrow \omega_{\oplus}$

The EOPs are generated and published by several sources, including the IERS,⁴ the National Geospatial-Intelligence Agency (NGA),⁵ and the United States Naval Observatory (USNO).⁶ CelesTrak has compiled EOPs from these sources to create a single file containing EOP data from 1962 up to predictions 1 year into the future.⁷ The file maintained by CelesTrak is used by Analytical Graphics Inc. Systems Tool Kit (STK) and is used for the EOP studies presented in this paper, unless otherwise mentioned. See Vallado and Kelso (2005, 2013, 2015) for more information on the differences between each agencies' EOP product and how the CelesTrak file is formed.

⁴ <http://www.iers.org/IERS/EN/DataProducts/EarthOrientationData/eop.html>

⁵ <http://earth-info.nga.mil/GandG/sathtml/eopp.html>

⁶ <http://www.usno.navy.mil/USNO/earth-orientation/eo-products>

⁷ <http://www.celestrak.com/SpaceData/>

Chapter 3

A Bandlimited Implicit Runge-Kutta Integration Scheme

Journal Article:

Bradley, B.K., Jones, B.A., Beylkin, G., Sandberg, K. and Axelrad, P., "Bandlimited Implicit Runge-Kutta Integration for Astrodynamics," *Celestial Mechanics and Dynamical Astronomy*, Vol. 119, 2014, pp. 143–168.

Extension of Conference Paper:

Bradley, B.K., Jones, B.A., Beylkin, G., and Axelrad, P., "A New Numerical Integration Technique in Astrodynamics," 22nd AAS/AIAA Space Flight Mechanics Meeting, AAS 12-216, Charleston, SC, Jan. 29–Feb. 2, 2012.

3.1 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.

3.2 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.

Motivated by the need to improve the computational performance of existing schemes as the number of objects to be tracked orbiting Earth is expected to increase significantly in the near future, we compare the performance of the new scheme with the traditional methods used in astrodynamics. 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 (Kelso and Alfano, 2005). Currently, the public space catalog consists of between 15,000 objects¹ and 17,000 objects² in Earth orbit that are at least 10 centimeters in diameter. Although conjunction assessment for the entire space catalog is manageable at this time, it will become

¹ Based on bulk TLE data sets from Space-Track.org in March of 2013.

² Based on NASA Orbital Debris Quarterly News, Vol. 18, Issue 1, January 2014.

difficult in the near future. In part, the expected difficulty is due to the planned improvements in sensing and computation capabilities. These new capabilities are anticipated to increase the space catalog to hundreds of thousands, making the current method for performing orbit determination and conjunction assessment challenging. Since orbit determination and propagation take up a majority of the computation time, faster numerical integration techniques are considered necessary. Furthermore, fast integrators may be used for tracking and propagation of asteroids and can also aid Monte Carlo analyses used in research and mission design (Parcher and Whiffen, 2011).

Recently, IRK methods have received a lot of attention for use in orbit and uncertainty propagation, mainly due to the fact that these methods can be parallelized and have improved stability properties when compared to the traditional methods (Barrio et al, 1999; Jones and Anderson, 2012; Jones, 2012; Bradley et al, 2012; Bai, 2010; Bai and Junkins, 2011a; Aristoff and Poore, 2012; Aristoff et al, 2012; Herman et al, 2013; Aristoff et al, 2014). Specifically, Gauss-Legendre implicit Runge-Kutta (GL-IRK) is symplectic, A -stable, B -stable, and has been shown to outperform explicit Runge-Kutta (ERK) methods for both orbit propagation and uncertainty propagation (Jones, 2012; Aristoff and Poore, 2012; Aristoff et al, 2012, 2014). Similarly, BLC-IRK is both symplectic and A -stable (see Beylkin and Sandberg (2014) for details). Collocation methods have also been used for boundary value problems in trajectory design and optimization (Herman and Conway, 1996, 1998; Betts and Erb, 2003; Ozimek et al, 2009, 2010; Grebow et al, 2010, 2011; Bai and Junkins, 2011b, 2012). Another method for parallelized evaluation of the force model, dubbed Modified Chebyshev-Picard Iteration (MCPI), uses the Gauss-Lobatto-Chebyshev nodes in an algorithm similar to collocation (Bai, 2010; Bai and Junkins, 2011a).

While a Runge-Kutta scheme with the Gauss-Legendre nodes provides an excellent discretization of a system of ordinary differential equations (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.

This behavior effectively puts an upper limit on useful step size and the number of nodes, since computations become increasingly wasted near time interval boundaries as the number of nodes increases. On the other hand, the node accumulation of 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 (further discussion may be found in Beylkin and Sandberg (2014)). The consequence of this fact is that the solution may be sought on a large time interval using a large number of nodes. Since BLC-IRK is parallelizable at the node level, using more nodes can improve the speed of the implementation if multiple processors are used. Additionally, the use of generalized Gaussian nodes for bandlimited functions minimizes the total number of nodes required to achieve a given accuracy (Beylkin and Sandberg, 2014; Beylkin and Monzón, 2002). The implementation of BLC-IRK for this paper takes advantage of speed improvements during force model evaluation. We reduce the computational cost associated with iteration at each node by employing a low-fidelity force model for a majority of the required force evaluations. Forms of this technique used in Jones (2012) and Aristoff et al (2014) have shown to vastly improve performance when applied to a GL-IRK scheme (see also Beylkin and Sandberg (2014)).

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 Beylkin and Monzón (2002) use a finite set of nodes 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 ϵ , where ϵ is arbitrarily small but finite. We note that if the accuracy ϵ is chosen to be around 10^{-16} , such quadratures are effectively exact within the double precision of machine arithmetic.

The class of functions well approximated by the bandlimited exponentials $\{e^{ibx}\}_{|b|\leq c}$ includes functions with the support of the Fourier transform restricted to the interval $[-c, c]$. A basis for such bandlimited functions was constructed in a series of seminal papers (Slepian and Pollak, 1961; Landau and Pollak, 1961, 1962; Slepian, 1964, 1965, 1978, 1983) the goal of which was to optimize (simultaneously) the localization of functions in the space and Fourier domains. These papers

showed that the time-limiting and band-limiting integral operator commutes with the differential operator whose eigenfunctions are the so-called Prolate Spheroidal Wave Functions (PSWFs) of classical mathematical physics, i.e., the integral and differential operators share the eigenfunctions. In spite of the importance of bandlimited functions, efficient quadratures for integrating and interpolating them were constructed only recently (Beylkin and Monzón, 2002; Xiao et al, 2001). These quadratures are essential for using bandlimited functions in numerical analysis and, in particular, in the BLC-IRK method.

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 them to other integration techniques. Three orbit types are used to compare results of 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 70×70 gravity field and lunisolar perturbations. The dense output capability of BLC-IRK is then detailed and we conclude with a summary of the results and recommended future work.

3.3 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 ODE

$$\mathbf{y}' = \mathbf{f}(t, \mathbf{y}), \quad \mathbf{y}(0) = \mathbf{y}_0, \quad t \geq 0. \quad (3.1)$$

The solution \mathbf{y} at some time h can then be written as a Picard integral

$$\mathbf{y}(h) = \mathbf{y}_0 + \int_0^h \mathbf{f}(s, \mathbf{y}(s)) \, ds. \quad (3.2)$$

Runge-Kutta methods are based on using quadratures for discretization of the integral in Eq. 3.2.

3.3.1 Runge-Kutta Methods

While ERK are commonly used in astrodynamics problems, the use of IRK methods is still infrequent. Runge-Kutta methods use M stages (nodes) within a time interval to solve Eq. 3.2 above. The basic form of Runge-Kutta methods uses quadratures to integrate from time $t = 0$ to time $t = h$ as

$$\mathbf{y}(h) = \mathbf{y}_0 + h \sum_{j=1}^M w_j \mathbf{f}(h\tau_j, \mathbf{y}(h\tau_j)), \quad \tau \in [0, 1] \quad (3.3)$$

with weights $\{w_j\}_{j=1}^M$ and nodes $\{\tau_j\}_{j=1}^M$. Using $\boldsymbol{\xi}_i$ to denote values of the solution at the nodes $\mathbf{y}(h\tau_j)$, we have

$$\boldsymbol{\xi}_i = \mathbf{y}_0 + h \sum_{j=1}^M S_{ij} \mathbf{f}(h\tau_j, \boldsymbol{\xi}_j), \quad (3.4)$$

where S is the integration matrix (Iserles, 2009), and find $\mathbf{y}(h)$ as

$$\mathbf{y}(h) = \mathbf{y}_0 + h \sum_{j=1}^M w_j \mathbf{f}(h\tau_j, \boldsymbol{\xi}_j). \quad (3.5)$$

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

$$\begin{array}{c|c} \boldsymbol{\tau} & S \\ \hline & \mathbf{w}^T \end{array} \quad (3.6)$$

which expands to

$$\begin{array}{c|ccc} \tau_1 & S_{1,1} & \cdots & S_{1,M} \\ \tau_2 & S_{2,1} & \cdots & S_{2,M} \\ \vdots & \vdots & & \vdots \\ \tau_M & S_{M,1} & \cdots & S_{M,M} \\ \hline & w_1 & \cdots & w_M \end{array} \quad (3.7)$$

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

In ERK methods, the integration matrix is lower triangular, $S_{ij} = 0$ for $j \geq i$, and, consequently, such methods are explicit. In IRK methods, the set of nonlinear equations in Eq. 3.4 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 (2002, 1993), 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 $\mathbf{y}(h\tau_j)$ 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 \mathbf{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.

We note that in the traditional use of Runge-Kutta methods the time interval (or step size), h , is small, typically between 15 and 60 seconds for orbit propagation. In the new method, the time interval does not have to be small since the number of nodes, M , may be selected to be large.

3.3.2 Collocation IRK

Among IRK methods of particular interest are those based on collocation. Consider the polynomial, $\mathbf{u}(t)$, matching the solution at the nodes,

$$\begin{aligned} \mathbf{u}(0) &= \mathbf{y}_0 \\ \dot{\mathbf{u}}(h\tau_j) &= \mathbf{f}(h\tau_j, \mathbf{u}(h\tau_j)) \end{aligned} \tag{3.8}$$

where $\mathbf{y}(h\tau_j) = \mathbf{u}(h\tau_j)$, $j = 1, \dots, M$. As demonstrated in e.g., Iserles (2009), this formulation leads to an IRK method. Introducing Lagrange interpolating polynomials $\{L_j(\tau)\}$ with nodes $\{\tau_j\}_{j=1}^M$, we approximate \mathbf{f} to a given accuracy ϵ on $[0, h]$,

$$\|\mathbf{f}(h\tau, \mathbf{y}(h\tau)) - \sum_{j=1}^M \mathbf{f}(h\tau_j, \mathbf{y}(h\tau_j)) L_j(\tau)\| \leq \epsilon, \quad \tau \in [0, 1]. \quad (3.9)$$

Equation 3.2 is then rewritten using Eq. 3.9 as

$$\mathbf{y}(h\tau_i) = \mathbf{y}_0 + h \sum_{j=1}^M \mathbf{f}(h\tau_j, \mathbf{y}(h\tau_j)) \int_0^{\tau_i} L_j(s) ds, \quad i = 1, \dots, M \quad (3.10)$$

or

$$\mathbf{y}(h\tau_i) = \mathbf{y}_0 + h \sum_{j=1}^M S_{ij} \mathbf{f}(h\tau_j, \mathbf{y}(h\tau_j)) \quad (3.11)$$

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

$$\mathbf{y}(h) = \mathbf{y}_0 + h \sum_{j=1}^M w_j \mathbf{f}(h\tau_j, \mathbf{y}(h\tau_j)) \quad (3.12)$$

yields the solution at time $t = h$ and, thus, Eq. 3.11 and 3.12 form a collocation IRK scheme.

The most commonly used polynomial-based quadratures are Gauss-Legendre (Butcher, 1964) and Gauss-Lobatto, although the use of Chebyshev quadratures (Barrio et al, 1999; Bai, 2010; Bai and Junkins, 2011a) has captured some attention in astrodynamics recently. We note that only the Gauss-Legendre quadratures yield symplectic, A -stable IRK schemes with the maximum order $2M$, where M is the number of stages (nodes).

3.3.3 New Scheme: BLC-IRK

Polynomial-based quadrature has a long history of use due to tradition, ease of use, and node/order optimality (Jones and Anderson, 2012; Iserles, 2009). Polynomial-based quadratures are constructed so that

$$\int_{-1}^1 f(x) W(x) dx = \sum_{j=1}^M w_j f(\tau_j), \quad (3.13)$$

for all polynomials f less than some fixed degree. Here $W(x) \geq 0$ is the weight, τ_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. 3.13 is exact. We note that Gauss-Legendre quadratures correspond to the weight $W(x) = 1$ whereas Chebyshev quadratures correspond to the weight $W(x) = 1/\sqrt{1-x^2}$.

The new scheme described in this paper is a collocation IRK method that uses generalized Gaussian quadratures for bandlimited exponentials instead of polynomials (Beylkin and Sandberg, 2014). Consult Beylkin and Monzón (2002) and Xiao et al (2001) for the development of generalized Gaussian quadratures for exponentials. These quadratures are constructed so that

$$\left| \int_{-1}^1 e^{2ictx} W(t) dt - \sum_{j=1}^M w_j e^{2ic\tau_j x} \right| < \epsilon^2, \quad x, \tau_j \in [-1, 1] \quad (3.14)$$

for the user-selected accuracy $\epsilon > 0$, bandlimit $2c > 0$, and weights $w_j > 0$. The nodes τ_j and weights w_j depend on the bandlimit and accuracy. In the BLC-IRK method the weight $W(t) = 1$ 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 $2c$ and accuracy ϵ^2 , we can generate an interpolating basis for bandlimited functions with bandlimit c and accuracy ϵ . These interpolating basis functions are defined as

$$R_j(x) = \sum_{l=1}^M r_{jl} e^{ic\tau_l x} \quad (3.15)$$

for $j = 1, \dots, M$ with

$$r_{jl} = \sum_{k=1}^M w_j \Psi_k(\tau_j) \frac{1}{\eta_k} \Psi_k(\tau_l) w_l, \quad (3.16)$$

where the matrix $\Psi_k(\tau_l)$ is obtained by solving an algebraic eigenvalue problem,

$$\sum_{l=1}^M w_l e^{ic\tau_l \tau_m} \Psi_k(\tau_l) = \eta_k \Psi_k(\tau_m), \quad k, m = 1, \dots, M. \quad (3.17)$$

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

$$\Psi_k(\tau) = \frac{1}{\eta_k} \sum_{l=1}^M w_l \Psi_k(\tau_l) e^{ic\tau_l \tau}, \quad k = 1, \dots, M. \quad (3.18)$$

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

$$S_{ij} = \int_0^{\tau_i} R_j(s) ds. \quad (3.19)$$

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

$$r(M, \epsilon) = \frac{\tau_2 - \tau_1}{\tau_{\lfloor M/2 \rfloor} - \tau_{\lfloor M/2 \rfloor - 1}}, \quad (3.20)$$

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. 3.20 yields a quantitative comparison of node accumulation property. 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 3.1 displays the behavior of the ratio as a function of the number of nodes for polynomial-based quadratures and quadratures for exponentials.

The ratio for polynomial-based quadrature nodes asymptotically approaches zero as the number of nodes increase. This ratio for nodes of quadratures for exponentials, however, approaches a finite limit. This asymptote is a function of the accuracy, ϵ , to which the quadrature is constructed,

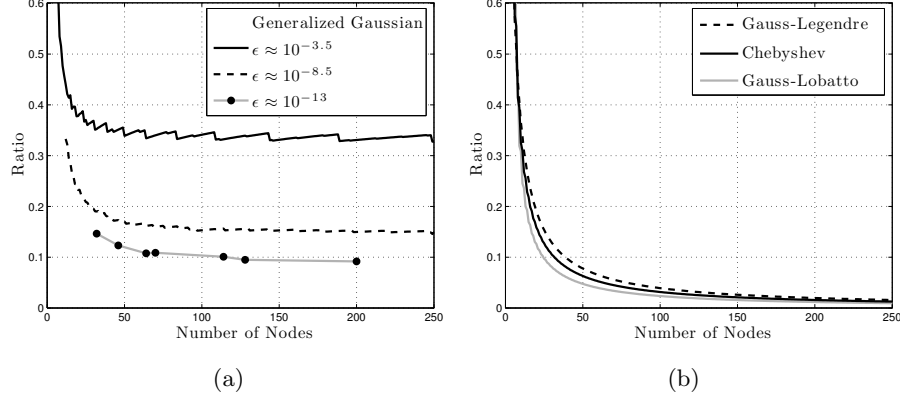


Figure 3.1: Comparison of node accumulation for exponential and polynomial-based quadratures. (a) Generalized Gaussian quadrature for bandlimited exponentials with different interpolation accuracies. Marker dots indicate values for quadratures used in this study. (b) Polynomial-based quadratures. Ratios approach zero as $1/M$.

as seen in Eq. 3.14. This property of generalized Gaussian quadratures for bandlimited functions allows us to use larger time intervals with a large number of nodes per interval when compared to polynomial-based methods.

We provide quadrature data needed to implement BLC-IRK numerical integration online. The accompanying data files are described in Appendix 3.7.1. Data files necessary to perform dense output, discussed in Section 3.5.2, are also given.

3.4 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. As mentioned previously, BLC-IRK is implemented using both a low- and high-fidelity force model to save computational effort during iteration. Figure 3.2 illustrates time intervals and nodes within these intervals to aid in our discussion.

The current implementation 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. We plan to develop an approach to determine appropriate values of each parameter automatically

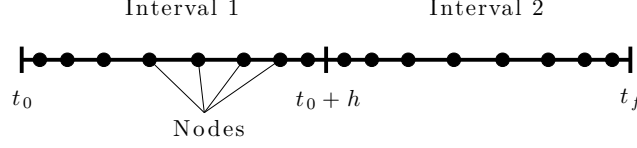


Figure 3.2: Example of nodes and intervals (for illustrative purposes only).

based on the orbit and force model.

- **Accuracy (ϵ):** Interpolation accuracy for which the generalized Gaussian quadratures are constructed. In the current implementation this accuracy is fixed to $\epsilon \approx 10^{-13}$. It may be made available to the user in future implementations.
- **Number of nodes per interval (bandlimit) (M):** For a given accuracy ϵ , the number of nodes per interval determines bandlimit and vice versa. More nodes per interval equates to a higher bandlimit.
- **Number of Intervals (N_I):** A time interval N_I is similar to a step size h in traditional integration schemes where $N_I = (t_f - t_0)/h$ and t_f 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 N_I 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 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, ξ , 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 low-fidelity and high-fidelity force models to reduce the number of evaluations of the high-fidelity force model. IRK methods use iteration to solve the nonlinear equations for ξ , thus involving several calls to the force model, \mathbf{f} , at each node. We first use a low-fidelity force model, \mathbf{f}_{low} , containing point-mass and 3x3 gravity field effects of the Earth, for the first few iterations to place the solution at each node close to the final value. The high-fidelity force model, \mathbf{f}_{high} , is then evaluated once and the difference between the low- and high-fidelity model, $\Delta\mathbf{f}$, is stored. The high-fidelity force model used in this study is comprised of a 70x70 EGM96 gravity model (Lemoine et al, 1998) and third-body gravitational effects from the Sun and Moon. 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, $\Delta\mathbf{f}$ 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. Algorithm 1 describes the overall iteration process used in this study in greater detail.

The results in this paper were generated using a second call to the high-fidelity force model. The second evaluation ensures satisfactory orbit propagation accuracies in the current setup. However, the low-fidelity force model used here is not necessarily the optimal choice. The low-/high-fidelity force models and the iteration implementation can be adjusted for different situations. For example, a satellite in the Jovian system might want to include approximate third-body effects in

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 \mathbf{f}_{low} and \mathbf{f}_{high} .

Note: This algorithm is to be used for each interval

```

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

for  $m = 1 \rightarrow M$  do
  Evaluate  $\mathbf{f}_{\text{high}}^m$  and store  $\Delta \mathbf{f}_m = \mathbf{f}_{\text{high}}^m - \mathbf{f}_{\text{low}}^m$ 
end for

for  $i_2 = 1 \rightarrow N_2$  do
  for  $m = 1 \rightarrow M$  do
    Evaluate  $\mathbf{f}_{\text{low}}^m$ 
    Update  $\xi_m$  with  $\mathbf{f}_{\text{low}}^m + \Delta \mathbf{f}_m$ 
  end for
end for

for  $m = 1 \rightarrow M$  do
  Update  $\xi_m$  by evaluating  $\mathbf{f}_{\text{high}}^m$ 
end for

```

the low-fidelity force model.

We also note that use of a single relative tolerance value for iteration instead of fixing the number of iterations N_1 and N_2 would improve the ease-of-use for the user and guarantee that excess computations were kept to a minimum. The use of a relative tolerance setting is common in many implementations of IRK schemes (e.g., for fixed-point iteration see 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).

3.4.1 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 3.1 lists the Keplerian orbital elements at epoch (0^h January 1st, 2011) for each of the three orbits and includes the perigee altitude, h_p .

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 to 130 intervals over the duration of the propagation as well as 1 to 3 iterations for both N_1 and N_2 . 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 (GJ 8) integration scheme using a 5-second

Table 3.1: Initial osculating Keplerian orbital elements and perigee altitude of each orbit investigated in this study. Epoch is 0^h January 1st, 2011.

Name	a (m)	e	i (deg)	Ω (deg)	ω (deg)	ν (deg)	h_p (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

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 (Jackson, 1924; Fox, 1984; Berry and Healy, 2004; SPADOC Computation Center, 1982). Note, however, that Gauss-Jackson is neither symplectic nor A -stable.

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 (2003) and Berry (2004) investigate 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 GJ 8 scheme with a fixed step size of 5 seconds and compare integration accuracy only. The implementation of GJ 8 follows that of Berry and Healy (2004). 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 reference trajectory with differences below 10^{-5} meters. This indicates that round-off error is not affecting our results for the accuracy range we are considering, i.e., 10^{-4} to 10^2 meters. Other truth trajectories were also assessed, including GJ 8 with 2 and 10 second step sizes, and the 8th-order Dormand and Prince scheme, DOPRI 8(7), with similar steps. Each of these trajectories match the 5-second GJ 8 truth trajectory to the order of 10^{-5} meters.

The interpolation strategy can have a notable impact on computing the error of an integration

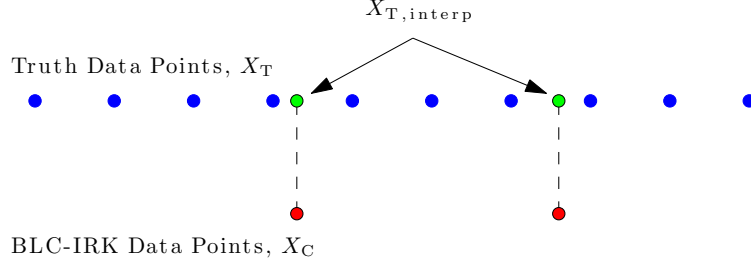


Figure 3.3: Illustration of interpolation strategy. X denotes position solutions. Error comparisons are made at solution points of the method we are testing (e.g., BLC-IRK). The dense truth trajectory is interpolated to these points using a cubic spline to eliminate interpolation error.

method. As depicted in Fig. 3.3, we interpolate the truth trajectory to times where we have a solution from the method we are comparing. A slightly different execution of interpolating at fixed 30-second intervals, however, has shown to introduce errors too large for this study on integration accuracy. 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 dense truth trajectory, error due to interpolation is essentially eliminated. Based on several tests, the observed maximum interpolation error is on the order of 10^{-5} meters for the LEO and MOL orbits, and down to 10^{-8} meters for GEO. These errors are below the accuracy range we are considering. The root-sum-square (RSS) position error at each time, t_i , is computed by

$$\Delta r_i = \left\| \mathbf{X}_{C_i} - \mathbf{X}_{T,interp_i} \right\| \quad (3.21)$$

where \mathbf{X} denotes position solutions. The root mean square (RMS) position error for the entire trajectory is then computed using all Δr_i . All propagation comparison plots report this RMS error for the entire trajectory. The maximum and mean errors were also considered, however, these values are of the same order of magnitude as the RMS error. Given the log plots and similar behavior of each propagation scheme, the general error magnitude and performance relations between each scheme stay approximately the same. For this reason, we display the RMS values only.

3.4.2 Intervals (step size)

First, we look at how the number of intervals affects propagation accuracy. Figure 3.4 shows the relationship between the number of intervals used per orbit and the RMS of position error for all three orbit types. When compared to a small number of intervals per orbit, adding intervals reduces the integration error significantly.

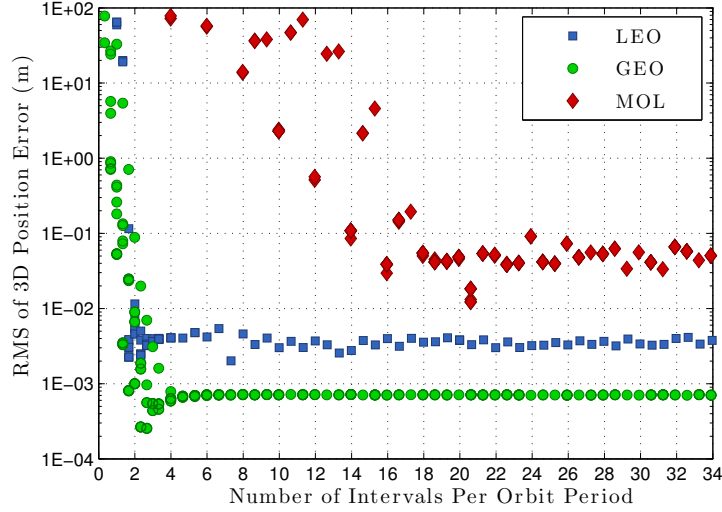


Figure 3.4: RMS values of position errors for propagations of the LEO, GEO, and MOL orbits using a range of number of intervals per orbit. Each propagation has a duration of 3 orbit revolutions and uses 64 nodes per interval. Every combination of $\{1, 2, 3\}$ N_1 iterations and $\{1, 2, 3\}$ N_2 iterations were used.

There reaches a point, however, where additional intervals do not reduce the integration error. Usually, this accuracy floor is caused by the finite precision of computing or the accumulation of roundoff error. In this case, it is due to the iteration algorithm being used for BLC-IRK, as described in Sect. 3.4. Standard implementations of IRK schemes used fixed-point, or Picard, iteration until convergence to some tolerance, e.g., 10^{-13} (Hairer et al, 2002; Jones and Anderson, 2012; Herman et al, 2013). However, a small amount of accuracy is sacrificed for faster evaluation time as fewer force model evaluations are performed, as is the case with the relatively few number of low-/high-fidelity force model evaluations used in this paper. This is acceptable when position accuracies below the micron or even centimeter level are not needed, or even possible, due to imperfect force

model knowledge.

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. Determining the location of this knee automatically and reliably requires additional analysis due to its dependence on the orbit, force model, and number of nodes used.

3.4.3 Nodes

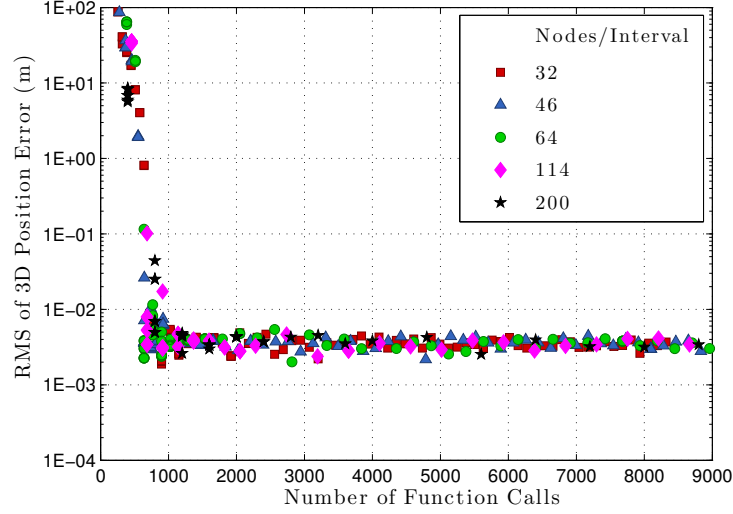
As mentioned previously, the number of nodes that are contained in each interval is tied to the bandlimit. Table 3.2 lists several node counts and their associated bandlimits. The displayed bandlimits are those that have been used to compute and store integration matrices, and are the only ones considered in this study.

Table 3.2: Number of nodes per interval and the corresponding bandlimits.

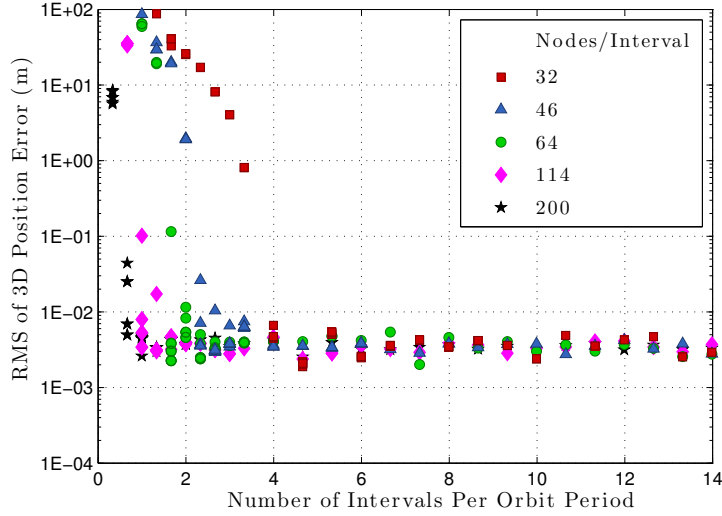
Nodes Per Interval	32	46	64	114	200
Bandlimit	5π	10π	17π	40π	81π

Figure 3.5(a) illustrates the impact that number of nodes has on the relationship between number of function calls and integration 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 the low-fidelity force model. This is mainly due to the high degree and order 70x70 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. 3.5(a) seems odd. However, this feature is actually a byproduct of the node accumulation ratio of the generalized Gaussian quadratures illustrated in Fig. 3.1(a). 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



(a)



(b)

Figure 3.5: RMS values of position errors for the LEO orbit using a range of number of nodes per interval. Each propagation has a duration of 3 orbit revolutions and used every combination of $\{1, 2, 3\}$ N_1 iterations and $\{1, 2, 3\}$ N_2 iterations.

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. 3.5(b). 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.

3.4.4 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)

$$w_i S_{ij} + w_j S_{ji} - w_i w_j = 0, \quad i, j = 1, \dots, M. \quad (3.22)$$

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, K , is computed by

$$\frac{V^2}{2} - \frac{\mu}{R} - U'(\mathbf{R}) = K = \mathbf{constant} \quad (3.23)$$

where μ is the gravitational parameter of the central body, R and V are the orbital radius and inertial velocity of the satellite, respectively, and $U'(\mathbf{R})$ is the gravitational potential of the Earth (without the point-mass contribution) (Tapley et al, 2004; Bond and Allman, 1996). Equation 3.23 is valid when the gravitational potential consists of zonal terms only. The inclusion of a time-varying gravity field, i.e., sectoral and tesseral terms, requires a slight modification to Eq. 3.23 (Bond and Allman, 1996).

The Jacobi Constant is an energy-like parameter that, in theory, remains constant over time when integrating a system involving a central gravity field. The purpose of a symplectic integrator is to enforce an approximate version of this property numerically since, otherwise, it is not maintained due to the finite precision of computation. The relative change in Jacobi Constant compared to its initial value is plotted in Fig. 3.6 for a 10-year propagation of the LEO orbit using BLC-IRK and

the explicit Runge-Kutta method DOPRI 8(7) (Prince and Dormand, 1981). BLC-IRK maintains a bounded Jacobi Constant over 10 years while the explicit Runge-Kutta method fails to maintain the Jacobi Constant over long integration times. It is known that non-symplectic integrators (e.g., all ERK methods (Sanz-Serna, 1988)) do not maintain a bounded energy, or Jacobi Constant, due to the accumulation of roundoff error. The symplectic property of the BLC-IRK method is of great benefit to long-term propagations where the accumulation of roundoff error is a problem. In particular, long-term asteroid propagation and debris field evolution with timescales on the order of tens to hundreds of years benefit from using symplectic integrators. Breiter and Métris (1999), Mikkola (1999), and Mikkola et al (2000) have also investigated symplectic schemes for use in space debris propagation and satellite tracking in Earth orbit.

