

Copyright

by

Noble Ariel Hatten

2016

The Dissertation Committee for Noble Ariel Hatten
certifies that this is the approved version of the following dissertation:

**Space Object Translational and Rotational State
Prediction and Sensitivity Calculation**

Committee:

Ryan P. Russell, Supervisor

Maruthi R. Akella

Srinivas V. Bettadpur

Brandon A. Jones

Ryan M. Weisman

**Space Object Translational and Rotational State
Prediction and Sensitivity Calculation**

by

Noble Ariel Hatten, B.S.As.E., M.S.E.

DISSERTATION

Presented to the Faculty of the Graduate School of
The University of Texas at Austin
in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

The University of Texas at Austin

December 2016

Dedicated to anyone who reads this document cover to cover. I hope it is
worth your while!

Acknowledgments

First (and second ... and third ...), I would like to extend supreme gratitude to my Ph.D. adviser, Dr. Ryan P. Russell. It is not an exaggeration to say that I may not have ever started, let alone finished, this dissertation had he not selflessly allowed me to join his research group following the completion of my master's degree. His supervising style struck a perfect balance between giving me the freedom to work independently and always being available to discuss progress and problems.

I thank the other members of my dissertation committee: Drs. Maruthi Akella, Srinivas Bettadpur, Brandon Jones, and Ryan Weisman. In addition to reviewing this document, these individuals have been invaluable as teachers, mentors, and collaborators through my undergraduate and graduate career. I am also grateful for all of the other exemplary UT professors from whose courses I have benefited mightily over the course of nearly a decade. In particular, Dr. Cesar Ocampo supervised my M.S. work and infected me with his contagious love of astrodynamics.

During my time at UT, I have also taken advantage of the non-professorial perks of matriculation, of which I must here particularly acknowledge the UT Libraries. That collection of books and online journal subscriptions saved me what could otherwise have been countless hours of frustration in the completion of this document.

I thank all of my peers who helped me navigate homework, qualifying exams, research, computer problems, graduation requirements, and post-graduation plans.¹ In particular, I will always have a special place for my fellow Orbital Elements of WRW 412A ... even if they did stick me with anomaly. I am also grateful to all of the students of Flight Dynamics Lab who tolerated my supervision (and my jokes) – especially those who showed me where to buy Aerospace Department T-shirts for my wife.

Outside of UT, I wish to acknowledge fruitful correspondence with Martin Lara and Toshio Fukushima, both of whom were more than helpful in answering questions stemming from their published works.

I am grateful for my wife, Gabby. I do not even want to think about what the experience of preparing this dissertation would have been like without her. I thank my family for their unceasing and unconditional love and support; less abstractly, I thank my late grandfather, William M. Hatten, whose generosity ensured that I was able to pursue higher education without the burden of scholastic debt. And I begrudgingly acknowledge Kevin Stout, whose, frankly, ridiculous accomplishments in academia, sports, and video games have motivated me (in all three arenas) for 20-plus years.

Finally, this dissertation would not have been possible without two specific sources of financial support: (1) a Phase I & II SBIR from the Air Force Research Laboratory, contract FA9453-14-C-0295, under a subcontract from Emergent Space Technologies, Inc.; and (2) teaching assistant and assistant

¹You know who you are.

instructor positions within the Cockrell School of Engineering's Department of Aerospace Engineering and Engineering Mechanics at UT Austin. With respect to the latter, I wish to express particular gratitude to Professor David G. Hull, who offered me the opportunity to serve as assistant instructor of the Flight Dynamics Laboratory course.

Space Object Translational and Rotational State Prediction and Sensitivity Calculation

by

Noble Ariel Hatten, Ph.D.

The University of Texas at Austin, 2016

SUPERVISOR: Ryan P. Russell

While computing power has grown monumentally during the space age, the demands of astrodynamics applications have more than kept pace. Resources are taxed by the ever-growing number of Earth-orbiting space objects (SOs) that must be tracked to maintain space situational awareness (SSA) and by increasingly popular but computationally expensive tools like Monte Carlo techniques and stochastic optimization algorithms.

In this dissertation, methods are presented to improve the accuracy, efficiency, and utility of SO state prediction and sensitivity calculation algorithms. The dynamical model of the low Earth orbit regime is addressed through the introduction of an upgraded Harris-Priester atmospheric density model, which introduces a smooth polynomial dependency on solar flux. Additional modifications eliminate singularities and provide smooth partial derivatives of the density with respect to SO state, time, and solar conditions.

The numerical solution of the equations of motion derived from dynamics models is also addressed, with particular emphasis placed on six-degree-of-

freedom (6DOF) state prediction. Implicit Runge-Kutta (IRK) methods are applied to the 6DOF problem, and customizations, including variable-fidelity dynamics models and parallelization, are introduced to maximize efficiency and take advantage of modern computing architectures.

Sensitivity calculation – a necessity for SSA and other applications – via RK methods is also examined. Linear algebraic systems for first- and second-order state transition matrix calculation are derived by directly differentiating either the first- or second-order form of the RK update equations. This approach significantly reduces the required number of Jacobian and Hessian evaluations compared to the ubiquitous augmented state vector approach for IRK methods, which can result in more efficient calculations. Parallelization is once again leveraged to reduce the runtime of IRK methods.

Finally, a hybrid special perturbation/general perturbation (SP/GP) technique is introduced to address the notoriously slow speed of fully coupled 6DOF state prediction. The hybrid method uses a GP rotational state prediction to provide low-fidelity attitude information for a high-fidelity 3DOF SP routine. This strategy allows for the calculation of body forces using arbitrary shape models without adding attitude to the propagated state or taking the small step sizes often required by full 6DOF propagation. The attitude approximation is obtained from a Lie-Deprit perturbation result previously applied to SOs in circular orbits subject to gravity-gradient torque and extended here to SOs in elliptical orbits. The hybrid method is shown to produce a meaningful middle ground between 3DOF SP and 6DOF SP methods in the accuracy vs. efficiency space.

Table of Contents

Acknowledgments	v
Abstract	viii
List of Tables	xv
List of Figures	xvii
Nomenclature	xxiv
Chapter 1. Introduction	1
1.1 Motivation	1
1.2 Literature Review	4
1.2.1 Space Object Dynamics Models	4
1.2.1.1 Thermospheric Modeling	5
1.2.1.2 Interpolated Space Environment Models	8
1.2.2 Numerical Solution of Ordinary Differential Equations	10
1.2.2.1 ODE Solution Algorithms for 3DOF Applications	11
1.2.2.2 Intelligent Initial Guesses for Implicit ODE Solvers	13
1.2.2.3 Implicit ODE Solvers and Variable-fidelity Dynamics Models	14
1.2.2.4 Parallelization of Implicit ODE Solvers	15
1.2.2.5 Classes of Implicit ODE Solvers	17
1.2.2.6 Six-Degree-of-Freedom State Propagation	21
1.2.3 Sensitivity and State Transition Matrix Calculation	24
1.2.3.1 Direct Methods of STM Calculation	26
1.2.3.2 Indirect Methods of STM Calculation	28
1.2.4 Analytical and Semianalytical Space Object State Prediction Techniques	31
1.2.4.1 Perturbation Methods	32

1.2.4.2	General Perturbation Methods for Space Object Translational Dynamics	36
1.2.4.3	Semianalytical Methods for Space Object Translational Dynamics	38
1.2.4.4	General Perturbation Methods for Space Object Rotational Dynamics	38
1.3	Organization of Dissertation	42
1.4	Summary of Contributions	43
Chapter 2. A Smooth and Robust Harris-Priester Atmospheric Density Model for Low Earth Orbit Applications		45
2.1	Introduction	45
2.2	Traditional Modified Harris-Priester Model	47
2.3	Continuity of Derivatives	48
2.3.1	Approach 1: Analytical Approximation of Heaviside Step Function	49
2.3.1.1	Derivatives with Respect to Space Object State	52
2.3.2	Approach 2: Polynomial Weighting Function	53
2.3.2.1	Derivatives with Respect to Space Object State	56
2.3.3	Derivatives with Respect to Time	57
2.4	Elimination of Singularities	58
2.4.1	Collinearity of Space Object and Diurnal Bulge Position Vectors	58
2.4.2	Space Object at Pole	59
2.4.3	Functional form of Latitude Dependence	61
2.5	Introduction of Environmental Parameter Dependencies	63
2.5.1	Calculation of Coefficients	65
2.5.2	Derivatives with Respect to $\bar{F}_{10.7}$	66
2.5.2.1	Heaviside-Based Continuity	67
2.5.2.2	Polynomial-Based Continuity	69
2.6	Summary of Harris-Priester Models	70
2.7	Comparisons with Other Atmospheric Models	70
2.7.1	GRACE Accelerometer-Derived Density Estimates	70
2.7.2	Parametric Sweep	72
2.7.3	Trajectory Propagation	75
2.7.4	Computational Efficiency	78
2.8	Conclusions	81

Chapter 3. Parallel Implicit Runge-Kutta Methods Applied to Coupled Orbit/Attitude Propagation	83
3.1 Introduction	83
3.2 Fully Coupled Orbit and Attitude Propagation	84
3.3 Implicit Runge-Kutta ODE Solver	88
3.3.1 Variable-Fidelity Dynamics Models	90
3.3.2 Variable-Step-Size Propagation	93
3.3.3 Parallelization	97
3.4 Results	99
3.4.1 Tumbling Object in Low Earth Orbit	100
3.4.2 High-Area-to-Mass-Ratio Object in Geosynchronous Orbit	110
3.5 Conclusions	118
Chapter 4. Decoupled Direct State Transition Matrix Calculation with Runge-Kutta Methods	121
4.1 Introduction	121
4.2 Second-Order Runge-Kutta Methods	123
4.3 STM Calculation with Runge-Kutta Methods	124
4.3.1 Coupled Direct Method	124
4.3.2 Decoupled Direct Method, Option A (DDMa): Multi-Iteration Approach	125
4.3.3 Decoupled Direct Method, Option B (DDMb): Linear-Algebra Approach	125
4.3.3.1 First-Order ODEs	127
4.3.3.2 Double Integrator	130
4.3.3.3 Variable-Step-Size Propagation	133
4.3.4 Variable-Fidelity Dynamics with IRK Methods	134
4.3.4.1 Coupled Direct Method	135
4.3.4.2 Decoupled Direct Method	135
4.3.5 Comparison of the Three Direct Methods: CDM, DDM Linear Algebra, and DDM Multi-Iteration	137
4.4 Results	140
4.4.1 Simple Dynamics Example: Damped Oscillator	142
4.4.2 Expensive Dynamics Examples: Space Object Trajectories	144
4.4.3 Explicit Runge-Kutta Methods	152
4.5 Conclusions	154

Chapter 5. The Eccentric Case of a Fast-Rotating, Gravity-Gradient-Perturbed Space Object Attitude Solution	160
5.1 Introduction	160
5.2 Solution Procedure	163
5.2.1 Extension to Elliptical Orbits	164
5.3 Numerical Studies	171
5.3.1 Parameter Sweep: Eccentricity	172
5.3.2 Parameter Sweep: Nominal ϵ	176
5.3.3 Comparison with Unaveraged Gravity-Gradient Torque	179
5.3.4 Application to Tumbling Rocket Body	183
5.4 Conclusions	189
Chapter 6. A Semianalytical Technique for Six-Degree-of-Freedom Space Object Propagation	191
6.1 Introduction	191
6.2 Hybrid SP/GP Method for 6DOF SO State Prediction	193
6.3 Results	200
6.3.1 Computational Speed of Closed-Form Attitude Solution	201
6.3.2 Tumbling Rocket Body in LEO	203
6.3.2.1 Impact of SO Rotation Rate	210
6.3.3 HAMR Plate in GTO	212
6.4 Conclusions	217
Chapter 7. Conclusions	221
7.1 Dissertation Summary and Primary Contributions	222
7.2 Future Work	226
Appendices	230
Appendix A. GLIRK Local Truncation Error Estimation	231
Appendix B. State Transition Matrix Calculation Via the Direct Method	233
B.1 Second-Order ODEs	235

Appendix C. The Lie-Deprit Method	237
C.1 Example	241
Appendix D. Lara-Ferrer Circular Orbit Equations	247
Appendix E. Lara-Ferrer Fast-Rotating Satellite Solution: Spin about Axis of Minimum Inertia	251
Appendix F. Publications	253
Bibliography	257
Vita	285

List of Tables

2.1	Summary of Harris-Priester model variants.	70
2.2	Geometric statistics of ratios of JB08 densities to HP densities. (Closeness to unity is a measure of accuracy for all columns.) .	74
2.3	Normalized execution times for density models; one normalized time unit is 8.05×10^{-4} ms.	80
3.1	Initial state for SO in LEO.	100
3.2	SO basic physical characteristics.	101
3.3	SO panel reflectance properties.	101
3.4	High-fidelity dynamics model for SO in LEO.	101
3.5	Low-fidelity dynamics model for SO in LEO.	101
3.6	Initial state for HAMR SO in GEO.	111
3.7	SO physical characteristics.	111
3.8	High-fidelity dynamics model for SO in GEO.	111
3.9	Low-fidelity dynamics model for SO in GEO.	112
4.1	Parallelization of dynamics model evaluations.	148
4.2	Initial state for space object in LEO.	148

4.3	ERK implementation details.	153
5.1	Rocket body inertia tensor elements along principal axes in body-fixed frame.	185
5.2	Initial rocket body state.	186
6.1	Summary of coordinate transformations.	204
6.2	Rocket body inertia tensor elements along principal axes in body-fixed frame.	207
6.3	Initial rocket body state.	207
6.4	Common logarithm (i.e., \log_{10}) of ODE solver relative tolerances corresponding to data points in Figure 6.5.	212
6.5	HAMR SO inertia tensor elements along principal axes in body- fixed frame.	214
6.6	Initial HAMR SO state.	215

List of Figures

2.1	Effect of new Heaviside-smoothed HP model on density and its derivatives near altitude layer boundary.	51
2.2	Effect of new polynomial-smoothed HP model on density and its derivatives near altitude layer boundary.	55
2.3	Comparison of scale height H'_{m_i} as function of altitude. Plot titles give k parameter value for Heaviside-smoothed HP and α parameter value for polynomial-smoothed HP.	56
2.4	Effect of weighting function on altitude and $\partial h / \partial r^E$ near pole. Vertical dashed lines indicate τ_p	61
2.5	Effect of weighting function on $\cos^n(\psi/2)$ and $\partial[\cos^n(\psi/2)]/\partial n$ near $\psi = \pi$. Vertical dashed lines indicate τ_n	63
2.6	Contour plots of the logarithms of ρ_m and ρ_M as functions of altitude and solar flux.	66
2.7	Bivariate histograms showing level of agreement between model densities and accelerometer-derived densities for GRACE A SO. Values on heavy line are in perfect agreement.	73

2.8	Bivariate histograms showing level of agreement between CHP densities and Jacchia-Bowman 2008 densities for parametric scan of input values. Values on heavy line are in perfect agreement.	75
2.9	Bivariate histograms showing level of agreement between LIHP densities and Jacchia-Bowman 2008 densities for parametric scan of input values. Values on heavy line are in perfect agreement.	76
2.10	Bivariate histograms showing level of agreement between NNHP densities and Jacchia-Bowman 2008 densities for parametric scan of input values. Values on heavy line are in perfect agreement.	77
2.11	Magnitude of position difference between trajectory propagated using each of several density models and trajectory propagated using Jacchia-Bowman 2008 model.	77
2.12	Orbit shape evolution due to atmospheric drag perturbation.	78
3.1	Rotational state evolution for three-orbit LEO scenario.	103
3.2	Performance of GLIRK propagator as a function of number of stages s using both variable-fidelity and high-fidelity-only dynamics models.	105

3.3	Fraction of total high-fidelity dynamics model CPU time spent evaluating each model constituent for three-orbit LEO scenario; alternative low-fidelity dynamics model time shown for comparison.	106
3.4	Accuracy vs. CPU time for propagation of tumbling SO in LEO; relative LTE tolerance of propagators is varied from $10^{-5} - 10^{-15}$.	107
3.5	Effect of more computationally expensive dynamics function. Geopotential is calculated using spherical harmonics.	108
3.6	Accuracy vs. CPU time for propagation of non-tumbling SO in LEO; relative LTE tolerance of propagators is varied from $10^{-5} - 10^{-15}$	109
3.7	Rotational state evolution for three-orbit GEO scenario. . . .	113
3.8	Performance of GLIRK propagator as a function of number of stages s using both variable-fidelity and high-fidelity-only dynamics models.	114
3.9	Fraction of total high-fidelity dynamics model CPU time spent evaluating each model constituent for three-orbit GEO scenario; alternative low-fidelity dynamics model time shown for comparison.	115
3.10	Accuracy vs. CPU time for propagation of HAMR SO in GEO; relative LTE tolerance of propagators is varied from $10^{-5} - 10^{-15}$.	116

3.11	Accuracy vs. CPU time for propagation of tumbling HAMR SO in GEO; relative LTE tolerance of propagators is varied from $10^{-5} - 10^{-15}$	117
3.12	Accuracy vs. relative LTE tolerance for propagation of tumbling HAMR SO in GEO; relative LTE tolerance of propagators is varied from $10^{-4.8} - 10^{-6.4}$	118
4.1	Conceptual diagram of one step of a two-stage RK method. . .	126
4.2	Time evolution of damped oscillator system.	144
4.3	CPU time required to propagate state and calculate STMs for damped oscillator.	145
4.4	Number of function, Jacobian, and Hessian evaluations required to propagate state and calculate STMs for damped oscillator. .	145
4.5	High-fidelity space object trajectory.	149
4.6	Time evolutions of representative elements of high-fidelity space object STMs.	149
4.7	CPU time required to propagate state and calculate STMs for space object in LEO.	157
4.8	Multiplicative speed increase vs. serial calculation as function of number of available parallel threads.	158

4.9	Normalized CPU time required to propagate state and calculate STMs for example scenarios using the single-integrator formulation of the RK equations and serial computation. Normalization is performed individually for each scenario.	159
5.1	Andoyer variables; adapted from Celletti [45].	164
5.2	Square of small parameter of perturbation procedure for elliptical orbit as function of mean anomaly.	173
5.3	Square of small parameter of perturbation procedure for elliptical orbit evaluated at $r = r_p$, $r = a$, and $r = r_a$ as function of eccentricity.	173
5.4	Differences between analytically obtained perturbation solution and numerically integrated fast-rotating truth solution (subscript T denotes truth); $e = 0.2$	174
5.5	Differences between analytically obtained perturbation solution and numerically integrated fast-rotating truth solution (subscript T denotes truth); $e = 0.8$	175
5.6	Root-mean-square of differences between analytically obtained perturbation solutions and numerically integrated fast-rotating solutions.	176
5.7	Semimajor axis as function of ϵ^2 , assuming constant SO rotational and physical properties.	178

5.8	Root-mean-square of differences between analytically obtained perturbation solutions and numerically integrated solutions as functions of ϵ^2	178
5.9	Differences between analytically obtained perturbation solution and numerically integrated truth solution (subscript T denotes truth); $e = 0$	181
5.10	Differences between analytically obtained perturbation solution and numerically integrated truth solution (subscript T denotes truth); $e = 0.2$	182
5.11	Root-mean-square of differences between analytically obtained perturbation solutions and numerically integrated solutions.	183
5.12	Root-mean-square of differences between analytically obtained perturbation solutions and numerically integrated solutions; $e =$ 0	184
5.13	Differences between analytically obtained perturbation solution and numerically integrated truth solution (subscript T denotes truth) for tumbling rocket body.	187
5.14	Differences between analytically obtained perturbation solution and numerically integrated truth solution (subscript T denotes truth) for tumbling rocket body.	188
6.1	Flow chart of hybrid method.	195
6.2	Relative CPU times for closed-form attitude theory operations.	203

6.3	RMS position error as a function of CPU time for 15-orbit propagation of a rocket body in LEO.	210
6.4	Simplification of Figure 6.3, featuring only the hybrid solutions and extending the CPU time axis to show all data points. . . .	210
6.5	Multiplicative speedups of hybrid and “Low” 6DOF propagations compared to “All” 6DOF propagation.	213
6.6	RMS position error as a function of CPU time for three-orbit propagation of a HAMR plate in GTO.	218
A.1	Position accuracy as a function of high-fidelity dynamics function evaluations for three LTE calculation methods; relative LTE tolerance of propagators is varied from $10^{-5} - 10^{-15}$	232
C.1	Time evolutions of states for system (C.28), calculated in three different manners. “Numerical” indicates numerical solution of unaveraged ODEs; “Avg.” indicates closed-form solution of averaged dynamics <i>without</i> recovery of periodic terms; “Avg. transformed” indicates closed-form solution of averaged dynamics <i>with</i> recovery of periodic terms.	246