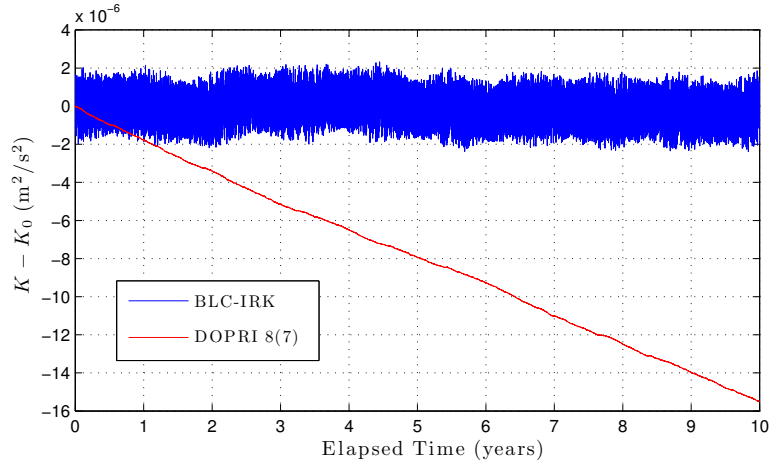


Figure 3.6: Change in Jacobi Constant during a 10-year GEO propagation using point-mass and zonals $J_2 - J_4$ only. BLC-IRK propagation performed using 4 intervals/orbit and 64 nodes/interval. DOPRI 8(7) propagation performed with a relative tolerance of 10^{-15} for step size control.

3.5 Performance Comparison

3.5.1 Orbit Propagation

In this section, we compare the propagation efficiency of BLC-IRK to commonly used integration methods for the three orbits given in Table 3.1. Three of the four integration methods are explicit Runge-Kutta schemes with step size control and the fourth is the 8th-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 (Fehlberg, 1968). The software package Satellite Tool Kit, by Analytical Graphics Inc., uses this as the default integrator (other options are available as well). The implementation of RKF 7(8) used in this study is not using local extrapolation (i.e., the 7th-order result is used as the solution).
- **Dormand & Prince 8(7) (DOPRI 8(7)13 or RK 8(7)13):** similar to the 13-stage RKF 7(8), but uses an 8th-order method for the solution and a 7th-order method for step size control (Prince and Dormand, 1981).
- **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 (Dormand and Prince, 1980). This integration scheme is available in MATLAB where it is known as ode45 (Shampine and Reichelt, 1997). 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 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 U.S. Space Surveillance Centers since the 1960's due to its highly efficient

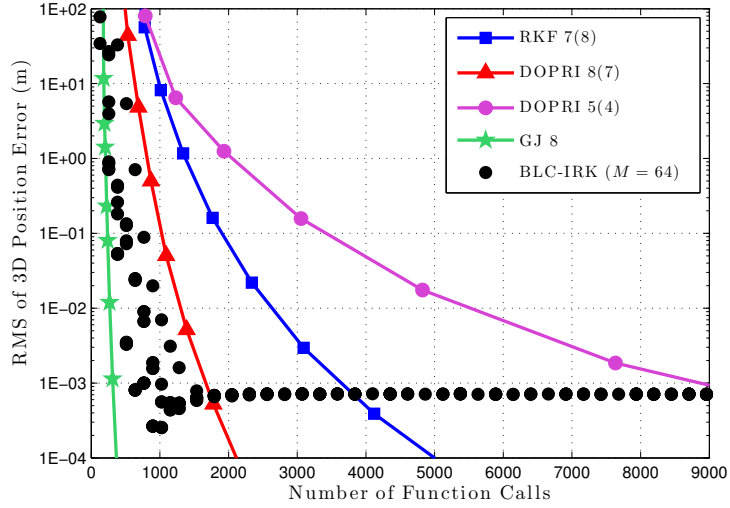
propagation of near-circular orbits (SPADOC Computation Center, 1982; Berry and Healy, 2004).

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 function calls for BLC-IRK is the number of high-fidelity force model evaluations only.

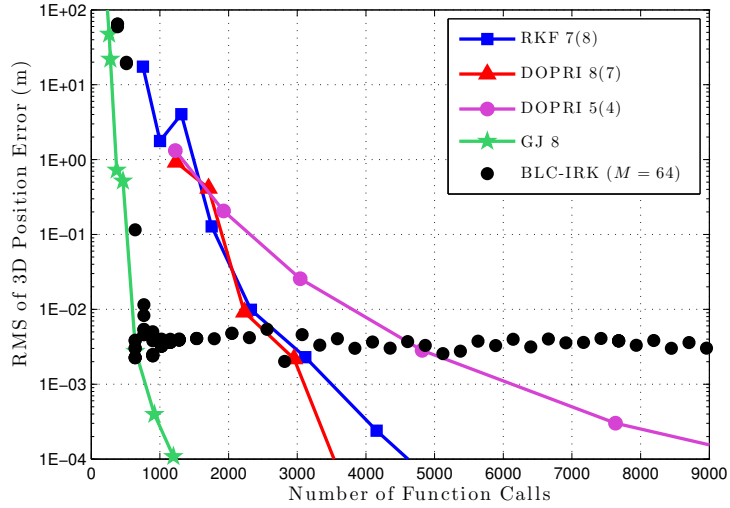
BLC-IRK is executed serially (without any parallelization) using 64 nodes per interval (M) and 2 function calls per node. Results for BLC-IRK are shown for propagations performed using $\{1, 2, \dots, 130\}$ intervals (N_I) and $\{1, 2, 3\}$ first set (N_1) and second set (N_2) iterations. Thus, the circular markers for BLC-IRK in the following figures are not connected by a line. Results for RKF 7(8), DOPRI 8(7), and DOPRI 5(4) are shown for propagations using relative tolerances ranging from 10^{-8} to 10^{-15} . Relative tolerance is used to adaptively control step size for these three embedded ERK methods. The implementation of step size control closely follows that of Dormand and Prince (1980). Results for GJ 8 were generated using a wide range of fixed step sizes while a 5-second time step is used as truth for all comparisons. Note that GJ 8 was forced to use only 1 iteration (i.e., force model evaluation) per step. Details on the interpolation of the reference trajectory for integrator comparison can be found in Sect. 3.4.1.

Figure 3.7(a) contains results for the GEO propagation (see Table 3.1 for orbital elements) and 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). BLC-IRK clearly outperforms all of the ERK methods in GEO, requiring many fewer function calls to achieve sub-centimeter accuracy. At the meter level, BLC-IRK closely matches the performance

of GJ 8, but requires about twice as many function calls at the centimeter to millimeter range. The performance of BLC-IRK in GEO may be improved with a better selection of low-fidelity force model. The low-fidelity model used here is just an example. As mentioned in Sect. 3.4.2, the higher accuracy floor of BLC-IRK is due to the fact that we are not iterating to a small relative tolerance, but are instead performing only a few force model evaluations. While this floor is greater than the floor for the other methods, it is still well within force model errors.



(a) GEO



(b) LEO

Figure 3.7: Comparison of RMS position errors over a 3-orbit GEO (a) and LEO (b) propagation.

Results of the LEO propagation, shown in Fig. 3.7(b), demonstrate a significantly different distribution of integration schemes than Fig. 3.7(a). Results for the ERK methods are now more clustered together and 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 spatial variation of perturbations, resulting in similar propagation accuracies. Note that BLC-IRK closely matches the efficiency of GJ 8 and is more efficient than the explicit methods.

Figure 3.8 demonstrates the differences in magnitude and temporal variation between the low- and high-fidelity force models for the LEO and GEO orbits. For both LEO and GEO, the difference in magnitude between the low- and high-fidelity models is several orders of magnitude smaller than that of the low-fidelity model itself. The fact that the low-fidelity model constitutes the bulk of the acceleration on a satellite is the reason for the benefit of using both low- and high-fidelity force models. Furthermore, the difference between the force models in LEO is greater than at GEO by about 1 order of magnitude. The smaller difference between the force models and the reduced spatial variation of acceleration in GEO allows the variable step methods to perform well. Since the Gauss-Jackson scheme uses a polynomial to generate an initial prediction of the solution at each time step, GJ 8 also performs well in the “smoothly” varying GEO regime.

As shown in Fig. 3.7(b), BLC-IRK has the ability to match Gauss-Jackson in LEO, largely due to the fact that the low-/high-fidelity force model scheme has a greater advantage in that region. BLC-IRK uses approximately the same number of force model evaluations as GJ 8 at centimeter to millimeter accuracy and outperforms the ERK methods significantly. Since the majority of objects in the space catalog reside in the LEO regime, this approach is very compelling. Furthermore, BLC-IRK can be massively parallelized, using a separate processor for each node in an interval.

We now consider a highly eccentric test case ($e = 0.74$), the Molniya orbit, shown in Fig. 3.9. With this orbit type the variable step size methods, particularly RKF 7(8) and DOPRI 8(7), show a vast improvement over the GJ 8 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 GJ 8 is forced to use a small step size for the duration of

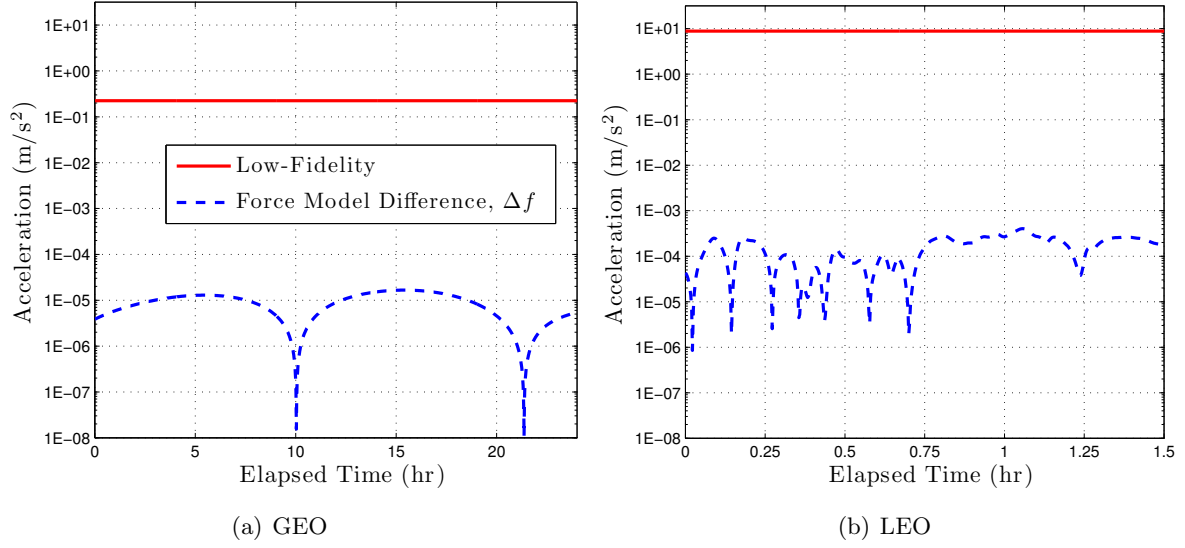


Figure 3.8: Comparison of low- and high-fidelity force models used in this study over 1 orbit period for GEO and LEO. The low-fidelity force model includes Earth point-mass and a 3×3 gravity field. The high-fidelity force model includes Earth point-mass, a 70×70 gravity field, and third-body gravitational forces from the Sun and Moon.

the propagation in order to deal with the high dynamics at perigee. Since BLC-IRK is currently implemented as a fixed-step integrator, it performs similarly to GJ 8.

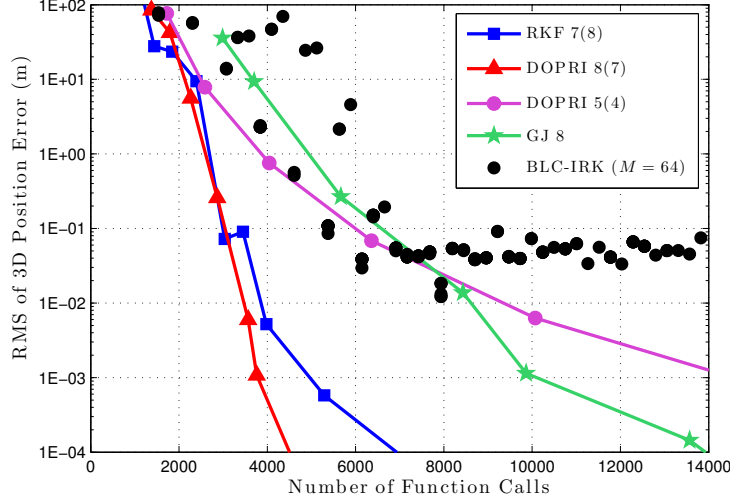


Figure 3.9: Comparison of RMS position errors over a 3-orbit Molniya propagation.

Table 3.3 provides a quantitative summary of the performance of each integration scheme. Additional entries are given for a 2-processor parallelized and an ideally parallelized implementation of BLC-IRK to demonstrate the ability of this IRK method when parallelization is taken advantage of. Since we are using 64 nodes per interval, ideally parallelized means the use of 64 processors and no communication overhead. This number of processors is easily taken care of if GPUs are utilized. Note that the term “ideal” is used because an actual parallel implementation would inherently contain added computation time due to communication/data transfer as well as memory management. This extra overhead could be quite large and is especially sensitive to implementation. Table 3.3 demonstrates that BLC-IRK outperforms Gauss-Jackson in all scenarios if only 2 processors are used. Using BLC-IRK with 2 processors even outperforms the explicit variable-step methods for the highly-elliptic Molniya orbit. A further improvement in efficiency can be gained by ideally parallelizing the implementation of BLC-IRK.

This paper compares BLC-IRK with common ERK methods and the multistep GJ 8 scheme, but does not look at other IRK methods. A recent study by Herman et al (2013), however, directly compares BLC-IRK with a fixed-step implementation of GL-IRK. Herman et al (2013) demonstrates that BLC-IRK always performs as well or better than GL-IRK (i.e., requiring fewer function calls for the same accuracy). The study ensures a fair comparison by operating both schemes in fixed-

Table 3.3: Performance summary of integration methods over three orbit periods. 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.

Orbit	Method	Function Calls ($< 1\text{m}$ Error)	Function Calls ($< 1\text{cm}$ 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 (3cm)
	BLC-IRK (2 processors)	2304	3070 (3cm)
	BLC-IRK (ideal parallel)	72	96 (3cm)

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 3.1 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 van der Houwen and Sommeijer (1990) with a GL-IRK scheme. Jones (2012) demonstrates that the variable-step algorithm, dubbed VGL-*s*, 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-*s*, 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).

3.5.2 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 $\{\tau_m\}_{m=1}^M$ to an arbitrary time τ . For clarity, we present the necessary equations for the case where the quadrature nodes τ_m 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 files necessary for executing the dense output algorithm are provided online and described in Appendix 3.7.1.

The input data for interpolation is given as the values $y(\tau_m)$ at the nodes $\tau_m \in [-1, 1]$,

$m = 1, 2, \dots, M$. The function $y(\tau)$ is then interpolated via

$$y(\tau) = \sum_{k=1}^M b_k \Psi_k(\tau), \quad (3.24)$$

where the approximate PSWFs $\Psi_k(\tau)$ are defined in Eq. 3.18 and the coefficients b_k satisfy the condition

$$y(\tau_m) = \sum_{k=1}^M b_k \Psi_k(\tau_m), \quad m = 1, 2, \dots, M, \quad (3.25)$$

so that

$$b_k = \sum_{m=1}^M B_{km} y(\tau_m), \quad (3.26)$$

where the matrix $\{B_{km}\}_{k,m=1}^M$ is the inverse of $\{\Psi_k(\tau_m)\}_{m,k=1}^M$. From Eq. 3.18, we also have

$$y(\tau) = \sum_{l=1}^M a_l e^{ic\tau_l \tau}, \quad (3.27)$$

where

$$a_l = \sum_{k=1}^M A_{lk} b_k, \quad A_{lk} = \frac{w_l}{\eta_k} \Psi_k(\tau_l). \quad (3.28)$$

In the event that the matrix $\Psi_k(\tau_m)$ may be ill-conditioned for large M , we note that this matrix can be pre-computed with extended precision (e.g., using Mathematica), and then tabulated. Computing and storing matrix $\Psi_k(\tau_m)$ and its inverse in advance, we compute the coefficients b_k and a_k from the values $\{y(\tau_m)\}_{m=1}^M$ by applying these matrices. Matrix $\Psi_k(\tau_m)$ and its inverse may have to be computed and applied with extra precision to avoid losing accurate digits.

We use Eq. 3.27 to evaluate the function y at arbitrary points in $[-1, 1]$. The advantage of Eq. 3.27 is that we can use Unequally Spaced Fast Fourier Transform (USFFT) (Dutt and Rokhlin, 1993; Beylkin, 1995) to compute values of y at N points in $[-1, 1]$ in $\mathcal{O}(N \log N) + \mathcal{O}(M)$ operations. However, if M is relatively small (e.g., $M \leq 64$), the direct evaluation via

$$y(\tau) = \sum_{l=1}^M \mathcal{R}e(a_l) \cos(c\tau_l \tau) - \mathcal{I}m(a_l) \sin(c\tau_l \tau), \quad (3.29)$$

requires $\mathcal{O}(NM)$ operations and may turn out to be faster.

Figure 3.10 displays the interpolation accuracy achieved for a GEO propagation performed using BLC-IRK and the collocation algorithm. A 6-hour segment of the 3-orbit propagation is

shown, revealing each of the 64 nodes used in this interval. Note that the y-axis has units of millimeters. The error grows over time because the actual BLC-IRK integrated trajectory is being interpolated and compared to the truth trajectory. Hence, the error already contained in the orbit propagation is still present. The BLC-IRK interpolation (blue line) yields a very smooth and continuous solution with errors similar to those of the nodes. This result highlights the benefit of collocation techniques.

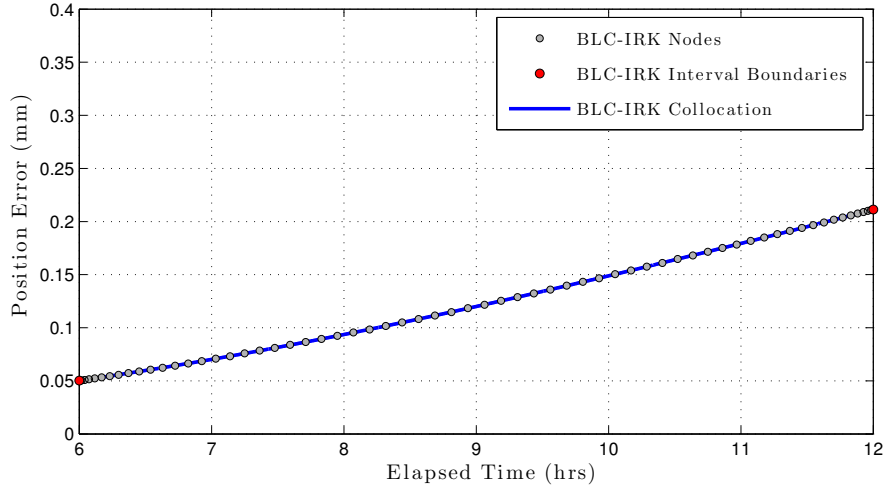
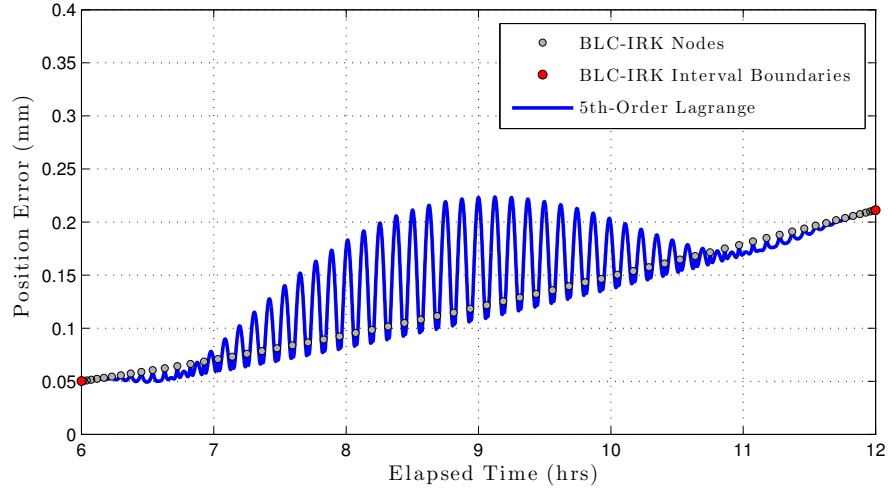


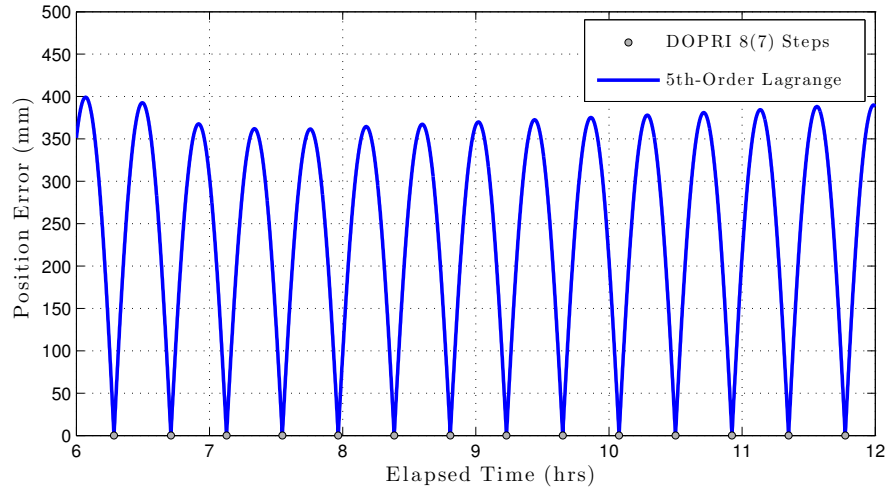
Figure 3.10: BLC-IRK collocation interpolation error for a GEO propagation. BLC-IRK propagation performed using 4 intervals/orbit and 64 nodes/interval. Interpolation is performed every 5 seconds. Note that the plotted error is due to both interpolation and integration.

For comparison, Fig. 3.11 demonstrates the accuracy of Lagrange interpolation used on orbit propagations generated by BLC-IRK and DOPRI 8(7). Lagrange interpolation was chosen for comparison due to the easy implementation and long history of use of the polynomial-based scheme. During the same 6-hour time span shown in Fig. 3.10, the 5th-order Lagrange interpolation yields slightly larger errors than the collocation algorithm for BLC-IRK. However, the interpolation error in Fig. 3.11(a) has a maximum of only 0.1 mm. Note that this is only a 5th-order implementation of Lagrange and that a higher order Lagrange method (e.g., 7 or 9) yields errors similar to that of BLC-IRK collocation.

Figure 3.11(b) demonstrates the degraded accuracy achieved by interpolating a DOPRI 8(7) trajectory using a 5th-order Lagrange scheme as compared to BLC-IRK. Polynomial-based interpo-



(a) BLC-IRK Trajectory



(b) DOPRI 8(7) Trajectory

Figure 3.11: 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.

lation performs much better when using the higher density BLC-IRK nodes rather than the sparse DOPRI 8(7) steps.

It should also be noted that explicit Runge-Kutta schemes can have dense output capabilities as well. Several researchers have developed dense output schemes for the 5th- and 8th-order methods used in this study. Two dense output schemes have been developed for the DOPRI 5(4) method. Dormand and Prince (1986) develop a 4th-order “free” interpolant (meaning additional function evaluations are not required), which is also discussed in Hairer et al (1993). Calvo et al

(1990) provides a 5th-order interpolant for the DOPRI 5(4) method, requiring 2 additional stage evaluations per step. Additionally, Bogacki and Shampine (1990) have developed a 5th-order “free” interpolant for the higher-order DOPRI 8(7) scheme. Tsitouras (2007) discusses several Runge-Kutta interpolants that have been developed and presents a new high-order interpolation method for the RK 9(8) developed by Tsitouras (2001). Other options include Runge-Kutta Triples which have dense output capability, including embedded ERK and Runge-Kutta-Nyström schemes (Montenbruck and Gill, 2000).

While a few of the dense output options for ERK methods are “free”, they are of lower order than the integration scheme itself. Collocation methods allow for the state to be computed at anytime on the trajectory for “free”, utilizing a continuous solution that is coupled to the integration scheme. Even if another interpolation strategy (e.g., Lagrange) is desired, the increased density of solutions due to the collocation nodes provides a more accurate interpolated solution, as seen in Fig. 3.11.

3.6 Conclusions and Future Work

This paper describes a new numerical integration scheme, bandlimited collocation implicit Runge-Kutta (BLC-IRK), for orbit propagation and outlined its implementation towards propagating orbits with perturbing forces. As a symplectic and *A*-stable IRK method, BLC-IRK is of particular interest because it is parallelizable. In addition, the generalized Gaussian quadratures for bandlimited functions on which 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 paral-

lization). 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.

The current implementation of the BLC-IRK scheme requires several user-defined tuning parameters. While these parameters can affect the efficiency and accuracy of the resulting orbit propagation, this paper gives a rough idea of satisfactory choices for these parameters in several orbit regimes. Future work will aim to: (1) determine optimal low- and high-fidelity force models for Earth orbiting objects, (2) develop a step size control algorithm for BLC-IRK, and (3) investigate its use in boundary value problems. The first two items will allow a more autonomous and efficient implementation of BLC-IRK while the third allows for its use in trajectory optimization. It is

also important to note that further systematic studies of the integration schemes compared in this paper and other recently developed integrators would aid in better defining the relative merits and capabilities of each scheme. We include data files online, containing all quadrature data, necessary for implementing BLC-IRK integration and interpolation. Details on the provided data can be found in Appendix 3.7.1.

3.7 Appendix

3.7.1 BLC-IRK Data Files

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

- `nodesM.txt`: Contains the quadrature nodes $\{\tau_k\}_{k=1}^M$ for the interval $[-1, 1]$.
- `weightsM.txt`: Contains the quadrature weights $\{w_k\}_{k=1}^M$ for the interval $[-1, 1]$.
- `integrationMatrixM.txt`: Contains the elements of the integration matrix S (stored row-wise) with respect to the interval $[-1, 1]$.
- `BmatrixM.txt`: Contains the elements of the matrix B (stored row-wise), Eq. (3.26), with respect to the interval $[-1, 1]$.
- `AmatrixRealPartM.txt`: Contains the elements of the real part of the matrix A (stored row-wise), Eq. (3.28) and (3.29), with respect to the interval $[-1, 1]$.
- `AmatrixImagPartM.txt`: Contains the elements of the imaginary part of the matrix A (stored row-wise), Eq. (3.28) and (3.29), with respect to the interval $[-1, 1]$.

These data files allow the reader to implement the BLC-IRK scheme for numerical integration and to perform the interpolation described in Sect. 3.5.2. The files can be downloaded in the online supplement from Springer.

3.8 Acknowledgements

This research was made possible with Government support under and awarded by DoD, Air Force Office of Scientific Research, National Defense Science and Engineering Graduate (NDSEG) Fellowship, 32 CFR 168a. B. Jones' contribution to this work was funded by Air Force Research Laboratories contract FA9453-08-C-0165. The research of G. Beylkin was partially supported by AFOSR grants FA9550-07-1-0135 and STTR Phase I grant 1118-001-01.

Chapter 4

ITRS/GCRS Implementation for Coordinate Transformations

Journal Version:

Bradley, B.K., Sibois, A., and Axelrad, P., "Influence of ITRS/GCRS Implementation for Astrodynamics: Coordinate Transformations," in press at Advances in Space Research, 2015.

Extension of Conference Paper:

Bradley, B.K., Vallado, D.A., Sibois, A., and Axelrad, P., "Earth Orientation Parameter Considerations for Precise Spacecraft Operations," AAS/AIAA Astrodynamics Specialist Conference, AAS 11-529, Girdwood, AK, July 31–August 4, 2011.

4.1 Abstract

Converting between the Geocentric Celestial Reference System (GCRS) and International Terrestrial Reference System (ITRS) is necessary for many applications in astrodynamics, such as orbit determination and analyzing geoscience data from satellite missions. The implementation of this frame transformation and the manner in which the Earth orientation parameters (EOPs) are used have a notable impact on station coordinates and satellite positions. After briefly reviewing the various theories and their mathematical description, we investigate the impact of EOP interpolation methods, ocean tide corrections, precession–nutation simplifications, and Julian date handling on the ITRS/GCRS coordinate transformation. Estimates of the impact on position concern a range of altitudes, from the Earth’s surface to geosynchronous orbit (GEO), and apply to a wide array of astrodynamics applications. We demonstrate that EOP interpolation methods and ocean tide corrections impact the ITRS/GCRS transformation between 5 cm and 20 cm on the surface of the Earth and at the Global Positioning System (GPS) altitude, respectively. We conclude with a

summary of recommendations on EOP usage and bias–precession–nutation model implementations for achieving a wide range of transformation accuracies at several altitudes. This comprehensive set of recommendations allows astrodynamacists, flight software engineers, and Earth scientists to make informed decisions when choosing the best implementation for their application, balancing accuracy and computational complexity.

4.2 Introduction

Satellite missions require orbit propagation and orbit determination to carry out their objectives, both of which encompass a wide range of accuracy and computational requirements. Real-time, on-board orbit determination requires low to moderate accuracy with low computational overhead and memory usage, while post processing orbit determination, e.g., for Earth science missions, requires the highest accuracy possible, with little consideration for computational cost. The numerical models and precise position solutions needed for orbit propagation, determination, and analysis of science data require accurate frame transformations, particularly between the Geocentric Celestial Reference System (GCRS) and the International Terrestrial Reference System (ITRS). This transformation includes precession, nutation, polar motion, and Earth’s proper rotation. The recommended procedures and models needed to transform between the ITRS and GCRS are maintained by the International Earth Rotation Service (IERS) and International Astronomical Union (IAU) and are published in their conventions/technical notes (McCarthy, 1992, 1996; McCarthy and Petit, 2004; Petit and Luzum, 2010). Unfortunately, the precise implementation of the ITRS/GCRS transformation is computationally expensive and is not necessary for every application.

Precise computation of the bias–precession–nutation parameters is the most demanding on computation time and memory, but the use of Earth orientation parameters (EOPs) and their ocean tide corrections adds to the complexity of the procedure. EOPs are crucial for correctly executing these frame transformations, affecting parameters such as ground station coordinates and satellite positions on the order of centimeters up to hundreds of meters. EOPs are published by the IERS once per day in their Bulletin A, effective at 0h UTC of that day. For maximum accuracy, these

daily EOP values must be interpolated to the desired time before they are incorporated into the ITRS/GCRS transformation. In literature to date, however, recommended interpolation methods to achieve specific accuracies are not mentioned. Additionally, the published values of the pole coordinates, UT1 time difference, and length-of-day are “tide-free”, meaning that users requiring high-precision need to apply additional corrections for diurnal and semi-diurnal ocean tides and libration effects after the daily values have been interpolated (Petit and Luzum, 2010). This fact is fairly unpublicized within astrodynamics literature. Nonetheless, Vallado (2013), Montenbruck and Gill (2000), Vallado and Kelso (2005) and Vallado and Kelso (2013) describe the general use of EOPs clearly. Additionally, Stamatakis et al (2008) demonstrate the effect of EOPs on satellite pointing accuracy requirements.

This paper investigates both precise and simplified approaches to the ITRS/GCRS transformation. We begin with an overview of the IAU-supported transformation methods and highlight several options for implementing the Celestial Intermediate Origin (CIO) scheme. The coordinate transformation effects of several commonly-used interpolation methods applied to EOPs, ocean tide corrections, and ignoring certain EOPs entirely are presented next. We follow with an analysis of several simplifications to the full ITRS/GCRS transformation, namely in the computation of bias-precession-nutation parameters, and the less common topic of Julian date formatting. We conclude with a set of recommendations for ITRS/GCRS transformation strategies to achieve a wide range of accuracies while balancing computational burdens for several altitude regimes. The resulting recommendations will allow astrodynamicists, flight software engineers, and Earth scientists to implement the best transformation procedure for their needs.

4.3 Mathematical Overview

This section gives an overview of the ITRS/GCRS transformation procedures, EOP behavior and usage, and EOP ocean tide corrections. A summary of several transformation procedures is shown in Fig. 4.5 and contains methods currently supported by the IERS 2010 Conventions (Petit and Luzum, 2010; SOFA, 2014).

4.3.1 ITRS/GCRS Transformation

Over the years, several recommendations have been published by the IERS on the precise transformation between the ITRS and GCRS reference systems (McCarthy, 1992, 1996; McCarthy and Petit, 2004; Petit and Luzum, 2010). Effective January 1, 2009 the IERS recommends the use of the new IAU models for nutation and precession to perform this ITRS/GCRS transformation (Petit and Luzum, 2010; SOFA, 2014; Hilton et al, 2006; Wallace and Capitaine, 2006; Capitaine and Wallace, 2008). The new model, as a whole, is called IAU 2000A/2006. This includes the IAU 2000A_{R06} nutation theory (Mathews et al, 2002; Buffett et al, 2002; Herring et al, 2002; Capitaine and Wallace, 2006) and the IAU 2006 precession theory (Capitaine et al, 2003a; Capitaine and Wallace, 2006; Fukushima, 2003). The inclusion of the “R06” subscript indicates that we refer to the 2000A nutation theory that has been updated to be compliant with the 2006 precession model. Several methods for implementing this new model have been developed over the past few years, but are all within the uncertainty of the IAU model itself (SOFA, 2014; Capitaine and Wallace, 2008; Hilton et al, 2006; Vallado, 2013; Coppola et al, 2009). The methods can be separated into two categories, the classical, equinox-based method and the modern CIO-based approach. Figure 4.5 depicts the procedure of one equinox-based approach and several CIO-based transformation options. SOFA (2014), Wallace and Capitaine (2006), and Vallado (2013) outline both equinox- and CIO-based procedures.

The CIO-based method is built on the direction cosine components of the CIP in the GCRS (X and Y) and the CIO locator (s). This method is preferred when the goal is the transformation between the terrestrial and celestial frames, because it requires only three angles instead of four and the ecliptic and equinox are not needed (Wallace and Capitaine, 2006). The benefits of the CIO-based method include the following:

- (1) The bias-precession-nutation matrix is formed directly, thus reducing potential errors.
- (2) The IERS will publish the dX and dY celestial pole offset values far into the future, while publication of the $\delta\Delta\psi_{2000A}$ and $\delta\Delta\epsilon_{2000A}$ corrections are likely to be discontinued (Kaplan,

2005).

- (3) X , Y , and s can be computed a priori, stored in a data table, and interpolated as needed (Coppola et al, 2009). Other methods would require interpolating four values instead of only three.

The transformation between the ITRS and GCRS consists of matrices for bias–precession–nutation (\mathbf{C}), proper motion (\mathbf{R}), and polar motion (\mathbf{W}). A position vector \vec{r} is transformed from the ITRS to the GCRS such that (Petit and Luzum, 2010)

$$\mathbf{C} = \begin{bmatrix} 1 - aX^2 & -aXY & X \\ -aXY & 1 - aY^2 & Y \\ -X & -Y & 1 - a(X^2 + Y^2) \end{bmatrix} \mathbf{R}_3(s) \quad (4.1)$$

$$\mathbf{R} = \mathbf{R}_3(-\theta_{ERA}) \quad (4.2)$$

$$\mathbf{W} = \mathbf{R}_3(-s')\mathbf{R}_2(x_p)\mathbf{R}_1(y_p) \quad (4.3)$$

$$\vec{r}_{\text{GCRS}} = \mathbf{C} \mathbf{R} \mathbf{W} \vec{r}_{\text{ITRS}} = \mathbf{T} \vec{r}_{\text{ITRS}}, \quad (4.4)$$

where $Z = \sqrt{1 - X^2 - Y^2}$ and $a = 1/(1 + Z)$. The TIO locator (s') varies linearly with respect to Julian centuries of Terrestrial Time (TT), $s' = -0.000047'' T_{\text{TT}}$, for use in polar motion.¹

A simplified equation for transforming a velocity vector \vec{v} between the ITRS and GCRS (that is sufficient for most applications) is (Vallado, 2013)

$$\vec{v}_{\text{GCRS}} = \mathbf{C} \mathbf{R} \left\{ \mathbf{W} \vec{v}_{\text{ITRS}} + \vec{\omega}_{\oplus} \times \mathbf{W} \vec{r}_{\text{ITRS}} \right\}, \quad (4.5)$$

where $\vec{\omega}_{\oplus} = \begin{bmatrix} 0 & 0 & \omega_{\oplus} \end{bmatrix}^T$ is the rotation vector of the Earth in the Terrestrial Intermediate Coordinate System (TIRS) and the instantaneous rotation rate of the Earth is²

$$\omega_{\oplus} = 7.2921151467064 \times 10^{-5} \left\{ 1 - \frac{\text{LOD}}{86400} \right\} \text{ rad/s}. \quad (4.6)$$

¹ $T_{\text{TT}} = (\text{JD}_{\text{TT}} - 2451545)/36525$ where JD_{TT} is the Julian date of TT.

² This definition is based on Aoki et al (1982), which defines the rotational period of the Earth as 86164.09890369732 seconds of UT1. An overview of the computation of rotation rate and definitions of UT1 and LOD can be found here:

<http://hpiers.obspm.fr/eop-pc/earthor/ut1lod/UT1.html>

The largest computational burden for CIO-based transformation is the computation of X, Y, s . Several methods for computing these parameters are available as shown in Fig. 4.5. The series method consists of around 4,000 coefficients to compute X and Y directly and the parameter $s + XY/2$ (Capitaine and Wallace, 2006; Capitaine et al, 2003b; Capitaine and Wallace, 2008; Petit and Luzum, 2010). Each series contains a polynomial and trigonometric portion

$$q = q_0 + q_1 T_{\text{TT}} + q_2 T_{\text{TT}}^2 + q_3 T_{\text{TT}}^3 + q_4 T_{\text{TT}}^4 + q_5 T_{\text{TT}}^5 + \sum_i \sum_{j=0}^4 [(a_{s,j})_i \sin(\text{ARG}) + (a_{c,j})_i \cos(\text{ARG})] T_{\text{TT}}^j, \quad (4.7)$$

where q stands for X, Y , or $s + XY/2$ and ARG stands for combinations of the luni-solar and planetary parts of the fundamental arguments. The time parameter T_{TT} is Julian centuries of Terrestrial Time past J2000.³ The large number of coefficients and trigonometric evaluations make the series evaluation burdensome, both on computation time and memory. However, the series can be truncated to speed up computation time at the loss of accuracy, as is shown by Wallace and Capitaine (2006) and Capitaine and Wallace (2008). These papers compare a few of these truncated series to the full accuracy models, demonstrating the ability of these approximations and highlighting the accuracy ranges they are useful for.

A second method is the “Full Theory”, which uses the Fukushima-Williams angles for precession ($\bar{\gamma}, \bar{\phi}, \bar{\psi}, \epsilon_A$) and the IAU 2000A_{R06} nutation theory ($\Delta\psi, \Delta\epsilon$) to compute X and Y (Wallace and Capitaine, 2006). The CIO locator must still be computed using the series for the quantity $s + XY/2$. This method has a high computational load as well, due to the series for $s + XY/2$ and the several thousand terms for nutation. More details on the IAU models and implementation processes can be found in SOFA (2014), Petit and Luzum (2010), Wallace and Capitaine (2006).

The third option for computing X, Y , and s is that of tabulation and interpolation. The “Full Theory” or series methods may be used to generate X, Y , and s values at 0h each day and stored in a data table. The user can then interpolate the values to any desired time. This procedure maintains nearly the full accuracy of the “Full Theory” and series methods while greatly reducing

³ Formally, the time argument is Barycentric Dynamical Time (TDB), but TT is used in practice because the resulting difference is below the precision of the IAU models themselves.

computation time. The method is developed and analyzed in detail in Coppola et al (2009) and used in this paper as one of the default recommended procedures.

4.3.2 Earth Orientation Parameters

EOPs are used to correct for errors in the IAU models and consist of the following:⁴

- Pole Coordinates (x_p, y_p): coordinates of the Celestial Intermediate Pole (CIP) with respect to the IERS Reference Pole (IRP) in the International Terrestrial Reference System (ITRS). The IRP is the location of the agreed upon terrestrial pole while the CIP is the approximate instantaneous rotation axis of the Earth.
- Celestial Pole Offsets (dX, dY): observed corrections to the conventional celestial pole. The conventional celestial pole position is defined by the IAU Precession and Nutation models.
- UT1 Time Difference (ΔUT1): the offset of Universal Time (UT1) from Universal Coordinated Time (UTC), where $\Delta\text{UT1} = \text{UT1} - \text{UTC}$.
- Length of Day (LOD): time difference between the observed duration of a mean solar day and 86,400 SI seconds. LOD is the time derivative of ΔUT1 and has units of seconds per day. Omitting ‘per day’ is common when discussing LOD and is left out in this paper.
- Atomic Time Offset (ΔAT): time difference between International Atomic Time (TAI) and Universal Coordinated Time (UTC), where $\Delta\text{AT} = \text{TAI} - \text{UTC}$ (i.e., leap seconds).⁵

The list below identifies which EOP contributes to each of the components needed for the ITRS/GCRS frame transformation.

- $X = X + dX, Y = Y + dY \rightarrow \mathbf{C}$
- $\text{UT1} = \text{UTC} + \Delta\text{UT1} \rightarrow \mathbf{R}$

⁴ Note that ΔUT1 and LOD are not independent parameters. LOD is the time derivative of ΔUT1 . Additionally, the ΔUT1 time series contains leap second jumps whenever ΔAT is updated.

⁵ Although the Atomic Time offset (leap seconds) is not an EOP, it is included in this list because it is commonly incorporated into tabulated EOP data files.

- $(x_p, y_p) \rightarrow \mathbf{W}$
- $\text{LOD} \rightarrow \omega_{\oplus}$

The EOPs are tabulated at 0h UTC of each day and published by several sources, including the IERS⁶ (Bizouard and Gambis, 2009), the National Geospatial-Intelligence Agency (NGA)⁷, and the United States Naval Observatory (USNO).⁸ CelesTrak has compiled EOPs from these sources to create a single file containing EOP data from 1962 to predictions of 1 year into the future.⁹ The file maintained by CelesTrak is used by Analytical Graphics Inc. Systems Tool Kit (STK) and is used for the EOP studies presented in this paper, unless otherwise mentioned. See Vallado and Kelso (2005, 2013, 2015) for more information on the differences between each agencies' EOP product and how the CelesTrak file is formed. Figure 4.6 illustrates the behavior of the daily tabulated EOPs to aid our analysis of EOP interpolation. The length-of-day and celestial pole offsets contain the most dramatic variation over time, while the pole coordinates and time offset parameters exhibit much smoother variations.

4.3.3 EOP Ocean Tide Corrections

Because sub-daily variations in the EOPs are not part of the values posted by the IERS and to be in full compliance with the models of the IERS Conventions (2010), the tabulated pole coordinates, UT1 time difference, and length-of-day values must first be interpolated to the appropriate time and then corrected for ocean tide and libration effects.

The libration corrections, e.g., $(\Delta x, \Delta y)_{\text{libration}}$, are not included in this investigation for two reasons: 1) they are an order of magnitude smaller than the ocean tide corrections and 2) the model has, to the best of the authors' knowledge, not been verified by observations as of yet. These libration corrections consist of diurnal and semi-diurnal nutations due to the tri-axiality of the Earth. The ocean tide corrections, denoted $(\Delta x, \Delta y)_{\text{ocean}}$, include diurnal and semi-diurnal

⁶ <http://www.iers.org/iers/EN/DataProducts/EarthOrientationData/eop.html>

⁷ <http://earth-info.nga.mil/GandG/sathtml/eopp.html>

⁸ <http://www.usno.navy.mil/USNO/earth-orientation/eo-products>

⁹ <http://www.celestrak.com/SpaceData/>

variations caused by ocean tides for polar motion, UT1 time difference, and length-of-day such that

$$(x_p, y_p) = (x_p, y_p)_{\text{IERS}} + (\Delta x, \Delta y)_{\text{ocean}}, \quad (4.8a)$$

$$\Delta\text{UT1} = \Delta\text{UT1}_{\text{IERS}} + \Delta\text{UT1}_{\text{ocean}}, \quad (4.8b)$$

$$\text{LOD} = \text{LOD}_{\text{IERS}} + \Delta\text{LOD}_{\text{ocean}}, \quad (4.8c)$$

where the subscript “IERS” indicates the daily published EOP values that have been interpolated to the desired time.

The computation of the ocean tide corrections is relatively simple and is purely a function of T_{TT} . Petit and Luzum (2010) provide the algorithm for computing the ocean tide corrections using a table of amplitudes and arguments for 71 tidal components based on Ray et al (1994). This investigation uses a MATLAB version of the `INTERP.F` Fortran source code¹⁰ for the computation of ocean tide corrections.

Figure 4.1 depicts examples of the behavior and magnitude of ocean tide corrections for pole coordinates, ΔUT1 , and LOD . The celestial pole offsets (dX, dY) are **observed** corrections to the precession–nutation theory and do not require additional corrections. The time window in Fig. 4.1 has been reduced to one day to clearly show the effect of the diurnal and semi-diurnal ocean tide corrections. The corrections to ΔUT1 have a root-mean-square (RMS) amplitude of about $25 \mu\text{s}$ and reach up to about $73 \mu\text{s}$.

¹⁰ `INTERP.F` available here: <ftp://hpiers.obspm.fr/eop-pc/models/interp.f>

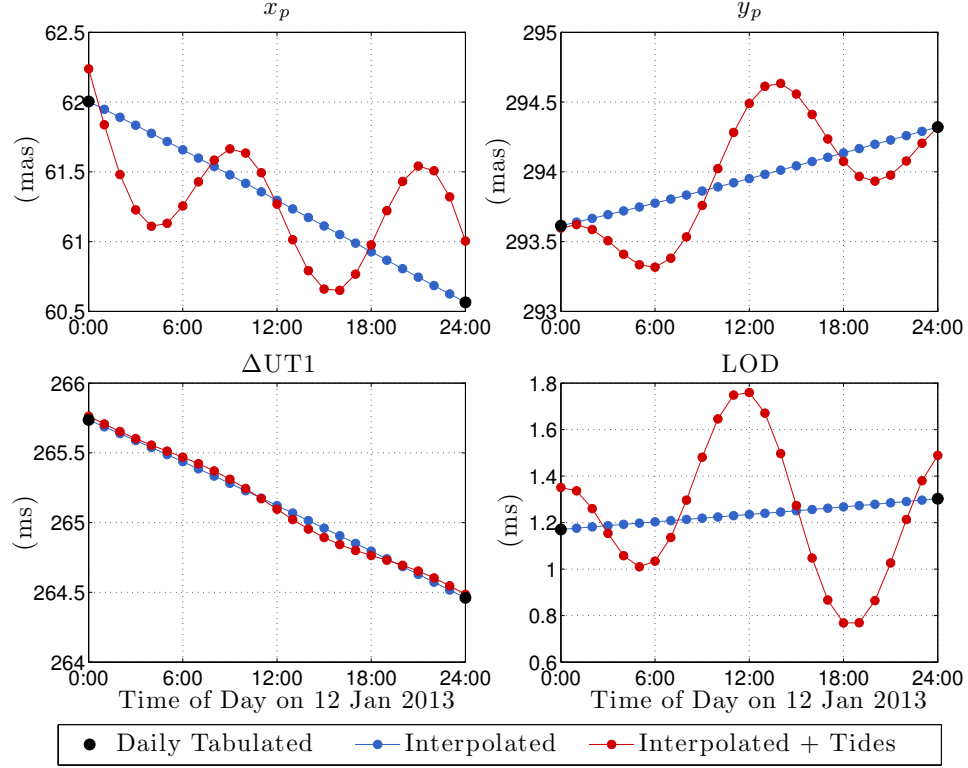


Figure 4.1: Ocean tide corrections for applicable EOPs on 12 January 2013.

4.3.4 Regularization of Δ UT1 and LOD

We have discussed the need to interpolate EOPs to the desired time and to apply ocean tide corrections for x_p , y_p , Δ UT1, and LOD for high-accuracy applications. An additional step may be included for Δ UT1 and LOD, however, to improve the interpolation process. Prior to the interpolation of Δ UT1 and LOD, the tabulated values should be smoothed through regularization to enhance the interpolation accuracy. Regularization is the removal of zonal tidal variations with frequencies ranging from 5 days to 18.6 years. Regularized UT1 is typically denoted UT1R.

After regularization and interpolation, the zonal tide value should be added back at the time of interpolation. At this point, the ocean tide corrections should be computed and added to the interpolated values, as discussed in the previous section. The zonal tide effects are based on models recommended by the IERS Conventions. Table 8.1 in Petit and Luzum (2010) provides the coefficients for one of the zonal tide models implemented in the software, which contains 62 tide

terms. Fortran source code, `RG_ZONT2.F`, is also available for computing the zonal tides for ΔUT1 and LOD.¹¹ The impact of regularization is evaluated in Sect. 4.4.2.

4.4 Position Transformation Analysis

This section investigates the effect on position transformations between the ITRS and GCRS due to EOP interpolation, ocean tide corrections, precession–nutation simplifications, and time handling. The coordinate transformation is evaluated in four altitude regimes from the Earth’s surface to GEO over a 15-year period, as outlined in Table 4.1. The duration and time interval are selected to ensure that the variability of EOPs and precession–nutation is captured. A grid of ITRS positions created for each altitude is rotated to the GCRS. The four altitudes selected are typical of ground stations, Earth science satellites, GPS satellites, and communication satellites. This analysis illustrates which EOPs have the greatest impact on frame transformations and how the interpolation method impacts the results. The “truth” position transformation is generated using the Full Theory transformation method with all EOPs interpolated using a cubic spline. Ocean tide corrections are omitted from the truth transformation in order to isolate interpolation errors from the effect of the tide corrections. Cubic spline interpolation is chosen for the reference because it yields a smooth and continuous function, which closely follows that of real geophysical processes (except during earthquakes).

Table 4.1: Summary of position transformation analysis setup.

Parameter	Value
Surface altitude	0 km
LEO altitude	800 km
GPS altitude	20,200 km
GEO altitude	35,786 km
Start Time	1 Jan 2000
Stop Time	15 Mar 2015
Time Step	0.1 days
Latitude Step	5 deg

¹¹ `RG_ZONT2.F` available here: <http://tai.bipm.org/iers/convupdt/convupdt.c8.html>

4.4.1 Ignoring EOPs

To assess the contribution of each EOP to the coordinate transformation, a comparison is made between the “truth” rotation and simplified versions. Each EOP is individually set to zero, followed by all EOPs, to investigate the impact each parameter has on the ITRS/GCRS frame transformation. Finally, an extreme simplification, often inappropriately used, is shown. This simplification is the sole use of Greenwich Mean Sidereal Time (GMST) to transform between the ITRS and GCRS, thus ignoring all EOPs and precession–nutation. The use of GMST is included here, because it is a commonly used and easy to implement scheme for approximating the frame transformation.

Figure 4.2 and Table 4.2 present results on the position error over time and the maximum errors resulting from the exclusion of each EOP. The ΔUT1 parameter has the largest effect, contributing to 99.6% of the error when compared to ignoring EOPs entirely. Excluding ΔUT1 can lead to a position error of approximately 310 m on the surface of Earth up to 2.1 km in GEO. The introduction of leap seconds are chosen to limit the parameter ΔUT1 to ± 0.9 seconds. This means that the effect of ΔUT1 is greatest near leap seconds. Though the impact of polar motion is smaller than the UT1 time difference, position errors can reach maximum values between 18 m and 115 m on the Earth’s surface and at GEO altitude, respectively. The celestial pole offsets contribute the smallest amount to the frame transformation with position differences on the order of centimeters.

The use of GMST-only results in tens of kilometers of error in position at LEO altitude and over 100 km of error at GEO altitude, growing over time due to neglected precession and nutation effects. This simplification should only be used if coarse positions are acceptable (e.g., visibility calculations or early phase design simulations).

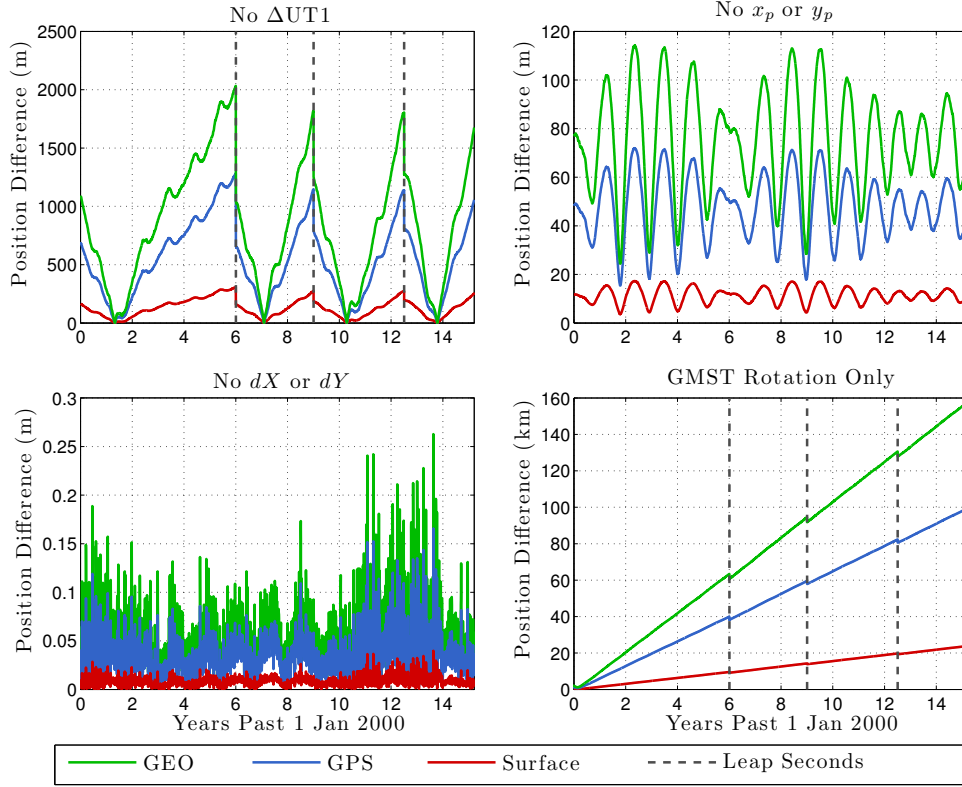


Figure 4.2: Position errors resulting from the exclusion of EOPs and the use of GMST alone. Note the GMST plot has units of kilometers while the other three have units of meters. LEO results are omitted for clarity because they are similar to the surface results.

Table 4.2: Maximum 3D position errors and angular errors of each reduced EOP set compared to using all EOPs with a cubic spline interpolation. Errors are based on transformations every 0.1 days from 2000 to 2015.

EOP Set	Surface	LEO	GPS	GEO	Angular Error
$\Delta UT1 = 0$	310 m	350 m	1.3 km	2.1 km	10 asec
$(x_p, y_p) = 0$	18 m	20 m	72 m	115 m	0.57 asec
$(dX, dY) = 0$	4.0 cm	4.5 cm	17 cm	27 cm	1.3 mas
All EOPs = 0	310 m	350 m	1.3 km	2.1 km	10 asec
GMST Only	24 km	27 km	100 km	160 km	780 asec

4.4.2 EOP Interpolation and Ocean Tide Corrections

Turning to the usage of EOPs for precise transformations, we consider various interpolation methods and ocean tide corrections. Each daily tabulated EOP should be interpolated for use in the ITRS/GCRS frame transformation. However, the desired accuracy of the transformation may vary due to the mission or simulation requirements. This section demonstrates the difference between several different orders of a polynomial interpolation (Lagrange), a linear interpolation, and using the nearest tabulated value. The differences with respect to the use of a cubic spline are then compared with the IERS published uncertainties of each EOP. Note that the “truth” values are generated using a cubic spline without ocean tide corrections. The exclusion of ocean tide corrections from truth allows us to separate out the differences in interpolation method alone.

Table 4.3 shows the maximum 3D position differences resulting from each interpolation method as compared to a cubic spline. As expected, the frame transformation error grows with altitude since transformation errors stem from differences in the central angle (whose vertex is the center of Earth). Using the nearest tabulated EOP performs the worst compared to using interpolation to retrieve EOPs at intermediate times. The position error decreases as the order of the interpolation is increased from linear (2 points) to 7th-order Lagrange (8 points). Linear interpolation results in a maximum position error of around 3 cm on the surface of the Earth and in LEO. Depending on the application, this magnitude of error could be acceptable. Increasing the order of interpolation from linear to 7th-order decreases the position difference from about 3 cm to less than 1.5 mm for locations on the Earth’s surface and in LEO.

Table 4.3: Maximum 3D position differences and angular errors resulting from interpolating EOPs with different schemes compared to the use of a cubic spline. RMS errors are shown in parenthesis. The effect of ocean tide corrections is also included here for ease of comparison. Errors are based on transformations every 0.1 days from 2000 to 2015.

Method	Surface	LEO	GPS	GEO	Angular Error
Nearest	54 cm (13 cm)	61 cm (14 cm)	2.3 m (0.52 m)	3.6 m (0.82 m)	18 mas (4.0 mas)
Linear	3.2 cm (0.64 cm)	3.6 cm (0.72 cm)	14 cm (2.7 cm)	22 cm (4.3 cm)	1.1 mas (0.21 mas)
Lagrange 3rd	5.7 mm (0.6 mm)	6.4 mm (0.7 mm)	2.4 cm (0.26 cm)	3.8 cm (0.42 cm)	0.19 mas (20 μ as)
Lagrange 5th	2.2 mm (0.3 mm)	2.5 mm (0.3 mm)	9.4 mm (1.1 mm)	1.5 cm (1.7 mm)	73 μ as (8.6 μ as)
Lagrange 7th	1.2 mm (0.1 mm)	1.3 mm (0.2 mm)	4.9 mm (0.6 mm)	7.8 mm (0.9 mm)	39 μ as (4.5 μ as)
Ocean Tide	4.9 cm (1.7 cm)	5.4 cm (1.9 cm)	20 cm (7.1 cm)	32 cm (12 cm)	1.6 mas (0.55 mas)

The effect of ocean tide corrections is also shown in Table 4.3 for comparison with interpolation errors. The position error can be up to several centimeters on the surface of Earth and in LEO and up to 20 cm at GPS altitude. Again, depending on the application, ocean tide corrections could either be ignored or may be necessary to be included, since some precise orbit determination applications require position accuracies down to the centimeter and even millimeter level (Bertiger et al, 2010; Tapley et al, 1994).

When choosing an interpolation method to meet accuracy requirements, it is important to consider the uncertainty of the published EOPs themselves. Matching a cubic spline interpolation to the sub-millimeter level, for example, would not yield a meaningful improvement if the uncertainties in the daily tabulated EOPs are larger. To investigate this aspect, the maximum interpolation error (compared to a cubic spline) and maximum ocean tide correction on each day are compared against the corresponding published EOP uncertainty for that day. Figure 4.3 illustrates the comparison between daily maximum interpolation errors, daily maximum ocean tide corrections, and the published IERS uncertainties for each EOP. Table 4.4 displays what percentage of the number of days each interpolation method yields errors larger than the published 1-sigma uncertainty values.

Various texts make different suggestions as to which interpolation scheme should be used with EOPs. The `INTERP.F` routine discussed in Sect. 4.3.3, incorporates EOP interpolation into its procedure along with the ocean tide corrections. The default setup uses 4-point Lagrangian interpolation (i.e., 3rd-order Lagrange). Although this is the default interpolation method, it does not mean it is the most accurate or the best choice. Montenbruck and Gill (2000) suggest that using a quadratic polynomial interpolation for the pole coordinates and ΔUT1 will yield sufficient accuracy and Artz et al (2012) suggests that linear interpolation is a common standard.

The results from Table 4.4 shows that linear interpolation performs worse than all of the polynomial-based Lagrange interpolation methods. Depending on the application, it may not be significant that 74% of the time the linear interpolation error of ΔUT1 is larger than its uncertainty. Precise users, however, will likely want to minimize how often this happens. Tables 4.3 and 4.4

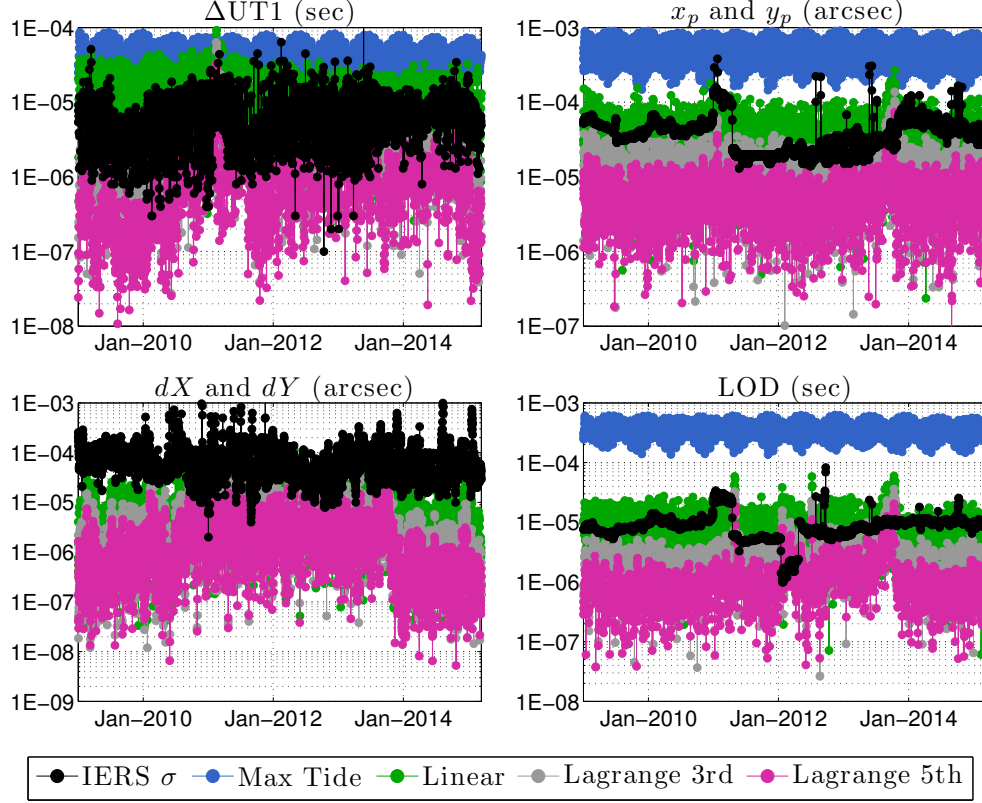


Figure 4.3: Comparison of EOP uncertainties published by the IERS, EOP interpolation errors, and EOP ocean tide corrections. The interpolation errors and ocean tide corrections displayed are daily maximum values. Time span is from 1 Jan 2009 to 15 Mar 2015.

indicate that the 5th-order Lagrange scheme is a good balance of accuracy and speed, achieving errors on the order of several millimeters with respect to the cubic spline and achieving a much lower percentage of errors as compared to the 3rd-order scheme. So despite the various recommendations scattered amongst literature and code, the results presented here illustrate that the 5th-order Lagrange interpolation scheme (at a minimum) should be recommended for interpolating EOPs for precise applications.