Nomenclature

- Unless otherwise noted in text, boldfaced symbols are arrays, and standard-thickness symbols are scalars.
 - Some symbols take on different meanings for different applications; explicit clarification is given in the relevant text if the appropriate definition is not obvious. Here, multiple definitions are separated by semicolons.
- (n) Superscript: n th-order contribution to solution for method of successive approximations
- 0 Subscript: initial
- 1, 2, 3 Subscripts: rectangular components of three-dimensional vector
- α Scale height weighting function parameter
- α_s Right ascension of the Sun
- $\bar{\mathbf{A}}, \bar{\mathbf{b}}, \bar{\mathbf{c}}$ Second-order Runge-Kutta Butcher tableau arrays
- $\bar{\mathbf{q}}$ Quaternion
- $\bar{F}_{10.7}$ 81-day centered average 10.7 cm solar flux index
- $\Delta \mathbf{x}$ Perturbation from nominal solution for method of successive approximations
- $\delta \mathbf{x}$ Variation of \mathbf{x}
- $\Delta \mathbf{f}$ High-fidelity-only contribution to dynamics model
- δ_s Declination of the Sun

δ	Finite-difference perturbation; geocentric latitude
δ_x	Norm of change of state components between consecutive iterations of implicit Runge-Kutta solution procedure
ϵ	Small parameter of perturbation methods
ϵ_n	Small perturbation in Harris-Priester exponent n
$\epsilon_1, \epsilon_2, \epsilon_3$	Variable-fidelity implicit Runge-Kutta convergence tolerances
γ_0	Local truncation error estimation parameter
$\hat{\mathbf{b}}$	Runge-Kutta tableau array \mathbf{b} for comparison solution (vs. \mathbf{x})
$\hat{\mathbf{x}}$	Comparison solution (vs. \mathbf{x}) for estimation of local truncation error
λ_{lag}	Lag angle between the Sun line and the apex of the diurnal bulge
$\lambda, \mu, \nu, \Lambda, M, N$	Andoyer variables
\mathbb{R}^n	n -dimensional space of real numbers
\mathcal{H}	Untransformed Hamiltonian
\mathcal{H}_s	Heaviside step function
\mathcal{K}	Transformed Hamiltonian
$\text{am}(u m)$	Jacobi amplitude function with input parameter m and argument u
$\text{sn}(u m), \text{cn}(u m), \text{dn}(u m)$	Jacobi elliptic functions with input parameter m and argument u
μ_*	Point-mass gravitational parameter
μ^*	Geometric mean
Φ	Hamiltonian of triaxial rigid body in torque-free rotation
ψ	Angle between Earth-centered inertial space object position vector and apex of diurnal bulge
ρ	Atmospheric density

σ^*	Geometric standard deviation
τ_n	Harris-Priester n singularity tolerance
τ_p	Polar singularity tolerance
$\tilde{\mathbf{h}}$	Higher-order portion of state equations
$\boldsymbol{\omega}$	Angular velocity vector
$\mathbf{A}, \mathbf{b}, \mathbf{c}$	Runge-Kutta Butcher tableau arrays
\mathbf{a}_p	Perturbing acceleration vector
\mathbf{C}_a	Vector aerodynamic coefficient
\mathbf{d}_{cm}	Vector from center of mass of space object to center of pressure of panel
\mathbf{I}_n	$n \times n$ identity matrix
\mathbf{J}_0	Inertia tensor
\mathbf{n}	Unit vector in outward normal direction of a space object panel
\mathbf{p}	Generalized coordinates of untransformed Hamiltonian system
\mathbf{q}	Generalized momenta of untransformed Hamiltonian system
\mathbf{R}	Transformation matrix
\mathbf{r}	Position vector
\mathbf{r}_B	Position vector in body-fixed reference frame
\mathbf{T}	Torque vector
\mathbf{u}_b	Unit vector toward apex of diurnal bulge in Earth-centered inertial frame
\mathbf{u}_r	Unit vector in direction of space object position vector
\mathbf{u}_{sun}	Unit vector pointing from space object to Sun
\mathbf{x}	State vector
$\dot{\mathbf{x}}$	First time derivative of \mathbf{x}

$\ddot{\mathbf{x}}$	Second time derivative of \mathbf{x}
$\mathbf{0}_{n \times n}$	$n \times n$ array of zeros
Φ^i	i th-order state transition matrix
ξ	Perturbation solution variable set
ξ'	Singly averaged perturbation solution variable set
ξ''	Doubly averaged perturbation solution variable set
g	Second-order state equations
H	Hessian
h	Orbital angular momentum vector; zeroth-order portion of state equations
J	Jacobian
P	Generalized coordinates of transformed Hamiltonian system
Q	Generalized momenta of transformed Hamiltonian system
r_B	Position vector of mass center of space object with respect to mass center of central body, expressed in body-fixed frame
\mathbf{y}	First-order portion of state vector consisting of state and its time derivatives (i.e., $\mathbf{x} \triangleq [\mathbf{y}^T \quad \dot{\mathbf{y}}^T]^T$)
ξ_H	Independent variable of scale height weighting function w_H
ξ_n	Independent variable of Harris-Priester n weighting function w_n
ξ_p	Independent variable of polar singularity weighting variable w_p
A	Area
A, B, C	Principal moments of inertia of space object; $A \leq B \leq C$

$a, e, i, \Omega, \omega, \theta, M_0$	Orbital elements: semimajor axis, eccentricity, inclination, right ascension of the ascending node, argument of periapsis, true anomaly, mean anomaly, respectively
A_p	Three-hourly (geomagnetic) planetary amplitude index
AU	One astronomical unit
c	Speed of light
C_D	Drag coefficient
C_d	Diffusive reflectivity coefficient
c_I	$\cos I$; $\cos^2 I = H^2/G^2$
c_m, c_M	Cubic Harris-Priester coefficients
C_s	Specular reflectivity coefficient
D	Distance from space object to Sun
E	Superscript: appended variable is expressed in Earth-centered, Earth-fixed reference frame
ell	Subscript: references elliptical orbit (as opposed to circular orbit)
$F(\Psi m), E(\Psi m)$	Incomplete elliptic integrals of the first and second kinds, respectively, with input parameter m and argument Ψ
f, \mathbf{f}	Scalar, vector first-order state equation(s)
$F_{10.7}$	Daily 10.7 cm solar flux index
F_{sun}	Total solar flux over all wavelengths
g	Subscript: Harris-Priester model: altitude calculated using general expressions
GG	Subscript: gravity-gradient
H	Scale height

h	Geodetic altitude; integration step size
H'	Smoothed scale height
H_1	Earth shadowing function
H_2	Fraction of panel of space object facing the Sun
<i>high</i>	Subscript: high-fidelity dynamics model
I	Inclination of plane orthogonal to rotational angular momentum with respect to inertial orbital frame
J	Inclination of equatorial plane of rigid body with respect to plane orthogonal to rotational angular momentum
J_2	Central-body oblateness
k	Index variable; Heaviside step function approximation parameter
$K(m), E(m), \Pi(f m)$	Complete elliptic integrals of the first, second, and third kinds, respectively, with input parameter m and characteristic f
$l, g, h, \tau, L, G, H, T$	Rotational perturbation solution variables
<i>low</i>	Subscript: low-fidelity dynamics model
M	Magnitude of rotational angular momentum; subscript: maximum
m	mass of space object; input parameter for elliptic integral; subscript: index; subscript: minimum (Harris-Priester model)
<i>max</i>	Subscript: maximum
n	Index variable; Harris-Priester exponent; mean motion; dimension of state vector
n'	Dimension of first-order portion of state vector (i.e., \mathbf{y})
p	Subscript: Harris-Priester model: altitude calculated assuming space object is exactly above a pole

r	Magnitude of \mathbf{r}
r_p, r_a	Periapsis and apoapsis distances, respectively
ref	Subscript: reference
S	Doubly averaged Hamiltonian
s	Number of internal stages of a Runge-Kutta method
s_I	$\sin I$
T	Superscript: transpose
t	Time
U	Gravity-gradient potential for fast-rotating space object
v_r	Magnitude of relative velocity between space object and atmosphere
W, V	Generating functions of first and second Lie transformations, respectively
w_H	Scale height weighting function value
w_n	Harris-Priester n singularity weighting variable
w_p	Polar singularity weighting variable
x	Scalar variable
atol	Absolute local truncation error tolerance
rtol	Relative local truncation error tolerance

Chapter 1

Introduction

1.1 Motivation

Space object (SO) state prediction is one of the fundamental problems of astrodynamics. State prediction is essential for tasks ranging from orbit determination (OD) of heavenly bodies [20] to interplanetary satellite mission design [47] to the maintenance of space situational awareness (SSA) for Earth-orbiting assets [8]. It is well-known that, in practical scenarios, all state prediction algorithms, analytical or numerical, are approximations. The nonexistence of a single “correct” solution makes it unsurprising that a prodigious wealth of methods has been produced, with development beginning hundreds of years ago and accelerating rapidly with the advent of artificial satellites in the mid-twentieth century. Even orders-of-magnitude increases in computational capabilities have not slaked the demand for and production of new techniques. On the contrary, driven both by necessity and computer hardware advances, astrodynamics software has become ever more ambitious. For example, the proliferation of artificial SOs placed in Earth orbit – and, even more importantly, the resulting debris field – makes necessary the accurate state prediction of tens of thousands of SOs to safeguard active satellites [196]. Additionally, population-based approaches to uncertainty propagation

(e.g., Monte-Carlo methods [147, 91]) and trajectory optimization (e.g., genetic algorithms [84]) allow for relaxations of some of the restrictive assumptions of less computationally intensive alternatives (e.g., extended Kalman filters [17] and gradient-based optimizers [81], respectively). The cost is a large number of individual SO state predictions for the uncertainty propagation of a single SO or the optimization of a single trajectory.

Outside of population-based methods, the need for innovative algorithms persists, in part due to the commonplace requirement of sensitivity calculation. In particular, the calculation of state transition matrices (STMs) – a vital element of sensitivity calculation for a dynamical system – can significantly increase computation time vs. the prediction of the state alone. Further, computational resource expansion has encouraged the development of more expensive techniques that improve performance by using second- and higher-order STMs as opposed to traditional first-order methods [160, 67, 159, 130, 197].¹

As a result, the creation of new state prediction and sensitivity calculation methods continues in the present day. Novel special perturbation (SP), general perturbation (GP), and semianalytical prediction methods are continually developed for SO orbital (i.e., three-degree-of-freedom or 3DOF) motion, rotational motion, and coupled orbital/rotational (i.e., 6DOF) motion

¹For notational convenience, the acronym STM is used to refer to both the first-order state transition matrix and higher-order state transition tensors. When appropriate, the modifiers “first-order,” “second-order,” etc. are applied to differentiate between STMs.

[64, 152].² Each new advancement seeks to improve on its forebears by accomplishing at least one of the following three objectives: (1) improve the fidelity of a prediction; (2) improve the utility of a prediction; (3) decrease the time required to make a prediction.

In this dissertation, four interrelated contributions to the astrodynamics field are presented, each of which addresses at least one of these objectives. The areas of contribution are:

1. The accuracy and efficiency of the dynamics model used by SP methods to produce the forces and torques acting on an SO; specifically, the approximation of thermospheric density, a critical component of aerodynamic force and torque calculations.
2. The efficiency of solving the initial value problem (IVP) generated by the EOMs produced by the dynamics model of an SP or semianalytical method. In particular, the work in this dissertation focuses on solving the ordinary differential equations (ODEs) of the coupled orbit/attitude prediction problem.
3. The efficiency of calculating first- and second-order STMs of a numerically propagated SO state.

²SP methods use numerical techniques to approximately solve differential equations of motion (EOMs) that have no analytical solution, while GP methods create closed-form state predictions by simplifying the EOMs. Semianalytical methods simplify EOMs to achieve more efficient numerical propagations rather than a fully closed-form solution [64]. Each method is introduced in further detail in Sections 1.2.1 and 1.2.2 (SP methods) and 1.2.4 (GP and semianalytical methods).

4. The accuracy and efficiency of 6DOF state prediction. SP orbital methods and GP rotational methods are combined to create a semianalytical method that provides improved accuracy over a 3DOF SP propagation and improved efficiency over a fully SP 6DOF propagation.

The remainder of Chapter 1 introduces background on each of these subjects, followed by an overview of the organization and contributions of the dissertation.

1.2 Literature Review

An introduction to the four areas of study listed in Section 1.1 is given here, with an emphasis on the material most directly related to the contributions of the dissertation.

1.2.1 Space Object Dynamics Models

SO state prediction using SP methods relies on the numerical approximation of the solution of a system of (usually nonlinear) ODEs. These numerical techniques generally require evaluating the dynamics model (i.e., the EOMs) at intervals along the propagated solution trajectory [89]. In SP, the computational cost of the necessary dynamics model evaluations tends to greatly overshadow the costs associated with constructing the approximate trajectory once the dynamics evaluations are performed – to the extent that the number of dynamics model evaluations is often used as a proxy for CPU

runtime [6]. Thus, reducing the cost of each evaluation while maintaining desired fidelity is a primary goal when constructing an efficient propagator.

In this dissertation, dynamics model efficiency is addressed by introducing an upgraded modified Harris-Priester (HP) atmospheric density model that uses interpolation techniques to improve accuracy and provide desirable numerical properties. Section 1.2.1.1 presents an overview of thermospheric modeling, and Section 1.2.1.2 provides background on the use of interpolation in space environment modeling.

1.2.1.1 Thermospheric Modeling

For SOs in low Earth orbit (LEO), aerodynamic forces create an important orbital perturbation that generally causes secular decreases in both semimajor axis and eccentricity [183]. Aerodynamic forces can also create torques on SOs. The aerodynamic force experienced by an SO is directly proportional to the density of the atmosphere. Thus, a method for approximating the density as a function of the SO state and time is vital to state propagation.

At the crudest level, density decreases approximately exponentially with altitude, and the most basic algorithms neglect all other variations:

$$\rho(h) = \rho(h_{ref}) \exp\left(-\frac{h - h_{ref}}{H}\right), \quad (1.1)$$

where ρ is the density, h is the geodetic altitude, H is a parameter called the scale height, and the subscript *ref* indicates a reference value [149]. While the simplicity of the exponential atmosphere results in a very fast computation, the

approximation is also very rough: The density also depends on local apparent solar time, geodetic latitude and longitude, time of year, solar radiation flux, and geomagnetic conditions. However, the understanding of these dependencies is far from complete, a situation that has resulted in the development of competing models based on both theoretical and empirical functional relationships. Even among the high-fidelity models, significant discrepancies between output values can exist, and algorithm development continues [191, 55]. This uncertainty motivates the use of fast, approximate methods for many analyses.

Examples of such algorithms include the HP model and TD-88. Like the exponential atmosphere, HP is fundamentally based on an assumed analytical relationship between density and altitude, but adds dependencies on local apparent solar time, latitude, and, in some implementations, solar flux conditions [90, 57, 184, 75, 76]. The HP model is discussed in more detail in Chapter 2. On the other hand, TD-88 and its successor, TD-88Up, offer a simple, analytical model based on a relatively small number of coefficients that are determined through a nonlinear least-squares fit of data generated by a higher-fidelity model [178, 182].

Among the most well-known of the higher-fidelity models are the Jacchia, Mass Spectrometer – Incoherent Scatter Radar (MSIS), Drag Temperature Model (DTM), and High Accuracy Satellite Drag Model (HASDM) variants, each of which is described briefly below.

Jacchia Atmosphere Models The earliest Jacchia models were published in the 1960s [107], and updates have been produced in the years since, among

them Jacchia-71 [108], Jacchia-Roberts [173], Jacchia-77 [109], Jacchia-Bowman 2006 [31], and Jacchia-Bowman 2008 [30]. The basic Jacchia model calculates density via an explicit numerical integration of the diffusion equation, based on an assumed, empirically determined temperature profile. Correction factors are used both prior to and subsequent to the integration to account for environmental factors. The parameters used to drive the model are based on analysis of the accelerations experienced by satellites due to aerodynamic forces.

MSIS Atmosphere Models The MSIS model has also seen several significant updates since its original publication in late 1970s [97], including MSIS-83 [95], MSIS-86 [96], and NRLMSISE-00 [164]. In contrast to the Jacchia models, the parameter values used by the MSIS model are derived from mass spectrometer and incoherent scatter radar observations [55]. In addition, MSIS relies on a large number of empirically determined coefficients to calculate density rather than direct integration of the diffusion equation.

DTM Atmosphere Models As with Jacchia and MSIS, several variations of the DTM exist [18], including DTM-94 [22], DTM-2000 [37], DTM-2009 [38], and DTM-2013 [36]. The DTM variants are based on the observed effects of aerodynamic forces on satellites (like the Jacchia models), but densities are calculated as explicit functions of tabulated coefficients (like the MSIS models). It is noted that DTM models are generally defined by significantly fewer coefficients than MSIS models [55].

HASDM Atmosphere Models The HASDM was developed by the U.S. Air Force Space Command [181, 41]. Though based on the Jacchia-70 model, HASDM adds a capability known as the Dynamic Calibration Atmosphere (DCA). The DCA updates up to 20 parameters of the Jacchia model in near real-time through analysis of the tracked motion of approximately 75 so-called calibration satellites. Thus, HASDM is able to respond dynamically to temporal atmospheric variations, such as those caused by geomagnetic storms. This capability sets HASDM apart from the other methods introduced in this section, which, though functions of time-varying environmental variables, rely on fixed internal parameters. For SO state prediction purposes, HASDM also produces three-day DCA coefficient projections. HASDM has been shown to more closely match density values derived from the accelerometer data of the CHAMP and GRACE satellites than Jacchia-Bowman 2008, NRLMSISE-00, or DTM-94 [55]. Unlike the other models discussed in this section, HASDM is not publicly available.

1.2.1.2 Interpolated Space Environment Models

Spurred by the ever-growing amount of runtime memory available in modern computers, several authors have turned to interpolation techniques to improve the efficiency of SO dynamics models. In these methods, a computationally intensive function is replaced by a less expensive approximation that relies on a precalculated data set that may be loaded in runtime memory prior to execution of the state propagation. For example, a high-fidelity geopotential model is traditionally calculated using a spherical harmonics formulation.

However, the computational cost grows quadratically with the degree/order of the model [12]. As an alternative, the spherical harmonics model may be sampled *a priori* to generate interpolation coefficients [111, 117, 12, 115]. For high-degree/order models, speedups of multiple orders of magnitude have been demonstrated, with differences between the interpolated and calculated models in the noise of the estimated precision of the spherical harmonics coefficients [12, 115].

Interpolation techniques have also been shown to effectively improve efficiency in ephemeris generation and thermospheric density calculations. For the former, the use of cubic splines produces propagation time speedups of multiple orders of magnitude compared to the commonly used Jet Propulsion Laboratory (JPL) Spacecraft Planet Instrument C-matrix Events (SPICE) ephemeris retrieval system [10, 2]. For the latter, the Jacchia-Gill model uses a fourth/fifth-order bi-polynomial representation of the standard density to replace the expensive Newton-Cotes five-point quadrature standard density calculator of the Jacchia 1971 model [80, 149, 108]. In this case, a one-order-of-magnitude speedup has been reported [149].

Aside from providing efficiency improvements, the interpolation scheme may be selected to endow the approximated function with attractive properties. In the context of SO state propagation, commonly desirable traits are higher-order continuity and precise numerical derivatives. A smooth dynamics function improves the performance of numerical ODE solvers [89], and gradient-based trajectory optimization routines require first-order (and sometimes second-order) derivatives [81, 82, 197, 130]. Similarly, the many esti-

mation and uncertainty propagation algorithms that use gradient information benefit from continuous and precise derivatives [183, 13]. Interpolation-based dynamics models can provide these important properties [111].

1.2.2 Numerical Solution of Ordinary Differential Equations

The SP method core is an algorithm for numerically approximating the solution of a system of ODEs, which may be described by

$$\frac{d\mathbf{x}}{dt} \triangleq \dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}), \quad \mathbf{x}(t_0) \triangleq \mathbf{x}_0, \quad (1.2)$$

where \mathbf{x} is the state and t is the independent variable of integration. The choice of algorithm used to approximate the solution of Eq. (1.2) can have a significant effect on the efficiency of an SP routine. The primary computational burden is the repeated evaluation of the dynamics model \mathbf{f} . Therefore, traditionally, the algorithm that requires the fewest dynamics model evaluations to achieve a given accuracy is generally preferable. However, owing to relatively recent improvements in algorithm design and the availability of parallel processing, strict dynamics model evaluation counts are no longer the only relevant factor in measuring propagator efficiency.

Implicit ODE solvers make up one class of methods that can take advantage of these new advancements, and have been applied in several forms to 3DOF SO propagation. In this dissertation, implicit methods are shown to produce efficiency improvements for certain classes of 6DOF propagation, as

well. Section 1.2.2.1 introduces some of the most popular ODE solution algorithms for 3DOF applications, while Sections 1.2.2.2–1.2.2.5 go into more detail regarding the implicit solvers that are the primary focus of the dissertation. Finally, literature specific to 6DOF propagation is discussed in Section 1.2.2.6.

1.2.2.1 ODE Solution Algorithms for 3DOF Applications

For 3DOF SO propagation, linear multistep methods have been considered state-of-the-practice (as well as state-of-the-art) for many years. These algorithms approximate the solution of Eq. (1.2) at a given time step ($\mathbf{x}_{m+1} = \mathbf{x}(t_{m+1})$) using saved state information from previous time steps: $\mathbf{x}_m, \mathbf{x}_{m-1}, \dots$ [24].³ Variable-order, variable-step-size linear multistep methods are the default propagators for mission analysis tools like Copernicus [155, 126] and DIVA [129], while the fixed-step⁴, eighth-order Gauss-Jackson method [110, 23] is used by Air Force Space Command (AFSPC) to maintain the catalog of Earth-orbiting SOs [152].

Single-step algorithms, which produce \mathbf{x}_{m+1} using only evaluations of \mathbf{f} in the interval $t \in [t_m, t_{m+1}]$, are also popular, primarily due to their ease of use. Notably, high-order, variable-step-size, explicit Runge-Kutta (ERK) methods like those of Prince and Dormand [166] and Fehlberg [60] are used

³The dependence on “backpoints” necessitates a startup procedure to initialize the solver. Popular methods include the use of (1) lower-order multistep solvers (increasing the number of backpoints results in a higher-order method), (2) a single-step solver, or (3) an analytical approximation coupled with an iterative correction procedure [23].

⁴Precision is maintained for high-eccentricity orbits by performing a Sundman transformation on the independent variable to concentrate steps near periapsis, where the state changes most rapidly [23, 152].

frequently. ERK-Nyström methods, which formulate Eq. (1.2) as $\ddot{\mathbf{x}} = \mathbf{g}(t, \mathbf{x})$, may provide enhanced efficiency when the dynamics model has no velocity dependence [148].

In recent years, focus in the astrodynamics community has turned toward *implicit* single-step propagation methods. From an operational standpoint, the distinguishing feature of an implicit method is the existence of a system of algebraic equations that must be solved at each propagation step. For example, an explicit or implicit Runge-Kutta (IRK) method advances the state one step in the independent variable t from t_m to t_{m+1} using the expressions [89]

$$\mathbf{x}_{m+1} = \mathbf{x}_m + h \sum_{j=1}^s b_j \mathbf{f}_j \quad (1.3)$$

$$\mathbf{f}_i \triangleq \mathbf{f}(t_i, \mathbf{x}_i), \quad i = 1, \dots, s \quad (1.4)$$

$$t_i \triangleq t_m + hc_i \quad (1.5)$$

$$\mathbf{x}_i \triangleq \mathbf{x}_m + h \sum_{j=1}^s a_{ij} \mathbf{f}_j. \quad (1.6)$$

The arrays $\mathbf{A}_{s \times s}$, $\mathbf{b}_{s \times 1}$, and $\mathbf{c}_{s \times 1}$ – whose elements are the a_{ij} , b_i , and c_i , respectively – define a Runge-Kutta (RK) method with s internal stages. In ERK methods, the a_{ij} are zero for $j \geq i$, which allows for the explicit, sequential calculation of the \mathbf{f}_i . On the other hand, IRK methods violate this restriction, and the \mathbf{f}_i that satisfy Eq. (1.4) must be determined by simultaneously solving the system of algebraic equations. For nonlinear systems of ODEs,

solving Eq. (1.4) requires an iterative method such as fixed-point iteration, the Newton-Raphson method, or the Gauss-Seidel method [88].

The iterative solves at each step often lead to inefficient propagation compared to ERK or linear multistep methods. As a result, traditionally, implicit methods have been primarily valued for the theoretical properties some possess (e.g., A -stability, symmetry, symplecticity, etc. [88]). Unfortunately, many of these properties are not always of the utmost importance for short- or medium-time-span SO propagation. However, advances in both algorithm development and computer architecture have resulted in implicit methods that have been shown to compete with or even outperform explicit alternatives in many scenarios. Three elements, in particular, must be mentioned: intelligent initial guesses, the use of variable-fidelity dynamics models, and parallelization. It is noted that the first two of these elements require propagator customizations specific to the problem under consideration. For SO state prediction, the derivation and implementation of such customizations may be justified by the frequency with which propagation is required for SSA and other applications.

1.2.2.2 Intelligent Initial Guesses for Implicit ODE Solvers

As an iterative solution procedure is required to solve Eq. (1.4), an initial guess for the \mathbf{x}_i is needed. For example, for 3DOF SO propagation, the existence of an analytical solution of the two-body IVP (i.e., the Kepler problem) provides an efficient, relatively accurate means of “warm-starting” the iterative solver [20]. Additional accuracy may be obtained via a higher-fidelity GP theory (see Section 1.2.4.2).

1.2.2.3 Implicit ODE Solvers and Variable-fidelity Dynamics Models

For Earth-orbiting SO propagation applications, the primary source of inefficiency for implicit solvers is the need to repeatedly evaluate a computationally expensive dynamics model within the iterative scheme used to solve Eq. (1.4). This cost may be substantially reduced if the full-fidelity dynamics model may be well approximated by a less computationally expensive low-fidelity model. For example, for many relevant Earth orbit regimes, the sum of the point-mass and J_2 terms of the gravitational attraction of the Earth dominates all other forces acting on an SO [183]. In this case, at early iterations of the solution process, while the \mathbf{x}_i are relatively far from the eventual converged values, evaluation of the low-fidelity model only is enough to steer the algorithm toward the solution. Even at later iterations, the full-fidelity model need not be evaluated at every iteration. Instead, the high-fidelity contribution from one iteration may be saved and used in subsequent iterations with an updated low-fidelity contribution. In such a manner, moderate precision has been obtained for 3DOF SO state prediction using a single full-fidelity iteration per step [33], and two or three full-fidelity iterations per step have produced results indistinguishable from the use of full-fidelity-only evaluations [141, 6]. Significant computational savings are achieved using variable-fidelity dynamics because, for Earth-orbiting SO propagation applications, the low-fidelity model requires significantly less CPU time to evaluate than the full-fidelity model.

If (1) the low-fidelity dynamics model does not adequately approximate the full-fidelity model or (2) the low-fidelity model is not significantly less expensive to evaluate than the full-fidelity model, the benefit of the variable-fidelity paradigm diminishes. Examples of such scenarios are discussed in Chapter 3 in the context of coupled orbit/attitude SO state prediction.

1.2.2.4 Parallelization of Implicit ODE Solvers

Implicit solvers can take advantage of parallel computing architectures within a single propagation because, at each iteration of the nonlinear algebraic solution procedure, all of the dynamics model evaluations are independent. In an idealized scenario⁵, the computation time of a propagation step is proportional only to the number of iterations required to achieve convergence of the nonlinear system; the number of stages is irrelevant. ERK methods, on the other hand, cannot take advantage of parallelization in this manner. Referencing Eqs. (1.3)–(1.6), the calculation of each \mathbf{f}_i requires precalculation of all \mathbf{f}_j , $j < i$.

Though this advantage of implicit ODE solvers is well-known [113, 5, 15, 33], the precise benefits are not as well-documented because practical implementations do not achieve ideal parallelization. In other words, the use of m cores does not generally result in an m -fold speedup in execution. In the context of the Modified Chebyshev-Picard Iteration (MCPI)⁶ solver, the impacts

⁵In this context, “ideal” means that the number of parallel threads is the same as the number of stages per step, and parallelization overhead is negligible.

⁶MCPI is discussed further in Section 1.2.2.5.

of parallelization for 3DOF SO propagation have been examined using graphics processing units (GPUs) and the NVIDIA CUDA architecture [15, 154]. While the GPU-accelerated MCPI produced significant performance gains in certain scenarios, this study was limited in scope.⁷ MCPI has also been parallelized across multiple CPU cores using the MATLAB `parfor` construct, with far-from-ideal observed performance gains [126]. Such limited data and conflicting results motivates more systematic study of the potential benefits of parallelization.

A caveat of this parallelization strategy is that, for some applications, more efficient uses of parallel computing resources may be possible. For example, if state prediction is required for a large number of SOs, then parallelizing at the trajectory-propagation level is likely to result in less overhead than parallelizing the dynamics function evaluations at each step of an implicit ODE solver [8]. (Parallelization overhead is discussed in more depth in Chapters 3 and 4.) Similarly, trajectory-level parallelization is possible when using uncertainty propagation techniques like sigma-point methods [194] or Monte Carlo methods [147, 91]. For these applications, nested parallelization is a possibility for taking advantage of both trajectory-level and IRK stage-level parallelization opportunities [154].

⁷(1) Single-precision arithmetic was used for the GPU implementation, providing superior computational speed at the expense of precision [154]. (2) The comparisons were limited to cases of 128 or more internal stages per propagation step. While this is not necessarily an unreasonable value for MCPI, not all applications require such a large number of stages, and other classes of implicit solvers typically use significantly fewer stages per step. This limitation is relevant because the speed gains available using GPUs generally increase as the number of necessary threads increases. (3) MATLAB, rather than a compiled language, was used to implement the serial comparison solver.

1.2.2.5 Classes of Implicit ODE Solvers

Though the implicit methods may be categorized in several different ways, for the purposes of this description, the algorithms are classified into two groups based on the qualitative step-size length generally used by the methods.

Long-Step-Size Implicit ODE Solvers “Long”-step-size methods are so-called because the propagation time steps of members of this group tend to be very significantly larger than those of ERK or linear multistep methods. Depending on precision requirements and dynamics model fidelity, 3DOF ERK and linear multistep propagators may take dozens, if not more than 100, time steps per orbit revolution for Earth-orbiting SO propagation. On the other hand, typical step sizes for long-step-size implicit methods range from one to five steps per orbit – even for high-fidelity LEO propagations, in which the dynamics vary rapidly [141, 33]. This difference is made possible by significantly increasing the number of points per step at which the dynamics model is evaluated. For example, the eighth-order Runge-Kutta-Dormand-Prince method with seventh-order error control (RKDP8(7)) uses 13 evaluation points per step (i.e., $s = 13$ in Eqs. (1.3)–(1.6)) [166], and linear multistep methods frequently require only a single evaluation point per step [23]. These values are in stark contrast with those typical of long-step-size implicit methods, for which 30 stages per step is a low number and multiple hundreds of stages per step is not an uncommon amount [170, 16, 15, 126, 141, 33]. Such large numbers

of stages per step make long-step-size implicit methods ideal candidates for parallelization, provided sufficient threads are available.

The two long-step-size methods that have received the greatest amount of attention for astrodynamics applications are MCPI [15] and bandlimited collocation implicit Runge-Kutta (BLC-IRK) [33]. MCPI uses Chebyshev polynomials to approximate both the dynamics and the solution of an ODE. Picard (i.e., fixed-point) iteration is then used to refine an initial guess of the solution at the internal stages of a step until some convergence criterion is met. Alternatively, BLC-IRK approximates the ODE using quadrature based on bandlimited exponential functions rather than polynomials. The method may then be formulated and executed using the standard RK equations (Eqs. (1.3)–(1.6)) [89]. As with MCPI, fixed-point iteration is typically used to solve the nonlinear system of algebraic equations that arises at each propagation step [33].

Both MCPI and BLC-IRK are fixed-step methods, though pseudo-adaptive techniques have been designed. For example, one implementation of MCPI uses *a priori* information of the initial SO orbit to divide each orbital period into several propagation steps, with greatest concentration at periapsis, where the state changes most rapidly [124, 141]. Such a strategy mimics the effect of regularizing the independent variable using, for example, a Sundman transformation [150]. The latter option, classically used with fixed-step methods of all types to efficiently propagate eccentric orbits, is also an option for implicit methods. However, neither predetermined step concentrations nor regularizations provide adaptive step size control in the sense of actively se-

lecting a step size to meet a user-prescribed truncation error tolerance. In fact, the use of a typical variable-step algorithm with a long-time-step implicit propagator is likely to be highly inefficient because the large number of stages (and, therefore, high computational expense) in each step makes recomputing a rejected step excessively expensive. This limitation reduces the applicability of long-step-size implicit methods to 6DOF SO propagation because of the multiple time scales inherent in the 6DOF problem. Nevertheless, both MCPI and BLC-IRK have been shown to perform efficiently for 3DOF propagation scenarios [15, 141, 33, 32, 98].

Short-Step-Size Implicit ODE Solvers The second class of implicit propagators encompasses the “short”-time-step methods, though it is noted that this designation is only intended to convey relation to the long-time-step implicit methods, *not* ERK or linear multistep methods. Of these, the most commonly used for 3DOF SO propagation is the Gauss-Legendre implicit Runge-Kutta (GLIRK) scheme [88], typically with 5 – 20 stages per step. GLIRK has been studied as a fixed-step method [98, 114], and, unlike the long-step-size solvers, as a variable-step method. The lower number of stages per step, coupled with the theoretical properties of GLIRK, make step size adaptation feasible. Two primary variable-step alternatives have been introduced: VGL-s [113] and VGL-IRK [5].

VGL-s uses the theoretical convergence rate of fixed-point iteration to generate propagations of varying order from different iterations of the solution process. The order of the solution at iteration i may be mathematically related

to the order of the solution at iteration $i + 1$, and, thus, the local truncation error (LTE) may be estimated from the differences between the two solutions [193]. The LTE may then be used to predict an appropriate step size in the same manner ubiquitously used by adaptive-step ERK algorithms. On the other hand, VGL-IRK estimates the LTE using a second propagation that is performed alongside the GLIRK propagation [4]. Knowledge of the order of the two propagations and the difference between the solutions allows for estimation of the LTE and selection of a new step size. This scheme is similar to the concept of generating a variable-step ERK algorithm through the use of embedded methods.

VGL-IRK has been further customized to improve performance when applied to uncertainty propagation [8]. Sigma-point filters such as the unscented Kalman filter require the propagation of a population of neighboring state vectors [194]. For relatively small initial uncertainties, the differences between the neighboring state vectors are also small. In this situation, one state vector (e.g., the mean) may be propagated using an unmodified implicit method, and the converged times and states of each internal stage of each step may be saved. Then, when propagating the remaining state vectors, the variable time steps produced by the initial propagation may be reused, eliminating costly rejected steps. This modification is not unique to VGL-IRK; however, each individual step of VGL-IRK is more computationally expensive than popular ERK and linear multistep methods because VGL-IRK is single-step and high-order (and therefore performs more dynamics function evaluations per step than explicit methods). Thus, the elimination of a rejected – and there-

fore recalculated – step is more meaningful for VGL-IRK than for competing methods. In addition, the saved states at each internal stage of each step may be used as accurate initial guesses for the internal stages of the neighboring propagations, potentially reducing the number of iterations required for convergence, depending on the proximity of the neighboring state vectors.

In a serial computing environment, both options, and, in particular, VGL-IRK, have been shown to compete with or outperform typical ERK methods for several typical Earth-orbit state propagation scenarios, though linear multistep methods still provide greatest efficiency in some cases [113, 5, 6]. The effects of parallelization have not been fully examined in published literature, which provides significant motivation for a subset of the material presented in Chapters 3 and 4.

1.2.2.6 Six-Degree-of-Freedom State Propagation

For some applications, propagation of an SO’s translational state alone is not sufficient, and prediction of the SO’s attitude is also required. Perhaps the most obvious example is the determination of the orientation evolution of a controllable SO with pointing requirements. Attitude knowledge of a non-controllable SO involved in proximity operations with another SO may also be important. In addition, body forces like aerodynamic drag and solar radiation pressure (SRP) are dependent on SO orientation. In a pure 3DOF propagation, these dependencies are suppressed by modeling the SO as a uniform sphere or “cannonball.” Under the cannonball assumption, body forces are independent of orientation and do not produce a net torque. However, a higher-fidelity

SO physical model – which frequently consists of some number of flat panels – allows for dynamic changes in time of, for example, the SO surface area exposed to the Sun. Each panel may have unique reflectance properties and aerodynamic coefficients, further improving the approximation of the true SO. As a result, even setting aside the desirability of attitude information for its own sake, the orbital state prediction produced by a 6DOF simulation may differ substantially from a 3DOF prediction if the assumptions of the cannonball model are violated [65]. In the context of SSA, 3DOF propagation may be insufficient to maintain custody of and perform conjunction analysis on such SOs.

Unfortunately, numerical propagation of a coupled orbit/attitude state is particularly challenging, and often requires significantly more compute time than a 3DOF-only propagation. The primary reasons include (1) the increased size of the state vector, (2) the use of potentially complicated SO shape models, and (3) differences between the characteristic time scales of the translational and rotational dynamics, which can result in increased stiffness compared to 3DOF propagation. As a result, 6DOF numerical propagation generally uses much smaller time steps than 3DOF propagation, and the advancement of the state at a single time step is more computationally intensive for 6DOF propagation than for 3DOF propagation.

Several methods have been shown to ease the computational burden of 6DOF propagation. For example, multiple variations of an Encke-type

algorithm have been proposed.⁸ In one such approach, forces independent and dependent on SO attitude are separated from one another in the translational equations of motion [202, 138]. The Encke reference trajectory is taken to be the numerically propagated solution obtained using only attitude-independent forces, which is assumed to take larger step sizes than a coupled orbit/attitude state due to the assumed high frequencies of the rotational motion. Once a step is taken in the reference propagation, the rotational state and corrections to the translational state are propagated using the necessary smaller step sizes, with the reference translational state obtained as needed via interpolation. In this way, unnecessary evaluations of attitude-independent forces are avoided. Naturally, efficiency improvements depend heavily on the relative expenses of the attitude-dependent and attitude-independent dynamics. For example, the combination of a high-degree/order geopotential and a simple SO physical model is likely to lead to strong efficiency gains, while the opposite situation may result in minimal speed gains.

A related method performs the “long-step” propagation using an assumption of constant angular velocity, while the attitude propagation is performed assuming a Keplerian orbit [65]. The two systems are recoupled each time a threshold value of a measure of separation between the two propagations is exceeded, and the split propagations are begun anew. Simulations of a slowly rotating, high-area-to-mass-ratio (HAMR) SO in a near-geosynchronous

⁸In the traditional Encke method for 3DOF propagation, a known reference solution is calculated analytically, and a numerical ODE solver is applied only to deviations from the reference trajectory [21].

orbit (GEO) have demonstrated CPU time savings of approximately 10 percent relative to a fully coupled propagation. As with the previously described IRK customizations, the potential gains of these custom 6DOF propagators must be weighted against the difficulty and expense of implementation. At the same time, the numerical solution of a system of ODEs remains fundamental to the Encke-type methods, and the use of more efficient ODE solvers benefits both fully coupled and semi-coupled propagation approaches.

1.2.3 Sensitivity and State Transition Matrix Calculation

While the importance of numerical algorithms for solving ODEs (Section 1.2.2) cannot be overstated, SO state prediction is often not a practitioner’s only goal. Sensitivities (i.e., partial derivatives) of output variables with respect to input parameters are frequently needed, as well. Two important fields of astrodynamics frequently rely on sensitivity information: trajectory optimization and OD.

Gradient-based root-solving and optimization algorithms use partial derivatives of constraint equations and a performance index with respect to free variables to iteratively drive an initial guess toward a solution [81, 82]. Examples including the Newton-Raphson method and sequential quadratic programming are commonly used to design SO trajectories. All such methods require first-order derivatives, and some benefit from knowledge of second-order derivatives, as well [130, 197].

Meanwhile, the OD process is based on dynamics models and measurements that are both imperfect, which means that nonlinear estimation

techniques are needed to develop a “best guess” of a true SO trajectory. Many of these techniques, such as the minimum-variance solution, rely on first-order sensitivity information [183]. Additionally, as with optimization routines, second-order methods can improve performance [13]. Uncertainty propagation also benefits from the use of higher-order derivatives. Due to the nonlinearity of the orbit problem, the traditional linear propagation of Gaussian uncertainty causes a breakdown in uncertainty realism over sufficiently long time spans [116]. One approach to mitigate this result is to use higher-order derivatives to capture nonlinearity in the uncertainty propagation [160, 67, 159].

When output variables are related to input parameters via a dynamical system, sensitivities are described in part by STMs [183, 156]. Particularly for nonlinear systems – like that of the SO state prediction problem – STM calculation is demanding, and the difficulty only increases if STMs beyond first order are desired. In this dissertation, the implementation of IRK methods for state propagation is augmented by the derivation and analysis of the decoupled direct method (DDM) of STM calculation for RK solvers. In particular, the method is newly derived for second-order STM calculation, as well as for both first- and second-order STM calculation using the double-integrator form of the RK update equations.⁹ Background for the decoupled and coupled direct methods of STM calculation is given in Section 1.2.3.1, while alternative methods of STM calculation are described in Section 1.2.3.2.

⁹The second-order RK update equations are given in Section 4.2.

1.2.3.1 Direct Methods of STM Calculation

The dynamics of the STMs are described using the partial derivatives of the state equations \mathbf{f} with respect to the state \mathbf{x} – i.e., the Jacobian and Hessian for first- and second-order derivatives, respectively. The direct method of STM calculation uses these derivatives to calculate the STMs. In particular, the **coupled direct method** (CDM) appends the physical state with the elements of the STMs. The entire augmented state vector is then propagated simultaneously using the state equations and ODEs for the time evolutions of the STMs [149].¹⁰ The CDM is most commonly used in practice due to ease of implementation, and generally performs well when applied to nonstiff problems using explicit ODE solvers.

An alternative to the CDM is the **decoupled direct method** (DDM) [58]. In the DDM, the state is propagated first, and relevant values (e.g., the time and state) are saved at each evaluation point. The STMs are then calculated after-the-fact in one of two ways:

1. The ODEs for the STMs are propagated using a numerical integration method. In an IRK method, this approach amounts to performing one iterative solve of the RK update equations to obtain the state at each step, then a second iteration procedure to obtain the STMs [171]. This strategy is therefore referenced as the “multi-iteration” approach in this dissertation.