Choice of interpolation scheme and ocean tide corrections have a notable impact on the ITRS/GCRS transformation. Regularization of ΔUT1 and LOD is an additional procedure that has a smaller effect on the overall transformation, but may be included for high-precision applications. As discussed in Sect. 4.3.4, regularization is the removal of zonal tides prior to interpolation.

Table 4.4: Percentage of the days EOP interpolation errors are greater than the IERS reported EOP uncertainties. Comparisons are made using each daily maximum interpolation error and the corresponding uncertainty for each day from 1 Jan 2009 to 15 Mar 2015.

Method	x_p (%)	y_p (%)	ΔUT1 (%)	LOD (%)	dX (%)	dY (%)
Linear	43	27	74	40	4.2	2.1
Lagrange 3rd	6.4	2.8	12	7.3	0.93	0.18
Lagrange 5th	0.97	0.57	4.8	3.8	0.09	0.0
Lagrange 7th	0.31	0.04	1.9	2.1	0.04	0.0

Zonal tides are then added back in at the time of interpolation. The position and angular errors resulting from the exclusion of this regularization process are specified in Table 4.5. The maximum position errors are at or below 1 mm for all altitude regimes, making this procedure only applicable for high-precision applications requiring frame transformation accuracies on the order of a few microarcseconds. The maximum ΔUT1 and LOD differences are $0.33 \mu\text{s}$ and $0.22 \mu\text{s}$, respectively.

Table 4.5: Maximum 3D position differences and angular errors resulting from not performing regularization on ΔUT1 and LOD. RMS errors are shown in parenthesis. Errors are based on transformations every 0.1 days from 2000 to 2015.

Surface	LEO	GPS	GEO	Angular Error
0.15 mm (33 μm)	0.17 mm (39 μm)	0.63 mm (0.14 mm)	1.0 mm (0.20 mm)	4.9 μas (0.97 μas)

4.4.3 Interpolation of UT1-UTC Over Leap Seconds

So far, we have shown that EOPs significantly impact coordinate transformations and should be interpolated using a moderate (5th or higher) order Lagrange interpolation scheme for precise applications. This section takes a closer look at interpolating the ΔUT1 parameter. ΔUT1 has the largest impact on the ITRS/GCRS frame transformation compared to all other EOPs. Most published EOP products contain ΔUT1 as the time difference UT1-UTC. However, the time difference may instead be published as UT1-TAI. It is important to know which value is in the data file being used, because UT1-UTC contains leap seconds within the data series. This ± 1 -second jump impacts how the data set should be interpolated and can lead to large errors if done incorrectly. Table 4.6 demonstrates how leap seconds in UTC take effect and create a jump in UT1-UTC.

Table 4.6: Relationship between UTC and UT1 during the addition of a leap second. Table layout based on SOFA (2013).

	UTC	ΔAT	UT1-UTC	UT1
2012 June 30	23 59 58	34	-0.587	23 59 57.413
	23 59 59	34	-0.587	23 59 58.413
	23 59 60	34	-0.587	23 59 59.413
2012 July 01	00 00 00	35	+0.413	00 00 00.413
	00 00 01	35	+0.413	00 00 01.413

To compensate for the discontinuity in UT1-UTC, interpolation should instead be done on the continuous UT1-TAI series. Once the interpolation is complete, the number of leap seconds ΔAT for the day in question should be added to return the interpolated value back to UT1-UTC. Given daily tabulated values $(\text{UT1-UTC})_n$ and ΔAT_n , where $n = 1, 2, \dots, N$, the correct interpolation algorithm is as follows:

(1) Construct continuous UT1-TAI:

$$\bullet (\text{UT1-TAI})_n = (\text{UT1-UTC})_n - \Delta\text{AT}_n$$

(2) Interpolate UT1-TAI to the desired time, t_k :

$$\bullet (\text{UT1-TAI})_k$$

(3) Add leap seconds for day in question, ΔAT_k :

$$\bullet (\text{UT1-UTC})_k = (\text{UT1-TAI})_k + \Delta\text{AT}_k$$

Figure 4.4 illustrates both an incorrect and correct interpolation of UT1-UTC during a leap second boundary using a cubic spline technique. Blindly interpolating ΔUT1 leads the interpolation to connect the large jump over the course of the day. Table 4.6 demonstrates that in reality, the 1-second jump occurs at a single instant like a step function. For the day in question, the incorrect interpolation results in ΔUT1 being just under 0.4 seconds towards the end of June 30th. The true value, however, should be approximately -0.6 seconds. This error in ΔUT1 is larger than if the parameter had been ignored entirely. This demonstrates the necessity to take care when using a

published EOP data set and to follow the interpolation algorithm outlined above if $\Delta\text{UT1}=\text{UT1}-\text{UTC}$.

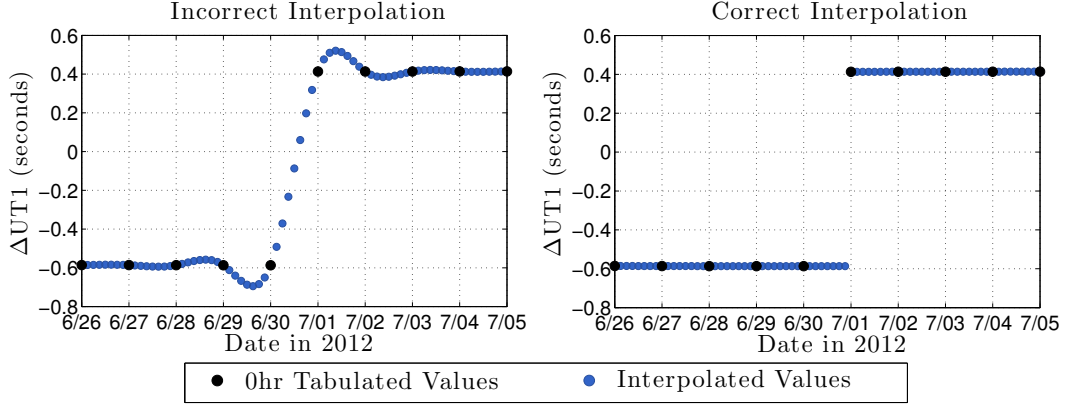


Figure 4.4: Demonstration of incorrectly interpolated (left) and correctly interpolated (right) UT1-UTC data series during the 2012 leap second. A cubic spline interpolation scheme is used here.

4.4.4 Precession–Nutation Simplifications

In the previous sections, we have restricted our analysis to the impact and recommended usage of EOPs. The studies yielded recommendations for a range of applications from high-precision to low-precision. This section focuses on a different aspect of the ITRS/GCRS transformation, the bias–precession–nutation matrix \mathbf{C} , which is the largest computational burden of the transformation. In the CIO–based framework, this entails the computation and usage of the CIP coordinates in the GCRS (X and Y) and the CIO locator (s). Section 4.3.1 overviewed the 3 methods available for the computation of X, Y, s : 1) Full Theory, 2) Series, and 3) tabulation and interpolation.

We now compare the accuracy of the interpolation method, truncated series representations, and several matrix simplifications to further flush out trades between accuracy, memory, and computation time. EOPs are left out entirely in order to isolate the effect of the bias–precession–nutation simplifications. Specifically we consider the following implementations:

- Interpolation of daily tabulated X, Y, s using an 11th-order Lagrange scheme
- Interpolation of daily tabulated X, Y, s using a 9th-order Lagrange scheme

- Interpolation of daily tabulated X, Y, s using a 7th-order Lagrange scheme
- Small angle approximation for s in \mathbf{C} matrix (i.e., $\cos(s) = 1$, $\sin(s) = s$) and X, Y, s 11th-order interpolation
- Exclusion of s entirely and 11th-order interpolation of X and Y
- Approximate 15-term series representation of X, Y with $s = 0$
- Approximate 6-term series representation of X, Y with $s = 0$
- Approximate 4-term series representation of X, Y with $s = 0$

Interpolation of tabulated X, Y , and s values was introduced by Coppola et al (2009) and offers superior computational efficiency as compared to the Full Theory or series methods, while maintaining nearly full accuracy. Accuracy degradation is largely determined by the order of Lagrange interpolation used. Errors in X, Y and s due to 11th-, 9th-, and 7th-order interpolation are contained in 4.8.3. The compromise with the interpolation method is the memory required to store the tabulated data, though for short timespans, the number of values needed is far less than the several thousand needed by the full series or Full Theory methods. Therefore, we present shortened series (4-, 6-, and 15-term approximations) for computing X and Y . These approximate series are fast and low on memory requirements, but come at the price of accuracy. The 4-term and 6-term truncations are published and discussed by Wallace and Capitaine (2006) and Capitaine and Wallace (2008), respectively. The 15-term approximation has been assembled specifically for use in this study. The values needed for each series and their accuracies for the computation of X and Y are presented in 4.8.5, 4.8.6, and 4.8.7.

A comparison of each bias–precession–nutation scheme is performed by transforming positions from the ITRS to the GCRS at noon everyday from 1990 to 2050, using the Full Theory as truth. Performing each transformation at noon (12h) is chosen to maximize interpolation errors for the 0hr tabulated data. Table 4.7 lists the position and angular errors resulting from each simplified procedure. The 11th-order Lagrange interpolation scheme yields the best accuracy, with errors

on the same level as the accuracy of the IAU models themselves (sub-microarcsecond). A small degradation in accuracy is seen by reducing the interpolation order to 9 and to 7.

Lines 3 and 4 in Table 4.7 show results from simplifications to the use of the CIO locator s . Both of the simplifications use 11th-order interpolation of X, Y, s , but either eliminate the trigonometric expressions by using a small angle approximation or remove s from the \mathbf{C} matrix entirely (see Eq. 5.4). We see that between 1990 and 2050, there is no need to use the full trigonometric terms when including s in the \mathbf{C} matrix. A small angle approximation does not degrade the transformation in a meaningful way as the value of s itself does not exceed 0.022 arcseconds between 1990 and 2050 (see Fig. 4.8). Excluding s entirely, however, introduces a sub-meter position error at the surface of the Earth and several meters in GEO.

The use of the truncated series approximations for X and Y produce larger position and angular errors than the other methods. There are many applications which only require sub-arcsecond accuracy, instead of milli-arcsecond or even micro-arcsecond accuracy, such as telescope pointing, radar dish pointing, or satellite attitude knowledge. The truncated series are very light on memory usage and computation time, making them ideal for situations when ultimate Earth orientation accuracy is not required.

Table 4.7: Maximum 3D position and angular differences resulting from bias–precession–nutration simplifications. RMS errors are shown in parenthesis. Transformations performed at noon each day from 1990 to 2050.

Method	Surface	LEO	GPS	GEO	Angular Error
Interp X, Y, s 11th-Order	11 μm (2.6 μm)	12 μm (2.9 μm)	44 μm (11 μm)	69 μm (17 μm)	0.34 μas (0.09 μas)
Interp X, Y, s 9th-Order	35 μm (8.7 μm)	40 μm (9.8 μm)	150 μm (37 μm)	240 μm (58 μm)	1.2 μas (0.28 μas)
Interp X, Y, s 7th-Order	0.15 mm (41 μm)	0.17 mm (45 μm)	0.63 mm (0.17 mm)	1.0 mm (0.27 mm)	4.9 μas (1.3 μas)
Small Angle Approx s	11 μm (2.6 μm)	12 μm (2.9 μm)	44 μm (11 μm)	69 μm (17 μm)	0.34 μas (0.09 μas)
No s ($s = 0$)	0.68 m (0.25 m)	0.76 m (0.28 m)	2.8 m (1.0 m)	4.5 m (1.7 m)	0.022 asec (8 mas)
15-Term Series for X, Y	4.3 m (1.7 m)	4.8 m (1.9 m)	18 m (7.2 m)	28 m (12 m)	0.14 asec (56 mas)
6-Term Series for X, Y	12 m (4.9 m)	14 m (5.5 m)	50 m (21 m)	79 m (33 m)	0.39 asec (0.16 asec)
4-Term Series for X, Y	28 m (18 m)	32 m (20 m)	120 m (74 m)	190 m (120 m)	0.91 asec (0.57 asec)

4.4.5 Julian Date Handling

With the ITRS/GCRS transformation procedure discussed in detail, one last important consideration (often overlooked) is the storage of time itself. During the implementation of the IAU models, time is the independent variable for EOP interpolation, bias-precession-nutation computations, and the computation of the Earth rotation angle (ERA). Each of these pieces of the transformation require time to be in either UT1 or TT, but always as a Julian date or Julian century. This section discusses the impact on precise ITRS/GCRS frame transformations due to how the Julian date is stored in the computer.

Storing the Julian date as one double-precision number limits the precision to which the date and time can be recorded due to the large integer part inherent to Julian dates. Current Julian dates stored in double precision only have a precision of $2\text{e}-9$ days, or about 0.2 milliseconds. The precision to which time is stored most notably affects the computation of θ_{ERA} (used in \mathbf{R}), however it also influences the computation (or interpolation) of X , Y , and s . For comparison, the ocean tide corrections for ΔUT1 are on the order of tens of microseconds, well below the precision maintained by a single Julian date. Four methods for Julian date storage are described and supported by IAU’s SOFA software suite (SOFA, 2013). Table 4.8 provides an example of each storage technique and the precision limit of each. The light-time equivalent is also given, since time stamp formats will dictate the resolution to which a range measurement, for example, may be stored.

Table 4.8: Several common examples of ways that Julian dates can be stored. Example date is 7 March 2013 18:00:00.0. This analysis assumes a computer floating-point accuracy of $2.2\text{e}-16$.

Format Name	JD1	JD2	Precision (s)	Light Dist. (m)
Julian Date Method	2456359.25	0	$1.9\text{e}-4$	57,000
MJD Method	2400000.5	56358.75	$1.9\text{e}-6$	570
J2000 Method	2451545	4814.25	$1.9\text{e}-7$	57
Date & Time Method	2456358.5	0.75	$1.9\text{e}-11$	0.006

Table 4.9 lists the maximum 3D position errors resulting from using a single Julian date value when performing a coordinate transformation. The same transformation simulation setup described in Table 4.1 is used. Position errors are on the order of centimeters. A significant improvement to

precision can be obtained by storing a Julian date in two parts, referred to as the “date and time” method in Table 4.8. In this scheme, one value is the 0h day portion (thus ending in 0.5) and the second value is the fraction of a day, which is always less than 1.0. This two-part formulation yields about $2.2\text{e-}16$ day precision, which is equivalent to $2\text{e-}11$ seconds.¹²

Table 4.9: Maximum 3D position differences in the ITRS/GCRS frame transformation due to storing Julian date as a single value instead of a two-part Julian date.

Time Storage Format	Surface	LEO	GPS	GEO	Angular Error
Single JD Value	2.6 cm	2.9 cm	11 cm	17 cm	0.82 mas

Proper handling of date/times also has a significant impact on orbit determination. The importance of precise timing in constructing radio ranging and Doppler measurements is clear, where a timing error of $1\text{ }\mu\text{s}$ leads to a range error of approximately 300 m due to the distance that radio signals travel over that interval. What is perhaps less appreciated, is the importance of the time stamp precision in marking these measurements. In this case, the timing error maps to orbit error based on the motion of the transmitter or receiver platform. For example, when incorporating a ranging measurement made on a LEO platform, a time resolution limited to only 0.2 ms (corresponding to the Julian date method in Table 4.8) could result in an along-track position error of 1.4 m due to the 7 km/s velocity of the platform.

4.5 Summary of Recommendations

The frame transformation analysis presented in this paper provides details describing the impact on accuracy of EOP usage and interpolation, tide corrections, time handling, and precession–nutation, including various implementation options. This section compiles results in order to present a set of recommended procedures to meet a range of accuracy and computation load requirements. While some users require the highest accuracy with little regard for computation time, others may find moderate to low accuracy perfectly acceptable, given more stringent limits on computational load.

¹² This assumes the floating-point accuracy of the computer is $2.2\text{e-}16$.

Table 4.10 summarizes our recommendations for the ITRS/GCRS frame transformation procedure to achieve specific levels of accuracy in several altitude regimes about the Earth. Each recommended procedure is aimed at a minimum computational burden to achieve the desired accuracy. Position and angular errors assume a truth transformation using “Full Theory” bias–precession–nutation, all EOPs interpolated using a cubic spline, interpolation of regularized ΔUT1 , and EOP ocean tide corrections included. Recommendations for using the X, Y, s interpolation method assume the X, Y, s table has been generated using either the “Full Theory” or full series methods for bias–precession–nutation with values stored every day at 0h TT.

In Table 4.10, the threshold line between 2.3 mas and 23 mas for maximum angular error indicates the breaking point for using a one-part versus a two-part Julian date. Below the line, a two-part Julian date format (the “date and time” method) should be used. Above this line, a single number for Julian date may be used without significant impact on the total error. The term “simple s ” in the X, Y, s column stands for the small angle approximation of s to be used in the bias–precession–nutation matrix formulation. The threshold for using a one- or two-part Julian date occurs near the switch between including s or ignoring s entirely and approximately when dX and dY are excluded. Lastly, an asterisk in the EOP Interpolation column indicates the use of the regularization process for the interpolation of ΔUT1 . Regularization has a small effect on the overall transformation and is only required for the highest accuracy applications.

Table 4.10: Recommendations for EOP usage and precession–nutation implementations to achieve specific levels of position transformation accuracy with minimum computational cost. Each row gives the recommended procedure to achieve position and angular accuracies below the maximum errors specified on the left. An asterisk in the EOP Interpolation column indicates the use of ΔUT1 regularization.

Angle	Maximum Allowed Error			X, Y, s Representation	EOPs	EOP Interpolation	Ocean Tide Corrections
	Surf/LEO	GPS	GEO				
0.7 deg	90 km	330 km	520 km	None (GMST Only to 2050)	none	–	–
0.37 deg	46 km	170 km	270 km	None (GMST Only to 2025)	none	–	–
10 asec	350 m	1.3 km	2.1 km	4-term X, Y , No s	none	–	–
1.3 asec	46 m	170 m	270 m	4-term X, Y , No s	ΔUT1	Nearest	No
0.8 asec	28 m	110 m	170 m	6-term X, Y , No s	ΔUT1	Nearest	No
0.66 asec	23 m	85 m	140 m	15-term X, Y , No s	ΔUT1	Nearest	No
0.14 asec	4.7 m	18 m	28 m	15-term X, Y , No s	$x_p, y_p, \Delta\text{UT1}$	Nearest	No
23 mas	0.80 m	3.0 m	4.7 m	Interp, Lagr 7, No s	$x_p, y_p, \Delta\text{UT1}$	Linear	No
2.3 mas	7.9 cm	29 cm	46 cm	Interp, Lagr 7, Simple s	$x_p, y_p, \Delta\text{UT1}$	Linear	No
0.19 mas	6.5 mm	2.4 cm	3.9 cm	Interp, Lagr 7, Simple s	All EOPs	Lagr 3	Yes
71 μas	2.5 mm	9.1 mm	15 mm	Interp, Lagr 7, Simple s	All EOPs	Lagr 5	Yes
37 μas	1.3 mm	4.8 mm	7.5 mm	Interp, Lagr 7, Simple s	All EOPs	Lagr 7	Yes
5.0 μas	0.18 mm	0.64 mm	1.1 mm	Interp, Lagr 9, Simple s	All EOPs	Cubic Spline	Yes
1.2 μas	39 μm	0.15 mm	0.23 mm	Interp, Lagr 9, Simple s	All EOPs	Cubic Spline*	Yes
0.34 μas	12 μm	43 μm	68 μm	Interp, Lagr 11, Simple s	All EOPs	Cubic Spline*	Yes

The method of interpolating a table of X, Y, s values is recommended for all moderate to high-accuracy applications. Interpolation, especially using a 9th- or 11th-order Lagrange scheme, yields accuracies that agree to within about 1 μas of the series method and “Full Theory” methods, but is nearly 600 times faster (see Coppola et al (2009) for more details). For moderate to low accuracy requirements, the 6- and 15-term series approximations yield sub-arcsecond accuracies and ΔUT1 is the only EOP that needs to be included. Beyond these, using a simple GMST rotation is a fast alternative that yields transformation errors below 0.4 degrees up to the year 2025 and below 1 degree up to 2050. The GMST method could be acceptable for visibility calculations or spacecraft attitude rotations, depending on the accuracy requirements. However, including the simple 4-term approximation for precession–nutation significantly improves the accuracy, reducing the transformation error by 99.3% with little additional computation time.

4.6 Conclusion

The impacts of EOP usage, interpolation, ocean tide corrections, bias–precession–nutation simplifications, and Julian date storage on the accuracy of coordinate transformations between

the ITRS and GCRS have been thoroughly investigated. Strategies are discussed for reducing computation time while achieving various levels of accuracy. EOP interpolation methods and ocean tide corrections are shown to have a notable impact on precise frame transformations. Poor interpolation of EOPs and the exclusion of ocean tide corrections can lead to position errors on the order of 5 cm near the surface of the Earth and 20-30 cm at GPS altitude. The effect of individual EOPs being excluded from the transformation is also presented. The results demonstrate the need for orbital analysts and scientists to carefully consider EOP implementation strategies based on the given task, including orbital altitude and desired position accuracy.

Simplifications to the CIO-based transformation method, namely in the computation of the CIP coordinates and CIO locator, are presented and illustrate the impact of each simplification on accuracy. Each simplification offers trade-offs between accuracy, computation time, and memory usage. It is demonstrated that the inclusion of the CIO locator can be simplified to using small angle approximations for precise applications, and ignored entirely for moderate accuracy applications. Interpolation of precession–nutration parameters offer fast and accurate capabilities, but at the cost of some memory. Truncated series representations offer moderate accuracy with the least computational burden.

Lastly, the commonly-used simplification of only using GMST to convert between the ITRS and GCRS is shown to yield position errors from 50 km to 270 km in LEO and GEO, respectively, through the year 2025. This approximation should only be used for early phase design studies or satellite visibility computations where some loss in accuracy is acceptable.

We provide a table of recommendations for the use of EOPs, interpolation methods, ocean tide corrections, and precession–nutration schemes to achieve various levels of position accuracy in several altitude regimes. This information may be used to guide the design of on-board and post-processing orbit estimation, orbit propagation, and attitude tasks.

4.7 Acknowledgements

This research was made possible with partial Government support under and awarded by DoD, Air Force Office of Scientific Research, National Defense Science and Engineering Graduate (NDSEG) Fellowship, 32 CFR 168a. Part of the research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. The authors also wish to thank the anonymous reviewers for their insightful recommendations which helped improve the paper.

4.8 Appendix

4.8.1 Transformation Procedure and EOPs

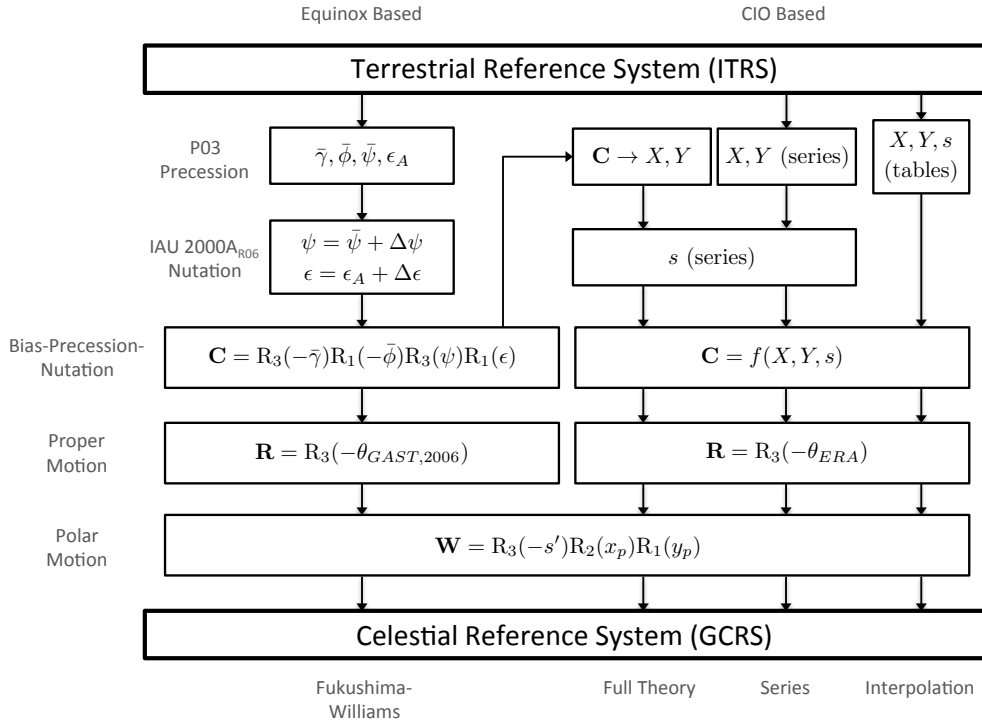


Figure 4.5: Flow chart for several ITRS/GCRS frame transformation procedures. The layout is based on Vallado (2013).

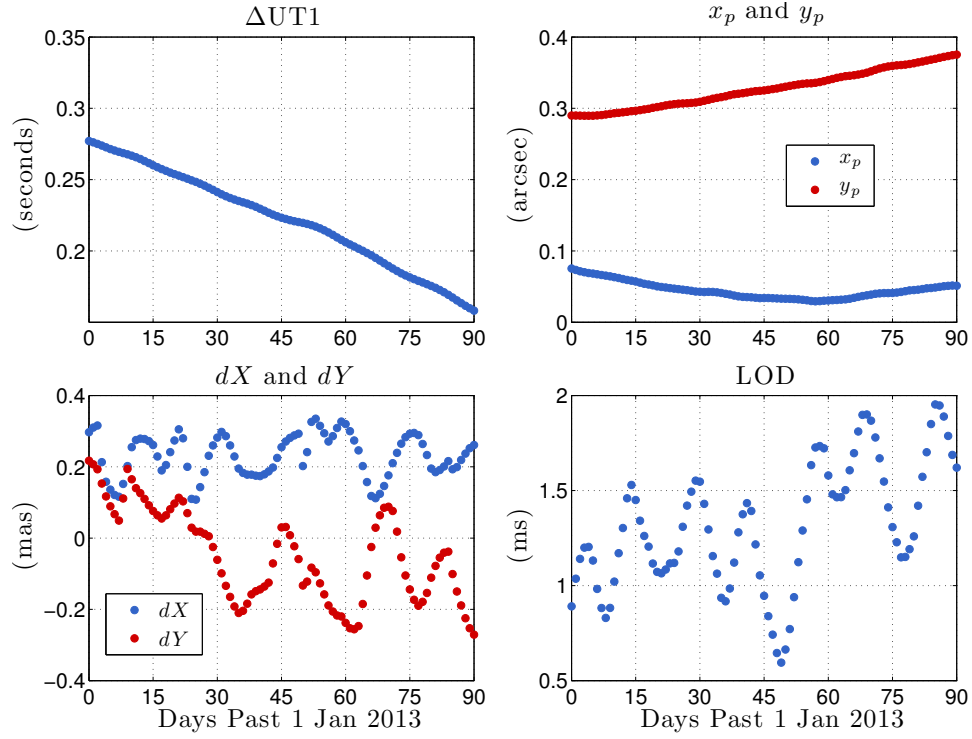


Figure 4.6: Behavior of daily tabulated EOP values for 90 days from 1 January 2013.

4.8.2 X, Y, s Behavior

To help understand the behavior and magnitude of the CIP coordinates (X and Y) and the CIO locator (s), Fig. 4.7 and 4.8 depict the value of these parameters between the years 1990 and 2050. The “Full Theory” has been used to compute X, Y and s has been computed using its full series representation. These models adhere to the IAU 2006/2000A_{R06} theory.

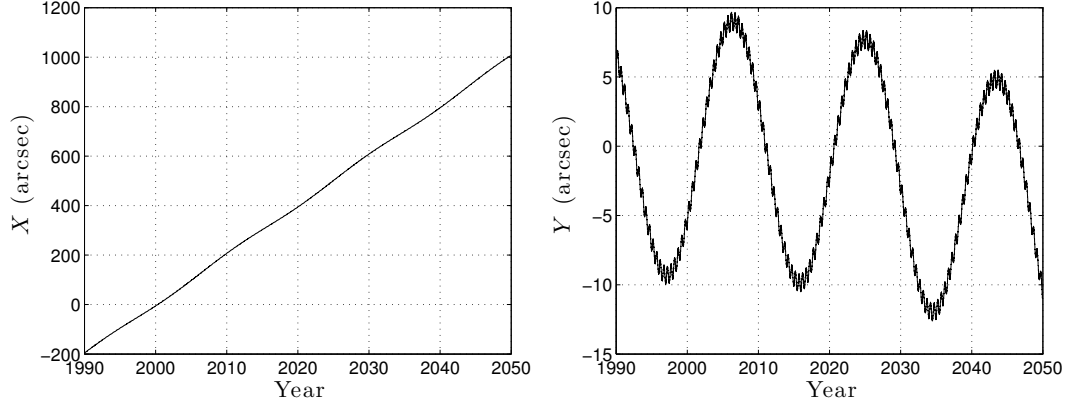


Figure 4.7: Actual values of X and Y between 1990 to 2050 generated using the Full Theory.

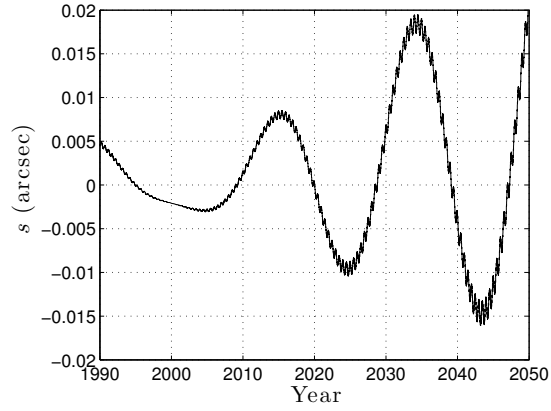


Figure 4.8: Actual values of CIO-locator s between 1990 to 2050 generated using the Full Theory.

4.8.3 X, Y, s Interpolation Accuracy

This section displays the errors associated with the tabulation and interpolation method for computing X, Y, s using 11th-, 9th-, and 7th-order Lagrange schemes. Values are tabulated at 0hr TT each day and interpolations are made at noon to yield worst cast interpolation errors. Truth is considered to be the use of the “Full Theory” to compute X, Y, s .

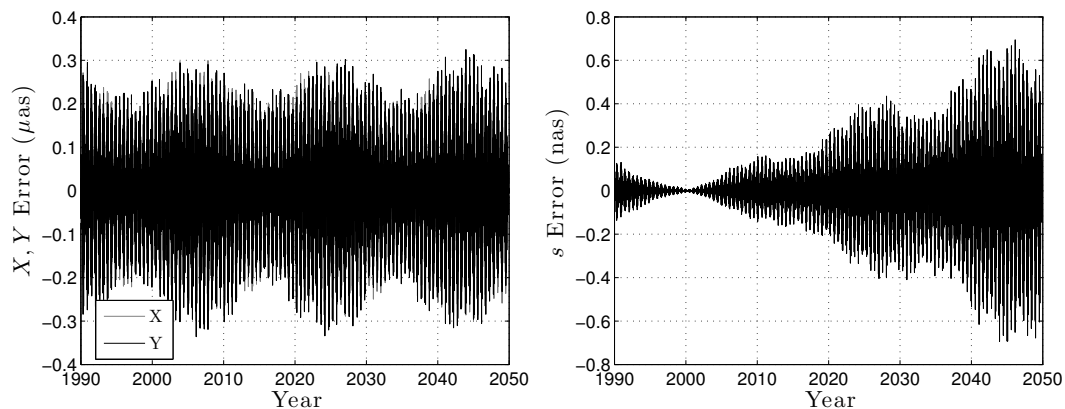


Figure 4.9: Error in CIO parameters X, Y, s from 11th-order Lagrange interpolation of daily tabulated values between 1990 to 2050. Interpolations are made each day at noon. Truth X, Y, s values are generated using the Full Theory. Note the unit for X, Y is microarcseconds and the unit for s is nanoarcseconds. Maximum errors are $0.30 \mu\text{as}$, $0.34 \mu\text{as}$, and 0.69 nas for X, Y , and s respectively. RMS errors are $0.057 \mu\text{as}$, $0.061 \mu\text{as}$, and 0.08 nas .