¹⁰The ODEs for first- and second-order STMs are given in Appendix B.

2. The STMs are calculated rather than “propagated” at each step by directly differentiating the ODE solver update equations with respect to the state to obtain expressions for $\partial \mathbf{x}_{m+1} / \partial \mathbf{x}_m$, etc. In an RK method, the differentiation leads to a set of linear equations for the STMs, which are derived in Chapter 4. This method is referenced as the “linear-algebra” approach.

Use of the DDM has been shown to improve calculation stability for stiff problems [58], and opens the door for several efficiency-enhancing modifications, which are discussed more fully in Chapter 4.

A primary disadvantage of either the coupled or decoupled direct method is the requirement of analytical expressions for the Jacobian and Hessian. Even the first-order derivatives of the state equations may be analytically intractable for complicated systems, and the problem only worsens when higher-order derivatives are needed. Further, the Jacobian and Hessian must be rederived if the system dynamics change. However, use of a symbolic manipulator like Mathematica or Maple can significantly reduce these burdens, while simultaneously improving computational efficiency through code generation and optimization. Additionally, for the SO trajectory problem, the same dynamics models are used with such frequency that the *a priori* derivation of a small number of expressions is justifiable.

1.2.3.2 Indirect Methods of STM Calculation

Indirect methods of STM calculation do not rely on analytical implementations of the derivatives of the function of interest with respect to the input parameters. The two main classes of indirect methods are finite-difference (FD) algorithms and algorithmic (also called automatic or computational) differentiation (AD).

Finite-Difference Methods Perhaps the conceptually simplest approach to the computation of sensitivities is the FD method, in which a problem parameter is numerically varied, and the resulting effect on the quantity of interest is observed [104]. The process is performed one or more times for each variable, and the results are substituted into a derivative approximation formula, from which first- and higher-order derivatives of model outputs with respect to inputs may be approximated. For example, the first-order forward-difference formula for a scalar function f of a scalar variable x is

$$\frac{df(x_0)}{dx} \approx \frac{f(x_0 + \delta) - f(x_0)}{\delta}, \quad (1.7)$$

where x_0 is the point of interest and δ is the user-selected perturbation size.

FD methods are primarily attractive for their ease of implementation: Analytical derivatives are not required, and the function of interest need not be modified in any way. These features are particularly important when the source code of the function of interest is not entirely available and editable.

However, there are drawbacks to the simplicity of the FD method. The approximation is subject to two sources of error: truncation error and roundoff error. Truncation error arises from the fact that any derivative approximation formula is only accurate to some finite order in the perturbation size [145]. Increasing the order of the method reduces truncation error at the cost of increasing the number of points at which the function of interest is evaluated – which increases compute time.

Roundoff error is caused by the subtraction of similar numbers (e.g., in the numerator of Eq. (1.7)), which results in precision loss due to the finite-length representation of numbers in computer memory [145]. As truncation error increases as the perturbation size increases, and roundoff error increases as the perturbation size decreases, the accuracy of an FD method is strongly dependent on the selection of a proper perturbation size for each independent variable. Methods that attempt to automatically select an optimal step size exist, though the appropriateness of a step size is local, and such an algorithm must be rerun if the independent variables change significantly [145, 172]. These methods also rely on repeated function evaluations to determine appropriate step sizes, increasing computational effort.

If complex arithmetic is available, complex-step differentiation (CX) may be used instead of traditional FD. In CX, the perturbation δ in Eq. (1.7) is multiplied by the imaginary unit $i \equiv \sqrt{-1}$ [144]. Then, it may be shown that

$$\frac{df(x_0)}{dx} \approx \frac{\text{Im}[f(x_0 + i\delta)]}{\delta}. \quad (1.8)$$

The lack of a subtractive term in Eq. (1.8) means that the perturbation δ may be chosen arbitrarily small without introducing roundoff errors. This feature both improves accuracy and eliminates the need for tuning δ . In turn, the selection of a very small perturbation (e.g., $\delta = 10^{-50}$), can practically eradicate truncation errors. Thus, CX offers significant benefits over traditional FD at the cost of requiring that all operations performed in the evaluation of f accept complex arguments and return appropriate output. Additionally, for higher-order derivatives, multicomplex arithmetic may be introduced [131].

CX may also be formulated as an example of AD, which is described in the following section [144, 143].

Algorithmic Differentiation Methods Algorithmic differentiation computes the derivative of a function by exploiting the fact that the computer implementation of any function may be broken down into a sequence of elementary operations. Each operation may be differentiated individually, and the chain rule is used to obtain the full derivative [85]. Higher-order derivatives may also be calculated [131]. Unlike the FD method, AD is not subject to roundoff error beyond that present in the calculation of the relevant function itself, and the accuracy of AD may be comparable to that of the direct method

[162]. AD can also be faster than FD methods, especially if higher-order finite-differencing is used to reduce truncation error [162, 39]. As implied previously, the primary argument in favor of FD as the indirect method of choice is practical rather than theoretical: AD requires that the function of interest be specially written and compiled to accommodate the desired implementation.

On the other hand, AD may not be as fast as a direct method, even before taking into account the potential savings available from using the DDM [162, 42]. Depending on the implementation¹¹, AD may incur overhead penalties from “carrying” derivatives alongside computations. Additionally, a well-written direct algorithm may be more capable than an AD algorithm of eliminating unnecessary calculations through the use of intermediate variables.

1.2.4 Analytical and Semianalytical Space Object State Prediction Techniques

Sections 1.2.1–1.2.3 focus on the use of numerical ODE solvers for SO state propagation and STM calculation (i.e., SP techniques), which form the basis for Chapters 2–4 of this dissertation. On the other hand, the material of Chapters 5 and 6 combines SP with GP techniques, and a brief overview of the latter topic is presented here.

Unlike SP, a GP method does not “propagate” the SO state by stepping sequentially from t_m to t_{m+1} to t_{m+2} Instead, the state at any time is calculated as an explicit function of the initial state. As a result, a GP method

¹¹When source code is editable, AD may be implemented in several ways (such as operator overloading), and a multitude of software packages are freely available.

can generate a state prediction much more rapidly than an SP method. The cost is that any analytical solution is necessarily a more coarse approximation of the true dynamics than a high-fidelity SP model.

In Section 1.2.4.1, an introduction to a selection of the methodologies commonly used in GP is presented. Subsequently, a background of GP for 3DOF SO state prediction is given in Section 1.2.4.2. Despite the fact that the focus of this dissertation is GP for rotational states, the translational motion application is worth discussing due to its considerable historical interest and the fact that it is still in common use today. Many of the concepts used by translational GP methods are applicable to the SO rotation problem, as well. The concept of 3DOF *semianalytical* methods is introduced in Section 1.2.4.3. Finally, GP methods for rotational motion are discussed in Section 1.2.4.4.

1.2.4.1 Perturbation Methods

Perturbation methods encompass a wide variety of techniques, but all are based on the idea that some difficult-to-solve problems are quantitatively similar to other, easier-to-solve problems [26, 153]. A difficult problem may then be treated as a perturbed version of its less-challenging counterpart. Serendipitously for astrodynamics applications, SO translational and rotational motion are both amenable to this type of analysis due to the existence of analytical solutions to the Kepler problem [20] and the Euler-Poinsot problem

[195], respectively.¹² What follows are brief descriptions of two of the most common approaches used for these applications.

Method of Successive Approximations The method of successive approximations [26] assumes the system given by Eq. (1.2) may be written as

$$\dot{\mathbf{x}} = \mathbf{h}(t, \mathbf{x}) + \epsilon \tilde{\mathbf{h}}(t, \mathbf{x}), \quad (1.9)$$

where ϵ is a small parameter. It is further assumed that the state may be approximated as

$$\mathbf{x} = \mathbf{x}^{(0)} + \Delta \mathbf{x} \quad (1.10)$$

$$\Delta \mathbf{x} = \sum_{n=1}^k \frac{\epsilon^n}{n!} \mathbf{x}^{(n)}, \quad (1.11)$$

for some k . Eq. (1.10) is substituted into Eq. (1.9), and the resulting \mathbf{h} and $\tilde{\mathbf{h}}$ are expanded in Taylor series about $\mathbf{x}^{(0)}$ with perturbation $\Delta \mathbf{x}$. Equating the coefficients of like powers of ϵ on either side of the expanded Eq. (1.9) results in a sequence of linear ODEs for the $\mathbf{x}^{(n)}$. If the ODEs are solved successively, each reduces to a quadrature, assuming $\mathbf{x}^{(0)}$ is known.

¹²The Kepler problem describes the translational motion of a particle subject to the gravitational attraction of a point mass. The Euler-Poinsot problem describes the rotational motion of a triaxial rigid body in the absence of external torques.

The method of successive approximations is frequently applied to a system of ODEs expressed using variation of parameters because this form is naturally likely to contain small perturbations to a known solution [26].

Hamiltonian (Canonical) Perturbation Methods If the dynamical system of interest is Hamiltonian, a wealth of specialized perturbation methods are applicable [26]. A Hamiltonian system is one whose dynamics may be characterized by Hamilton's equations, given by

$$\dot{\mathbf{p}}^T = -\frac{\partial \mathcal{H}}{\partial \mathbf{q}} \quad (1.12)$$

$$\dot{\mathbf{q}}^T = \frac{\partial \mathcal{H}}{\partial \mathbf{p}}, \quad (1.13)$$

where $\mathcal{H} = \mathcal{H}(\mathbf{q}, \mathbf{p}; t)$ is the Hamiltonian, and \mathbf{q} and \mathbf{p} are the generalized coordinates and momenta, respectively. In general, the goal of such methods is the elimination of some or all of the elements of \mathbf{q} from the Hamiltonian, achievement of which simplifies the equations of motion.¹³ If \mathcal{H} is completely independent of \mathbf{q} , all \mathbf{p} are constant in t , and all \mathbf{q} vary linearly in t .

The simplest method of eliminating some or all dependency on a coordinate is the direct averaging of terms in \mathcal{H} that are periodic in that coordinate [128]. More sophisticated methods explicitly construct a canonical transformation from $\{\mathcal{H}, \mathbf{q}, \mathbf{p}\}$ to $\{\mathcal{K}, \mathbf{Q}, \mathbf{P}\}$ such that the new system has desirable

¹³A coordinate that is not present in \mathcal{H} is said to be cyclic or ignorable.

properties (such as \mathcal{K} simpler than \mathcal{H}). In von Zeipel's method [26], the original Hamiltonian is organized such that

$$\mathcal{H}(\mathbf{q}, \mathbf{p}; t; \epsilon) = \sum_{n=0}^{\infty} \epsilon^n \mathcal{H}_{n,0}(\mathbf{q}, \mathbf{p}; t). \quad (1.14)$$

The canonical transformation is then derived via a generating function that depends on both old and new variables. Like \mathcal{H} , the generating function is organized in powers of ϵ , and is calculated term-by-term using the Hamilton-Jacobi partial differential equation (PDE). At each step, the corresponding term of the new Hamiltonian \mathcal{K} is selected (e.g., as an average value of periodic terms in \mathcal{H}), and the PDE is solved for the generating function contribution. Once \mathcal{K} and the generating function are determined to the desired orders in ϵ (not necessarily equal), the equations of motion of the transformed system are (ideally) solved, and the partial derivatives of the generating function are used to transform the solution into the original variables (\mathbf{q} and \mathbf{p}).

Another popular means of constructing the necessary canonical transformation(s) is the Lie-Deprit method [26], which uses a generating function that does not depend on mixed old and new variables (unlike von Zeipel's method). As the Lie-Deprit method is the basis for the perturbation procedures used in this dissertation, the precise procedure is discussed in more detail in Appendix C. It is noted here, however, that this procedure has been extended to apply to non-Hamiltonian systems by Kamel [119].

1.2.4.2 General Perturbation Methods for Space Object Translational Dynamics

When 3DOF GP models were first introduced in the 1950s and '60s, the derivations of analytical methods were significantly motivated by the lack of available computational resources. More than 50 years later, despite exponential increases in computing power, GP methods remain valuable for several reasons. First, the speed of GP methods is still unmatched by SP alternatives. Thus, for applications in which many trajectory predictions are required – and extreme accuracy is not – GP is strongly preferable to SP. Perhaps the most obvious and important example of such an application is conjunction assessment for Earth-orbiting SOs [190]. Performing an initial analysis using GP to determine which pairs of SOs clearly pose no threat to one another over a given time frame – then performing SP propagations for only those SOs deemed at risk – requires significantly less computational time than using SP only for all calculations [123]. Similarly, GP may be applied to initial orbit-design trade studies or rapid orbit determination [189]. Finally, the analytical expressions produced by GP can often provide more insight into the general behavior of a class of SO than the coupled ODEs used by SP methods.

The seminal works of 3DOF GP are the papers of Brouwer [34] and Kozai [128], published in the same issue of *The Astronomical Journal* in 1959. Each author approximates the SO trajectory prediction problem as a Hamiltonian system, taking into account only accelerations due to low-order terms of the spherical harmonics representation of the gravitational field of the central body [121]. Perturbation techniques are then used to obtain an approximate,

analytical solution for the evolution of the trajectory. Specifically, Brouwer’s solution uses von Zeipel’s method [79] to produce two consecutive canonical transformations that result in an integrable transformed Hamiltonian. The central-body oblateness coefficient (J_2) is used as the small parameter that facilitates the perturbation-based solution, and the zonal coefficients J_3 through J_5 are considered to be of order J_2^2 .¹⁴ Periodic perturbations to elliptic motion are retained to order J_2 , and secular perturbations are retained to order J_2^2 .

Alternatively, Kozai directly averages the Hamiltonian to eliminate undesirable terms, then uses the Lagrange planetary equations [183] to obtain the time rates of change of the orbital elements. As in Brouwer’s method, J_2 is used as the small parameter, and periodic and secular terms are retained to first and second order, respectively.

Extensions and improvements of these two works have resulted in numerous GP methods. For example, drag effects, assuming a simple exponential atmosphere, were added to Brouwer’s theory in 1961 [35]. The Lie-Deprit method [52, 119] has been used as an effective alternative to the averaging techniques employed by Brouwer and Kozai [53, 19, 136]. The most well-known method in modern times is Simplified General Perturbations-4 (SGP4) [102, 190], which is based on Brouwer’s work but also takes into consideration the effects of aerodynamic drag and lunar and solar gravity [103]. SGP4 is still employed in the present day to calculate the two-line element sets produced

¹⁴For an Earth-orbiting SO, the J_2 contribution to the Hamiltonian is $\approx 10^{-3}$ relative to the point-mass term.

and made available to the public by the North American Aerospace Defense Command (NORAD) [1].

1.2.4.3 Semianalytical Methods for Space Object Translational Dynamics

Semianalytical methods make up another important class of prediction models. These methods attempt to provide greater accuracy than fully analytical GP methods without the high computational cost of SP methods. For instance, Draper Semianalytic Satellite Theory (DSST) uses both analytical and numerical averaging procedures to derive a system of ODEs for a set of mean orbital elements [44]. The method of multiple scales [153] has also been used to produce semianalytical models [14]. Unlike in a GP method, the ODEs are not solvable analytically; however, the states vary slowly enough that the step sizes used by a numerical propagator may be much larger than those of an SP method for comparable truncation errors [179]. In this regard, the goals of the semianalytical approach are similar to those of the combined application of GP and SP presented in this dissertation. In the latter case, the analytical prediction of the fast-changing rotational state allows for larger propagation step sizes than would be achievable if numerically propagating the full 6DOF state.

1.2.4.4 General Perturbation Methods for Space Object Rotational Dynamics

The rotation of an SO subject to gravity-gradient torque has been studied frequently due to the importance of this disturbance in LEO [83]. Holland

and Sperling [101] directly average Euler’s equation for the time derivative of the rotational angular momentum of a rigid body [94] to obtain a first-order solution for the evolution of the angular momentum vector of an SO subject to gravity-gradient and geomagnetic torques (under certain assumptions). Unfortunately, the theory does not predict the actual orientation of the SO. Crenshaw and Fitzpatrick [48] use the Hamilton-Jacobi equation to obtain a transformation from Euler angles and momenta to a set of variables that are constant for a symmetric rigid body in torque-free rotation. In these variables, the authors use the method of successive approximations to obtain the same first-order solution for an SO subject to gravity-gradient torque as Holland and Sperling [101], but add expressions for the SO orientation. Hitzl and Breakwell [99] use a canonical transformation to write the Hamiltonian for a triaxial rigid body in torque-free rotation independent of coordinates. The gravity-gradient perturbation (assuming an elliptical orbit) is added to the Hamiltonian, and short-period terms are removed through direct averaging, which allows for the determination of the secular rates of the transformed variables. Cochran [46] extends the work of Hitzl and Breakwell [99] to the case of a secularly precessing elliptical orbit. Each of the preceding theories assumes that the rotation rate of the SO is fast compared to the orbital rate, and a value related to the ratio of the mean motion to the rotational speed of the SO is used as the small parameter of the perturbation theories.

More recently, Lara and Ferrer [134], explicitly referring to this problem as that of a fast-rotating satellite, derived a form of the coordinate-free Hamiltonian in terms of action-angle variables, under the assumption of a tri-

axial SO in a circular orbit. The Lie-Deprit transformation procedure [52] is then used to compute a first-order solution that, unlike the previous works, retains the short-period terms. Additionally, the use of action-angle variables eliminates the appearance of mixed secular-periodic terms in the transformation equations. Lara, Ferrer, and coauthors have also applied perturbation theory to other, similar cases, including the rotation of a symmetric SO in a circular orbit subject to gravity-gradient torque [135]. A first-order solution is derived equivalently using both the method of successive approximations and the Lie-Deprit method. Additionally, Lara et al. [137] investigate the rotation of the dwarf planet Ceres due to the gravity-gradient torque of the Sun. The Lie-Deprit method is used to derive an analytical solution in which the non-sphericity of Ceres is treated as a perturbation and the eccentricity of Ceres's orbit is taken into account.

Meanwhile, San-Juan et al. [174] use the Lie-Deprit method to obtain a higher-order solution for a symmetric SO in a circular orbit. Despite the fact that two is the maximum order present in the original perturbed Hamiltonian, improved accuracy in the perturbation solution is achieved by retaining terms up to fourth-order in the transformed Hamiltonian.

GP solutions for other scenarios have also been put forth. For example, Van der Ha [192] uses the method of successive approximations to derive a second-order solution for an SO subject to a torque that is constant in the body-fixed reference frame. Zanardi and coauthors have presented several analytical theories including a first-order solution assuming gravity-gradient and SRP torques (with no Earth shadow) [205]; an orbit-averaged solution for a

spin-stabilized SO subject to magnetic residual torque [206]; and a first-order solution for a gravity-gradient-perturbed, spin-stabilized SO in nonsingular variables [204]. In each case, the perturbation solution is obtained using variation of parameters and the method of successive approximations. Bois and Kovalevsky [29] also utilize the method of successive approximations, producing first-order [27] and second-order [28] solutions for the rotation of a triaxial satellite subject to arbitrary torques. The solution represents torques as Fourier series in the Euler angles describing the orientation of the body and additional angles (such as the SO-Sun angle) that are assumed to vary linearly in time; the coefficients of the series must be determined by the practitioner for each application. Though the theory does not assume fast rotation, deviations from rotation about a single spin axis are assumed to remain small.

In addition to the work previously described for predicting translational and rotational motion individually, the perturbation framework has been applied to the roto-translational problem to obtain a GP solution for full 6DOF motion. These theories are not specifically the subject of this dissertation due to a desire for higher-fidelity prediction of translational motion. However, as with the translation-only GP methods, it is appropriate to mention several examples because of similarities between the 6DOF theories and the rotation-only theories developed in this dissertation.

Ferrandiz and Sansaturio [61] use the Lie-Deprit method to obtain a first-order, average 6DOF solution for a quasi-spherical SO in a Keplerian orbit. This solution is subsequently extended to the case of an oblate central body [62]. Ferrer and Lara [63] use both the Lie-Deprit method and trans-

formations based on the Hamilton-Jacobi equation to address the case of an axisymmetric SO in a Keplerian orbit. Finally, the Lie-Deprit method is also used by Hohmmed et al. [100] to develop a theory for a cylindrical SO that takes into account central-body oblateness and geomagnetic torque. The non-conservative geomagnetic torque is introduced to the Hamiltonian through a “potential-like” function.

As may be expected, the increased dimensionality of the 6DOF problem compared to the independent translational or rotational problems generally leads to the appearance of significantly more terms in the perturbation solutions.

1.3 Organization of Dissertation

Following the introductory Chapter 1, the methods of improving the accuracy, efficiency, and utility of SO state prediction and STM calculation are described in Chapters 2–6. A publication history of the material is given in Appendix F.