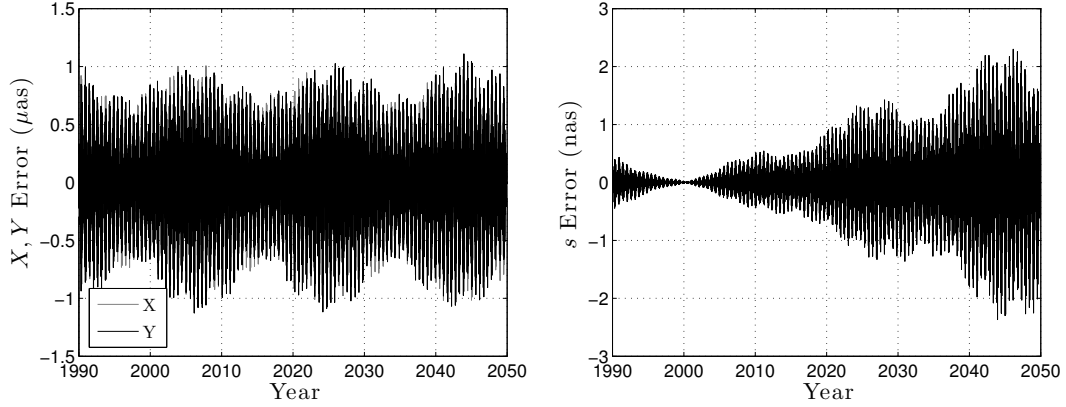


Figure 4.10: Error in CIO parameters X, Y, s from 9th-order Lagrange interpolation of daily tabulated values between 1990 to 2050. Interpolations are made each day at noon. Truth X, Y, s values are generated using the Full Theory. Note the unit for X, Y is microarcseconds and the unit for s is nanoarcseconds. Maximum errors are $1.0 \mu\text{as}$, $1.1 \mu\text{as}$, and 2.4 nas for X, Y , and s respectively. RMS errors are $0.19 \mu\text{as}$, $0.21 \mu\text{as}$, and 0.27 nas .

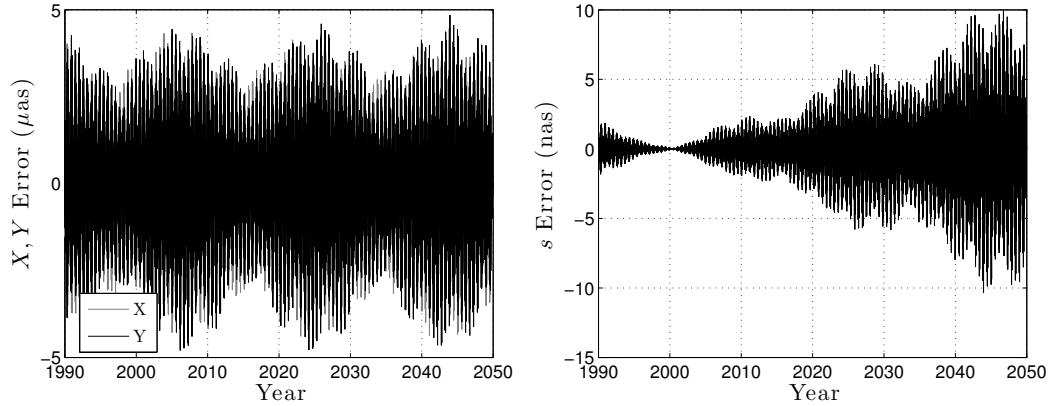


Figure 4.11: Error in CIO parameters X, Y, s from 7th-order Lagrange interpolation of daily tabulated values between 1990 to 2050. Interpolations are made each day at noon. Truth X, Y, s values are generated using the Full Theory. Note the unit for X, Y is microarcseconds and the unit for s is nanoarcseconds. Maximum errors are $4.4 \mu\text{as}$, $4.8 \mu\text{as}$, and 11 nas for X, Y , and s respectively. RMS errors are $0.89 \mu\text{as}$, $0.96 \mu\text{as}$, and 1.3 nas .

4.8.4 Fundamental Arguments

The linear approximations for the Delaunay variables are given in Table 4.11, where T_{TT} is Julian centuries of TT past J2000. The higher order T_{TT}^2 , T_{TT}^3 , and T_{TT}^4 terms have been omitted.

Table 4.11: Linear approximations of the Delaunay variables from Capitaine and Wallace (2008). Variables l and l' are omitted from this table because they are not used in the 4-, 6-, or 15-term X, Y approximations discussed in this paper. Jcy stands for Julian century.

Angle	(rad)	T_{TT} (rad/Jcy)
F	1.6279050815	8433.4661569164
D	5.1984665887	7771.3771455937
Ω	2.1824391966	−33.7570459536

4.8.5 4-Term X, Y Series Approximation

This section outlines the 4-term series truncation for computing the CIP coordinates, X and Y . This series truncation is originally presented in Wallace and Capitaine (2006). The values for computing the Delaunay variables are contained above in 4.8.4.

Table 4.12: Series for the 4-term X, Y approximation from Wallace and Capitaine (2006). Maximum errors between 1990 and 2050 are 0.90 arcsec and 0.80 arcsec for X and Y , respectively. RMS errors are 0.39 arcsec and 0.42 arcsec.

Term	Amplitude	l	l'	F	D	Ω
	(μas)					
X t	2004191898					
Y t^2	−22407275					
X sin	−6844318	0	0	0	0	1
Y cos	+9205236	"	"	"	"	"

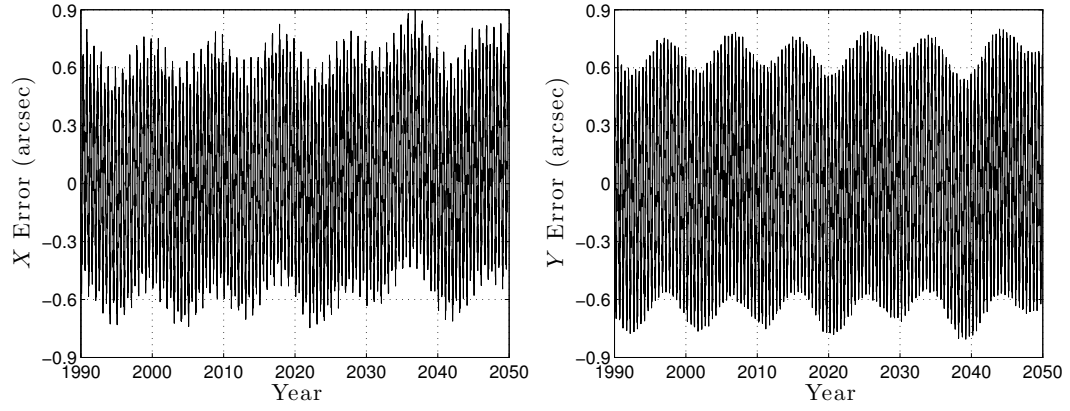


Figure 4.12: Error in CIO parameters X and Y from the 4-term series approximation between 1990 to 2050. Truth X, Y values are generated using the Full Theory.

4.8.6 6-Term X, Y Series Approximation

The parameters of the 6-term series truncation for computing X and Y are given below in Table 4.13. The 6-term truncation is originally presented by Capitaine and Wallace (2008).

Table 4.13: Series for the 6-term X, Y approximation from Table G.1 in Capitaine and Wallace (2008). Table layout follows that of Capitaine and Wallace (2008). Maximum errors between 1990 and 2050 are 0.38 arcsec and 0.27 arcsec for X and Y , respectively. RMS errors are 0.12 arcsec and 0.10 arcsec.

Term		Amplitude	l	l'	F	D	Ω
		(μas)					
X	t	2004191898					
Y	t^2	-22407275					
X	sin	-6844318	0	0	0	0	1
Y	cos	+9205236	"	"	"	"	"
X	sin	-523908	0	0	2	-2	2
Y	cos	+573033	"	"	"	"	"

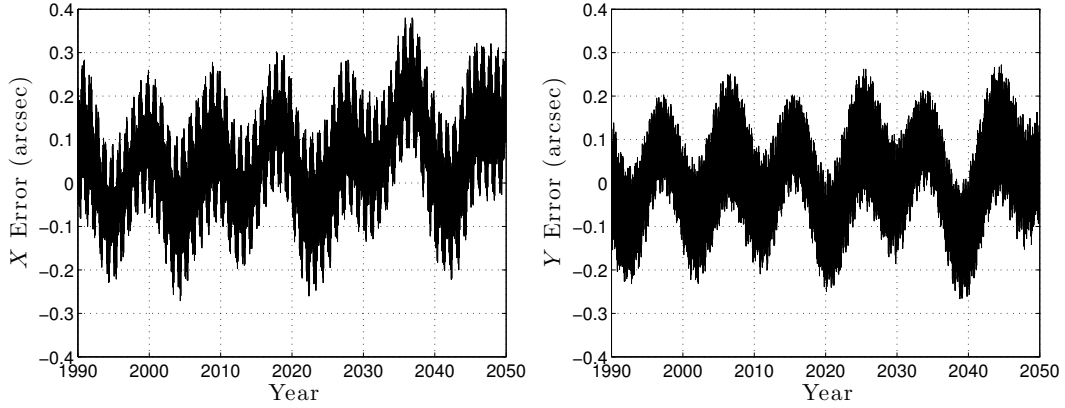


Figure 4.13: Error in CIO parameters X and Y from the 6-term series approximation between 1990 to 2050. Truth X, Y values are generated using the Full Theory.

4.8.7 15-Term X, Y Series Approximation

This section presents the parameters and the accuracy of the 15-term series truncation for X and Y . The 15-term series is presented for this paper and is a truncation of the full series adhering to the IAU 2006/2000A_{R06} theory.

Table 4.14: Series for the 15-term X, Y approximation presented in this paper. Maximum errors between 1990 and 2050 are 0.13 arcsec and 0.094 arcsec for X and Y , respectively. RMS errors are 0.048 arcsec and 0.028 arcsec.

Term		Amplitude (μas)	l	l'	F	D	Ω
X		-17251					
X	t	2004191898					
X	t^2	-429783					
Y	t	-25896					
Y	t^2	-22407275					
X	\sin	-6844318	0	0	0	0	1
X	$t \cos$	+205833	"	"	"	"	"
Y	\cos	+9205236	"	"	"	"	"
Y	$t \sin$	+153042	"	"	"	"	"
X	\sin	+82169	0	0	0	0	2
Y	\cos	-89618	"	"	"	"	"
X	\sin	-523908	0	0	2	-2	2
Y	\cos	+573033	"	"	"	"	"
X	\sin	-90552	0	0	2	0	2
Y	\cos	+97847	"	"	"	"	"

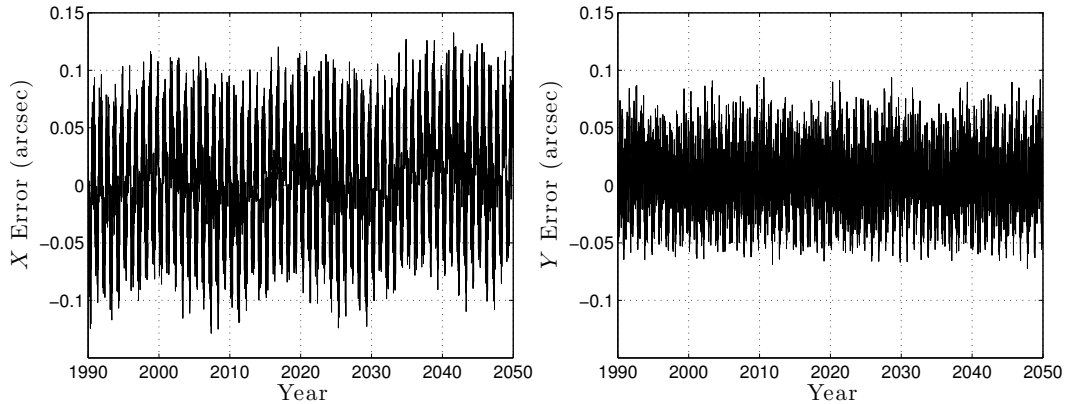


Figure 4.14: Error in CIO parameters X and Y from the 15-term series approximation between 1990 to 2050. Truth X, Y values are generated using the Full Theory.

Chapter 5

ITRS/GCRS Impact on Orbit Propagation

Journal Version:

Bradley, B.K. and Axelrad, P., “Influence of ITRS/GCRS Implementation for Astrodynamics: Orbit Propagation,” in preparation for Advances in Space Research, 2015.

5.1 Abstract

Propagation of satellite orbits is required for astrodynamics applications including mission design, orbit determination in support of operations and payload data analysis, and conjunction assessment. Each application has unique requirements in terms of precision, latency, and computational load. This research investigates the impact of the fidelity of the transformation between the Geocentric Celestial Reference System (GCRS) and the International Terrestrial Reference System (ITRS) on orbit propagation. This transformation is required to evaluate and rotate the acceleration vector due to Earth’s gravity field and the time-derivative of the transformation is used for orbit propagation in the ITRS. Motivated by the growing space catalog and the demands for precise orbit determination with shorter latency for science and reconnaissance missions, this research improves the computational performance of orbit propagation through more efficient and precise ITRS/GCRS implementations. This paper provides simplifications to the Celestial Intermediate Origin (CIO) transformation scheme and Earth orientation parameter (EOP) storage for use in orbit propagation. A complete derivation of the frame transformation time-derivative is detailed for use in velocity transformations between the GCRS and ITRS and is applied to orbit propagation in the rotating ITRS. A range of orbit propagation simulations are performed to analyze

the impact of transformation complexity on propagation accuracy. We consider low-Earth orbits (LEO), global positioning system (GPS) orbits, and geosynchronous orbits (GEO) with a variety of inclinations and initial conditions. A significant savings in computation time and memory can be made by the use of polynomial and trigonometric series approximations for CIO parameters and EOPs with a small number of terms. These techniques yield maximum propagation errors of around 2 cm and 1 m are achieved after 15 minutes and 6 hours in LEO, respectively. For orbit propagation in the ITRS, we demonstrate a simplified scheme that yields propagation errors under 5 cm after 15 minutes in LEO. This approach is beneficial for orbit determination based on GPS measurements. The result is a comprehensive summary yielding accuracy versus algorithm complexity for inertial and Earth-fixed orbit propagations so that users may choose the best approach which balances accuracy, computational load, and memory for their application.

5.2 Introduction

Orbit propagation is a key component in astrodynamics. It is used in orbit determination for commercial, military, and scientific satellites, and in conjunction assessment for the growing catalog of objects in Earth orbit. Today, the public space catalog consists of approximately 17,000 objects.¹ Maintaining the growing catalog, performing conjunction assessment to prevent collisions, and operating on-going satellite missions all require orbit propagation encompassing a wide range of accuracy and computational requirements. The central piece to performing orbit propagation around Earth is a transformation between the Geocentric Celestial Reference System (GCRS) and the International Terrestrial Reference System (ITRS). This transformation can be a costly procedure if performed to full accuracy. The computational complexity required for this transformation depends on the application and can be adjusted such that the precision is met while minimizing memory and computation time. This paper seeks to determine the influence of the ITRS/GCRS on orbit propagation and follows the study in Bradley et al (2015), which focused on coordinate rotation accuracy, e.g., for use with ground station coordinates, satellite positions, and

¹ Based on NASA Orbital Debris Quarterly News, Vol. 19, Issue 3, July 2015.

spacecraft attitude.

The procedures and models needed to transform between the GCRS and ITRS are maintained by the International Earth Rotation Service (IERS) and International Astronomical Union (IAU) (Petit and Luzum, 2010). The transformation includes precession, nutation, polar motion, Earth’s proper rotation, and requires Earth orientation parameters (EOPs) for moderate to high accuracy. The full accuracy procedures involve series with several thousand terms, requiring substantial computation time and memory (Wallace and Capitaine, 2006; Petit and Luzum, 2010). This transformation is used within orbit propagation during the evaluation and subsequent rotation of the acceleration vector due to the central body’s gravity field. The computation of the acceleration due to Earth’s gravity field (e.g., through spherical harmonics) must be performed multiple times per numerical integration step when propagating orbits using special perturbation theory.

We conduct a range of orbit propagation simulations with the goal of generating recommendations for the minimum ITRS/GCRS transformation complexity needed to achieve specific orbit propagation accuracies. To this end, simplifications to bias–precession–nutation, namely interpolation and truncated series approximations, are outlined and a polynomial fit to precession–nutation quantities is presented. These methods are faster and use less memory than the full accuracy procedures.

We then perform a detailed analysis of orbit propagation in the ITRS. While orbit propagation is commonly performed in an inertial system (e.g., the GCRS), there are situations when propagating an orbit in a rotating frame may be beneficial. Such is the case for satellites utilizing GPS measurements to perform precise orbit determination on-board a satellite, because the position of each GPS satellite is given in the ITRS. Propagating an orbit in the ITRS requires that each perturbing acceleration be expressed in the ITRS and that additional centripetal, Coriolis, and Euler accelerations be included. Computing the additional acceleration terms requires the angular velocity vector of the ITRS. Montenbruck and Ramos-Bosch (2008) describe a simple approach for computing this rotation vector, but the full expression is complex and requires time-derivatives of

polar motion, CIP coordinates, and the time-derivative of the rotation vector itself.

A detailed derivation of the angular velocity vector necessary for ITRS propagation is presented next. Each time-varying component of precession–nutation, polar motion, and proper motion of Earth is analyzed. Orbit propagation simulations with a variety of rotation vector simplifications are used to generate recommendations which balance accuracy and computational complexity. The resulting recommendations give satellite operators and scientists a concrete set of options to decide if ITRS propagation is appropriate for their application, and if so, how best to implement it. This research demonstrates that ITRS propagation is viable for both real-time on-board orbit determination and long-term orbit propagation.

This paper begins with an overview of the ITRS/GCRS transformation procedure based on the Celestial Intermediate Origin (CIO). Simplifications for computing the CIP parameters and EOPs are detailed next. Section 4 investigates inertial orbit propagation and the impact that frame transformation procedure has on propagation accuracy. Next, the complete time-derivative of the ITRS/GCRS transformation matrix and rotation vector of the ITRS are derived in full detail. Velocity transformation accuracy is analyzed to ensure correct initial conditions for ITRS propagation. The paper concludes with results from ITRS orbit propagation simulations using various levels of simplification to the ITRS rotation vector. The result is a comprehensive summary yielding accuracy versus algorithm complexity for inertial and Earth-fixed orbit propagations so that users may easily decide the best method for their needs.

5.3 ITRS/GCRS Transformation

This section outlines the ITRS to GCRS transformation process used in orbit propagation and orbit determination. The current recommendations for the transformation procedure are the International Astronomical Union (IAU) models contained within the IERS 2010 Conventions (Petit and Luzum, 2010; Capitaine and Wallace, 2008). The new model, as a whole, is called IAU 2000A/2006. This includes the IAU 2000A_{R06} nutation theory and the IAU 2006 precession theory. The methods for implementing this transformation can be separated into two categories, equinox-

based and Celestial Intermediate Origin (CIO-based). This work focuses on the use of the CIO-based method which is based on the direction cosine components of the Celestial Intermediate Pole (CIP) in the GCRS (X and Y) and the CIO locator (s). This is a convenient method because the CIP is approximately the instantaneous rotation vector of the Earth in the GCRS, the bias–precession–nutation matrix is formed directly from X, Y, s , and the IERS will continue to publish the dX and dY celestial pole offsets in the future (Wallace and Capitaine, 2006; Kaplan, 2005). The transformation of a position vector, \vec{r} , from the ITRS to the GCRS is performed by

$$\vec{r}_{\text{GCRS}} = \mathbf{C} \mathbf{R} \mathbf{W} \vec{r}_{\text{ITRS}}, \quad (5.1)$$

and the reverse by

$$\vec{r}_{\text{ITRS}} = \mathbf{W}^T \mathbf{R}^T \mathbf{C}^T \vec{r}_{\text{GCRS}}. \quad (5.2)$$

We define the total transformation such that

$$\mathbf{T} = \mathbf{C} \mathbf{R} \mathbf{W}, \quad (5.3)$$

where the CIO-based equations for bias–precession–nutation (\mathbf{C}), Earth rotation (\mathbf{R}), and polar motion (\mathbf{W}) are given by

$$\mathbf{C} = \begin{bmatrix} 1 - aX^2 & -aXY & X \\ -aXY & 1 - aY^2 & Y \\ -X & -Y & 1 - a(X^2 + Y^2) \end{bmatrix} \mathbf{R}_3(s) \quad (5.4)$$

$$\mathbf{R} = \mathbf{R}_3(-\theta_{ERA}) \quad (5.5)$$

$$\mathbf{W} = \mathbf{R}_3(-s') \mathbf{R}_2(x_p) \mathbf{R}_1(y_p). \quad (5.6)$$

$$(5.7)$$

The z-component of the CIP vector $[X \ Y \ Z]^T$ is given by

$$Z = \sqrt{1 - X^2 - Y^2}, \quad (5.8)$$

and is used to compute

$$a = 1/(1 + Z). \quad (5.9)$$

The TIO locator (s'), also known as the instantaneous prime meridian, is given in arcseconds by $s' = -0.000047'' T_{\text{TT}}$ for use in the polar motion matrix. The parameter T_{TT} is Julian centuries of Terrestrial Time (TT) past J2000². Within the CIO-based approach, the largest computational burden comes from the calculation of X , Y , and s .

An additional aspect of the ITRS/GCRS transformation is the inclusion of EOPs. EOPs correct for irregularities in the IAU models and are estimated from observations using Very Long Baseline Interferometry (VLBI), Satellite Laser Ranging (SLR), and the Global Positioning System (GPS) (Artz et al, 2012; Sibois et al, 2015). EOPs consist of the UT1 time difference (ΔUT1), pole coordinates (x_p, y_p), and celestial pole offsets (dX, dY). Length of day (LOD), the time derivative of ΔUT1 , is usually contained within published EOP files as well. EOPs are published in data tables which contain values at 0h UTC each day and typically include predicted values at least one month into the future (Vallado and Kelso, 2013). An overview of EOP behavior and usage, along with ocean tide corrections for EOPs, is given by Bradley et al (2015). That work showed that the largest effect on coordinate transformations is ΔUT1 , yielding a position error of up to 350 m at an altitude of 800 km if neglected (Bradley et al, 2015).

5.4 EOP and CIO Simplifications

The computation of CIO-based parameters X , Y , and s as well as the storage and maintenance of EOP data tables come at the cost of computation time and memory. Two full accuracy methods to compute X , Y , and s outlined by Wallace and Capitaine (2006), SOFA (2014), and Vallado (2013) are the (1) series method and (2) full theory. The series method contains a polynomial and trigonometric portion with approximately 4,000 coefficients, that yields X and Y directly and the parameter $s + XY/2$ (Petit and Luzum, 2010; Capitaine and Wallace, 2008). The “full theory” approach uses the Fukushima-Williams angles for precession and the IAU 2000A_{R06} nutation theory to compute X and Y from the resulting bias–precession–nutation matrix (Wallace and Capitaine, 2006). The CIO locator s must still be computed using the series for the quantity

² $T_{\text{TT}} = (\text{JD}_{\text{TT}} - 2451545)/36525$ where JD_{TT} is the Julian date of TT.

$s + XY/2$. This method also has a high computational load and memory requirement due to the series for $s + XY/2$ and the several thousand term expression for nutation. In addition, the use of EOPs for a full accuracy frame transformation requires more memory for the storage of EOP data tables and the complication of updating the data table to retrieve the latest estimated values.

This section presents procedures for computing X, Y, s and EOPs which are faster than the standard methods and use less memory. Two effective simplifications of these X, Y, s calculations that have been previously published employ a truncated series approximation and tabulation/interpolation (Capitaine and Wallace, 2008; Coppola et al, 2009; Bradley et al, 2015). We introduce a third method, polynomial fitting, for both X, Y, s and EOPs and compare the performance with the other methods. Polynomial fitting is advantageous for applications where high-precision frame transformations may not be required or for on-board applications that have stringent memory requirements.

5.4.1 X, Y Series Approximation

The full accuracy series method consists of approximately 4,000 coefficients to compute X and Y directly and the parameter $s + XY/2$ (Petit and Luzum, 2010; Capitaine and Wallace, 2008). Each series contains a polynomial and trigonometric portion (Capitaine and Wallace, 2008)

$$q = q_0 + q_1 T_{\text{TT}} + q_2 T_{\text{TT}}^2 + q_3 T_{\text{TT}}^3 + q_4 T_{\text{TT}}^4 + q_5 T_{\text{TT}}^5 + \sum_i \sum_{j=0}^4 [(a_{s,j})_i \sin(\text{ARG}) + (a_{c,j})_i \cos(\text{ARG})] T_{\text{TT}}^j, \quad (5.10)$$

where q stands for X , Y , or $s + XY/2$ and ARG stands for combinations of the luni-solar and planetary parts of the fundamental arguments. The large number of coefficients and trigonometric evaluations make the full series evaluation burdensome, both on computation time and memory. However, series can be truncated to speed up computation time at the loss of some accuracy, as is shown by Capitaine and Wallace (2008), Bradley et al (2015), and Wallace and Capitaine (2006). Bradley et al (2015) demonstrates the achievable accuracy of coordinate transformations using truncated series for X and Y for applications on the surface of the Earth and at several orbital

altitudes. Coefficients for 4-, 6-, 15-, and higher-term series methods can be found in Capitaine and Wallace (2008) and Bradley et al (2015). Approximations for s are usually not beneficial because the value of s is on the order of milliarcseconds (below 10 mas before 2030), a magnitude many times smaller than the errors in X and Y approximations.

The truncated series approach requires very little computation time and memory while yielding moderate accuracy. This method is thus well suited for on-board satellite operations. An added benefit is that each set of coefficients provide X and Y through the year 2050, never having to be replaced through an upload via a ground station.

5.4.2 X, Y, s Interpolation

Tabulating and interpolating X , Y , and s is an option that can preserve full accuracy (to within the uncertainty of the IAU models) while reducing computation time and memory (Coppola et al, 2009; Bradley et al, 2015). The full theory or series methods may be used to generate X , Y , and s values at 0h TT each day and stored in a data table. The user can then interpolate the values to any desired time. The method is developed and analyzed in Coppola et al (2009) and used in this paper as one of the default recommended procedures. Bradley et al (2015) analyzes the accuracy of several interpolation methods for X, Y, s in the context of coordinate rotations. Maximum errors of 12 μm and 40 μm in LEO are achieved with 11th-order and 9th-order Lagrange interpolation, respectively. These are equivalent to 0.34 μas and 1.2 μas , respectively. One disadvantage compared to the truncated series representation, however, is the need for a modest amount of memory. If the series contains X , Y , s , and modified Julian date (MJD), one year's worth of data amounts to 1,460 values. This is still a reduction in memory compared to the full accuracy methods and is faster to evaluate, but applications constrained in memory may require a different approach.

5.4.3 X, Y, s Polynomial Fitting

A third option for computing X , Y , and s is to use a polynomial fit to a short time interval of full accuracy values. As with the tabulation and interpolation scheme, full accuracy values can be

computed ahead of time. A polynomial can then be fit to the desired time window of data (e.g., 30 days). This method does not contain any trigonometric terms, which are present in the truncated series method, and is low on memory demands. The disadvantages are the finite time interval the fit is valid for and the moderate accuracy. Table 5.1 provides maximum and root mean square (RMS) errors resulting from various fit intervals and degrees of polynomials. Polynomial fits were made each day from 1 Jan 2014 to 1 Jan 2015 using the three specified fit windows and polynomial degrees. Evaluations of X , Y , and s were made every 0.1 days to compare the polynomial fit to full accuracy calculations of each parameter.

Table 5.1: Maximum and RMS errors of X , Y , and s for various polynomial fit windows and degrees. Fits were made from 1 Jan 2015 to 1 Jan 2015 and error calculations were made every 0.1 days.

Parameter	Polynomial	15-Day Fit		30-Day Fit		60-Day Fit	
		Max	RMS	Max	RMS	Max	RMS
X (asec)	linear	0.14	0.050	0.19	0.058	0.28	0.081
	quadratic	0.17	0.034	0.11	0.052	0.14	0.057
	cubic	0.13	0.019	0.13	0.052	0.13	0.056
Y (asec)	linear	0.15	0.052	0.16	0.059	0.28	0.082
	quadratic	0.16	0.036	0.11	0.054	0.14	0.058
	cubic	0.14	0.020	0.11	0.054	0.11	0.057
s (mas)	linear	0.11	0.037	0.12	0.042	0.20	0.059
	quadratic	0.11	0.026	0.077	0.038	0.11	0.041
	cubic	0.093	0.014	0.086	0.039	0.071	0.040

Table 5.1 shows that the 30-day fits have the lowest maximum errors for quadratic and cubic polynomials, but exhibit a small increase in RMS error as compared to the 15-day fits. However, each fit window has roughly the same order of magnitude of errors (maximum of about 0.15 arcseconds and RMS around 0.055 arcseconds for X and Y). Because of this error stability, a fit window of 60 days or even longer may be the best choice for applications where computing, and possibly uploading, new fit parameters is undesirable. For the 30- and 60-day fits, using a quadratic polynomial yields a noticeable improvement over a linear fit, while the use of a cubic provides a relatively small benefit. The use of a 30-day fit with a quadratic polynomial is a good compromise between update frequency, accuracy, and number of terms needed. Figure 5.1 illustrates the errors in X and Y resulting from a 30-day quadratic fit for the same analysis window as Table 5.1.

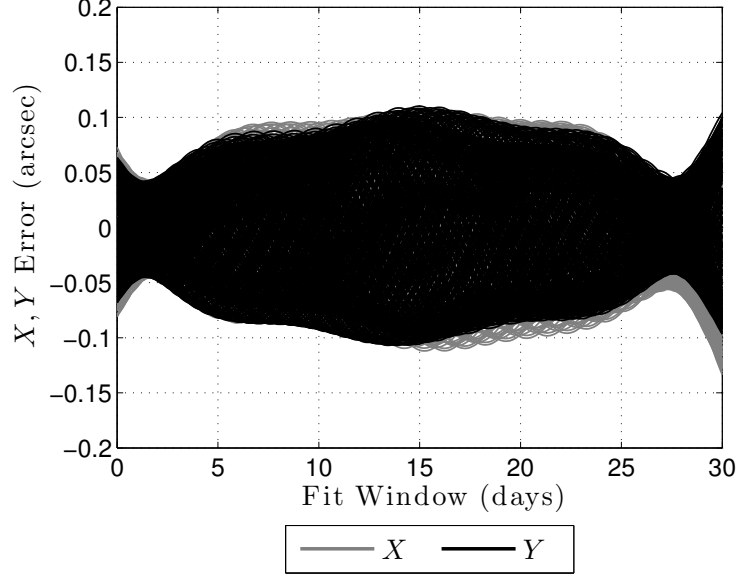


Figure 5.1: Polynomial fit errors for X and Y using a 30-day quadratic fit. Comparisons are made every 0.1 days from 1 Jan 2014 to 1 Jan 2015.

The errors shown in Fig. 5.1 remain consistent for the majority of the fit window and are comparable to the errors achieved with the 15-term series approximation (Bradley et al, 2015). Near the outer 1-2 days of the fit interval, however, the errors decrease and then increase sharply. While these outer portions are included in the maximum and RMS errors in Table 5.1, in practice it is advisable to not use the polynomial fit during the first and last 1-2 days. Note that as with the series representations, which also achieve X and Y errors near a tenth of an arcsecond, the CIO locator (s) can be ignored entirely because its value is at least an order of magnitude smaller than 0.1 arcseconds. We present results for fitting s here for completeness.