Chapter 2 addresses the efficiency and smoothness of the dynamics model by upgrading the modified HP atmospheric density model. Chapter 3 focuses on the numerical ODE solver used by SP and semianalytical methods to propagate the state. Specifically, a parallelized, variable-step-size GLIRK method is applied to 6DOF state prediction. The use of RK ODE solvers for STM calculation using the direct method is then addressed in Chapter 4. Chapters 5 and 6 further concentrate on 6DOF state prediction by presenting

a methodology for using GP-based SO attitude prediction to improve upon the accuracy of cannonball-type 3DOF state prediction without resorting to inefficient fully coupled 6DOF propagation. Finally, Chapter 7 presents a summary of the dissertation and discusses avenues for further study.

1.4 Summary of Contributions

The primary contributions of the dissertation are summarized below.

- A smooth, singularity-free version of the modified HP atmospheric density model is presented. Associated partial derivatives are derived to facilitate sensitivity calculations (Chapter 2).
- The accuracy and smoothness of the upgraded HP model are further improved through the addition of a cubic polynomial dependency on the 81-day average solar flux index $\bar{F}_{10.7}$ (Chapter 2).
- A parallelized, custom, variable-step-size GLIRK solver is presented and applied to coupled orbit/attitude state propagation (Chapter 3).
- Variable-fidelity dynamics models for IRK methods are derived for 6DOF SO state prediction (Chapter 3) and first- and second-order STM calculation (Chapter 4).
- The linear-algebra- and multi-iteration-based DDMs are derived for second-order STM calculation using an RK ODE solver. The two DDM methods are also derived for both first- and second-order STM calculation for the

double-integrator form of the RK update equations. Both methods are implemented in serial and parallel (Chapter 4).

- A closed-form perturbation solution for the rotational motion of a fast-rotating SO in a circular orbit subject to gravity-gradient torque is extended to the case of an elliptical orbit (Chapter 5).
- A semianalytical hybrid SP/GP algorithm, in which the SO translational state is propagated numerically, but body attitude is predicted analytically, is described (Chapter 6).
- Using the perturbation solution presented in Chapter 5 for attitude prediction, the hybrid method is shown to be capable of improving the accuracy of cannonball-based translational state predictions by an order of magnitude or more, while requiring several times less computational effort than a fully numerical 6DOF propagation for certain precision ranges (Chapter 6).

Chapter 2

A Smooth and Robust Harris-Priester Atmospheric Density Model for Low Earth Orbit Applications¹

2.1 Introduction

As introduced in Section 1.2.1, accurate and efficient modeling of thermospheric density is an important component of the prediction of LEO SO trajectories due to the effects of aerodynamic drag. Unfortunately, among the models introduced, accuracy, robustness, and computational efficiency are often competing goals [149]; high-fidelity implementations¹ like Jacchia-Bowman 2008 may take an order of magnitude or more longer to run than the simple 1976 U. S. Standard Atmosphere or TD-88 models.

The modified HP model may be considered as a middle ground between the two extremes [90, 140]. Like the Standard Atmosphere, HP relies on expo-

¹Work from this chapter was presented as:

- **Noble Hatten** and Ryan P. Russell. A smooth and robust Harris-Priester atmospheric density model. Paper AAS 16-406. In *26th AAS/AIAA Space Flight Mechanics Meeting*, Napa, CA, February 2016.

A manuscript has been accepted for publication in a peer-reviewed journal:

- **Noble Hatten** and Ryan P. Russell. A smooth and robust Harris-Priester atmospheric density model for low Earth orbit applications. *Adv Space Res.* doi: 10.1016/j.asr.2016.10.015. (online first November 2016)

In each instance, Ryan P. Russell supervised the work.

nential interpolation of density between values tabulated at discrete altitudes. However, HP also uses functional dependencies to model latitudinal and diurnal effects. Further, HP may be revised to take into account varying levels of solar activity. This effect has been achieved by including a set of 10 tables, each of which corresponds to a different value of the 81-day centered average 10.7 cm solar flux index $\bar{F}_{10.7}$. Given a value of $\bar{F}_{10.7}$, an interpolation scheme, such as nearest-neighbor [57] or linear [184], is used to calculate density values. For example, a linear interpolation scheme is used in the HP implementation available in the Java Astrodynamics Toolkit (JAT) [75, 76, 77].

Thus, HP may produce significantly more accurate density values than a simple exponential atmospheric model while executing in a fraction of the time of more complex models [149]. Such balance makes HP a suitable candidate for use in preliminary studies in which a combination of high speed and reasonable accuracy is paramount. However, even in this context, the HP model is not without its deficiencies. The work in this chapter addresses several of the most important shortcomings – most importantly, the lack of continuity in the first derivatives of the density with respect to the SO state. Additionally, several singularities present in common implementations of HP are identified and addressed. A mechanism for adding smooth, fully functional dependencies on environmental conditions is also introduced to improve the performance of the HP model across environmental conditions. The densities produced by the upgraded HP model are compared to two other HP models: one whose $\bar{F}_{10.7}$ dependence is based on nearest-neighbor interpolation and the JAT linear-interpolation-based model. The performance of each of the three

HP models is also compared to several other commonly used atmospheric density models.

2.2 Traditional Modified Harris-Priester Model

In the traditional modified HP model [140, 149], the complete expression for density ρ as a function of altitude h is given by

$$\rho(h) = \rho_m(h) + [\rho_M(h) - \rho_m(h)] \cos^n \left(\frac{\psi}{2} \right). \quad (2.1)$$

The minimum and maximum densities are obtained from

$$\rho_m(h) = \rho_m(h_i) \exp \left(\frac{h_i - h}{H_{m_i}} \right), \quad h_i \leq h \leq h_{i+1} \quad (2.2)$$

$$\rho_M(h) = \rho_M(h_i) \exp \left(\frac{h_i - h}{H_{M_i}} \right), \quad h_i \leq h \leq h_{i+1}, \quad (2.3)$$

where h_i , $\rho_m(h_i)$, and $\rho_M(h_i)$ are pre-tabulated values. The exponent n varies between 2 for equatorial orbits and 6 for polar orbits, and is intended to roughly take into account latitudinal variations in density. The scale heights H_{m_i} and H_{M_i} are calculated by exponential interpolation to maintain continuity in $\rho(h)$ when passing between altitude layers:

$$H_{m_i} = \frac{h_i - h_{i+1}}{\ln \left(\frac{\rho_m(h_{i+1})}{\rho_m(h_i)} \right)} \quad (2.4)$$

$$H_{M_i} = \frac{h_i - h_{i+1}}{\ln \left(\frac{\rho_M(h_{i+1})}{\rho_M(h_i)} \right)}. \quad (2.5)$$

The sinusoidal term takes into account diurnal density variations, and depends on the position of the Sun through

$$\cos^n\left(\frac{\psi}{2}\right) = \left(\frac{1}{2} + \frac{\mathbf{r}^T \mathbf{u}_b}{2r}\right)^{\frac{n}{2}}, \quad (2.6)$$

where \mathbf{u}_b is a unit vector toward the apex of the diurnal bulge in inertial geocentric (ECI) coordinates; \mathbf{r} is the SO's ECI position vector; and r is the Euclidian norm of \mathbf{r} . The vector \mathbf{u}_b is given by

$$\mathbf{u}_b = \begin{pmatrix} \cos(\delta_s) \cos(\alpha_s + \lambda_{lag}) \\ \cos(\delta_s) \sin(\alpha_s + \lambda_{lag}) \\ \sin(\delta_s) \end{pmatrix}, \quad (2.7)$$

where α_s is the right ascension of the Sun, δ_s is the declination of the Sun, and λ_{lag} is the lag angle between the Sun line and the apex of the diurnal bulge, usually taken to be a constant 30 deg.

2.3 Continuity of Derivatives

The exponential interpolation used by HP ensures continuity of density when passing between discrete altitude layers. However, the discontinuous changes in reference density values create discontinuities in the first derivatives of the density with respect to the state. Discontinuous derivatives can create difficulties for applications ranging from variable-step-size orbit propagation to trajectory estimation and optimization. Expressions for the first derivatives for HP – assuming a specific, approximate expression for geodetic altitude as

a function of SO state – have been given previously [140]. Here, the HP model is revised to ensure continuity of derivatives for all possible states. Two approaches are implemented; the first provides arbitrarily high-order continuity at the expense of computation time, while the second provides more efficient, lower-order continuity.

2.3.1 Approach 1: Analytical Approximation of Heaviside Step Function

The first approach makes the scale heights H_{m_i} and H_{M_i} continuous in altitude by introducing factors based on analytical approximations of the Heaviside step function. (Note that, in the following descriptions, only the revisions made to H_{m_i} are given explicitly; corresponding revisions for H_{M_i} are exactly analogous.)

As a first step, Eq. (2.2) is modified to be a scaled $\rho_m(h_{i+1})$ if h is near h_{i+1} :

$$\rho_m(h) = \rho_m(h_{i+1}) \exp\left(\frac{h_{i+1} - h}{H'_{m_i}}\right), \quad h_i + \frac{h_{i+1} - h_i}{2} \leq h \leq h_{i+1}. \quad (2.8)$$

Regardless of the position of h within an altitude layer, H'_{m_i} replaces H_{m_i} in Eq. (2.2), and is given by

$$H'_{m_i} = \begin{cases} H_{m_{i-1}} + (H_{m_i} - H_{m_{i-1}}) \cdot \\ \left[\mathcal{H}_s\left(\frac{h-h_i}{h_{i+1}-h_i}\right) - \mathcal{H}_s\left(\frac{h-h_{i+1}}{h_{i+1}-h_i}\right) \right], & h_i \leq h \leq h_i + \frac{h_{i+1}-h_i}{2} \\ H_{m_{i+1}} + (H_{m_i} - H_{m_{i+1}}) \cdot \\ \left[\mathcal{H}_s\left(\frac{h-h_i}{h_{i+1}-h_i}\right) - \mathcal{H}_s\left(\frac{h-h_{i+1}}{h_{i+1}-h_i}\right) \right], & h_i + \frac{h_{i+1}-h_i}{2} \leq h \leq h_{i+1} \end{cases}. \quad (2.9)$$

\mathcal{H}_s indicates the Heaviside step function, which may be approximated by

$$\mathcal{H}_s(x) \approx \frac{1}{1 + \exp(-2kx)}, \quad (2.10)$$

where k is a parameter that controls the steepness of the transition from $\mathcal{H}_s(0^-) = 0$ to $\mathcal{H}_s(0^+) = 1$; $k = 100$ is determined empirically to give suitable results for typical applications. Arguments to the Heaviside functions in Eq. (2.9) are scaled to prevent numerical underflow/overflow.

When using the approximation of Eq. (2.10), the H'_{m_i} and their derivatives with respect to altitude are mathematically only continuous in the limit as $k \rightarrow \infty$ due to the subtraction of shifted Heavisides in Eq. (2.9). However, for implementation purposes, $k = 100$ provides continuity to well below double precision machine epsilon. (This situation is depicted graphically in Fig. 2.3.) As shown in Fig. 2.1, $k = 100$ preserves the original values of the scale heights over nearly the entirety of each atmospheric layer.² The relative difference between the original scale height and the Heaviside-modified scale height is less than 1 percent over more than 95 percent of the altitude layer. Given the relatively low fidelity of the HP model and the uncertainty inherent in atmospheric modeling in general, this approximation is more than adequate.

For $h_i \leq h \leq h_{i+1}$, Eq. (2.9) makes use of reference values at either index $(i - 1)$ or index $(i + 2)$, depending on whether h is greater than or less

²Note that the lack of smoothness in Fig. 2.1b is caused by the traditional HP model, not the smoothed version.

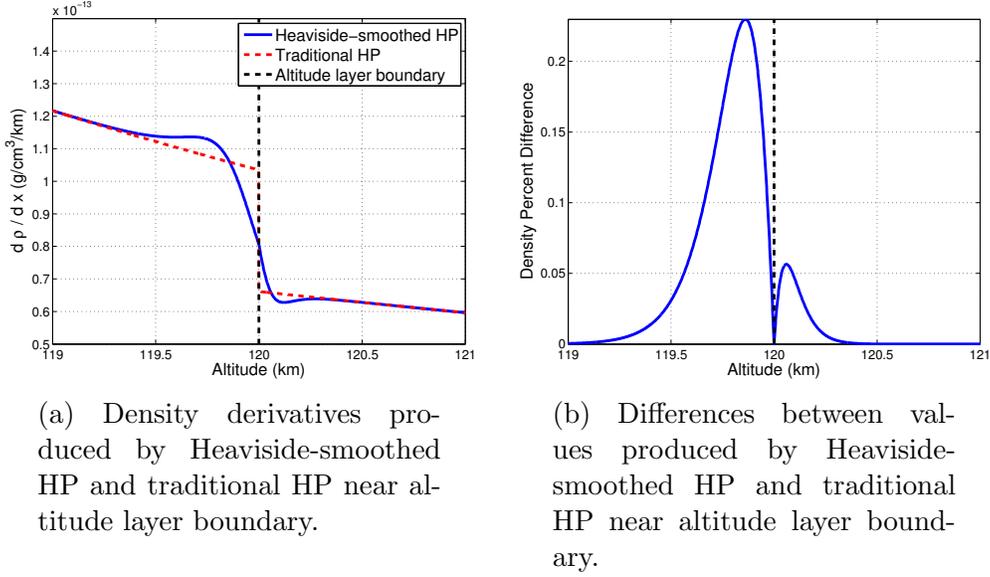


Figure 2.1: Effect of new Heaviside-smoothed HP model on density and its derivatives near altitude layer boundary.

than $h_i + \frac{1}{2}(h_{i+1} - h_i)$. This “knowledge” of scale heights from an adjacent altitude layer allows for the scale heights in the layer of interest to vary to maintain continuity near layer boundaries. However, special measures must be taken when $h \leq h_2$ or $h \geq h_{p-1}$, where p is the total number of indices. If $h \leq h_2$, set

$$h_{i-1} = h_1 - (h_2 - h_1) \quad (2.11)$$

$$\rho_m(h_{i-1}) = \rho_m(h_i) \exp\left(\frac{h_i - h_{i-1}}{H_{m_i}}\right). \quad (2.12)$$

Analogously, if $h \geq h_{p-1}$, set

$$h_{i+2} = h_p + (h_p - h_{p-1}) \quad (2.13)$$

$$\rho_m(h_{i+2}) = \rho_m(h_{i+1}) \exp\left(\frac{h_{i+1} - h_{i+2}}{H_{m_i}}\right). \quad (2.14)$$

The dummy indices simply add a “fake” altitude layer for the model to use and have a negligible impact on the output density values. This method is applicable if the altitude is outside the nominal altitude bounds – that is, if $h < h_1$ or $h > h_p$. In either case, the density and its derivatives are calculated as if the scale heights of the lowermost (or uppermost) altitude layer continue to zero (or infinite) altitude. This strategy smoothly varies the density as altitude changes while avoiding the discontinuous derivatives that would arise if the density were set to a uniform value for an out-of-bounds altitude.

2.3.1.1 Derivatives with Respect to Space Object State

The derivatives of the Heaviside-smoothed HP model with respect to altitude are calculated using

$$\frac{\partial \rho_m(h)}{\partial h} = \begin{cases} -\frac{\rho_m}{H'_{m_i}} \left[1 + \frac{(h_i - h)}{H'_{m_i}} \frac{\partial H'_{m_i}}{\partial h} \right] & h_i \leq h \leq h_i + \frac{h_{i+1} - h_i}{2} \\ -\frac{\rho_m}{H'_{m_i}} \left[1 + \frac{(h_{i+1} - h)}{H'_{m_i}} \frac{\partial H'_{m_i}}{\partial h} \right] & h_i + \frac{h_{i+1} - h_i}{2} \leq h \leq h_{i+1} \end{cases} \quad (2.15)$$

$$\frac{\partial H'_{m_i}}{\partial h} = \begin{cases} (H_{m_i} - H_{m_{i-1}}) \cdot \left[\frac{d\mathcal{H}_s\left(\frac{h-h_i}{h_{i+1}-h_i}\right)}{dh} - \frac{d\mathcal{H}_s\left(\frac{h-h_{i+1}}{h_{i+1}-h_i}\right)}{dh} \right], & h_i \leq h \leq h_i + \frac{h_{i+1}-h_i}{2} \\ (H_{m_i} - H_{m_{i+1}}) \cdot \left[\frac{d\mathcal{H}_s\left(\frac{h-h_i}{h_{i+1}-h_i}\right)}{dh} - \frac{d\mathcal{H}_s\left(\frac{h-h_{i+1}}{h_{i+1}-h_i}\right)}{dh} \right], & h_i + \frac{h_{i+1}-h_i}{2} \leq h \leq h_{i+1} \end{cases}, \quad (2.16)$$

where, generally,

$$\frac{d\mathcal{H}_s(x)}{dx} = \frac{2k \exp(-2kx)}{[1 + \exp(-2kx)]^2}. \quad (2.17)$$

Explicit expressions for the derivatives of traditional HP that are reused in the revised HP have been given previously [140].

2.3.2 Approach 2: Polynomial Weighting Function

The computational efficiency of the Heaviside-based approach suffers relative to the nonsmooth modified HP model because of the additional required evaluations of the exponential function. Alternatively, low-order continuity may be obtained using less expensive polynomial weighting functions [111]. In this approach, near an altitude layer boundary h_i , the scale height is a polynomial-weighted combination of the scale height valid over $[h_{i-1}, h_i]$ and the scale height valid over $[h_i, h_{i+1}]$ (i.e., $H_{m_{i-1}}$ and H_{m_i} , respectively). Far from any h_i , the weighting is unnecessary, and the CPU time required to calculate the scale heights decreases accordingly.

As in the Heaviside-based approach, Eq. (2.2) is modified to be a scaled $\rho_m(h_{i+1})$ if h is near h_{i+1} , and the calculation of the scale heights proceeds slightly differently depending on the relative location of h within the region $[h_i, h_{i+1}]$:

$$\rho_m(h) = \rho_m(h_{i+1}) \exp\left(\frac{h_{i+1} - h}{H'_{m_i}}\right), \quad h_{i+1} - \alpha \leq h \leq h_{i+1} \quad (2.18)$$

$$H'_{m_i} = \begin{cases} H_{m_{i-1}} + w_H (H_{m_i} - H_{m_{i-1}}), & h_i \leq h \leq h_i + \alpha \\ H_{m_i} + w_H (H_{m_{i+1}} - H_{m_i}), & h_{i+1} - \alpha \leq h \leq h_{i+1} \\ H_{m_i}, & h_i + \alpha \leq h \leq h_{i+1} - \alpha \end{cases} \quad (2.19)$$

The parameter α controls the size of the region in which weighting is applied, and is set to 0.5 km for all regions. (Altitude regions vary in size from 10 km to 40 km in the current implementation.) The precise expression for the weighting function w_H depends on the desired order of continuity. For example, for third-order continuity,

$$w_H = \xi_H^4 (-20\xi_H^3 + 70\xi_H^2 - 84\xi_H + 35) \quad (2.20)$$

$$\xi_H = \begin{cases} \frac{h - (h_i - \alpha)}{2\alpha}, & h_i \leq h \leq h_i + \alpha \\ \frac{h - (h_{i+1} - \alpha)}{2\alpha}, & h_{i+1} - \alpha \leq h \leq h_{i+1} \end{cases} \quad (2.21)$$

Unlike the Heaviside-smoothed HP, proximity to the bottommost or topmost altitude index does not present a special case; the unweighted HP is used in these instances. If the altitude is greater than the highest reference altitude or less than the lowest reference altitude, the extrapolation technique used in the Heaviside-smoothed model is again employed, though no additional “fake” altitude layers are required.

Fig. 2.2 displays the results of enforcing third-order continuity in the scale heights using the polynomial weighting function approach. The weighting function is applied only when $h \in [h_i - 0.5 \text{ km}, h_i + 0.5 \text{ km}]$. The effect closely

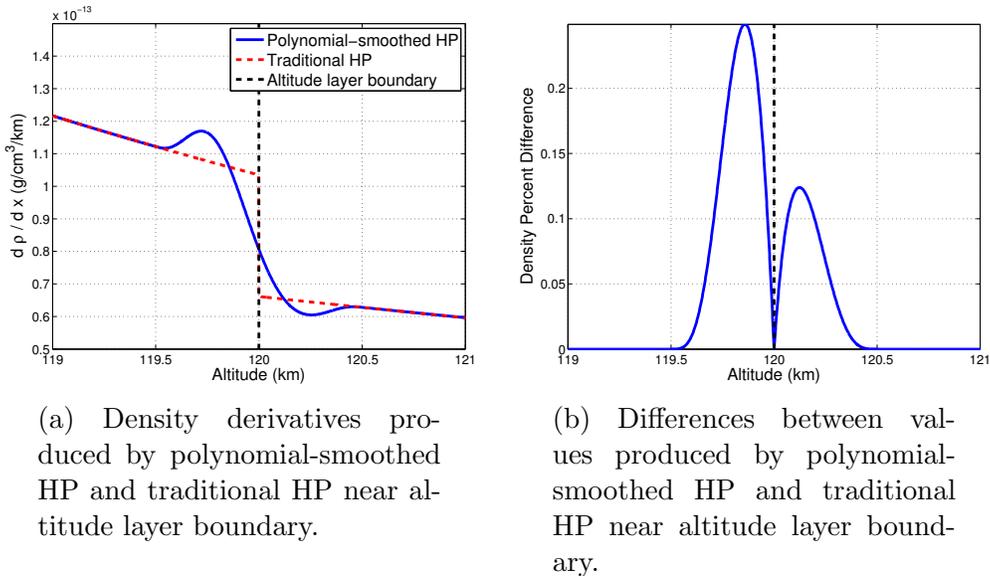


Figure 2.2: Effect of new polynomial-smoothed HP model on density and its derivatives near altitude layer boundary.

mimics that of the Heaviside-based smoothing approach (Fig. 2.1), with only a marginally larger overshoot in the partial derivatives on either side of the altitude layer boundary.