5.4.4 Speed and Memory Comparison: X, Y, s

Selecting a method for computing the bias-precession-nutation quantities X , Y , and s requires the consideration of computation time, memory usage, and accuracy. Table 5.2 compares each of the methods discussed in this paper and Bradley et al (2015) for computing X and Y . The CIO locator (s) is not included in the error column because the value of s is small compared to X

and Y . Interpolating X, Y, s values using a 9th-order Lagrange scheme is used as the baseline for relative timing and speed-up factors. The number of terms listed for the tabulation/interpolation methods is based on daily tabulated data for MJD_{TT} , X , Y , and s for 30 days. An additional number is contained in parentheses for the storage of a 1-year data table. The number of terms for the quadratic fit assumes 3 coefficients each for X and Y and the MJD of the epoch. While considerable consideration was given towards developing a fair runtime comparison, results may vary slightly due to differences in hardware, coding style, and programming language used.³ However, Table 5.2 is still useful as a guide to the relative computation time required by each method.

Table 5.2: Normalized time and memory required to compute CIO parameters X , Y , and s . Entries for relative timing and the number of terms are based on the computation of X , Y , and s while maximum errors are only reported for X and Y . The number of terms reported for the interpolation methods are for 30 days with a 1-year set of terms in parentheses. Normalized speed factor is the inverse of normalized computation time.

Method	Normalized Computation Time	Normalized Speed Factor	Coefficients	Max Error X, Y
Full Series	42	0.024	47,117	-
Full Theory	41	0.025	39,753	-
Interp 11th-Order	1.2	0.83	120 (1,460)	0.30, 0.34 μas
Interp 9th-Order	1.0	1.0	120 (1,460)	1.0, 1.1 μas
Interp 7th-Order	0.90	1.1	120 (1,460)	4.4, 4.8 μas
15-Term Series	0.17	6.0	21	0.13, 0.094 asec
6-Term Series	0.16	6.5	12	0.38, 0.27 asec
4-Term Series	0.023	43	6	0.9, 0.8, asec
30-Day Quad Fit	0.0071	141	7	0.13, 0.11 asec

Table 5.2 demonstrates the significant improvement in computation time and memory usage that the tabulation/interpolation methods have over the full theory and full series schemes. The tabulation/interpolation methods also have the ability maintain microarcsecond and even sub-microarcsecond agreement with the rigorous formulations. A greater improvement in speed and memory is achievable through the truncated series and quadratic polynomial fitting schemes at the cost of accuracy on the order of tenths of arcseconds.

³ Relative timing results were generated using MATLAB. Matrix operations were used for many of the multiplication and summation operations in the full series and full theory methods. The number of coefficients reported for the full series, full theory, and truncated series schemes include all coefficients for the fundamental arguments (i.e., Delaunay variables and mean heliocentric longitudes of the planets) and data files tab5.2a.txt, tab5.2b.txt, tab5.2d.txt, tab5.3a, and tab5.3b from <ftp://tai.bipm.org/iers/conv2010/chapter5/>.

5.4.5 EOP Polynomial Fitting

EOPs are an important component in the ITRS/GCRS transformation. Many applications have ready access to full EOP data tables which can be downloaded from one of several sources for the most up to date values and predictions (Vallado and Kelso, 2013). Some applications (e.g., on-board satellite operations) may want EOP data uploads occurring less frequently than once per day or desire less memory storage than a full data table spanning several weeks. As with the parameters X , Y , and s , EOPs can benefit from polynomial fits in the same way, utilizing a small number of terms to represent a week’s or a month’s worth of data. For ground-based simulations or processing, polynomial fits can be generated for rapid or final estimates of EOPs. Real-time EOPs for use on board a satellite require a fit to a combination of a few days worth of rapid EOP estimates and EOP predictions.

Table 5.3 provides maximum and root mean square (RMS) errors resulting from various EOP fit intervals and degrees of polynomials. Polynomial fits were made each day from 1 Jan 2014 to 1 Jan 2015 using the specified fit windows and polynomial degrees. Evaluations of each polynomial were made every 0.1 days and compared to cubic spline interpolation of each parameter. To put each error into perspective, x_p, y_p values are typically on the order of tens to hundreds of milliarcseconds and ΔUT1 varies between ± 0.8 seconds.⁴

Table 5.3: Maximum and RMS errors of x_p , y_p , and ΔUT1 for various polynomial fit windows and degrees. Fits were made from 1 Jan 2014 to 1 Jan 2015 and error calculations were made every 0.1 days using a cubic spline as truth.

Parameter	Polynomial	7-Day Fit		15-Day Fit		30-Day Fit	
		Max	RMS	Max	RMS	Max	RMS
x_p (mas)	linear	2.1	0.33	4.4	0.70	7.2	1.4
	quadratic	1.8	0.20	3.0	0.44	5.0	0.79
y_p (mas)	linear	1.6	0.24	2.2	0.47	5.1	1.0
	quadratic	1.4	0.15	1.6	0.30	2.6	0.53
ΔUT1 (ms)	linear	1.1	0.18	2.4	0.47	4.2	0.85
	quadratic	0.54	0.06	1.6	0.27	2.5	0.57

Table 5.3 demonstrates the increase in error as the length of fit window increases. However,

⁴ Care should be taken when fitting ΔUT1 due to the presence of leap seconds. A safer approach would be to fit UT1-TAI (which is continuous) and use this value in the transformation algorithms instead of UT1-UTC .

errors are approximately the same order of magnitude throughout. A significant improvement can be made by using a quadratic fit instead of linear for 15- and 30-day fit windows. For comparison, RMS errors for x_p and y_p are of the same order of magnitude as polar motion ocean tide corrections. This is approximately one order of magnitude greater than errors resulting from linear interpolation of an EOP data table. RMS errors for ΔUT1 quadratic fits are less than one order of magnitude greater than ocean tide corrections for ΔUT1 (which are in the 0.07 ms neighborhood). These errors are adequate, since polynomial fitting is meant for applications which require moderate accuracy and benefit from low memory usage.

A good compromise between accuracy and fit length is a 15-day fit window with a quadratic polynomial. Another EOP consideration for on-board operations is the method being used for X, Y, s . If X, Y, s data are also being uploaded to the spacecraft on some regular basis (either a data table for interpolation or a polynomial fit), the update frequency of both EOPs and X, Y, s should be considered together. Figures 5.2 and 5.3 illustrate the errors due to a quadratic fit for x_p, y_p and errors due to linear and quadratic fits of ΔUT1 , respectively.

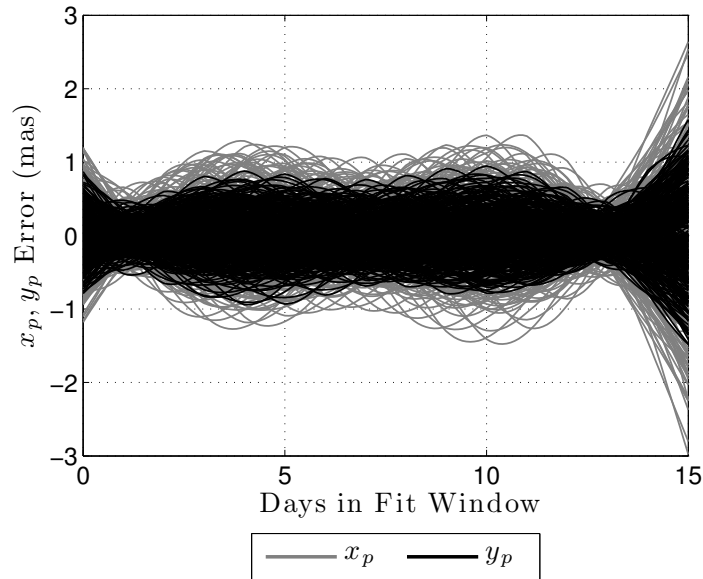


Figure 5.2: Polynomial fit errors for x_p and y_p using a 15-day quadratic fit. Comparisons are made every 0.1 days from 1 Jan 2014 to 1 Jan 2015.

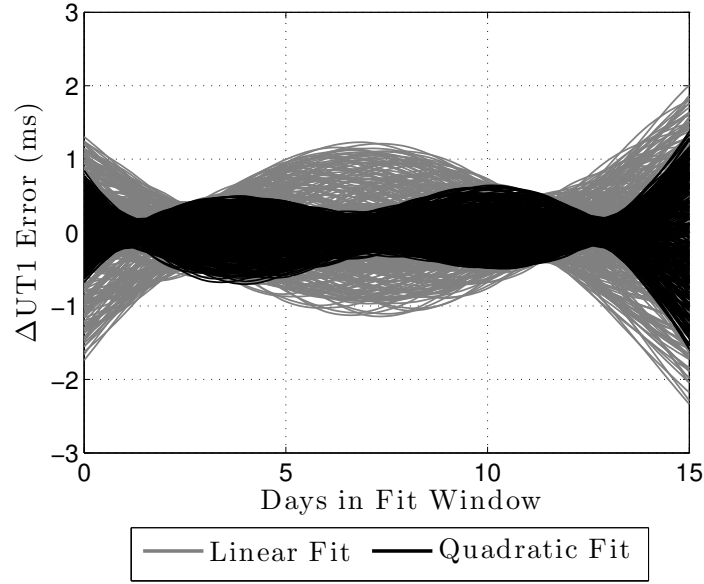


Figure 5.3: Polynomial fit errors for ΔUT1 using a 15-day linear and quadratic fit. Comparisons are made every 0.1 days from 1 Jan 2014 to 1 Jan 2015.

It should be noted the x_p , y_p , and ΔUT1 will become available on the modernized GPS signals in the near future (the current year is 2015). These three parameters and their derivatives, which include all zonal, diurnal, and semidiurnal effects, will be broadcast within the Civil Navigation (CNAV) message type 32 on the L2C and L5 GPS signals (GPSD, 2013). This is a great benefit for spacecraft flying GPS receivers, because it eliminates the need for EOPs to be supplied via other means. Unfortunately, the CNAV messages currently being broadcast are considered pre-operational. Until the CNAV messages are broadcast operationally by the whole constellation, GPS-based orbit determination as well as other applications may benefit from using polynomial fits (e.g., 15-day quadratic). Polynomial fits may also be of benefit for storing a backup set of EOPs in the event of failure to decode the relevant GPS navigation message.

5.5 Impact on Orbit Propagation in the GCRS

This section investigates the impact of the ITRS/GCRS transformation on orbit propagation around Earth in the GCRS. Specifically, the ITRS/GCRS transformation is required for the evaluation of the Earth’s non-spherical gravity field and the subsequent rotation of the computed acceleration during each orbit propagation step. A spherical harmonic gravity field is traditionally used with special perturbation techniques for orbit propagation. Using Cowell’s formulation, the total acceleration of a satellite in Earth orbit can be expressed as a sum of the different forces acting on the satellite

$$\ddot{\vec{r}} = \ddot{\vec{r}}_{\text{pm}} + \ddot{\vec{r}}_{\text{ns}} + \ddot{\vec{r}}_{\text{d}} + \ddot{\vec{r}}_{\text{3b}} + \ddot{\vec{r}}_{\text{srp}}, \quad (5.11)$$

where each acceleration is in the GCRS (Montenbruck and Gill, 2000; Vallado, 2013). Equation 5.11 contains accelerations due to the Earth as a point mass (pm), non-spherical gravity of Earth (ns), atmospheric drag (d), third-body gravity from the Sun and Moon (3b), and solar radiation pressure (srp). Most of these terms are most naturally computed and expressed in the GCRS. However, the evaluation of the non-spherical gravity field must be computed in the ITRS, because gravity models are tied to the respective central body’s body-fixed coordinate system. For Earth, the body-fixed system is the ITRS. The acceleration due to the non-spherical Earth is a function of the ITRS position of the satellite

$$\ddot{\vec{r}}_{\text{ns}}^{\text{ITRS}} = \frac{\partial U'(\vec{r}_{\text{ITRS}})}{\partial \vec{r}_{\text{ITRS}}}, \quad (5.12)$$

where U' is the gravitational potential of the Earth (excluding the point-mass contribution). Once the non-spherical gravitational acceleration is computed, it must be rotated from the ITRS to the GCRS for inclusion into the total acceleration by

$$\ddot{\vec{r}}_{\text{ns}} = \mathbf{T} \ddot{\vec{r}}_{\text{ns}}^{\text{ITRS}}. \quad (5.13)$$

The fidelity to which \mathbf{T} is constructed impacts orbit propagation. Applications of orbit propagation include high-precision post-processing orbit determination, propagation of a large population of objects for conjunction assessment, and on-board real-time navigation. Each of these applications

demand unique precision, memory, and computation time limitations. This section demonstrates the impact of EOP usage and bias–precession–nutation simplifications that support each level of requirements through the use of orbit propagations of four common orbit types: LEO at 400 km altitude, LEO at 800 km altitude, GPS, and GEO. Table 5.4 lists the propagation settings and set of osculating classical orbital elements for each orbit type. Every orbital element combination was used for propagations, beginning each day at noon from 1 June 2012 to 15 June 2012. Appropriate gravity field sizes for each orbit regime were implemented using the EGM2008 gravity field with the permanent tide included (Petit and Luzum, 2010). A fixed-step 8th-order Dormand and Prince integration scheme (DOPRI8) was used for numerical integration (Prince and Dormand, 1981). The month of June of 2012 was chosen for our study because a leap second occurs on 1 July, making the ΔUT1 parameter near its maximum value.

Table 5.4: Orbital elements and propagation settings. Each orbital element combination was used to propagate orbits beginning at noon each day from 1 June to 15 June 2012.

Orbit	Altitude (km)	i ($^\circ$)	Ω ($^\circ$)	e	ν_0 ($^\circ$)	Duration	Step (s)	Gravity Field
LEO	400	{25, 98.6}	{0, 90}	0.001	{0, 90}	6 hrs	15	90×90
LEO	800	{25, 98.6}	{0, 90}	0.001	{0, 90}	6 hrs	30	90×90
GPS	20,200	{55}	{0, 90}	0.001	{0, 90}	12 hrs	300	12×12
GEO	35,786	{0.001, 15}	{0, 90}	0.001	{0, 90}	3 days	600	8×8

Each reference, or truth, trajectory is generated using all EOPs interpolated with a cubic spline, EOP ocean tide corrections applied, and 11th-order Lagrange interpolation of full accuracy X , Y , and s values. The step size selected for each orbit was chosen slightly below the step size settled on by a variable-step DOPRI8(7) scheme. This step size is used by both the truth trajectory and comparison trajectories so that interpolation error is not introduced when performing error calculations. We compare the reference trajectories to propagations performed using simplified X, Y, s models (as discussed in Sect. 5.4) and ignoring EOPs all together. We also include the use of a 15-day quadratic fit of ΔUT1 , x_p , and y_p . A polynomial fit offers efficient storage and computation of EOPs when maximum accuracy is not required and memory is limited. The specification of “Interp X, Y ” indicates the tabulation and interpolation method for computing X and Y using a

9th-order Lagrange scheme. Included in this study is the use of Greenwich Mean Sidereal Time (GMST) alone for transforming between the ITRS and GCRS. This simplification is commonly used and taught in the astrodynamics community due to its ease of computation and implementation.

Table 5.5: Maximum orbit propagation errors for each orbit type and ITRS/GCRS transformation method. Position errors are given for different propagation durations. The reference propagation was performed using interpolated X, Y, s values and EOPs interpolated using a cubic spline with ocean tide corrections applied.

Orbit	Duration	6-term X, Y	Interp X, Y	6-term X, Y	4-term X, Y	GMST Only
		Fit $\Delta\text{UT1}, x_p, y_p$ $s = 0$	No EOPs $s = 0$	No EOPs $s = 0$	No EOPs $s = 0$	
LEO (400 km)	1 min	0.05 mm	0.16 mm	0.16 mm	0.2 mm	7.7 cm
	15 min	1.2 cm	2.3 cm	2.4 cm	4.6 cm	16 m
	6 hrs	0.75 m	1.1 m	1.1 m	2.9 m	1.1 km
LEO (800 km)	1 min	0.04 mm	0.10 mm	0.11 mm	0.16 mm	6.1 cm
	15 min	0.93 cm	1.8 cm	1.8 cm	3.7 cm	13 m
	6 hrs	0.64 m	0.85 m	0.90 m	2.5 m	930 m
GPS	15 min	0.05 mm	0.12 mm	0.12 mm	0.23 mm	7.7 cm
	12 hrs	5.8 cm	10 cm	11 cm	23 cm	83 m
GEO	1 day	2.0 cm	8.6 cm	8.5 cm	9.7 cm	28 m
	3 days	5.7 cm	53 cm	53 cm	57 cm	78 m

The results in Table 5.5 can be summarized by the following:

- Using GMST only, therefore ignoring precession–nutation and EOPs entirely, results in large propagation errors. A similar computation cost of the 4- or 6-term X, Y approximation (along with the Earth rotation angle (ERA)) improves the propagation accuracy significantly.
- For durations shown, EOPs (and the CIO locator, s) do not significantly impact orbit propagations. This is especially true for short durations.
- Using analytic simplifications to compute X and Y does not yield significant propagation errors and are similar in complexity and computation time to using GMST only. They also eliminate the need for data tables of X and Y to be stored on-board a spacecraft.
- The 4-term X, Y approximation does not yield drastically different results as compared to the 6-term representation. But, using the 2 extra terms approximately doubles the accuracy

for LEO and GPS orbits. Therefore, the 6-term X, Y approximation is recommended for precession and nutation.

- Given that a quadratic fit of X and Y using a 30-day window yields errors less than the 6-term series approximation for X and Y , the quadratic fit will produce similar propagation errors.
- If even greater accuracy is desired, using X, Y, s interpolation and either nearest, linear interpolation, or a linear fit for EOPs is sufficient.

In conclusion, use of the 6-term X, Y approximation and exclusion of s and all EOPs is recommended for nearly all applications. This method is computationally efficient and eliminates the memory requirement for storing data tables of X, Y, s and EOPs. This method is sufficient even for precise, real-time orbit determination (e.g., using GPS measurements in LEO every 10-60 seconds).

5.6 Orbit Propagation in the ITRS

This section discusses the benefits and implementation of performing orbit propagation in the ITRS instead of the GCRS. Typically, orbit determination is performed by estimating the inertial position and velocity of the satellite. However, this requires a rotation of the satellite position to the ITRS when evaluating the spherical harmonic gravity field, as discussed above, and when computing an expected pseudorange measurement from either a ground station or GNSS satellite. If the application requires a high level of accuracy, this transformation must be done precisely, requiring extra data tables for EOPs and precession–nutation. Using the ITRS for orbit propagation and orbit determination avoids some of these complications and is potentially preferred when the spacecraft position is required in the ITRS (instead of the GCRS) and measurements are ground– or GNSS–based. The following list provides a set of advantages and additional costs of using the ITRS instead of the GCRS for state estimation and orbit propagation.

Advantages

- Eliminates requirement for precise rotation of ground station and GNSS positions from ITRS to GCRS for range and pseudorange measurement processing.
- Eliminates requirement for rotation of satellite position to ITRS for gravity field evaluation.

Costs

- Requires the rotation of small perturbing forces (e.g., third-body and solar radiation pressure) from the GCRS to the ITRS. However, this may be done with a simple rotation.
- Requires the addition of Coriolis and centripetal accelerations to the force model, which requires low-fidelity knowledge of x_p and y_p .

5.6.1 Velocity Transformation

To assess the accuracy of ITRS orbit propagation implementations, an accurate velocity transformation between the GCRS and ITRS is necessary. Reference trajectories are generated using orbit propagations performed in the GCRS. The initial state for each corresponding ITRS propagation requires a transformation of the velocity vector. Any error in the initial velocity can lead to large errors in the propagated trajectory. This section presents both simplified and detailed approaches for transforming velocity vectors. A simplified implementation for transforming velocity is presented in Vallado (2013) (and similarly in Montenbruck and Gill (2000)), which accounts only for the Earth's proper rotation about its rotation axis by transforming the objects position vector into the Terrestrial Intermediate Reference System (TIRS) when applying the angular velocity cross product where

$$\vec{v}_{\text{GCRS}} = \mathbf{C} \mathbf{R} \left\{ \mathbf{W} \vec{v}_{\text{ITRS}} + \vec{\omega}_{\oplus} \times \mathbf{W} \vec{r}_{\text{ITRS}} \right\}, \quad (5.14)$$

and vice-versa

$$\vec{v}_{\text{ITRS}} = \mathbf{W}^T \left\{ \mathbf{R}^T \mathbf{C}^T \vec{v}_{\text{GCRS}} - \vec{\omega}_{\oplus} \times \mathbf{W} \vec{r}_{\text{ITRS}} \right\}. \quad (5.15)$$

In this form, the rotation vector of the Earth $\vec{\omega}_\oplus$ is given by

$$\vec{\omega}_\oplus = \begin{bmatrix} 0 & 0 & \omega_\oplus \end{bmatrix}^T, \quad (5.16)$$

where the instantaneous rotation rate of the Earth ω_\oplus is⁵

$$\omega_\oplus = 7.2921151467064 \times 10^{-5} \left\{ 1 - \frac{\text{LOD}}{86400} \right\} \text{ rad/s}. \quad (5.17)$$

The velocity transformation above is a good approximation to the full equations, which is demonstrated later in this section. However, our analysis requires the use of a more precise velocity transformation. The complete velocity transformation requires the time-derivative of the transformation matrix and is written

$$\vec{v}_{\text{GCRS}} = \mathbf{T} \vec{v}_{\text{ITRS}} + \dot{\mathbf{T}} \vec{r}_{\text{ITRS}}, \quad (5.18)$$

and

$$\vec{v}_{\text{ITRS}} = \mathbf{T}^T \vec{v}_{\text{GCRS}} + \dot{\mathbf{T}}^T \vec{r}_{\text{GCRS}}. \quad (5.19)$$

To simplify the derivative calculations, the three z-axis rotations (θ_{ERA} , s , and s') can be combined as is done in Capitaine et al (2006). The new matrices, denoted by an asterisk in the subscript, are then

$$\mathbf{C}_* = \begin{bmatrix} 1 - aX^2 & -aXY & X \\ -aXY & 1 - aY^2 & Y \\ -X & -Y & 1 - a(X^2 + Y^2) \end{bmatrix} \quad (5.20)$$

$$\mathbf{R}_* = \mathbf{R}_3(s)\mathbf{R}_3(-\theta_{\text{ERA}})\mathbf{R}_3(-s') = \mathbf{R}_3(-\phi) \quad (5.21)$$

$$\mathbf{W}_* = \mathbf{R}_2(x_p)\mathbf{R}_1(y_p) \quad (5.22)$$

where

$$\phi \equiv \theta_{\text{ERA}} + s' - s. \quad (5.23)$$

⁵ This definition is based on Aoki et al (1982), which defines the rotational period of the Earth as 86164.09890369732 seconds of UT1. An overview of the computation of rotation rate and definitions of UT1 and LOD can be found here:

<http://hpiers.obspm.fr/eop-pc/earthor/ut1lod/UT1.html>

The new matrix definitions result in the same overall transformation matrix

$$\mathbf{T} = \mathbf{T}_* = \mathbf{C}_* \mathbf{R}_* \mathbf{W}_*, \quad (5.24)$$

where rotation matrices are now in a more manageable form for computing their respective time-derivatives. The time derivative of the transformation matrix and its transpose needed in Eqs. 5.18 and 5.19 are given by

$$\dot{\mathbf{T}} = \dot{\mathbf{T}}_* = \dot{\mathbf{C}}_* \mathbf{R}_* \mathbf{W}_* + \mathbf{C}_* \dot{\mathbf{R}}_* \mathbf{W}_* + \mathbf{C}_* \mathbf{R}_* \dot{\mathbf{W}}_*, \quad (5.25)$$

and

$$\dot{\mathbf{T}}^T = \dot{\mathbf{T}}_*^T = \dot{\mathbf{W}}_*^T \mathbf{R}_*^T \mathbf{C}_*^T + \mathbf{W}_*^T \dot{\mathbf{R}}_*^T \mathbf{C}_*^T + \mathbf{W}_*^T \mathbf{R}_*^T \dot{\mathbf{C}}_*^T. \quad (5.26)$$

where the time-derivative of each matrix is detailed explicitly in Sec. 5.9.1. The rates of change of X , Y , θ_{ERA} , s' , s , x_p , and y_p are required for the rigorous computation of $\dot{\mathbf{T}}$. Figure 5.4 shows the magnitude of each parameter's rate of change from the year 2000 to 2050 to demonstrate the relative importance of each parameter in the total transformation derivative. Note that the derivative of θ_{ERA} is simply ω_{\oplus} . The remaining derivatives are computed using a 4th-order (i.e., 5-point) central difference scheme (Fornberg, 1988). A step size of 900 seconds was used for \dot{X} , \dot{Y} , and \dot{s} and a step of 90 seconds is used for the computation of \dot{x}_p and \dot{y}_p .⁶ Ocean tide corrections are incorporated into x_p and y_p when computing their derivatives, which actually dominate the derivative calculation due to their relatively steep oscillations compared to the slowly varying uncorrected values.

Figure 5.4 reveals that the rotation rate of the Earth through the angle θ_{ERA} (i.e., ω_{\oplus}) is approximately 7 orders of magnitude greater than the next largest derivatives, \dot{X} and \dot{Y} . Polar motion rates are then approximately an order of magnitude smaller than the CIP coordinate rates. Finally, the rates \dot{s} and \dot{s}' are several orders of magnitude smaller than polar motion. This provides an idea about the importance of each parameter, and thus each matrix derivative, in the transformation of velocity between the ITRS and GCRS.

⁶ Different step sizes for the numerical differentiation of each parameter were chosen due to the different variability of each parameter. Each step size was optimized to minimize machine round-off error while maximizing accuracy of the derivative calculation.

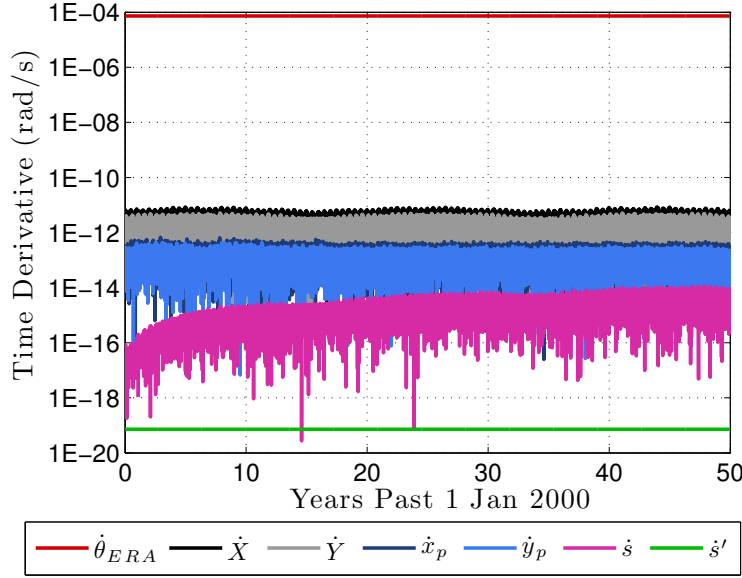


Figure 5.4: Comparison of the time derivative of individual CIO and EOP parameters over time. Derivatives are absolute values and are computed using a 4th-order central difference technique with optimized step sizes for each parameter.

The next step is to analyze the magnitude of each matrix time derivative directly for \mathbf{W}_* , \mathbf{R}_* , and \mathbf{C}_* . This can be done by computing the rotation vector of each contributing factor. This is analogous to the Earth rotation vector used in Eqs. 5.14 and 5.15. The rotation vector equivalent of $\dot{\mathbf{T}}_*^T$ is given by

$$\dot{\mathbf{T}}_*^T \mathbf{T}_* = \begin{bmatrix} \vec{\omega} \end{bmatrix}_{\times} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \quad (5.27)$$

where $\begin{bmatrix} \vec{\omega} \end{bmatrix}_{\times}$ is a skew symmetric matrix (SSM) representing the cross product ($\vec{\omega} \times$). This SSM is the Earth's instantaneous rotation vector and is the same one needed for the computation of the Coriolis and centripetal accelerations discussed in Sec. 5.6.2. Equation 5.27 can then be written out as

$$\dot{\mathbf{T}}_*^T \mathbf{T}_* = \dot{\mathbf{W}}_*^T \mathbf{W}_* + \mathbf{W}_*^T \dot{\mathbf{R}}_*^T \mathbf{R}_* \mathbf{W}_* + \mathbf{W}_*^T \mathbf{R}_*^T \dot{\mathbf{C}}_*^T \mathbf{C}_* \mathbf{R}_* \mathbf{W}_*. \quad (5.28)$$

Each group containing one matrix derivative can be computed separately and used to form its own

SSM such that

$$\begin{bmatrix} \vec{\omega} \end{bmatrix}_{\times} = \begin{bmatrix} \vec{\omega}_W \end{bmatrix}_{\times} + \begin{bmatrix} \vec{\omega}_R \end{bmatrix}_{\times} + \begin{bmatrix} \vec{\omega}_C \end{bmatrix}_{\times}. \quad (5.29)$$

Fortunately, each SSM component is able to be simplified greatly into

$$\vec{\omega}_W = \begin{bmatrix} \dot{y}_p \\ \dot{x}_p \cos y_p \\ \dot{x}_p \sin y_p \end{bmatrix} \quad (5.30)$$

$$\vec{\omega}_R = \dot{\phi} \begin{bmatrix} -\sin x_p \\ \cos x_p \sin y_p \\ -\cos x_p \cos y_p \end{bmatrix} \quad (5.31)$$

$$\vec{\omega}_C = \frac{\mathbf{W}_*^T \mathbf{R}_*^T}{Z} \begin{bmatrix} qX + \dot{Y} \\ qY - \dot{X} \\ -qZ \end{bmatrix} \quad (5.32)$$

where $q = a(Y\dot{X} - X\dot{Y})$. We now have a set of simple equations to compute the contribution of each transformation matrix derivative (polar motion, proper motion, and bias–precession–nutation) expressed in terms of the standard CIO parameters, EOPs, and their time rates of change. This process of creating individual rotation vector components for Earth’s net rotation is based upon Capitaine et al (2006), which presents a similar quantity (using a slightly different transformation definition) towards the goal of developing dynamical equations of Earth’s rigid body rotation. The resulting rotation vector components of Capitaine et al (2006), however, are different enough from $\dot{\mathbf{T}}_*^T \mathbf{T}_*$ to warrant the derivation presented here.

Figure 5.5 depicts the magnitude of each rotation vector component (ω_W , ω_R , and ω_C) between the year 2000 and 2050. A similar trend as in Fig. 5.4 is depicted where Earth’s proper motion (ω_R) contributes to the total rotation the most and the rotation due to polar motion (ω_W) contributes the least.

We now have a detailed expression for transforming velocity vectors between the ITRS and GCRS and an understanding of the relative contribution of each component of Earth’s rotation vec-

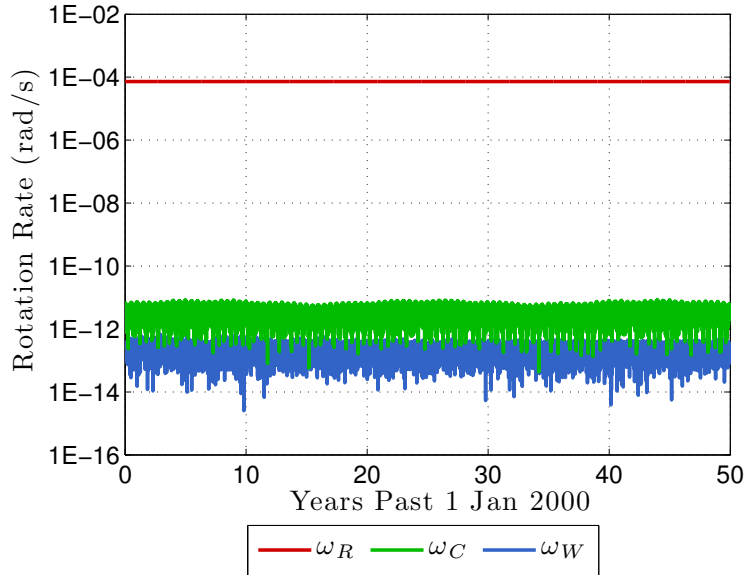


Figure 5.5: Magnitudes of each component of Earth’s total rotation vector from the year 2000 to 2050.

tor. This detailed velocity transformation is necessary for comparing orbit propagations performed in the ITRS versus propagations performed in the GCRS.