Fig. 2.3 shows the effect of the two smoothing methods on the scale height H'_{m_i} as a function of altitude. For the nominal parameter values $k = 100$ and $\alpha = 0.5$ km (Fig. 2.3a), the transition between altitude layers is nearly identical for the two smoothing methods, and both closely mimic the behavior of the traditional HP model. Fig. 2.3b depicts the previously-discussed discontinuities that develop if an unsuitably small value of k is used with the Heaviside-smoothing technique. On the other hand, Fig. 2.3b also shows that increasing α when using the polynomial-smoothing method does not create

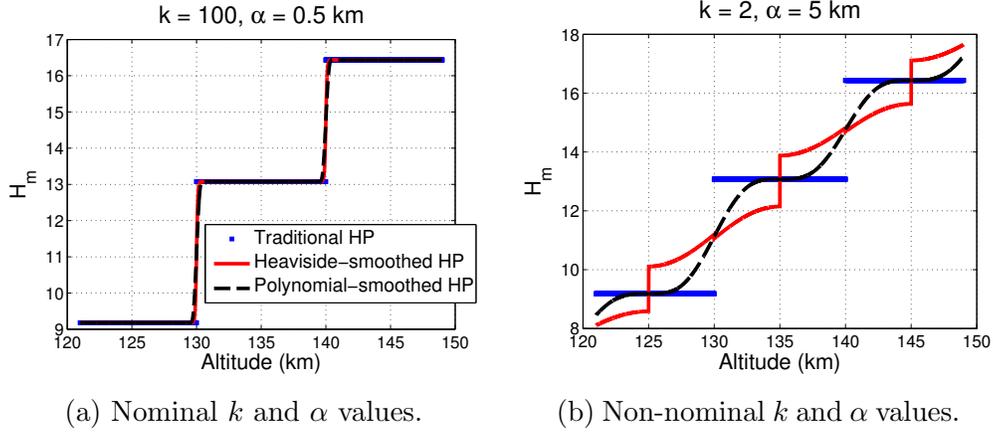


Figure 2.3: Comparison of scale height H'_{m_i} as function of altitude. Plot titles give k parameter value for Heaviside-smoothed HP and α parameter value for polynomial-smoothed HP.

discontinuities, and instead introduces a more gradual transition between the scale heights of adjacent altitude layers.

2.3.2.1 Derivatives with Respect to Space Object State

The derivatives of the polynomial-smoothed HP model with respect to altitude are calculated using

$$\frac{\partial \rho_m(h)}{\partial h} = \begin{cases} -\frac{\rho_m}{H'_{m_i}} \left[1 + \frac{(h_i - h)}{H'_{m_i}} \frac{\partial H'_{m_i}}{\partial h} \right], & h_i \leq h \leq h_i + \alpha \\ -\frac{\rho_m}{H'_{m_i}} \left[1 + \frac{(h_{i+1} - h)}{H'_{m_i}} \frac{\partial H'_{m_i}}{\partial h} \right], & h_{i+1} - \alpha \leq h \leq h_{i+1} \end{cases} \quad (2.22)$$

$$\frac{\partial H'_{m_i}}{\partial h} = \begin{cases} \left(\frac{H_{m_i} - H_{m_{i-1}}}{2\alpha} \right) \frac{dw_H}{d\xi_H}, & h_i \leq h \leq h_i + \alpha \\ \left(\frac{H_{m_{i+1}} - H_{m_i}}{2\alpha} \right) \frac{dw_H}{d\xi_H}, & h_{i+1} - \alpha \leq h \leq h_{i+1} \end{cases}. \quad (2.23)$$

If the altitude is not near an altitude layer boundary, weighting is not required, and derivatives are calculated using nonsmoothed expressions.

2.3.3 Derivatives with Respect to Time

Under the assumptions that (1) the Earth may be modeled as an ellipse of revolution and (2) the ECI and Earth-centered, Earth-fixed (ECEF) reference frames differ only by a rotation about a common polar axis, the position of the Sun represents the only explicit time dependence of the HP density model. The position of the Sun with respect to the Earth may be approximated as an explicit function of time whose time derivative is easily derived analytically; thus, $d\mathbf{u}_b/dt$ (with \mathbf{u}_b defined in Eq. (2.7)) and, therefore, $\partial\rho/\partial t$ are readily calculable [187]. On the other hand, if a higher-fidelity relationship between the ECI and ECEF frames is modeled and assumption (2) is violated, then $\partial h/\partial t$ is generally nonzero. Further, as is described in Section 2.5, the Harris-Priester model may be modified to include dependencies on environmental parameters such as $\bar{F}_{10.7}$, which may be approximated as explicit functions of time. Therefore, $\partial\rho/\partial t$ may be written as

$$\frac{\partial\rho}{\partial t} = \frac{\partial\rho}{\partial\mathbf{u}_b} \frac{d\mathbf{u}_b}{dt} + \frac{\partial\rho}{\partial h} \frac{\partial h}{\partial t} + \frac{\partial\rho}{\partial\bar{F}_{10.7}} \frac{d\bar{F}_{10.7}}{dt}. \quad (2.24)$$

The first subexpression of Eq. (2.24) is given by

$$\frac{\partial\rho}{\partial\mathbf{u}_b} = \frac{n}{4} \left(\frac{1}{2} + \frac{\mathbf{u}_r^T \mathbf{u}_B}{2} \right)^{\left(\frac{n}{2}-1\right)} \mathbf{u}_r^T. \quad (2.25)$$

The expression for $\partial\rho/\partial h$ is obtained by combining Eq. (2.1) with either Eq. (2.15) (for the Heaviside weighting approach) or Eq. (2.22) (for the

polynomial weighting approach) and applying the chain rule. The expression for $\partial h/\partial t$ depends on the specific method used to calculate h but generally requires the time derivative of the transformation matrix relating the ECI and ECEF reference frames – another quantity that is implementation-specific. Similarly, $d\bar{F}_{10.7}/dt$ depends on the model used to approximate the time evolution of solar activity.

2.4 Elimination of Singularities

Common implementations of the HP model derivatives contain singularities if one or more of several conditions occur [140]. In the interest of a fully robust implementation of the model, these singularities are identified and nonsingular alternative expressions are presented.

2.4.1 Collinearity of Space Object and Diurnal Bulge Position Vectors

The partial derivative of density with respect to altitude [140] may be singular when \mathbf{u}_r and \mathbf{u}_b are collinear, depending on the implementation of the chain rule for differentiation. The singularity may be avoided, if, rather than differentiating with respect to the angle ψ directly, the relevant expression is differentiated as

$$\frac{\partial \rho(h)}{\partial [\cos^n(\frac{\psi}{2})]} = \rho_M(h) - \rho_m(h) \quad (2.26)$$

$$\frac{\partial [\cos^n(\frac{\psi}{2})]}{\partial \mathbf{r}} = \frac{n}{4} \left(\frac{1}{2} + \frac{\mathbf{u}_r^T \mathbf{u}_B}{2} \right)^{\left(\frac{n}{2}-1\right)} \left(\frac{\mathbf{u}_B^T}{r} - \frac{\mathbf{u}_r^T \mathbf{u}_B}{r} \mathbf{u}_r^T \right), 2 < n \leq 6. \quad (2.27)$$

The strict inequality on the lower bound for n ensures that the exponent $(n/2 - 1)$ is never exactly zero. Because 0^0 is undefined, $(n/2 - 1) = 0$ would lead to discontinuities if $\cos(\psi/2) = 0$. This limitation on n prevents similar computational difficulties in Eq. (2.25).

2.4.2 Space Object at Pole

Many methods exist for converting geocentric coordinates to geodetic coordinates. Given the desirability of high speed with moderate accuracy when using the HP model, it is natural that a previous HP implementation relies on a simple, explicit, approximate equation for obtaining the SO's altitude above the reference Earth ellipsoid [140]. However, the expressions for the derivatives of h with respect to the SO state are singular for ECEF position vectors directly above a geographic pole. Fortunately, the singularity may be avoided by switching from the general expressions to alternative, correct expressions for derivatives at the poles if $r_1^E = r_2^E = 0$. (The superscript E represents that the position vector is expressed in the ECEF frame.) If at a pole, then the correct expression is

$$\frac{\partial h}{\partial \mathbf{r}^E} = \begin{pmatrix} 0 \\ 0 \\ \text{sign}(1, r_3^E) \end{pmatrix}^T. \quad (2.28)$$

In a manner directly analogous to that implemented in the polynomial-smoothing scheme for the scale heights, a polynomial weighting function [111] is used near the poles to (1) maintain continuity and smoothness in $\partial h / \partial \mathbf{r}^E$

and (2) ensure that $\partial h/\partial \mathbf{r}^E$ is the true derivative of h . If the cosine of the geocentric latitude δ is less than a preset tolerance τ_p (e.g., $\tau_p = 10^{-4}$), then

$$h = w_{p,1} [h]_g + w_{p,2} [h]_p \quad (2.29)$$

$$\frac{\partial h}{\partial \mathbf{r}^E} = \frac{\partial w_{p,1}}{\partial \mathbf{r}^E} [h]_g + w_{p,1} \left[\frac{\partial h}{\partial \mathbf{r}^E} \right]_g + \frac{\partial w_{p,2}}{\partial \mathbf{r}^E} [h]_p + w_{p,2} \left[\frac{\partial h}{\partial \mathbf{r}^E} \right]_p \quad (2.30)$$

$$\frac{\partial w_{p,i}}{\partial \mathbf{r}^E} = \frac{\partial w_{p,i}}{\partial \xi_p} \frac{\partial \xi_p}{\partial \mathbf{r}^E}, \quad i = 1, 2 \quad (2.31)$$

$$w_{p,1} = \xi_p^3 (6\xi_p^2 - 15\xi_p + 10) \quad (2.32)$$

$$w_{p,2} = 1 - w_{p,1} \quad (2.33)$$

$$\xi_p = \frac{\cos \delta}{\tau_p}, \quad (2.34)$$

where the subscript g refers to the altitude calculated using the general altitude expression, and the subscript p refers to the altitude calculated assuming the SO is exactly above a pole. The specific expressions for $w_{p,1}$ and $w_{p,2}$ ensure second-order continuity of the altitude. Fig. 2.4 shows the effect of using the weighting-function scheme on h (Fig. 2.4a) and $\partial h/\partial \mathbf{r}^E$ (Fig. 2.4b). The results are generated by setting $r_2^E = 0$, $r_3^E = 6478$ km, and varying r_1^E between -1 km and 1 km. Errors in altitude are on the order of millimeters, and the weighted $\partial h/\partial \mathbf{r}^E$ varies smoothly near the pole. Additionally, by applying the weighting functions to the altitude function itself rather than $\partial h/\partial \mathbf{r}^E$, the partial derivatives are analytically correct.

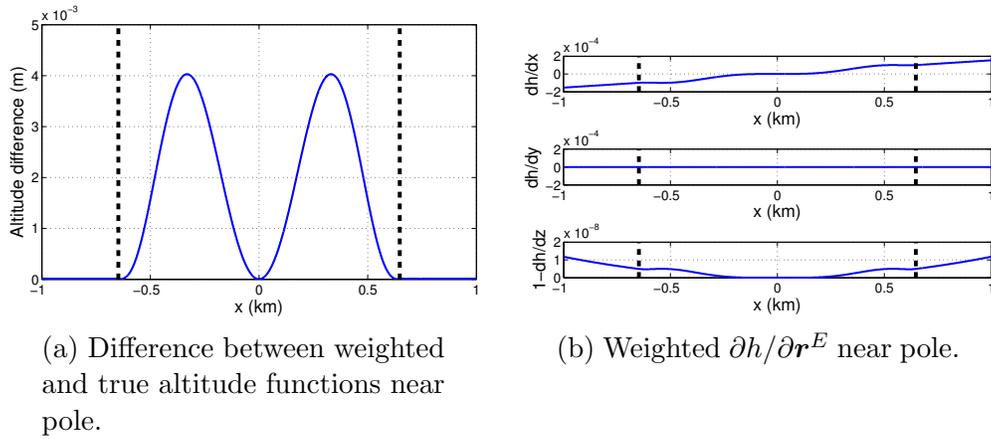


Figure 2.4: Effect of weighting function on altitude and $\partial h/\partial \mathbf{r}^E$ near pole. Vertical dashed lines indicate τ_p .

2.4.3 Functional form of Latitude Dependence

In order to vary the latitudinal dependence parameter n continuously from 2 for equatorial orbits to 6 for polar orbits while retaining continuity and continuous partial derivatives, n is set using the square of the sine of the orbital inclination. Calculating n using the inclination itself [75, 184] or the sine of the inclination introduces a singularity in the derivative of the density with respect to the SO state for equatorial orbits. The new formulation, given by Eq. (2.35), only produces a singularity if the magnitude of the angular momentum vector is zero (degenerate case). The derivatives of n are given in Eqs. (2.36)–(2.37); Eq. (2.38) provides the derivative of the density with respect to n .

$$n = 2 + \epsilon_n + 4 \sin^2 i \quad (2.35)$$

$$\frac{\partial n}{\partial \mathbf{r}} = 4 \frac{\partial \sin^2 i}{\partial \mathbf{r}} = \frac{8h_3}{h^4} \begin{pmatrix} -h_2 h_3 v_3 - (h_1^2 + h_2^2) v_2 \\ h_1 h_3 v_3 + (h_1^2 + h_2^2) v_1 \\ -h_1 h_3 v_2 - h_2 h_3 v_1 \end{pmatrix}^T \quad (2.36)$$

$$\frac{\partial n}{\partial \mathbf{v}} = 4 \frac{\partial \sin^2 i}{\partial \mathbf{v}} = \frac{8h_3}{h^4} \begin{pmatrix} h_2 h_3 r_3 - (h_1^2 + h_2^2) r_2 \\ -h_1 h_3 r_3 - (h_1^2 + h_2^2) r_1 \\ h_1 h_3 r_2 - h_2 h_3 r_1 \end{pmatrix}^T \quad (2.37)$$

$$\frac{\partial \rho}{\partial n} = (\rho_M(h) - \rho_m(h)) \cos^n \left(\frac{\psi}{2} \right) \ln \left[\cos \left(\frac{\psi}{2} \right) \right], \quad (2.38)$$

where $\mathbf{v} = [v_1 \ v_2 \ v_3]^T$ is the SO's ECI velocity vector and $\mathbf{h} = \mathbf{r} \times \mathbf{v} = [h_1 \ h_2 \ h_3]^T$ is the SO's ECI angular momentum vector (h is the Euclidean norm of \mathbf{h}). As implied previously, ϵ_n is a small, positive quantity used to prevent $n = 2$ (e.g., $\epsilon_n = 10^{-3}$).

A problem arises in Eq. (2.38) if $\psi = \pi$ because $\ln(0) = -\infty$. (There is no risk of $\psi > \pi$.) As in the polar position case, an alternative expression combined with a weighting function to ensure continuity and smoothness is used to produce correct results for $\partial \rho / \partial n$ for ψ near π . Here,

$$\cos^n \left(\frac{\psi}{2} \right) = \begin{cases} 0, & \cos \left(\frac{\psi}{2} \right) = 0 \\ [\cos^n \left(\frac{\psi}{2} \right)]_{\mathbf{g}}, & \cos \left(\frac{\psi}{2} \right) > \tau_n \\ w_{n,1} [\cos^n \left(\frac{\psi}{2} \right)]_{\mathbf{g}}, & 0 < \cos \left(\frac{\psi}{2} \right) < \tau_n \end{cases} \quad (2.39)$$

$$\frac{\partial [\cos^n \left(\frac{\psi}{2} \right)]}{\partial n} = \begin{cases} 0, & \cos \left(\frac{\psi}{2} \right) = 0 \\ [\cos^n \left(\frac{\psi}{2} \right)]_{\mathbf{g}} \ln [\cos \left(\frac{\psi}{2} \right)], & \cos \left(\frac{\psi}{2} \right) > \tau_n \\ w_{n,1} [\cos^n \left(\frac{\psi}{2} \right)]_{\mathbf{g}} \ln [\cos \left(\frac{\psi}{2} \right)], & 0 < \cos \left(\frac{\psi}{2} \right) < \tau_n \end{cases} \quad (2.40)$$

$$w_{n,1} = \xi_n^3 (6\xi_n^2 - 15\xi_n + 10) \quad (2.41)$$

$$\xi_n = \frac{\cos \left(\frac{\psi}{2} \right)}{\tau_n}, \quad (2.42)$$

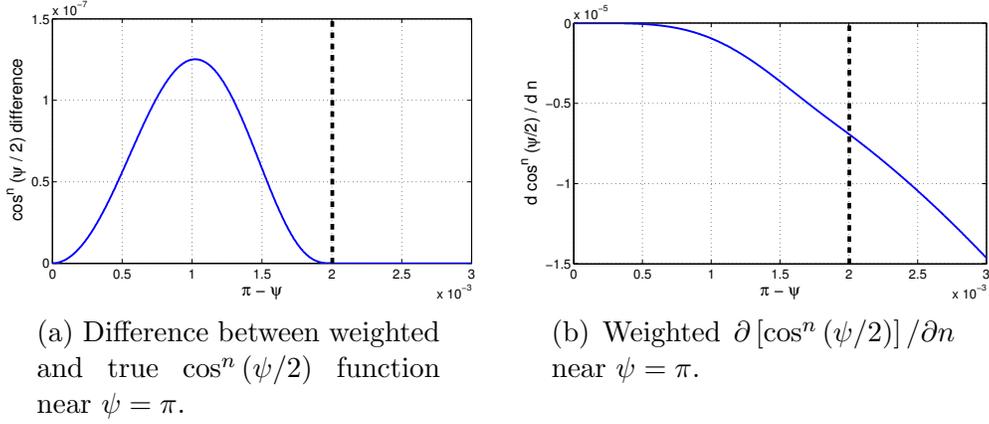


Figure 2.5: Effect of weighting function on $\cos^n(\psi/2)$ and $\partial[\cos^n(\psi/2)]/\partial n$ near $\psi = \pi$. Vertical dashed lines indicate τ_n .

where τ_n is a preset tolerance (e.g., $\tau_n = 10^{-3}$), and $[\cos^n(\frac{\psi}{2})]_g$ is obtained from Eq. (2.6). The derivative of $w_{n,1}$ with respect to n is zero, and the corresponding term is omitted from Eq. (2.40). Additionally, the specific expression for $w_{n,1}$ ensures second-order continuity of $\cos^n(\frac{\psi}{2})$. Fig. 2.5 shows the effect of using the weighting-function scheme on $\cos^n(\psi/2)$ (Fig. 2.5a) and $\partial[\cos^n(\psi/2)]/\partial n$ (Fig. 2.5b). For comparison, the alternative unweighted solution switches instantaneously between $[\cos^n(\frac{\psi}{2})]_g$ and $[\cos^n(\frac{\psi}{2})] = 0$ at $\cos(\psi/2) = \tau_n$.

2.5 Introduction of Environmental Parameter Dependencies

As an alternative to the traditional multi-table, interpolation-based approach to modeling effects caused by variations in $\bar{F}_{10,7}$, this chapter introduces a strictly functional dependence of $\rho_m(h_i)$ and $\rho_M(h_i)$ on any combination

of environmental parameters. An example that closely mimics the effect of the previously described linear-interpolation strategy (hereafter referenced as LIHP) is to use a cubic dependency on $\bar{F}_{10.7}$:

$$\rho_m(h_i) = \sum_{j=0}^3 c_{m_i,j} \bar{F}_{10.7}^j, \quad (2.43)$$

where the $c_{m_i,j}$ are preset constants. (Calculation of $\rho_M(h_i)$ proceeds exactly analogously with different coefficient values: $c_{M_i,j}$.) This approach (referenced as cubic Harris-Priester or CHP) yields several advantages over LIHP and the nearest-neighbor interpolation strategy (NNHP). First, derivatives of the density with respect to $\bar{F}_{10.7}$ are now continuous throughout the $\bar{F}_{10.7}$ domain. Second, a cubic interpolation scheme leads to a more efficient memory management requirement, given the same accuracy level. Third, and perhaps most importantly, the functional approach is easily upgradable: The dependence on $\bar{F}_{10.7}$ may be changed and/or further dependencies on other parameters may be added in a straightforward manner. The inclusion of additional environmental parameters in the model, of course, increases the dimensionality of the fit and may require nonlinear fitting techniques. However, the great majority of additional computational effort created by this complication lies in coefficient generation rather than model execution.

An alternative to the implementation presented in this chapter is to combine the fitting of any environmental parameters with a global fit in altitude. Such a methodology moves further away from the traditional HP model by eliminating the piecewise exponential interpolation in altitude (Eqs. (2.2)

and (2.3)) and the use of altitude layers. Depending on the functional dependencies used, a global fit would likely reduce the resolution of the model while potentially improving computational speed. A global altitude fit is avoided in the current work, which endeavors to improve the quality of the HP model while maintaining its general structure.

2.5.1 Calculation of Coefficients

The $c_{m_i,j}$ in Eq. (2.43) are calculated via a linear least-squares procedure, using “measurements” produced by a higher-fidelity density model. (In the current implementation, NRLMSISE-2000 [164] is used for this purpose due to its direct dependence on $\bar{F}_{10.7}$ and lack of dependence on the year.) For each of ρ_m and ρ_M , and for each reference altitude h_i , the higher-fidelity model is evaluated within a series of nested loops to obtain density values at representative combinations of input parameters: $\bar{F}_{10.7}$ (varied between 60 sfu and 240 sfu), longitude, day-of-year, latitude, and local apparent solar time.³ To mimic the original HP model, the densities are averaged over longitude, day-of-year, and latitude, and extremized over the set of local apparent solar times (minimized for ρ_m , maximized for ρ_M). This process leads to a single density value for each of ρ_m and ρ_M at each $\bar{F}_{10.7}$ input value at each reference altitude h_i . Once the process has been completed at all input parameter combinations for a given h_i , a set of measurements is available, and a linear least-squares fit is used to obtain the coefficients necessary to evaluate Eq. (2.43).

³For all evaluations, the daily value of the solar flux index $F_{10.7}$ is set equal to $\bar{F}_{10.7}$, and the geomagnetic index is set such that $A_p = 11$ (a moderate value).

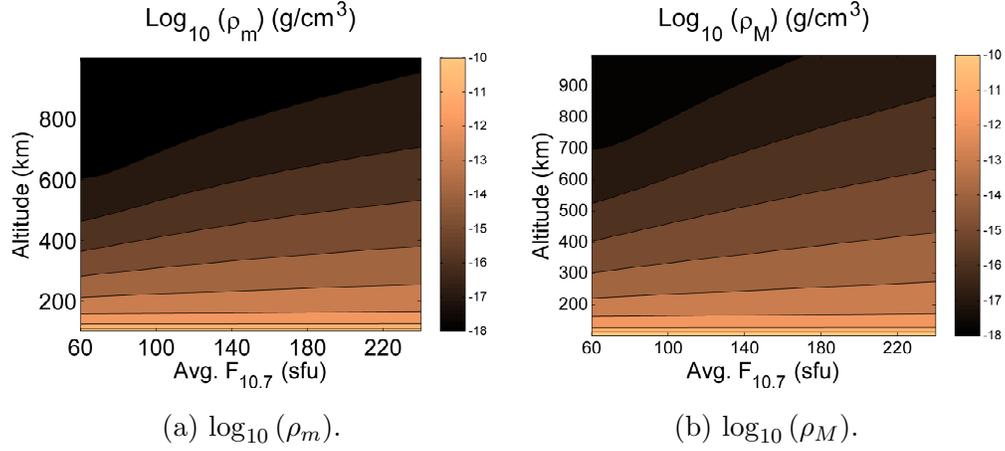


Figure 2.6: Contour plots of the logarithms of ρ_m and ρ_M as functions of altitude and solar flux.

Fig. 2.6 shows ρ_m and ρ_M as functions of both altitude and $\bar{F}_{10.7}$ in the current implementation. The dependence of density on the solar flux is clearly visible, particularly at higher altitudes, demonstrating the importance of including this relationship in the model.

2.5.2 Derivatives with Respect to $\bar{F}_{10.7}$

To facilitate the calculation of sensitivities with respect to solar activity – and, possibly, time (see Section 2.3.3) – expressions for $\partial\rho/\partial\bar{F}_{10.7}$ are developed for the CHP model. Differentiation of Eq. (2.1) gives

$$\frac{\partial\rho(h)}{\partial\bar{F}_{10.7}} = \frac{\partial\rho_m(h)}{\partial\bar{F}_{10.7}} + \left(\frac{\partial\rho_M(h)}{\partial\bar{F}_{10.7}} - \frac{\partial\rho_m(h)}{\partial\bar{F}_{10.7}} \right) \cos^n\left(\frac{\psi}{2}\right). \quad (2.44)$$

At this point, the derivation diverges depending on which continuity-preserving method is used and on the location of h within an altitude layer. However, for all cases, three useful expressions hold:

$$L_{m_{i+1},i} \triangleq \log \left(\frac{\rho_m(h_i)}{\rho_m(h_{i-1})} \right) \quad (2.45)$$

$$\frac{\partial H_{m_i}}{\partial \bar{F}_{10.7}} = \left(\frac{h_i - h_{i+1}}{L_{m_{i+1},i}^2} \right) \left(\frac{\partial \rho_m(h_i) / \partial \bar{F}_{10.7}}{\rho_m(h_i)} - \frac{\partial \rho_m(h_{i+1}) / \partial \bar{F}_{10.7}}{\rho_m(h_{i+1})} \right) \quad (2.46)$$

$$\frac{\partial \rho_m(h_i)}{\partial \bar{F}_{10.7}} = c_{m_{i,1}} + 2c_{m_{i,2}} \bar{F}_{10.7} + 3c_{m_{i,3}} \bar{F}_{10.7}^2. \quad (2.47)$$

These relations and the following derivations are given for the minimum density case only, with the understanding that the substitution $m \leftarrow M$ may be used to obtain the expressions for the maximum density case.

2.5.2.1 Heaviside-Based Continuity

For both the lower and upper portions of an altitude region, the difference between the Heaviside functions referenced to h_i and h_{i+1} is abbreviated as

$$\Delta \mathcal{H}_s \triangleq \mathcal{H}_s \left(\frac{h - h_i}{h_{i+1} - h_i} \right) - \mathcal{H}_s \left(\frac{h - h_{i+1}}{h_{i+1} - h_i} \right). \quad (2.48)$$

$h_i \leq h \leq h_i + \frac{h_{i+1} - h_i}{2}$ In the lower portion of an altitude region, differentiating Eq. (2.2) gives

$$\frac{\partial \rho_m(h)}{\partial \bar{F}_{10.7}} = \frac{\partial \rho_m(h_i)}{\partial \bar{F}_{10.7}} E_m + \rho_m(h_i) \frac{\partial E_m}{\partial \bar{F}_{10.7}} \quad (2.49)$$

$$E_m \triangleq \exp\left(\frac{h_i - h}{H'_{m_i}}\right). \quad (2.50)$$

Continuing to differentiate gives

$$\frac{\partial E_m}{\partial \bar{F}_{10.7}} = \left(\frac{h - h_i}{H'^2_{m_i}}\right) E_m \frac{\partial H'_{m_i}}{\partial \bar{F}_{10.7}}. \quad (2.51)$$

The derivative of H'_{m_i} is given by

$$\frac{\partial H'_{m_i}}{\partial \bar{F}_{10.7}} = \frac{\partial H_{m_{i-1}}}{\partial \bar{F}_{10.7}} + \Delta \mathcal{H}_s \left(\frac{\partial H_{m_i}}{\partial \bar{F}_{10.7}} - \frac{\partial H_{m_{i-1}}}{\partial \bar{F}_{10.7}} \right). \quad (2.52)$$

$h_i + \frac{h_{i+1} - h_i}{2} \leq h \leq h_{i+1}$ In the upper portion of an altitude region, differentiating Eq. (2.2) gives

$$\frac{\partial \rho_m(h)}{\partial \bar{F}_{10.7}} = \frac{\partial \rho_m(h_{i+1})}{\partial \bar{F}_{10.7}} E_m + \rho_m(h_{i+1}) \frac{\partial E_m}{\partial \bar{F}_{10.7}} \quad (2.53)$$

$$E_m \triangleq \exp\left(\frac{h_{i+1} - h}{H'_{m_i}}\right). \quad (2.54)$$

Continuing to differentiate gives

$$\frac{\partial E_m}{\partial \bar{F}_{10.7}} = \left(\frac{h - h_{i+1}}{H'^2_{m_i}}\right) E_m \frac{\partial H'_{m_i}}{\partial \bar{F}_{10.7}}. \quad (2.55)$$

The derivative of H'_{m_i} is given by

$$\frac{\partial H'_{m_i}}{\partial \bar{F}_{10.7}} = \frac{\partial H_{m_{i+1}}}{\partial \bar{F}_{10.7}} + \Delta \mathcal{H}_s \left(\frac{\partial H_{m_i}}{\partial \bar{F}_{10.7}} - \frac{\partial H_{m_{i+1}}}{\partial \bar{F}_{10.7}} \right). \quad (2.56)$$

2.5.2.2 Polynomial-Based Continuity

$h_i \leq h \leq h_i + \alpha$ In the lower portion of an altitude region, Eqs. (2.49)–(2.51) hold if H'_{m_i} is calculated using Eq. (2.19). Continuing to differentiate gives

$$\frac{\partial H'_{m_i}}{\partial \bar{F}_{10.7}} = \frac{\partial H_{m_{i-1}}}{\partial \bar{F}_{10.7}} + w_H \left(\frac{\partial H_{m_i}}{\partial \bar{F}_{10.7}} - \frac{\partial H_{m_{i-1}}}{\partial \bar{F}_{10.7}} \right). \quad (2.57)$$

$h_{i+1} - \alpha \leq h \leq h_{i+1}$ In the upper portion of an altitude region, Eqs. (2.53)–(2.55) hold if H'_{m_i} is calculated using Eq. (2.19). Continuing to differentiate gives

$$\frac{\partial H'_{m_i}}{\partial \bar{F}_{10.7}} = \frac{\partial H_{m_i}}{\partial \bar{F}_{10.7}} + w_H \left(\frac{\partial H_{m_{i+1}}}{\partial \bar{F}_{10.7}} - \frac{\partial H_{m_i}}{\partial \bar{F}_{10.7}} \right). \quad (2.58)$$

$h_i + \alpha \leq h \leq h_{i+1} - \alpha$ In an unweighted region, the differentiation procedure used in the lower region of an altitude layer may be reused with $w_H = 1$. That is,

$$\frac{\partial H'_{m_i}}{\partial \bar{F}_{10.7}} = \frac{\partial H_{m_i}}{\partial \bar{F}_{10.7}}. \quad (2.59)$$

2.6 Summary of Harris-Priester Models

Table 2.1 summarizes the properties of the three HP model implementations described in this chapter.

Table 2.1: Summary of Harris-Priester model variants.

	CHP	LIHP	NNHP
$\bar{F}_{10.7}$ dependence	Cubic	Linear interpolation	Nearest-neighbor interpolation
Constants per altitude layer	8	20	20
Continuous derivatives	Yes	No	No
Reference	This dissertation	JAT code [75]	Dowd & Tapley [57]

2.7 Comparisons with Other Atmospheric Models

The density values and execution time of the new CHP are compared against NNHP and LIHP, as well as several popular high-fidelity models: NRLMSISE-2000, DTM 2013 [36], Jacchia-Bowman 2008 (JB08) [30], and, for some data sets, HASDM [181].

2.7.1 GRACE Accelerometer-Derived Density Estimates

Comparing the densities produced by various atmospheric models is complicated by the absence of a universally recognized “truth” model. One method is to use, as reference values, densities estimated based on the tra-

jectory of a specific SO rather than calculated using an atmospheric model. Accelerometer-derived density estimates have been computed for the CHAMP mission SO and both GRACE mission SOs [55, 54]. While these reference values are subject to errors of their own (albeit generally smaller than generic atmosphere models), they provide an outside data set to which results of the models may be compared. Fig. 2.7 shows the level of agreement between each of the atmospheric models and accelerometer-derived densities for the GRACE A SO for more than 6,000 samples taken hourly during the year 2003. (Densities are in kg/m^3 .) Following Doornbos [55], Fig. 2.7 also summarizes the geometric mean (μ^*) and standard deviation (σ^*) of the ratios of the GRACE A accelerometer-derived densities to the model densities. (For the geometric standard deviation, a value of unity represents perfect agreement between data sets. Note this definition is contrasted with the common additive standard deviation, for which a value of zero represents perfect agreement between data sets.)

As expected, the variance in model agreement is clearly larger for all three HP variants⁴ than for the higher-fidelity models. However, for the intended application of CHP – preliminary studies that value speed and smoothness more highly than accuracy – the results are nonetheless promising. CHP’s cubic functional dependency on $\bar{F}_{10.7}$ is shown to be at least as capable as LIHP and NNHP at responding to variations in $\bar{F}_{10.7}$ over the values at which the models are evaluated. Additionally, many of the most anomalous density val-

⁴Table values for NNHP and LIHP were taken from the Java Astrodynamics Toolkit implementation [75].

ues produced by the HP models are caused by extreme fluctuations in either geomagnetic or solar activity. As previously mentioned, modifying Eq. (2.43) to take into account, for example, the daily $F_{10.7}$ solar flux or the geomagnetic index A_p is straightforward and could involve the addition of a small number of additional coefficients at each reference altitude. On the other hand, upgrading NNHP or LIHP in a similar manner requires the generation, storage, and runtime indexing of an ever-growing array of tables so that successive interpolations may be performed. Finally, the Heaviside-smoothed and third-order polynomial-smoothed versions of CHP produce very nearly identical results. This finding supports the use of the more efficient polynomial-smoothed CHP unless arbitrarily high-order continuity is desired.

2.7.2 Parametric Sweep

The GRACE density data provide a representative sampling of parameters like latitude, longitude, local apparent solar time, day of year, and geomagnetic activity. However, the SO's orbit limits the data to a near-constant altitude and orbital inclination – two input parameters for the HP model. Additionally, the year 2003 is characterized by low-to-moderate $\bar{F}_{10.7}$ values. In order to gain a broader perspective of the capabilities of CHP, further evaluations are performed using a parametric sweep of input conditions over the time span 1 Jan 2000 to 3 Jan 2011. At each day within the time span, latitude, longitude, and local apparent solar time are varied over their respective domains of possible values at each of a range of altitudes (10,800 density evaluations per altitude). The SO's orbital inclination is set equal to the geo-

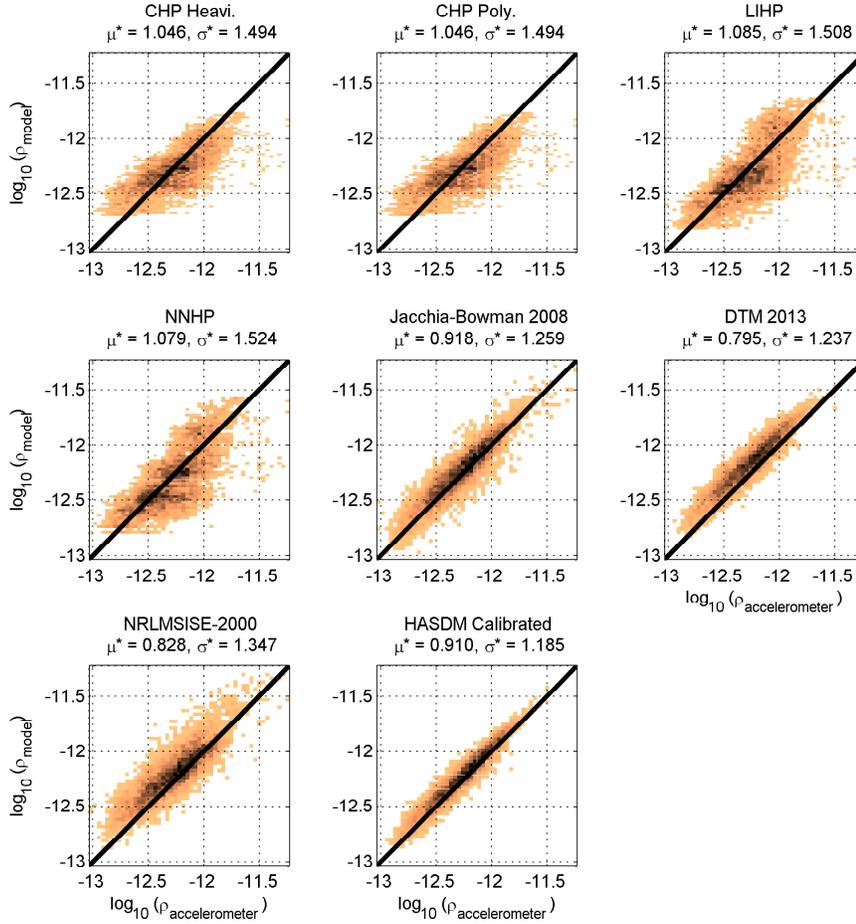


Figure 2.7: Bivariate histograms showing level of agreement between model densities and accelerometer-derived densities for GRACE A SO. Values on heavy line are in perfect agreement.

centric latitude. An altitude increment of 96.1 km is used to sample different regions within altitude layers (e.g., near h_i and near h_{i+1}). The results for CHP, LIHP, and NNHP are compared to the results for the high-fidelity JB08 model in Figs. 2.8, 2.9, and 2.10, respectively. (Densities are in kg/m^3 .) The geometric statistics of the ratios of the JB08 densities to the HP densities are

Table 2.2: Geometric statistics of ratios of JB08 densities to HP densities. (Closeness to unity is a measure of accuracy for all columns.)

Altitude (km)	CHP		LIHP		NNHP	
	μ^*	σ^*	μ^*	σ^*	μ^*	σ^*
204.0	0.996	1.176	1.031	1.191	1.041	1.191
300.1	0.900	1.321	0.981	1.379	1.007	1.380
396.2	0.860	1.493	0.838	1.616	0.872	1.615
492.3	0.810	1.667	0.686	1.849	0.724	1.847
588.4	0.798	1.704	0.603	1.917	0.642	1.915
684.5	0.859	1.618	0.618	1.821	0.659	1.821

given in Table 2.2. Only the results for the Heaviside-smoothed version of CHP are shown because results of polynomial-smoothed CHP are identical to the precision of Fig. 2.8 and Table 2.2.

It is observed that the relative agreement between all three HP models and JB08 increases as altitude decreases – an important trend given the inverse exponential relationship between altitude and the drag perturbation acceleration. Additionally, when comparing CHP, LIHP, and NNHP, the three models are seen to produce densities of similar fidelity relative to JB08 at low altitudes. Meanwhile, at higher altitudes, CHP shows significantly better agreement with JB08 than LIHP or NNHP. This outcome lends further credence to the use of the proposed method of calculating the HP scale heights as continuously differentiable functions of environmental parameters.

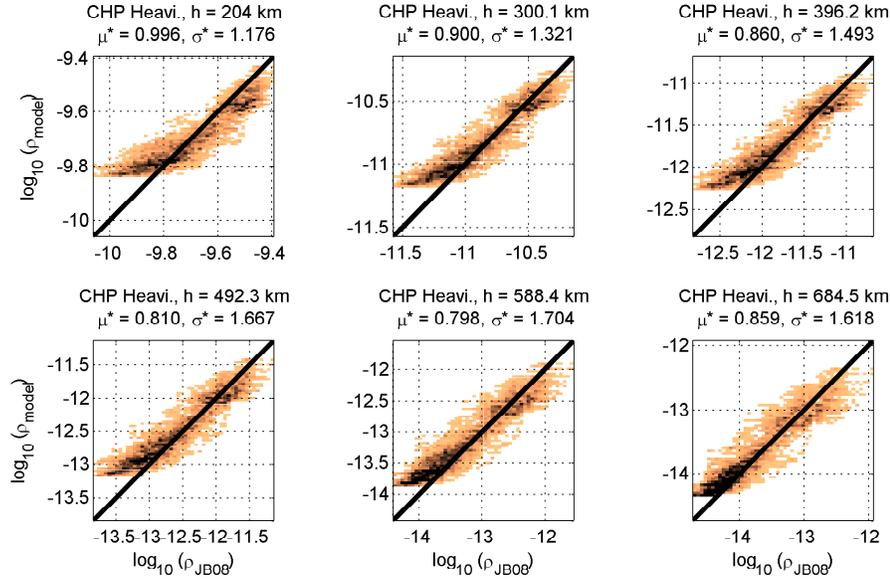


Figure 2.8: Bivariate histograms showing level of agreement between CHP densities and Jacchia-Bowman 2008 densities for parametric scan of input values. Values on heavy line are in perfect agreement.

2.7.3 Trajectory Propagation

An example of the effect of density model choice on the propagation of a specific trajectory is shown in Figs. 2.11 and 2.12. The trajectory is a 300-km altitude ($a_0 = 6678$ km), near-circular ($e_0 = 0.001$), near-polar ($i_0 = 86^\circ$) orbit propagated for four days using an 8th-order Runge-Kutta-Dormand-Prince integration scheme with a fixed 30-second step size. The SO is modeled with $C_D = 2.52$ and area-to-mass ratio 0.0054842 m²/kg. The force model consists only of the two-body gravitational force and the drag perturbation. The low altitude and simple force model are chosen to isolate the effects of atmospheric density on the SO's trajectory. As in the parametric sweep, only the results for Heaviside-smoothed CHP are shown.