We should also consider the impact that various simplifications to this detailed model have on velocity transformations for cases that do not require the highest accuracy. The computation of the full Earth rotation vector is notably more time consuming than using the basic proper motion assumption ($\vec{\omega}_{\oplus}$) due to the computation of the derivatives of each CIO parameter and EOP.⁷

For this analysis, velocity transformations from the GCRS to the ITRS are performed at points along each orbit described in Table 5.4. These velocity transformations are made every 5 degrees in true anomaly and every 4.66 hours in time between 1 Jan 2000 and 1 Jan 2015 to avoid repeating times and obtain good spatial and temporal resolution. The reference, or “truth”, ITRS velocity vector is computed using the full form of Eq. 5.26, including EOP ocean tide corrections and the derivative of each parameter in the model. Table 5.6 shows the maximum errors of the

⁷ Previously, we noted that the GPS constellation will broadcast EOPs and their rates as part of the CNAV message in the near future. For a satellite with a GPS receiver, this would eliminate the computation of EOP ocean tide corrections and \dot{x}_p, \dot{y}_p . However, the computation of \dot{X} and \dot{Y} , which are an order of magnitude greater than polar motion would still need to be computed.

transformed ITRS velocities due to several possible implementations. Each reported error is the error in the velocity magnitude. The results show the following:

- Excluding EOP ocean tide corrections results in $\mu\text{m/s}$ errors.
- Excluding $\vec{\omega}_W$, \dot{s} , and \dot{s}' has little impact on the velocity magnitude.
- Excluding the LOD parameter, which modifies ω_\oplus , is acceptable, especially if $\vec{\omega}_C$ and $\vec{\omega}_W$ are also excluded.

We conclude that the use of Eqs. 5.14 and 5.15 to transform velocity yields sufficient accuracy for most applications and the exclusion of the LOD are both warranted. If a higher accuracy is required, the best choice is to add the $\vec{\omega}_C$ component (precession–nutation rates, $\dot{\mathbf{C}}$). Note that even precise orbit determination of spacecraft in LEO produces velocity uncertainties on the order of mm/s, further justifying the use of the simplified recommendation made here.

Table 5.6: Maximum velocity transformation errors (magnitude) due to simplifications of Earth’s rotation vector. Values are the maximum error in velocity magnitude found in simulations covering transformations made every 5 degrees in true anomaly and every 4.66 hours from 1 Jan 2000 to 1 Jan 2015.

Orbit	$\vec{\omega}_R, \vec{\omega}_C, \vec{\omega}_W$	$\vec{\omega}_R, \vec{\omega}_C$	$\vec{\omega}_R$	$\vec{\omega}_R$
	No EOP tides	No EOP tides $\dot{s}, \dot{s}' = 0$	No EOP tides $\dot{s}, \dot{s}' = 0$	No EOP tides LOD, $\dot{s}, \dot{s}' = 0$
LEO (400 km)	4.9 $\mu\text{m/s}$	5.2 $\mu\text{m/s}$	58 $\mu\text{m/s}$	58 $\mu\text{m/s}$
LEO (800 km)	5.2 $\mu\text{m/s}$	5.5 $\mu\text{m/s}$	61 $\mu\text{m/s}$	62 $\mu\text{m/s}$
GPS	19 $\mu\text{m/s}$	23 $\mu\text{m/s}$	0.23 mm/s	0.23 mm/s
GEO	30 $\mu\text{m/s}$	36 $\mu\text{m/s}$	0.36 mm/s	0.37 mm/s

5.6.2 ITRS Orbit Propagation Analysis and Results

This section details the additional force model terms needed for orbit propagation in the ITRS and presents orbit propagation accuracy as a function of the fidelity of these models. Propagating a satellite orbit and performing orbit determination (OD) using ITRS position and velocity has several advantages over the use of the GCRS for certain applications, namely near real-time OD through the use of GNSS or ground-based measurements. Montenbruck and Ramos-Bosch (2008) successfully demonstrate the use of ITRS-based orbit propagation for OD of several flight missions using a GPS receiver.

The consequence from using a rotating frame when propagating an orbit is the need to include Euler, Coriolis, and centripetal accelerations into the force model. The additional consideration is that any other acceleration that is naturally expressed and computed in a different frame (e.g., the GCRS) must be rotated to the ITRS. The total acceleration of the satellite expressed in the ITRS, analogous to Eq. 5.11, is given by

$$\ddot{\vec{r}}_{\text{ITRS}} = \ddot{\vec{r}}_{\text{pm}} + \ddot{\vec{r}}_{\text{ns}} + \ddot{\vec{r}}_{\text{d}} + \mathbf{T}^T(\ddot{\vec{r}}_{\text{3b}} + \ddot{\vec{r}}_{\text{srp}}) + \ddot{\vec{r}}_{\text{cc}}, \quad (5.33)$$

where the acceleration due to the Earth as a point mass and non-spherical gravity field are computed directly in the ITRS. Atmospheric drag can be computed in the ITRS by using the satellite's ITRS velocity vector for the relative velocity vector needed in the calculation (see Sec. 5.9.2.7 for details). Equation 5.33 shows that third-body (e.g., lunisolar) and solar radiation pressure accelerations (for example) are computed in the GCRS and must be transformed to the ITRS for inclusion into the total acceleration. The last acceleration term in Eq. 5.33, denoted $\ddot{\vec{r}}_{\text{cc}}$, accounts for rotating frame being used and is given by

$$\ddot{\vec{r}}_{\text{cc}} = -\dot{\vec{\omega}} \times \vec{r}_{\text{ITRS}} - 2\vec{\omega} \times \vec{v}_{\text{ITRS}} - \vec{\omega} \times \vec{\omega} \times \vec{r}_{\text{ITRS}}, \quad (5.34)$$

which is composed of three components: (1) the tangential acceleration (also known as the Euler force) $(-\dot{\vec{\omega}} \times \vec{r}_{\text{ITRS}})$, (2) Coriolis acceleration $(-2\vec{\omega} \times \vec{v}_{\text{ITRS}})$, and (3) centripetal acceleration $(-\vec{\omega} \times \vec{\omega} \times \vec{r}_{\text{ITRS}})$. The study presented here provides an in depth investigation into the effects of $\vec{\omega}$

simplifications on orbit propagation as well as demonstrates that using the rigorous computation of $\vec{\omega}$ yields accurate long-term propagations in the ITRS. We also provide all of the necessary partial derivatives to be able to perform ITRS-based OD using a Kalman filter in Sec. 5.9.2. This includes measurement sensitivity matrix partials for kinematic GPS point solution measurements and pseudorange observations, as well as the state transition mapping matrix partials.

As in Sec. 5.6.1, $\vec{\omega}$ is the instantaneous rotation vector of the Earth. For the computation of acceleration in Eq. 5.34, however, we need the negative of Eqs. 5.28 and 5.29 such that

$$\left[\vec{\omega} \right]_{\times} = - \left(\dot{\mathbf{T}}_*^T \mathbf{T}_* \right). \quad (5.35)$$

Montenbruck and Ramos-Bosch (2008) recommend a simplified Earth rotation vector

$$\vec{\omega} \approx \mathbf{W}^T \vec{\omega}_{\oplus}, \quad (5.36)$$

which is equivalent to $-\vec{\omega}_R$ without \dot{s} or \dot{s}' . The inclusion of \mathbf{W}^T transforms the rotation axis, $\vec{\omega}_{\oplus}$, from the Terrestrial Intermediate Reference System (TIRS) to the ITRS.

Figure 5.6 illustrates the magnitude of the Euler/Coriolis/centripetal acceleration compared to other acceleration models in order to demonstrate its importance in ITRS orbit propagation.⁸

The magnitude of the acceleration is approximately within an order of magnitude of the point mass acceleration.

Figure 5.7 depicts the magnitude of the Euler/Coriolis/centripetal acceleration an 800-km LEO orbit at four different inclinations. Non-equatorial orbits exhibit a sinusoidal pattern, as expected, where higher inclination orbits show a greater variation in the acceleration. The peaks of the inclined orbit accelerations that rise above the equatorial orbit acceleration are caused by: (1) higher ITRS velocities due to the inclination and (2) velocity vectors nearly perpendicular to the Earth's rotation vector. This behavior indicates the importance of choosing an appropriate step size for fixed-step numerical integration or the use of a variable-step scheme.

The orbits listed in Table 5.4 are used to investigate the accuracy of ITRS orbit propagation

⁸ Spacecraft properties: $C_r = 1.2$ (coefficient of reflectivity), $C_d = 2.2$ (drag coefficient), area-to-mass ratio of 0.01 m²/kg. Harris-Priester atmospheric density model.

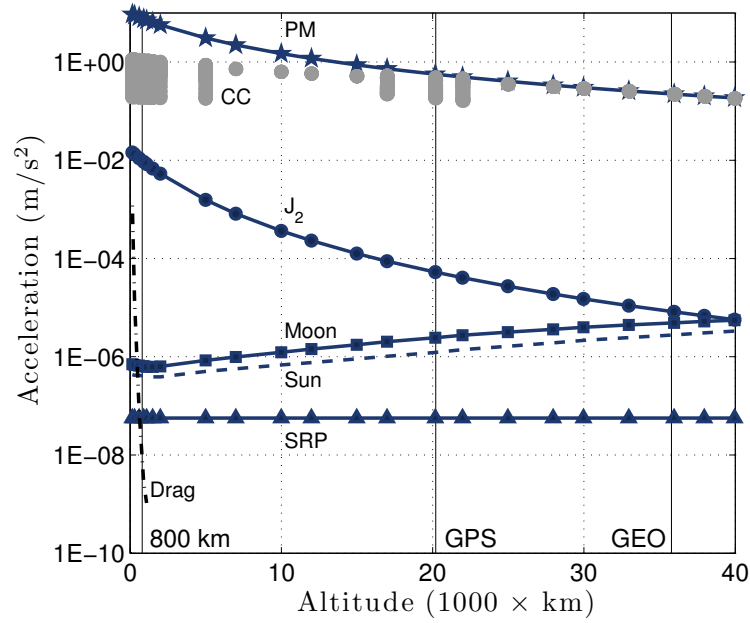


Figure 5.6: Comparison of force models at different altitudes. Plot style based on Montenbruck and Gill (2000).

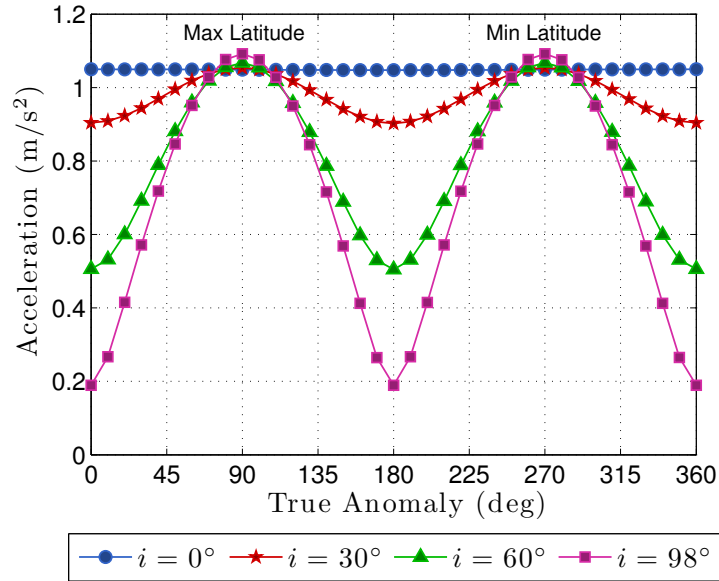


Figure 5.7: Behavior of Euler/Coriolis/centripetal acceleration for an 800-km altitude LEO orbit with several different orbit inclinations.

and to determine the effect of Euler/Coriolis/centripetal acceleration modeling on ITRS propagation accuracy. Each reference orbit is generated using orbit propagation in the GCRS with the

same settings as was done in Sec. 5.5. Each ITRS orbit propagation and associated maximum position error are computed using the following methodology:

- (1) The initial GCRS position and velocity are transformed to the ITRS using the full accuracy procedure.⁹
- (2) Orbit propagation is carried out in the ITRS using the schemes specified in each column of Table 5.7 for the computation of the Euler/Coriolis/centripetal acceleration.
- (3) The ITRS position at each propagation step is transformed to the GCRS using the full accuracy procedure.
- (4) Each position is compared to the reference orbit position and the maximum 3D position error is displayed in Table 5.7.

The results presented in Table 5.7 yield several conclusions. First, performing orbit propagation in the ITRS using the rigorous formulation for Earth's rotation vector, Eq. 5.35, and including the Euler acceleration in Eq. 5.34 (i.e., referring to the leftmost column of results) yields accurate propagations for relatively long period of time (hours for LEO and days for GEO).

⁹ GCRS to ITRS transformation for epoch state: X, Y, s tabulation/interpolation using 9th-order Lagrange scheme, all EOPs used and interpolated using a 5th-order Lagrange scheme, EOP ocean tide corrections applied, all EOPs and their rates included in the velocity transformation.

Table 5.7: Maximum ITRS orbit propagation position errors for each orbit type and Earth rotation vector computation method. Position errors are given for multiple propagation durations. Each reference propagation was performed in the GCRS using all EOPs and their respective ocean tide corrections.

Orbit	Duration	$\vec{\omega}_R, \vec{\omega}_C, \vec{\omega}_W, \vec{\omega}$	$\vec{\omega}_R$	$\vec{\omega}_R$	$\vec{\omega}_\oplus$
		$\dot{s}' = 0$	No EOP tides $\dot{s}, \dot{s}' = 0$	No EOP tides LOD, $\dot{s}, \dot{s}' = 0$	LOD = 0
LEO (400 km)	1 min	0.064 mm	0.20 mm	0.21 mm	4.1 mm
	15 min	0.93 mm	4.9 cm	4.9 cm	1.1 m
	6 hrs	6.6 cm	4.1 m	4.1 m	81 m
LEO (800 km)	1 min	0.068 mm	0.20 mm	2.0 mm	4.0 mm
	15 min	0.98 mm	4.7 cm	4.7 cm	0.98 m
	6 hrs	6.8 cm	4.3 m	4.3 m	85 m
GPS	15 min	5.2 mm	2.2 cm	2.2 cm	0.41 m
	12 hrs	0.66 m	27 m	28 m	46 m
GEO	1 day	0.41 m	37 m	43 m	420 m
	3 days	1.2 m	110 m	130 m	1300 m

The assumption that Earth's rotation axis is aligned with the z-axis of the ITRS (i.e., using $\vec{\omega}_\oplus$) produces orbit propagation errors that grow rapidly in every orbit regime. However, the inclusion of only the polar motion matrix, even while excluding ocean tide corrections, their rates, as well as LOD, \dot{s} , and \dot{s}' , produces errors less than 5 cm in LEO and less than 2.5 cm in GPS over 15 minutes. This refers to the second column from the right in Table 5.7 and is equivalent to Eq. 5.36. Many satellites which employ a GPS receiver for orbit determination take record pseudorange and carrier-phase observations at 10- or 30-second intervals. At this update frequency, the use of $\vec{\omega}_R$ with ITRS orbit propagation does not introduce any meaningful error into the propagation. This viable ITRS-based implementation for orbit propagation, and thus orbit determination, eliminates the need to compute precession–nutation quantities and only requires rough values for x_p and y_p , such as polynomial fits. Section 5.9.2 contains the necessary partial derivatives for performing orbit determination with ITRS position and velocity instead of the traditional GCRS. Measurement sensitivity matrix partials are given for both kinematic GPS position/clock bias measurements and pseudorange measurements. State transition mapping matrix partials are also given for a variety of force models.

5.7 Conclusion

In this paper, we have investigated the impact of bias–precession–nutation simplifications and EOP usage on orbit propagation for several orbit regimes. Specifically, their impact is contained to the evaluation and rotation of the acceleration vector due to Earth’s gravity field for inertial orbit propagation. Simplifications to the CIO-based ITRS/GCRS transformation method, such as tabulation/interpolation and truncated series, are discussed and short-term polynomial fits to CIP coordinates (X and Y) are introduced. The results demonstrate that EOPs as well as bias–precession–nutation simplifications have little effect on orbit propagation at any altitude. Ignoring EOPs entirely and using a 6-term series approximation for X and Y yield orbit propagation errors of 1 m after 6 hours in LEO and only 53 cm after 3 days in GEO. This robustness to EOPs and bias–precession–nutation simplifications allows for a faster and lower memory ITRS/GCRS transformation. It also makes orbit propagation in real-time and in the future robust to errors in EOP predictions or excluding them entirely. Note, however, that an over simplified approach of using only a GMST rotation to transform between the ITRS and GCRS results in large orbit propagation errors on the order of 1 km in LEO and 80 m in GEO after 6 hours and 3 days, respectively. Therefore, the 4-term series approximation of X and Y should be used at the minimum. The CIO locator, s , can be safely ignored for all inertial orbit propagations about Earth.

In addition to orbit propagation in an inertial system, we detail propagation in the rotating ITRS. This requires additional acceleration terms in the force model to account for centripetal, Coriolis, and Euler forces. The time-derivative of the ITRS/GCRS transformation matrix for velocity transformations and the ITRS rotation vector are both presented in detail. The results demonstrate that precise orbit propagation can be performed in the ITRS, but it comes with a high computational burden. The high computational load is mainly due to the time-derivatives of bias–precession–nutation and polar motion, along with the Euler force. However, a simplified computation of the ITRS rotation vector and neglect of the Euler force achieves orbit propagation errors less than 5 cm in LEO and 2.2 cm for GPS orbits in 15 minutes. Since propagation in the

ITRS is mainly beneficial to GPS-based orbit determination, typical propagation durations between measurements is only 30 seconds, making ITRS propagation tractable. This simplified approach only requires approximate polar motion coordinates, which can be provided, for example, via short data tables or 15-day quadratic fits. In the near future, polar motion parameters and the $\Delta UT1$ time difference will be broadcast operationally as part of GPS civilian navigation messages on the modernized L2C and L5 signals.

Overall, this work provides simplifications to the ITRS/GCRS transformation that enable orbit propagation without meaningful loss of accuracy in addition to improved efficiency and lower memory usage. We detail procedures for orbit propagation in both the inertial GCRS and rotating ITRS. The results allow astrodynamists and scientists to employ efficient algorithms in simulations, on-board and post-processing orbit determination, and conjunction assessment.

5.8 Acknowledgements

A portion of this research was made possible with Government support under and awarded by DoD, Air Force Office of Scientific Research, National Defense Science and Engineering Graduate (NDSEG) Fellowship, 32 CFR 168a. We would like to thank Aurore Sibois for her continuous support and useful conversations throughout this work.

5.9 Appendix

5.9.1 ITRS/GCRS Transformation Derivatives

This section contains the time derivatives of each rotation matrix in the ITRS/GCRS transformation. Specifically, the derivatives are given for each of the modified rotations in Eqs. 5.20–5.22. Each C and S refer to cosine and sine, respectively. The time derivative of the modified polar motion

matrix is given by

$$\dot{\mathbf{W}}_* = \begin{bmatrix} -S(x_p)\dot{x}_p & C(x_p)S(y_p)\dot{x}_p + S(x_p)C(y_p)\dot{y}_p & -C(x_p)C(y_p)\dot{x}_p + S(x_p)S(y_p)\dot{y}_p \\ 0 & -S(y_p)\dot{y}_p & C(y_p)\dot{y}_p \\ C(x_p)\dot{x}_p & S(x_p)S(y_p)\dot{x}_p - C(x_p)C(y_p)\dot{y}_p & -S(x_p)C(y_p)\dot{x}_p - C(x_p)S(y_p)\dot{y}_p \end{bmatrix}, \quad (5.37)$$

and the derivative of the modified Earth rotation matrix (or proper motion matrix) is

$$\dot{\mathbf{R}}_* = \dot{\phi} \begin{bmatrix} -S(\phi) & -C(\phi) & 0 \\ C(\phi) & -S(\phi) & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad (5.38)$$

where

$$\phi = \theta_{\text{ERA}} + s' - s, \quad (5.39)$$

$$\dot{\phi} = \omega_{\oplus} + \dot{s}' - \dot{s}. \quad (5.40)$$

The rotation rate of the Earth, ω_{\oplus} , can be computed using Eq. 5.17. Finally, the time derivative of the modified CIO-based bias-precession-nutation matrix is

$$\dot{\mathbf{C}}_* = \begin{bmatrix} -\dot{a}X^2 - 2aX\dot{X} & -\dot{a}XY - a(Y\dot{X} + X\dot{Y}) & \dot{X} \\ -\dot{a}XY - a(Y\dot{X} + X\dot{Y}) & -\dot{a}Y^2 - 2aY\dot{Y} & \dot{Y} \\ -\dot{X} & -\dot{Y} & -(X\dot{X} + Y\dot{Y})/Z \end{bmatrix}, \quad (5.41)$$

where the z-component of the CIP is

$$Z = \sqrt{1 - X^2 - Y^2}, \quad (5.42)$$

and the the parameter, a , and its derivative are

$$a = 1/(1 + Z), \quad (5.43)$$

and

$$\dot{a} = \frac{a^2(X\dot{X} + Y\dot{Y})}{Z}. \quad (5.44)$$

5.9.2 Partial Derivatives for ITRS Orbit Determination

Filters commonly used for orbit determination, such as the batch filter, conventional Kalman filter (CKF), and the extended Kalman filter (EKF), require partial derivatives of several quantities with respect to the state being estimated. This sections provides the partial derivatives needed for performing orbit determination with a filter set up to estimate ITRS position and velocity instead of the commonly-used GCRS. The partial derivatives include those for the measurement sensitivity matrix and the state transition mapping matrix. The filter state to be estimated is assumed to be

$$\mathbf{X} = \begin{bmatrix} \vec{r}_{\text{ITRS}} & \vec{v}_{\text{ITRS}} & b & C_d & C_r \end{bmatrix}^T, \quad (5.45)$$

where b is the receiver clock bias, C_d is the satellite drag coefficient, and C_r is the satellite coefficient of reflectivity.

5.9.2.1 Measurement Sensitivity Matrix, $\tilde{\mathbf{H}}$

This section provides partial derivatives of the measurement sensitivity matrix for two measurement types: (1) GPS kinematic point solutions and (2) GPS pseudorange observations. The measurement sensitivity matrix, $\tilde{\mathbf{H}}$, is the partial derivative of the measurement model, G , with respect to the filter state being estimated, \mathbf{X} . The measurement sensitivity matrix generalized to j observations and a filter state of length n is

$$\tilde{\mathbf{H}} = \frac{\partial G(\mathbf{X}, t)}{\partial \mathbf{X}} = \begin{bmatrix} \frac{\partial G_1}{\partial X_1} & \frac{\partial G_1}{\partial X_2} & \cdots & \frac{\partial G_1}{\partial X_n} \\ \frac{\partial G_2}{\partial X_1} & \frac{\partial G_2}{\partial X_2} & \cdots & \frac{\partial G_2}{\partial X_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial G_j}{\partial X_1} & \frac{\partial G_j}{\partial X_2} & \cdots & \frac{\partial G_j}{\partial X_n} \end{bmatrix}. \quad (5.46)$$

5.9.2.2 GPS Point Solution Measurements

A common observation set that is automatically generated by most GPS receivers is a kinematic ITRS position solution and receiver clock bias estimate. This observation set is denoted

$$G(\mathbf{X}, t) = \begin{bmatrix} \vec{r}_{\text{ITRS}} & b \end{bmatrix}^T, \quad (5.47)$$

where G is really a set of four observations at one time. Given the filter state in Eq. 5.45, the measurement sensitivity matrix is composed as

$$\tilde{\mathbf{H}} = \begin{bmatrix} \frac{\partial G}{\partial \vec{r}_{\text{ITRS}}} & \frac{\partial G}{\partial \vec{v}_{\text{ITRS}}} & \frac{\partial G}{\partial b} & \frac{\partial G}{\partial C_d} & \frac{\partial G}{\partial C_r} \end{bmatrix}, \quad (5.48)$$

where each component of the matrix can be computed separately with

$$\frac{\partial G}{\partial \vec{r}_{\text{ITRS}}} = \begin{bmatrix} I_{3 \times 3} \\ 0 & 0 & 0 \end{bmatrix}, \quad (5.49)$$

$$\frac{\partial G}{\partial \vec{v}_{\text{ITRS}}} = \begin{bmatrix} 0_{4 \times 3} \end{bmatrix}, \quad (5.50)$$

$$\frac{\partial G}{\partial b} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad (5.51)$$

and

$$\frac{\partial G}{\partial C_d} = \frac{\partial G}{\partial C_r} = \begin{bmatrix} 0_{4 \times 1} \end{bmatrix}. \quad (5.52)$$

If the filter state contains position and velocity in the GCRS instead, the only change is to the partial of the observations with respect to the position such that

$$\frac{\partial G}{\partial \vec{r}_{\text{GCRS}}} = \begin{bmatrix} \mathbf{T}^T \\ 0 & 0 & 0 \end{bmatrix}, \quad (5.53)$$

where the transformation matrix takes the place of the identity matrix in Eq. 5.49.

5.9.2.3 Pseudorange Measurements

This section provides the partial derivatives for a GPS-based pseudorange measurement, ρ . This measurement can be modeled using the simple expression

$$G(\mathbf{X}, t) = \rho = R + b - c\delta t_{\text{GPS}}, \quad (5.54)$$

where R is the geometric distance between the GPS satellite position (\vec{s}_{ITRS}) at the time of signal transmission (t_t) and the receiver position (\vec{r}_{ITRS}) at the time of signal reception (t) (i.e., $R = \|\vec{r}_{\text{ITRS}}(t) - \vec{s}_{\text{ITRS}}(t_t)\|$). The last term in Eq. 5.54 ($c\delta t_{\text{GPS}}$) is the GPS satellite clock bias (speed of light times clock bias). The ITRS position of the GPS satellite and GPS clock bias can be computed using parameters from the GPS navigation message or through the use of precise ephemerides published by the International GNSS Service (IGS). Note that Eq. 5.54 does not include terms for ionospheric delay, tropospheric delay, or multipath. These excluded terms, however, do not alter the partial derivatives given here. The partial derivatives for a single pseudorange measurement with respect to the ITRS filter state, \mathbf{X} , are

$$\frac{\partial G}{\partial \vec{r}_{\text{ITRS}}} = \frac{(\vec{r}_{\text{ITRS}} - \vec{s}_{\text{ITRS}})^T}{R}, \quad (5.55)$$

$$\frac{\partial G}{\partial \vec{v}_{\text{ITRS}}} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}, \quad (5.56)$$

$$\frac{\partial G}{\partial b} = 1, \quad (5.57)$$

and

$$\frac{\partial G}{\partial C_d} = \frac{\partial G}{\partial C_r} = 0. \quad (5.58)$$

If the filter state being estimated contains position in the GCRS instead, the relevant partial becomes

$$\frac{\partial G}{\partial \vec{r}_{\text{GCRS}}} = \frac{\partial G}{\partial \vec{r}_{\text{ITRS}}} \frac{\partial \vec{r}_{\text{ITRS}}}{\partial \vec{r}_{\text{GCRS}}}, \quad (5.59)$$

where

$$\frac{\partial \vec{r}_{\text{ITRS}}}{\partial \vec{r}_{\text{GCRS}}} = \mathbf{T}^T. \quad (5.60)$$

All other partial derivatives remain unchanged.

5.9.2.4 State Transition Mapping Matrix, \mathbf{A}

The state transition matrix (STM), Φ , maps state deviations from one time to another. It is the partial derivative of the state at one time with respect to the state at some epoch and is used in a Kalman filter for mapping the measurement sensitivity matrix as well as the state's covariance. For the application of Kalman filtering in orbit determination, the STM is propagated through time, which is typically done through numerical integration. The time derivative of the STM is given by

$$\dot{\Phi} = \mathbf{A}\Phi, \quad (5.61)$$

where \mathbf{A} is the state transition mapping matrix (Tapley et al, 2004). The state transition mapping matrix is the partial derivative of the state derivative with respect to the state

$$\mathbf{A} = \frac{\partial \dot{\mathbf{X}}}{\partial \mathbf{X}} = \begin{bmatrix} \frac{\partial \dot{X}_1}{\partial X_1} & \frac{\partial \dot{X}_1}{\partial X_2} & \cdots & \frac{\partial \dot{X}_1}{\partial X_n} \\ \frac{\partial \dot{X}_2}{\partial X_1} & \frac{\partial \dot{X}_2}{\partial X_2} & \cdots & \frac{\partial \dot{X}_2}{\partial X_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \dot{X}_n}{\partial X_1} & \frac{\partial \dot{X}_n}{\partial X_2} & \cdots & \frac{\partial \dot{X}_n}{\partial X_n} \end{bmatrix}, \quad (5.62)$$

where the derivative of state in this example is

$$\dot{\mathbf{X}} = \begin{bmatrix} \vec{v}_{\text{ITRS}} & \ddot{\vec{r}}_{\text{ITRS}} & \dot{b} & \dot{C}_d & \dot{C}_r \end{bmatrix}^T. \quad (5.63)$$

For our purposes here, we will assume that b , C_d , and C_r are treated as constant parameters, each with a derivative of zero. The first entry in Eq. 5.63 is the satellite's ITRS velocity vector and the second is the acceleration vector. The main calculation involved in Eq. 5.62 is the computation of the acceleration with respect to each state parameter. The following subsections provide the equations for each force model's contribution to the state transition mapping matrix. We include partial derivatives for the following accelerations: point-mass central body, J_2 zonal harmonic gravity contribution, atmospheric drag, and the Coriolis/centripetal acceleration.

5.9.2.5 Point-Mass Gravity

The satellite acceleration due to the central body acting as a point-mass is

$$\ddot{\vec{r}}_{\text{pm}} = -\frac{\mu}{r^3}\vec{r}, \quad (5.64)$$

where μ is the gravitational parameter of the central body. This acceleration is purely a function of the satellite's position and the equation is the same for both GCRS and ITRS position vectors. The only non-zero partial derivative of Eq. 5.64 is with respect to the position vector (Montenbruck and Gill, 2000)

$$\frac{\partial \ddot{\vec{r}}_{\text{pm}}}{\partial \vec{r}} = \frac{\mu}{r^5} \begin{bmatrix} 3x^2 - r^2 & 3xy & 3xz \\ 3xy & 3y^2 - r^2 & 3yz \\ 3xz & 3yz & 3z^2 - r^2 \end{bmatrix}, \quad (5.65)$$

$$= -\mu \left(\frac{I_{3 \times 3}}{r^3} - 3 \frac{\vec{r} \vec{r}^T}{r^5} \right). \quad (5.66)$$

The use of a position vector in the ITRS will yield a derivative in the ITRS. Similarly, the use of a GCRS position vector will yield the derivative in the GCRS.

5.9.2.6 J_2 Zonal Gravity

The satellite acceleration due to the J_2 zonal harmonic is computed by

$$\ddot{x}_{\text{ns}} = \frac{3}{2} \frac{\mu}{r^3} J_2 \left(\frac{R_{\oplus}}{r} \right)^2 x \left\{ 5 \left(\frac{z}{r} \right)^2 - 1 \right\}, \quad (5.67)$$

$$\ddot{y}_{\text{ns}} = \frac{3}{2} \frac{\mu}{r^3} J_2 \left(\frac{R_{\oplus}}{r} \right)^2 y \left\{ 5 \left(\frac{z}{r} \right)^2 - 1 \right\}, \quad (5.68)$$

$$\ddot{z}_{\text{ns}} = \frac{3}{2} \frac{\mu}{r^3} J_2 \left(\frac{R_{\oplus}}{r} \right)^2 z \left\{ 5 \left(\frac{z}{r} \right)^2 - 3 \right\}, \quad (5.69)$$

where R_{\oplus} is the equatorial radius of Earth and J_2 is the value of the 2nd-order zonal harmonic (e.g., $J_2 \approx 0.001082635$ for Earth). The ITRS position of the satellite should be used to evaluate this expression because gravity models are tied to the body-fixed frame of the central body. Even though zonal harmonics are not functions of longitude, differences between GCRS latitude and ITRS

latitude exist due to polar motion and will affect the computed acceleration. The components which make up the 3×3 partial derivative of the zonal harmonic acceleration with respect to the satellite position are given by

$$\frac{\partial \ddot{x}_{\text{ns}}}{\partial x} = \frac{3}{2} \frac{\mu}{r^3} J_2 \left(\frac{R_{\oplus}}{r} \right)^2 \left\{ 5 \left(\frac{z}{r} \right)^2 \left(1 - 7 \left(\frac{x}{r} \right)^2 \right) - 1 \right\}, \quad (5.70)$$

$$\frac{\partial \ddot{x}_{\text{ns}}}{\partial y} = \frac{3}{2} \frac{\mu}{r^3} J_2 \left(\frac{R_{\oplus}}{r} \right)^2 \left(\frac{5xy}{r^2} \right) \left\{ 1 - 7 \left(\frac{z}{r} \right)^2 \right\}, \quad (5.71)$$

$$\frac{\partial \ddot{x}_{\text{ns}}}{\partial z} = \frac{3}{2} \frac{\mu}{r^3} J_2 \left(\frac{R_{\oplus}}{r} \right)^2 \left(\frac{5xz}{r^2} \right) \left\{ 3 - 7 \left(\frac{z}{r} \right)^2 \right\}, \quad (5.72)$$

$$\frac{\partial \ddot{y}_{\text{ns}}}{\partial x} = \frac{\partial \ddot{x}_{\text{ns}}}{\partial y}, \quad (5.73)$$

$$\frac{\partial \ddot{y}_{\text{ns}}}{\partial y} = \frac{3}{2} \frac{\mu}{r^3} J_2 \left(\frac{R_{\oplus}}{r} \right)^2 \left\{ 5 \left(\frac{z}{r} \right)^2 \left(1 - 7 \left(\frac{y}{r} \right)^2 \right) - 1 \right\}, \quad (5.74)$$

$$\frac{\partial \ddot{y}_{\text{ns}}}{\partial z} = \frac{3}{2} \frac{\mu}{r^3} J_2 \left(\frac{R_{\oplus}}{r} \right)^2 \left(\frac{5yz}{r^2} \right) \left\{ 3 - 7 \left(\frac{z}{r} \right)^2 \right\}, \quad (5.75)$$

$$\frac{\partial \ddot{z}_{\text{ns}}}{\partial x} = \frac{3}{2} \frac{\mu}{r^3} J_2 \left(\frac{R_{\oplus}}{r} \right)^2 \left(\frac{5zx}{r^2} \right) \left\{ 3 - 7 \left(\frac{z}{r} \right)^2 \right\}, \quad (5.76)$$

$$\frac{\partial \ddot{z}_{\text{ns}}}{\partial y} = \frac{3}{2} \frac{\mu}{r^3} J_2 \left(\frac{R_{\oplus}}{r} \right)^2 \left(\frac{5zy}{r^2} \right) \left\{ 3 - 7 \left(\frac{z}{r} \right)^2 \right\}, \quad (5.77)$$

$$\frac{\partial \ddot{z}_{\text{ns}}}{\partial z} = \frac{3}{2} \frac{\mu}{r^3} J_2 \left(\frac{R_{\oplus}}{r} \right)^2 \left\{ -35 \left(\frac{z}{r} \right)^4 - 30 \left(\frac{z}{r} \right)^2 - 3 \right\}. \quad (5.78)$$

The use of a higher-order spherical harmonics model (e.g., a 10×10 gravity field) for the STM requires that the partial derivatives be computed using an ITRS position. This is ideal for propagation in the ITRS because an additional frame transformation is not required. However, if the propagation is performed in the GCRS, the partial derivative can be transformed using

$$\frac{\partial \ddot{\vec{r}}}{\partial \vec{r}_{\text{GCRS}}} = \mathbf{T} \frac{\partial \ddot{\vec{r}}}{\partial \vec{r}_{\text{ITRS}}} \mathbf{T}^T. \quad (5.79)$$

This transformation can also be used to transform partial derivatives that are defined in the GCRS and must be rotated to the ITRS, such as those for third-body and solar radiation pressure accelerations.

5.9.2.7 Atmospheric Drag

The acceleration due to atmospheric drag acting on a satellite is given by

$$\ddot{\vec{r}}_d = -\frac{1}{2}C_d \frac{A}{m} \rho_{\text{atm}} v_r \vec{v}_r, \quad (5.80)$$

where \vec{v}_r is the relative velocity vector of the satellite with respect to the velocity of the atmosphere and ρ_{atm} is the atmospheric density at the location of the satellite. Typically, the atmosphere is assumed to be co-rotating with the central body. For a typical orbit propagation using position and velocity in the GCRS, the relative velocity may be computed as

$$\vec{v}_r = \vec{v}_{\text{GCRS}} - \vec{\omega} \times \vec{r}_{\text{GCRS}}, \quad (5.81)$$

which is equivalent to the satellite velocity with a magnitude of an ITRS velocity, but in the GCRS frame. This is the traditional equation given in textbooks, which also assume that $\vec{\omega}$ is equal to $\vec{\omega}_{\oplus}$. However, as shown in this paper, that is not the case and a more appropriate rotation axis of the Earth should be used. The rotation axis vector should instead be

$$\vec{\omega} = \omega_{\oplus} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}, \quad (5.82)$$

where X , Y , and Z are the CIP coordinates in the GCRS. These parameters may be computed using a moderate accuracy scheme, such as the 4- or 6-term truncated series. As detailed in this paper, other methods exist for computing an appropriate rotation vector of the Earth and for computing the cross product, but the formulation in Eq. 5.82 is easy to implement and easy to modify existing atmospheric drag code if need be.

First, assuming the use of the typical GCRS formulation, the relevant partial derivatives for the state transition mapping matrix are

$$\frac{\partial \ddot{\vec{r}}_d}{\partial \vec{v}_{\text{GCRS}}} = -\frac{1}{2}C_d \frac{A}{m} \rho_{\text{atm}} \left(\frac{\vec{v}_r \vec{v}_r^T}{v_r} + v_r I_{3 \times 3} \right), \quad (5.83)$$

and

$$\frac{\partial \ddot{\vec{r}}_d}{\partial \vec{r}_{\text{GCRS}}} = -\frac{1}{2}C_d \frac{A}{m} v_r \vec{v}_r \left(\frac{\partial \rho_{\text{atm}}}{\partial \vec{r}_{\text{GCRS}}} \right) + \left(\frac{\partial \vec{r}}{\partial \vec{v}_{\text{GCRS}}} \right) \left(\frac{\partial \vec{v}_r}{\partial \vec{r}_{\text{GCRS}}} \right), \quad (5.84)$$

where the final term is simply the negative of the skew symmetric matrix form of Eq. 5.82

$$\frac{\partial \vec{v}_r}{\partial \vec{r}_{\text{GCRS}}} = -[\vec{\omega}]_{\times}. \quad (5.85)$$

For the drag coefficient partial, we have

$$\frac{\partial \ddot{\vec{r}}_d}{\partial C_d} = \frac{\ddot{\vec{r}}_d}{C_d}. \quad (5.86)$$

For the case where orbit propagation and state estimation are performed in the ITRS, the computation of atmospheric drag can be formulated in a more convenient way, where the relative velocity is simple the ITRS velocity vector of the satellite

$$\vec{v}_r = \vec{v}_{\text{ITRS}}. \quad (5.87)$$

Fortunately, this formulation change has no effect on $\partial \ddot{\vec{r}}_d / \partial \vec{v}_{\text{ITRS}}$. The partial derivative with respect to position, however, now becomes

$$\frac{\partial \ddot{\vec{r}}_d}{\partial \vec{r}_{\text{ITRS}}} = -\frac{1}{2} C_d \frac{A}{m} v_r \vec{v}_r \left(\frac{\partial \rho_{\text{atm}}}{\partial \vec{r}_{\text{ITRS}}} \right). \quad (5.88)$$

If the atmospheric density is a function of position in the GCRS and not the ITRS, the partial derivative can be transformed such that

$$\frac{\partial \rho_{\text{atm}}}{\partial \vec{r}_{\text{ITRS}}} = \left(\frac{\partial \rho_{\text{atm}}}{\partial \vec{r}_{\text{GCRS}}} \right) \mathbf{T}. \quad (5.89)$$

5.9.2.8 Coriolis and Centripetal Acceleration

The Coriolis and centripetal acceleration required for orbit propagation in the ITRS described in this paper is repeated here for convenience. Excluding the small Euler term, the acceleration is

$$\ddot{\vec{r}}_{\text{CC}} = -\vec{\omega} \times \vec{v}_{\text{ITRS}} - \vec{\omega} \times \vec{\omega} \times \vec{r}_{\text{ITRS}}, \quad (5.90)$$

where the rotation vector of the Earth for this equation may be sufficiently modeled by

$$\vec{\omega} = \mathbf{W}^T \begin{bmatrix} 0 \\ 0 \\ \omega_{\oplus} \end{bmatrix}, \quad (5.91)$$

as determined by the analysis described in this paper. The partial derivatives of the Coriolis/centripetal acceleration with respect to ITRS position and velocity are then

$$\frac{\partial \ddot{\vec{r}}_{CC}}{\partial \vec{r}_{ITRS}} = -[\vec{\omega}]_{\times} [\vec{\omega}]_{\times}, \quad (5.92)$$

and

$$\frac{\partial \ddot{\vec{r}}_{CC}}{\partial \vec{v}_{ITRS}} = -2[\vec{\omega}]_{\times}, \quad (5.93)$$

where $[\vec{\omega}]_{\times}$ is the skew symmetric matrix generated using the Earth rotation vector from Eq. 5.91.

Chapter 6

Conclusions

This dissertation develops efficient implementations of orbit propagation and coordinate transformations, impacting applications from precise post-processing orbit determination to real-time on-board orbit determination and conjunction assessment. A new implicit Runge-Kutta numerical integration scheme, called Bandlimited Collocation Implicit Runge-Kutta, is presented. The new technique features an efficient force model evaluation strategy applicable to astrodynamics. BLC-IRK features node spacing that is more efficient than traditional polynomial-based quadrature (e.g., Gauss-Legendre). This promotes the use of large time steps and a large number of nodes per interval, reducing the computational load near the clustered endpoints as with polynomial-based quadratures. Orbit propagation simulations for a variety of orbital regimes are used to characterize the efficiency and accuracy of BLC-IRK compared to commonly used explicit Runge-Kutta techniques and multistep 8th-order Gauss-Jackson scheme. We demonstrate 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). In addition, the dense output capability of BLC-IRK is outlined. Simulations demonstrate that interpolating a BLC-IRK trajectory using its collocation algorithm yields a high accuracy, smooth, and continuous solution. This is especially important for conjunction assessment where solutions are required at various points in time along a trajectory.

Efficient procedures for implementing the ITRS/GCRS transformation within orbit propagation are developed to reduce computation time and memory usage. Several simplifications to bias–

precession–nutation and EOP usage are presented and utilized in orbit propagation simulations. Improvements to costly full accuracy transformation procedures include tabulation/interpolation, truncated series representations, and polynomial fitting. Ignoring EOPs entirely and using a 6-term series approximation for CIP coordinates yield worst-case orbit propagation errors of 1 m after 6 hours in LEO and only 53 cm after 3 days in GEO. Due to the improved efficiency while maintaining accurate orbit propagations, these procedures are applicable to both on-board orbit propagation and post-processing applications.

In addition to orbit propagation in an inertial system (GCRS), performing orbit propagation in the rotating ITRS is detailed. This requires additional acceleration terms in the force model to account for centripetal, Coriolis, and Euler forces. I derive the detailed time-derivative of the ITRS/GCRS transformation matrix for velocity transformations and a detailed ITRS rotation vector. This detailed derivation is analyzed for dominating terms and computational time sinks. Comprehensive orbit propagation simulations demonstrate that a simplified computation of the ITRS rotation vector and neglect of the Euler force achieves orbit propagation errors less than 5 cm in LEO and 2.2 cm for GPS orbits in 15 minutes. The results indicate that this scheme is beneficial and tractable for use in propagating orbits on-board satellites using GPS-based orbit determination.

Lastly, the impacts of EOP usage, interpolation, ocean tide corrections, bias–precession–nutation simplifications, and Julian date storage on the accuracy of coordinate transformations between the ITRS and GCRS have been thoroughly investigated. These coordinate transformations are used in the computation of expected measurements (e.g., pseudorange) during orbit determination. Strategies are discussed for reducing computation time while achieving various levels of accuracy. This dissertation provides a table of recommendations for the use of EOPs, interpolation methods, ocean tide corrections, and bias–precession–nutation schemes to achieve various levels of position accuracy in several altitude regimes. The comprehensive summary guides astrodynamists and scientists through the design of on-board and post-processing orbit estimation, orbit propagation, and attitude tasks.

Bibliography

- Aoki S, Guinot B, Kaplan GH, Kinoshita H, McCarthy DD, Seidelmann PK (1982) The new definition of universal time. *Astronomy and Astrophysics* 105:359–361
- Aristoff JM, Poore AB (2012) Implicit Runge-Kutta methods for orbit propagation. In: *AIAA/AAS Astrodynamics Specialist Conference*, Minneapolis, MN, AIAA 2012-4880
- Aristoff JM, Horwood JT, Poore AB (2012) Implicit Runge-Kutta methods for uncertainty propagation. In: *Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS)*, Maui, HI
- Aristoff JM, Horwood JT, Poore AB (2014) Orbit and uncertainty propagation: A comparison of Gauss-Legendre-, Dormand-Prince-, and Chebyshev-Picard-based approaches. *Celestial Mechanics and Dynamical Astronomy* 118:13–28
- Artz T, Bernhard L, Nothnagel A, Steigenberger P, Tesmer S (2012) Methodology for the combination of sub-daily Earth rotation from GPS and VLBI observations. *Journal of Geodesy* 86(3):221–239
- Atkinson KE, Han W, Stewart DE (2009) *Numerical Solution of Ordinary Differential Equations*. John Wiley & Sons Inc
- Bai X (2010) Modified Chebyshev-Picard iteration methods for solution of initial value and boundary value problems. PhD thesis, Texas A&M University
- Bai X, Junkins JL (2011a) Modified Chebyshev-Picard iteration methods for orbit propagation. *Journal of the Astronautical Sciences* 58(4):583–613
- Bai X, Junkins JL (2011b) Modified Chebyshev-Picard iteration methods for solution of boundary value problems. *Journal of the Astronautical Sciences* 58(4):615–642
- Bai X, Junkins JL (2012) Modified Chebyshev-Picard iteration methods for station-keeping of translunar halo orbits. *Mathematical Problems in Engineering* 2012:1–18
- Barrio R, Palacios M, Elipse A (1999) Chebyshev collocation methods for fast orbit determination. *Applied Mathematics and Computation* 99:195–207
- Berry MM (2004) A variable-step double integration multi-step integrator. PhD thesis, Virginia Polytechnic Institute

- Berry MM, Healy LM (2003) Comparison of accuracy assessment techniques for numerical integration. In: 13th Annual AAS/AIAA Space Flight Mechanics Meeting, Ponce, Puerto Rico, AAS 03-171
- Berry MM, Healy LM (2004) Implementation of Gauss-Jackson integration for orbit propagation. *Journal of the Astronautical Sciences* 52(3):331–357
- Bertiger W, Desai S, Dorsey A, Haines BJ, Harvey N, Kuang D, Sibthorpe A, Weiss JP (2010) Sub-centimeter precision orbit determination with GPS for ocean altimetry. *Marine Geodesy* 33:363–378
- Betts JT, Erb SO (2003) Optimal low thrust trajectories to the Moon. *SIAM Journal on Applied Dynamical Systems* 2(2):144–170
- Beylkin G (1995) On the fast Fourier transform of functions with singularities. *Appl Comput Harmon Anal* 2(4):363–381
- Beylkin G, Monzón L (2002) On generalized Gaussian quadratures for exponentials and their applications. *Applied and Computational Harmonic Analysis* 12(3):332–373
- Beylkin G, Sandberg K (2005) Wave propagation using bases for bandlimited functions. *Wave Motion* 41(3):263–291
- Beylkin G, Sandberg K (2014) ODE solvers using band-limited approximations. *Journal of Computational Physics* 265:156–171
- Bizouard C, Gambis D (2009) The combined solution C04 for Earth orientation parameters consistent with international terrestrial reference frame 2005. In: *Geodetic Reference Frames*, Springer, pp 265–270
- Blanes S, Iserles A (2012) Explicit adaptive symplectic integrators for solving Hamiltonian systems. *Celestial Mechanics and Dynamical Astronomy* 114:297–317
- Blanes S, Casas F, Farrés A, Laskar J, Makazaga J, Murua A (2013) New families of symplectic splitting methods for numerical integration in dynamical astronomy. *Applied Numerical Mathematics* 68:58–72
- Bogacki P, Shampine LF (1990) Interpolating high-order Runge-Kutta formulas. *Computers and Mathematics with Applications* 20(3):15–24
- Bond V, Allman M (1996) *Modern Astrodynamics*. Princeton University Press
- Bradley BK, Vallado DA, Sibois A, Axelrad P (2011) Earth orientation parameter considerations for precise spacecraft operations. In: *AAS/AIAA Astrodynamics Specialist Conference*, Girdwood, AK, AAS 11-529
- Bradley BK, Jones BA, Beylkin G, Axelrad P (2012) A new numerical integration technique in astrodynamics. In: *22nd AAS/AIAA Space Flight Mechanics Meeting*, Charleston, SC, AAS 12-216
- Bradley BK, Jones BA, Beylkin G, Sandberg K, Axelrad P (2014) Bandlimited implicit Runge-Kutta integration for astrodynamics. *Celestial Mechanics and Dynamical Astronomy* 119:143–168

- Bradley BK, Sibois AE, Axelrad P (2015) Influence of ITRS/GCRS implementation for astrodynamics: Coordinate transformations. in press at *Advances in Space Research*
- Breiter S, Métris G (1999) Symplectic mapping for satellites and space debris including nongravitational forces. *Celestial Mechanics and Dynamical Astronomy* 71(2):79–94
- Buffett B, Mathews P, Herring T (2002) Modeling of nutation and precession: Effects of electromagnetic coupling. *Journal of Geophysical Research* 107(B4):2070
- Butcher JC (1964) Implicit Runge-Kutta processes. *Mathematics of Computation* 18:50–64
- Calvo M, Montijano JJ, Randez L (1990) A fifth-order interpolant for the Dormand and Prince Runge-Kutta method. *Computational and Applied Mathematics* 29(1):91–100
- Capitaine N, Wallace P (2006) High precision methods for locating the celestial intermediate pole and origin. *Astronomy and Astrophysics* 450:855–872
- Capitaine N, Wallace P (2008) Concise CIO based precession-nutation formulations. *Astronomy and Astrophysics* 478:277–284
- Capitaine N, Wallace P, Chapront J (2003a) Expressions for IAU 2000 precession quantities. *Astronomy and Astrophysics* 412(2):567–586
- Capitaine N, Wallace P, McCarthy DD (2003b) Expressions to implement the IAU 2000 definition of UT1. *Astronomy and Astrophysics* 406(3):1135–1149
- Capitaine N, Folgueira M, Souchay J (2006) Earth rotation based on the celestial coordinates of the celestial intermediate pole. *Astronomy and Astrophysics* 445:347–360
- Coppola V, Seago JH, Vallado DA (2009) The IAU 2000A and IAU 2006 precession-nutation theories and their implementation. In: *AAS/AIAA Astrodynamics Specialist Conference*, Pittsburgh, PA, paper AAS 09–159
- Dormand JR, Prince PJ (1980) A family of embedded Runge-Kutta formulae. *Journal of Computational and Applied Mathematics* 6(1):19–26
- Dormand JR, Prince PJ (1986) Runge-Kutta triples. *Computers and Mathematics with Applications* 12A(9):1007–1017
- Dutt A, Rokhlin V (1993) Fast Fourier transforms for nonequispaced data. *SIAM J Sci Comput* 14(6):1368–1393
- Farrés A, Laskar J, Blanes S, Casas F, Makazaga J, Murua A (2013) High precision symplectic integrators for the Solar System. *Celestial Mechanics and Dynamical Astronomy* 116:141–174
- Fehlberg E (1968) Classical fifth-, sixth-, seventh-, and eighth-order Runge-Kutta formulas with stepsize control. Tech. Rep. NASA TR R-287, NASA Technical Report
- Fornberg B (1988) Generation of finite difference formulas on arbitrarily spaced grids. *Mathematics of Computation* 51(184):699–706
- Fox K (1984) Numerical integration of the equations of motion of celestial mechanics. *Celestial Mechanics* 33:127–142

- Fukushima T (2003) A new precession formula. *The Astronomical Journal* 126:494–534
- GPSD (2013) Navstar GPS space segment/navigation user interface. Tech. Rep. Revision H (IS-GPS-200H: 24-Sep-2013), Global Positioning Systems Directorate
- Grebow DJ, Ozimek MT, Howell KC (2010) Advanced modeling of optimal low-thrust lunar pole-sitter trajectories. *Acta Astronautica* 67(7-8):991–1001
- Grebow DJ, Ozimek MT, Howell KC (2011) Design of optimal low-thrust lunar pole-sitter missions. *Journal of the Astronautical Sciences* 58(1):55–79
- Hairer E, Wanner G (1996) *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*, second revised edn. Springer
- Hairer E, Nørsett S, Wanner G (1993) *Solving Ordinary Differential Equations I: Nonstiff Problems*, second revised edn. Springer-Verlag
- Hairer E, Lubich C, Wanner G (2002) *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*. No. 31 in Springer Series in Computational Mathematics, New York: Springer
- Herman AL, Conway BA (1996) Direct optimization using collocation based on high-order Gauss-Lobatto quadrature rules. *Journal of Guidance, Control, and Dynamics* 19(3):592–599
- Herman AL, Conway BA (1998) Optimal, low-thrust, earth-moon orbit transfer. *Journal of Guidance, Control, and Dynamics* 21(1):141–147
- Herman JF, Jones BA, Born GH, Parker JS (2013) A comparison of implicit integration methods for astrodynamics. In: *AAS/AIAA Astrodynamics Specialist Conference*, Hilton Head, SC, AAS 13-905
- Herring T, Mathews P, Buffett B (2002) Modeling of Nutation-Precession: Very Long Baseline Interferometry Results. *Journal of Geophysical Research* 107(10.1029)
- Hilton J, et al (2006) Report of the International Astronomical Union Division I Working Group on Precession and the Ecliptic. *Celestial Mechanics and Dynamical Astronomy* 94(3):351–367
- van der Houwen PJ, Sommeijer DP (1990) Parallel iteration of high-order Runge-Kutta methods with stepsize control. *Journal of Computational and Applied Mathematics* 29(1):111–127
- Hubaux C, Lemaître A, Delsate N, Carletti T (2012) Symplectic integration of space debris motion considering several Earth's shadowing models. *Advances in Space Research* 49(10):1472–1486
- Iserles A (2009) *A First Course in the Numerical Analysis of Differential Equations*, 2nd edn. Cambridge University Press
- Jackson J (1924) Note on the numerical integration of $d^2x/d^2t = f(x, t)$. *Monthly Notes of the Royal Astronomical Society* 84:602–606
- Johnson NL, Stansbery E, Whitlock DO, Abercromby KJ, Shoots D (2008) *History of on-orbit satellite fragmentations* 14th edition. Tech. Rep. TM-2008-214779, NASA

- Jones BA (2012) Orbit propagation using Gauss-Legendre collocation. In: AIAA/AAS Astrodynamics Specialist Conference, Minneapolis, MN, AIAA 2012-4967
- Jones BA, Anderson RL (2012) A survey of symplectic and collocation integration methods for orbit propagation. In: 22nd Annual AAS/AIAA Space Flight Mechanics Meeting, Charleston, SC, AAS 12-214
- Kaplan GH (2005) Celestial pole offsets: Conversions from (dx, dy) to $(d\psi, d\epsilon)$. Tech. rep., U.S. Naval Observatory
- Kelso TS, Alfano S (2005) Satellite orbital conjunction reports assessing threatening encounters in space (SOCRATES). In: 15th AAS/AIAA Space Flight Mechanics Conference, Copper Mountain, CO, AAS 05-124
- Landau HJ, Pollak HO (1961) Prolate spheroidal wave functions, Fourier analysis and uncertainty II. Bell System Tech J 40:65–84
- Landau HJ, Pollak HO (1962) Prolate spheroidal wave functions, Fourier analysis and uncertainty III. Bell System Tech J 41:1295–1336
- Lemoine F, Kenyon S, Factor J, Trimmer R, Pavlis N, Chinn D, Cox C, Klosko S, Luthcke S, Torrence M, et al (1998) The development of the joint NASA GSFC and the national imagery and mapping agency (NIMA) geopotential model EGM96. NASA
- Mathews P, Herring T, Buffett B (2002) Modeling of nutation and precession: new nutation series for nonrigid Earth and insights into the Earth's interior. Journal of Geophysical Research 107(B4):2068
- McCarthy DD (1992) IERS conventions (1992). Tech. Rep. 13, Paris: Central Bureau of IERS - Observatoire de Paris
- McCarthy DD (1996) IERS conventions (1996). Tech. Rep. 21, Paris: Central Bureau of IERS - Observatoire de Paris
- McCarthy DD, Petit G (2004) IERS conventions (2003). Tech. Rep. 32, Frankfurt am Main: Verlag des Bundesamts für Kartographie und Geodäsie
- Mikkola S (1999) Efficient symplectic integration of satellite orbits. Celestial Mechanics and Dynamical Astronomy 74(4):275–285
- Mikkola S, Palmer P, Hashida Y (2000) A symplectic orbital estimator for direct tracking on satellites. Journal of the Astronautical Sciences 48(2):109–125
- Montenbruck O (1992) Numerical integration methods for orbital motion. Celestial Mechanics and Dynamical Astronomy 53:59–69
- Montenbruck O, Gill E (2000) Satellite Orbits: Models, Methods and Applications, corrected 3rd printing 2005 edn. Springer-Verlag, Netherlands
- Montenbruck O, Ramos-Bosch P (2008) Precision real-time navigation of LEO satellites using global positioning system measurements. GPS Solutions 12:187–198

- Nguyen-Ba T, Desjardins SJ, Sharp PW, Vaillancourt R (2013) Contractivity-preserving explicit Hermite-Obrechhoff ODE solver of order 13. *Celestial Mechanics and Dynamical Astronomy* 117:423–434
- Ozimek MT, Grebow DJ, Howell KC (2009) Design of solar sail trajectories with applications to lunar south pole coverage. *Journal of Guidance, Control, and Dynamics* 32(6):1884–1897
- Ozimek MT, Grebow DJ, Howell KC (2010) A collocation approach for computing solar sail lunar pole-sitter orbits. *The Open Aerospace Engineering Journal* 3:65–75
- Parcher DW, Whiffen GJ (2011) Dawn statistical maneuver design for vesta operations. In: 21st Annual AAS/AIAA Spaceflight Mechanics Meeting, New Orleans, Louisiana
- Petit G, Luzum B (2010) IERS conventions (2010). Tech. Rep. 36, Frankfurt am Main: Verlag des Bundesamts für Kartographie und Geodäsie
- Prince PJ, Dormand JR (1981) High order embedded Runge-Kutta formulae. *Journal of Computational and Applied Mathematics* 7:67–75
- Ray R, et al (1994) Diurnal and Semi-Diurnal Variations in the Earth's Rotation Rate Induced by Oceanic Tides. *Science* 264(5160):830–832
- Rose D, Dullin HR (2013) A symplectic integrator for the symmetry reduced and regularised planar 3-body problem with vanishing angular momentum. *Celestial Mechanics and Dynamical Astronomy* 117:169–185
- Sanz-Serna JM (1988) Runge-Kutta schemes for Hamiltonian systems. *BIT Numerical Mathematics* 28(4):877–883
- Shampine LF, Reichelt MW (1997) The MATLAB ODE suites. *SIAM Journal of Scientific Computing* 18:1–22
- Sibois A, Bertiger W, Desai S, Haines B (2015) GPS-based estimation of sub-daily and rapid polar motion at 15-minute temporal resolution. In: EGU General Assembly Geophysical Research Abstracts, Vienna, Austria
- Slepian D (1964) Prolate spheroidal wave functions, Fourier analysis and uncertainty IV. Extensions to many dimensions; generalized prolate spheroidal functions. *Bell System Tech J* 43:3009–3057
- Slepian D (1965) Some asymptotic expansions for prolate spheroidal wave functions. *J Math and Phys* 44:99–140
- Slepian D (1978) Prolate spheroidal wave functions, Fourier analysis, and uncertainty - v: The discrete case. *Bell System Technical Journal* 57(5):1371–1430
- Slepian D (1983) Some comments on Fourier analysis, uncertainty and modeling. *SIAM Review* 25(3):379–393
- Slepian D, Pollak HO (1961) Prolate spheroidal wave functions, Fourier analysis and uncertainty I. *Bell System Tech J* 40:43–63
- SOFA (2013) SOFA Time Scale and Calendar Tools. International Astronomical Union-Standards of Fundamental Astronomy

- SOFA (2014) SOFA Tools for Earth Attitude. International Astronomical Union-Standards of Fundamental Astronomy
- SPADOC Computation Center (1982) Mathematical foundation for SCC astrodynamic theory. NORAD Technical Publication TP-SCC-008, Headquarters North American Aerospace Defense Command
- Stamatakis N, Luzum B, Wooden W (2008) The other leg of the triangle - how knowledge of Earth orientation parameters could affect Earth observing satellite missions. In: Advances in the Astronautical Sciences, Cambridge, MA, AAS 08-296, vol 132
- Tapley B, Ries JC, Davis GW, Eanes RJ, Schutz BE, Shum CK, Watkins MM (1994) Precision orbit determination for TOPEX/POSEIDON. *Journal of Geophysical Research* 99(C12):24,383–24,404
- Tapley BD, Schutz BE, Born GH (2004) *Statistical Orbit Determination*. Elsevier Inc
- Tsitouras C (2001) Optimized explicit Runge-Kutta pair of orders 9(8). *Applied Numerical Mathematics* 38(1-2):123–134
- Tsitouras C (2007) Runge-Kutta interpolants for high precision computations. *Numerical Algorithms* 44(3):291–307
- Vallado D, Kelso T (2013) Earth orientation parameter and space weather data for satellite operations. In: AAS/AIAA Space Flight Mechanics Meeting, Maui, HI, AAS 13-373
- Vallado D, Kelso T (2015) New consolidated files for Earth orientation parameters and space weather data. In: AAS/AIAA Astrodynamics Specialist Conference, Vail, CO, AAS 15-500
- Vallado DA (2013) *Fundamentals of Astrodynamics and Applications*, 4th edn. Published jointly by Microcosm Press and Springer
- Vallado DA, Kelso T (2005) Using EOP and space weather data for satellite operations. In: 15th AAS/AIAA Astrodynamics Specialist Conference, Lake Tahoe, CA, AAS 05-406
- Wallace PT, Capitaine N (2006) Precession-nutation procedures consistent with IAU 2006 resolutions. *Astronomy and Astrophysics* 459:981–985
- Xiao H, Rokhlin V, Yarvin N (2001) Prolate spheroidal wavefunctions, quadrature and interpolation. *Inverse Problems* 17(4):805–838