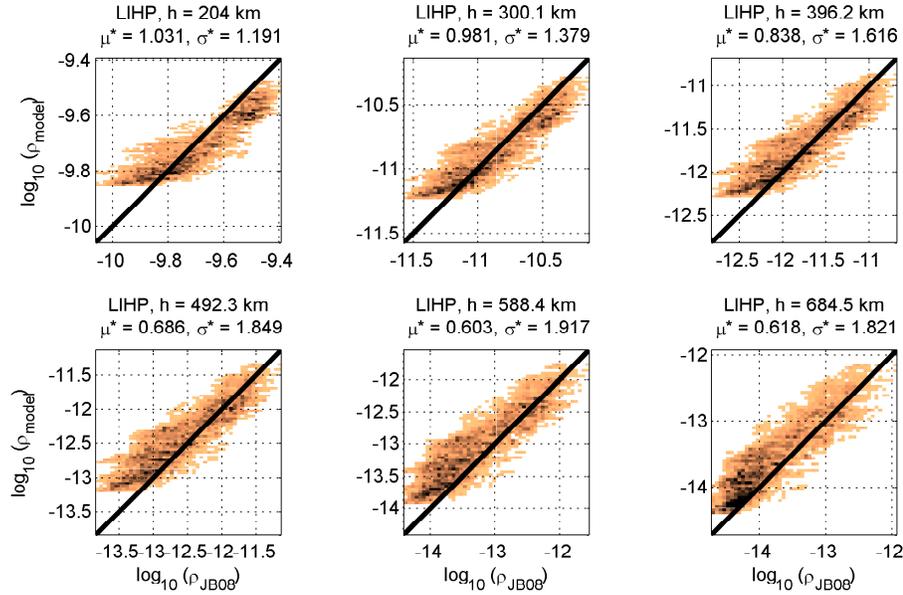


Figure 2.9: Bivariate histograms showing level of agreement between LIHP densities and Jacchia-Bowman 2008 densities for parametric scan of input values. Values on heavy line are in perfect agreement.

It is emphasized that the differences presented in Figs. 2.11 and 2.12 should not be taken at face value.⁵ These results are merely intended to convey three observations: First, CHP, LIHP, and NNHP produce results of comparable accuracy; second, all three HP alternatives produce qualitatively reasonable results relative to higher-fidelity models over relatively short propagation spans; and, finally, the potentially large variability created by switching between higher-fidelity models supports the use of fast, low-fidelity models like HP in preliminary studies.

⁵See Vallado and Finkleman [191] for a survey of the outcomes of propagating with various density models – and with a single model varying methods used to calculate input parameters.

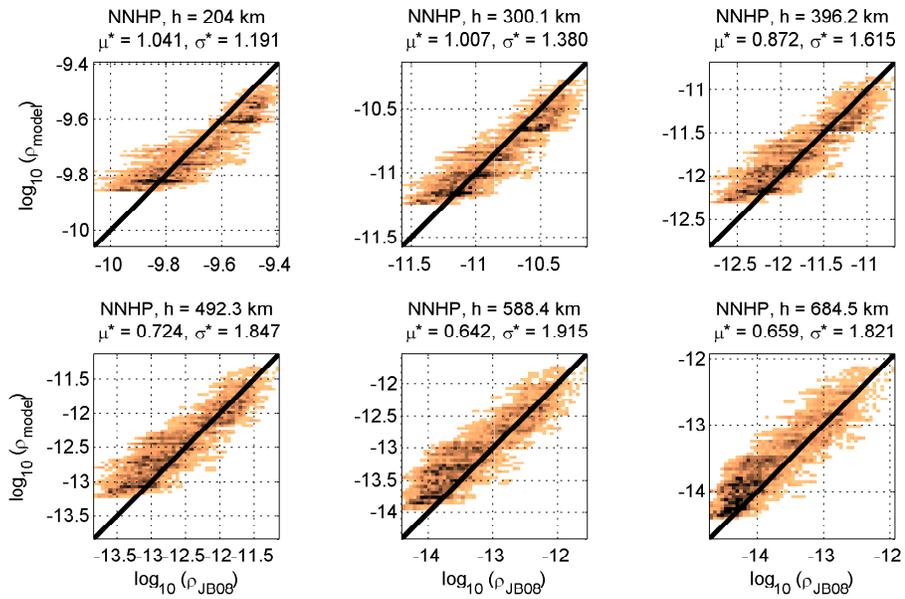


Figure 2.10: Bivariate histograms showing level of agreement between NNHP densities and Jacchia-Bowman 2008 densities for parametric scan of input values. Values on heavy line are in perfect agreement.

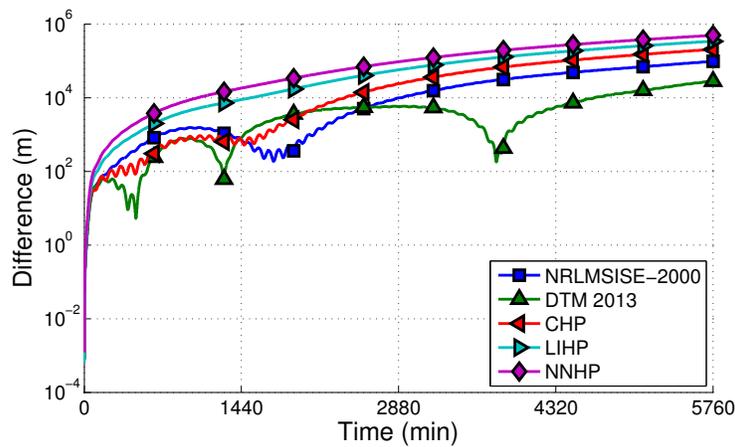


Figure 2.11: Magnitude of position difference between trajectory propagated using each of several density models and trajectory propagated using Jacchia-Bowman 2008 model.

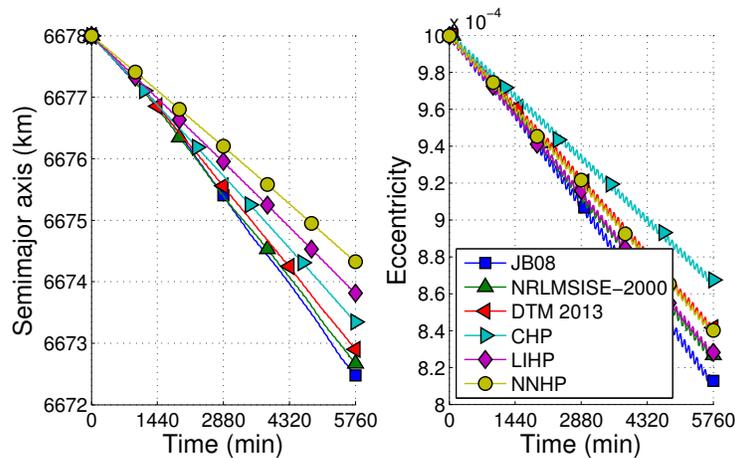


Figure 2.12: Orbit shape evolution due to atmospheric drag perturbation.

2.7.4 Computational Efficiency

The greatest value of the HP model lies in its ability to efficiently produce moderately accurate – and, in the case of the new CHP model, smooth and robust – densities. Relative CPU timing results for each of the density models tested in this dissertation are given in Table 2.3. All models are implemented in Fortran and compiled using the Intel Visual Fortran Compiler XE 14.0.0.103 for the 32-bit Windows platform with optimization flag -O3. An exception is the DTM 2013 model, which is available only as a static library and was therefore not compiled by the authors. Additionally, it is noted that Jacchia-Bowman 2008, NNHP, LIHP, and CHP are written using double-precision arithmetic, while NRLMSISE-2000 and DTM 2013 use single-precision arithmetic.⁶ Timing results are obtained by repeatedly evaluating each model at a fixed set of input conditions (493 km altitude, moderate values

⁶The use of single-precision arithmetic can reduce the solution precision obtainable from an optimization routine [82].

for environmental parameters) and averaging over the number of trials.⁷ The polynomial-weighted CHP provides third-order continuity of the scale heights in altitude. In order to test the efficiency of the polynomial-smoothed CHP both when the altitude is near and far from an altitude layer boundary, the tests are also run at 499.9 km altitude, which triggers the calculation of the polynomial weighting functions.

For CHP, derivatives of density with respect to the SO ECI position and velocity vectors, time, and $\bar{F}_{10.7}$ are computed analytically. For all other models, derivatives of density with respect to the ECI state vector, time, and $\bar{F}_{10.7}$ are computed using a first-order finite-difference approximation. Thus, five additional density evaluations are required to obtain approximate derivative information for NRLMSISE-2000, DTM 2013, and Jacchia-Bowman 2008, while eight additional evaluations are required for LIHP and NNHP. (The HP model is dependent on SO velocity, while the other models are not.)

The overall simplicity of all the HP models leads to execution speeds more than three times faster than any of the higher-fidelity models tested. Further, unlike NNHP and LIHP, CHP facilitates efficient, analytical computation of continuous partial derivatives of density with respect to the SO state, time, and input environmental parameters – a useful feature for many estimation and optimization algorithms. If arbitrarily high-order continuity is desired, then Heaviside-weighted CHP may be used to calculate density and its partial derivatives. On the other hand, if third-order continuity is acceptable,

⁷10,000,000 for CHP, LIHP, and NNHP; 2,000,000 for NRLMSISE-2000 and DTM 2013; 1,000,000 for Jacchia-Bowman 2008.

Table 2.3: Normalized execution times for density models; one normalized time unit is 8.05×10^{-4} ms.

Model	Normalized CPU time	
	Density	Density and derivatives
Nearest-neighbor Harris-Priester (NNHP)	0.80	6.40
Linear-interpolation Harris-Priester (LIHP)	0.81	6.48
Cubic Harris-Priester (CHP): Heaviside continuity	1.00	1.27
CHP: polynomial continuity, $h \in [h_i - 0.5 \text{ km}, h_i + 0.5 \text{ km}]$	0.94	1.22
CHP: polynomial continuity, $h \notin [h_i - 0.5 \text{ km}, h_i + 0.5 \text{ km}]$	0.81	1.02
DTM 2013	3.99	19.95
NRLMSISE-2000	7.92	39.60
Jacchia-Bowman 2008	15.90	79.50

polynomial-weighted CHP provides very similar density and derivative values at a reduced computational cost. The efficiency gain is particularly large when the altitude is more than 0.5 km from a reference altitude. As altitude layers are at least 10 km thick in the present implementation, a vast majority of cases benefit from the larger speedup. Additionally, it is reemphasized that the functional dependencies of the new CHP model may be easily upgraded to take into account first-order effects due to other environmental parameters with only marginal speed sacrifices in the calculation of density and its first derivatives.

More efficient methods of calculating the derivatives of NNHP, LIHP, and the high-fidelity models may exist. For example, the derivatives of the

NNHP and LIHP models are similar to those of the CHP model – through, as described, basic implementations of HP derivatives may encounter discontinuities and singularities. Additionally, automatic differentiation has been used to calculate the sensitivities of the MSIS-86 model, which is similar to NRLMSISE-2000 [39]. When calculating derivatives with respect to four input parameters, the execution time was reported to increase by a factor of 4.3, as opposed to the factor of 5 increase necessitated by a first-order finite difference approximation. Nevertheless, efficiency gains of this magnitude significantly lag the capability of the CHP model to calculate derivatives with a compute time increase of less than 30 percent.

2.8 Conclusions

The modified HP atmospheric density model is upgraded to remove singularities and to ensure continuous derivatives with respect to the SO state and time. These modifications lead to improved robustness and utility, particularly when the model is used in conjunction with a trajectory estimation or optimization routine. Additionally, a functional dependence of the scale heights on environmental parameters is introduced. When used to model the effect of the 81-day centered solar flux index $\bar{F}_{10.7}$, similar accuracy and efficiency is obtained relative to a linear-interpolation-based HP model using fewer than half as many stored coefficients. Further, the new model may be upgraded in a straightforward manner to take into account first-order effects due to other parameters with only marginal speed decreases by adding a small number of coefficients at each altitude. Conversely, adapting an interpolation-

based strategy in this manner would require successive interpolations and the addition of arrays of tables of stored values.

The accuracy of the HP model is not intended to directly compete with more computationally expensive modern high-fidelity density models. Nevertheless, fast, approximate models are attractive over short time spans due to the relatively large uncertainties inherent in atmospheric modeling – particularly when projecting environmental conditions into the future for predictive trajectory propagation. The efficiency of the HP model combined with the improved robustness and accuracy of the newly upgraded version make the model suitable for preliminary mission design, trajectory estimation, optimization, and other SSA applications.

Chapter 3

Parallel Implicit Runge-Kutta Methods Applied to Coupled Orbit/Attitude Propagation¹

3.1 Introduction

The need for efficient algorithms for numerically solving the ODEs generated by SP and semianalytical methods is introduced in Section 1.2.2, while the formulation and potential benefits of IRK methods are discussed in Sections 1.2.2.1–1.2.2.5. In this chapter, concepts for propagating 3DOF SO trajectories using IRK methods are modified and extended to the propagation of coupled SO trajectory and attitude, the importance and challenge of which is described in Section 1.2.2.6. A variable-step-size Gauss-Legendre IRK (GLIRK) ODE solver is applied to 6DOF state prediction. Customizations are introduced that greatly improve the efficiency of the GLIRK solver without

¹Work from this chapter was presented as:

- **Noble Hatten** and Ryan P. Russell. Parallel implicit Runge-Kutta methods applied to coupled orbit/attitude propagation. Paper AAS 16-395. In *26th AAS/AIAA Space Flight Mechanics Meeting*, Napa, CA, February 2016.

A manuscript has been accepted for publication in a peer-reviewed journal:

- **Noble Hatten** and Ryan P. Russell. Parallel implicit Runge-Kutta methods applied to coupled orbit/attitude propagation. *J Astronaut Sci.* doi: 10.1007/s40295-016-0103-3. (accepted October 2016)

In each instance, Ryan P. Russell supervised the work.

sacrificing accuracy for certain classes of SOs and dynamic environments. Two example scenarios are given, and the applicability of the customizations – and the corresponding effects on propagator efficiency – are discussed. In addition, the effects of parallelizing the dynamics model evaluations of the GLIRK solver using a multicore CPU are examined, and accuracy and efficiency are compared to linear multistep and ERK solvers.

3.2 Fully Coupled Orbit and Attitude Propagation

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

$$\mathbf{x} = \begin{pmatrix} \mathbf{r} \\ \mathbf{v} \\ \bar{\mathbf{q}} \\ \boldsymbol{\omega} \end{pmatrix}_{13 \times 1}, \quad (3.1)$$

where \mathbf{r} and \mathbf{v} are the SO's ECI position and velocity vectors, respectively, $\bar{\mathbf{q}}$ is a quaternion representing the orientation of a reference frame fixed to the SO body with respect to the ECI frame, and $\boldsymbol{\omega}$ is the angular velocity vector of the body-fixed frame with respect to the ECI frame, expressed in the body-fixed frame.² While the attitude may be expressed in several ways, the quaternion and angular velocity combination is chosen due to the robustness and precision of this formulation [74]. The state equations are [142]

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

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}) = \begin{pmatrix} \mathbf{v} \\ \ddot{\mathbf{r}} \\ \dot{\bar{\mathbf{q}}} \\ \dot{\boldsymbol{\omega}} \end{pmatrix}, \quad (3.2)$$

where

$$\ddot{\mathbf{r}} = -\frac{\mu_*}{r^3} \mathbf{r} + \mathbf{a}_p \quad (3.3)$$

$$\dot{\bar{\mathbf{q}}} = \frac{1}{2} \boldsymbol{\Omega} \bar{\mathbf{q}} \quad (3.4)$$

$$\dot{\boldsymbol{\omega}} = \mathbf{J}_0^{-1} [\mathbf{T} - \boldsymbol{\omega} \times (\mathbf{J}_0 \boldsymbol{\omega})]. \quad (3.5)$$

In Eqs. (3.3)–(3.5), μ_* is the point-mass gravitational parameter of the Earth, r is the magnitude of \mathbf{r} , \mathbf{a}_p is the sum of all non-two-body accelerations acting on the SO in the ECI frame, \mathbf{J}_0 is the inertia tensor of the SO in the body-fixed frame (assumed constant), and \mathbf{T} is the sum of external torques acting on the SO in the body-fixed frame. (Note that the time derivative of $\boldsymbol{\omega}$, $\dot{\boldsymbol{\omega}}$, is expressed in the body-fixed frame.) $\boldsymbol{\Omega}$ is a matrix given by

$$\boldsymbol{\Omega} = \begin{pmatrix} 0 & \omega_3 & -\omega_2 & \omega_1 \\ -\omega_3 & 0 & \omega_1 & \omega_2 \\ \omega_2 & -\omega_1 & 0 & \omega_3 \\ -\omega_1 & -\omega_2 & -\omega_3 & 0 \end{pmatrix}, \quad (3.6)$$

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

The vectors \mathbf{a}_p and \mathbf{T} are sums of individual perturbing accelerations and torques, respectively. The elements of the sums are problem-specific and

depend primarily upon the dynamical regime (e.g., LEO vs. GEO) and the desired model fidelity. For the purposes of the test cases presented in this chapter, the perturbing accelerations considered are the non-two-body component of the geopotential, two-body gravitational accelerations caused by the Sun and Moon, aerodynamic acceleration (i.e., drag), and SRP acceleration. Meanwhile, \mathbf{T} consists of the two-body approximation of the gravity-gradient torque, aerodynamic torque, and SRP torque. In the current model, the gravitational accelerations depend only on the 3DOF state of the SO, and the gravity-gradient torque depends on the 6DOF state and \mathbf{J}_0 . On the other hand, the aerodynamic and SRP accelerations and torques act on the SO external body surfaces, and are functions of the SO geometry in addition to the 6DOF state. The external SO surfaces are modeled as a set of single-sided flat panels, each of which may be given unique physical properties. The position, orientation, and surface area of each panel is fixed in the body-fixed reference frame. In this model, the SRP acceleration (in km/s^2) on a single panel i is given by [199, 65]

$$\mathbf{a}_{SRP,i} = -\frac{F_{sun}AU^2}{mcD^2}H_1H_{2,i}A_i \cos \phi_i \cdot \left\{ (1 - C_{s,i}) \mathbf{u}_{sun} + 2 \left[\frac{1}{3}C_{d,i} + C_{s,i} \cos \phi_i \right] \mathbf{n}_i \right\}, \quad (3.7)$$

where F_{sun} is the total solar flux over all wavelengths (W/m^2), AU is one astronomical unit (km), m is the mass of the SO (kg), c is the speed of light (km/s), D is the distance from the SO to the Sun (km), H_1 is an Earth

shadowing function³, $H_{2,i}$ is the fraction of panel i facing the Sun, A_i is the area of panel i (km²), $C_{s,i}$ is the specular reflectivity coefficient for panel i , $C_{d,i}$ is the diffusive reflectivity coefficient for panel i , \mathbf{u}_{sun} is a unit vector pointing from the SO to the Sun, \mathbf{n}_i is a unit vector in the outward normal direction of panel i , and $\cos \phi_i = \mathbf{u}_{sun} \cdot \mathbf{n}_i$. The total SRP acceleration experienced by the SO is the sum of the $\mathbf{a}_{SRP,i}$. The torque due to SRP on panel i is then calculated in the body-fixed frame by

$$\mathbf{T}_{SRP,i} = m [\mathbf{d}_{cm,i} \times (\mathbf{R}^{I \rightarrow B} \mathbf{a}_{SRP,i})], \quad (3.8)$$

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

The aerodynamic acceleration and torque are also calculated for each panel individually following the method of Doornbos [55]. In the interest of brevity, the full details of the method are not presented here; at the most basic level, the method calculates the vector aerodynamic coefficient $\mathbf{C}_{a,i}$ for each panel i based on atmospheric properties. Then, the aerodynamic acceleration on panel i is given by

³A cylindrical Earth shadowing function is assumed in the examples presented in this chapter.

$$\mathbf{a}_{aero,i} = \frac{1}{2} \frac{A_{ref} \rho v_r^2}{m} \mathbf{C}_{a,i}, \quad (3.9)$$

where A_{ref} is a constant reference area⁴, ρ is the atmospheric density, and v_r is the magnitude of the relative velocity between the SO and the atmosphere. Aerodynamic torque is then found following Eq. (3.8).

3.3 Implicit Runge-Kutta ODE Solver

A GLIRK ODE solver is written to propagate the 6DOF equations of motion. A brief overview of IRK methods is given in Section 1.2.2.1. A more detailed description of the mathematics behind the GLIRK method in particular and IRK methods in general is given by Hairer et al. [88], while further descriptions within the context of 3DOF SO propagation are found in Jones [113], Aristoff et al. [6], and Aristoff and Poore [5].⁵ The referenced works also discuss the motivations for selecting GLIRK amongst the range of available short-time-step IRK methods for astrodynamics applications. For example, the GLIRK method is superconvergent: An implementation using s internal stages per step produces a solution accuracy of order $2s$, the highest possible for an RK method. Additionally, the GLIRK method is both symmetric and symplectic, and exhibits both A and B stability. The symplectic property is

⁴The area and orientation of each individual panel i is taken into account in the calculation of $\mathbf{C}_{a,i}$.

⁵The defining arrays of the GLIRK method are calculable for arbitrary s , and may be obtained to high precision using, for example, the Mathematica function `NDSolve[ImplicitRungeKuttaGaussCoefficients [201]`.

particularly advantageous for long-time-span propagations of Hamiltonian systems. Errors in the Hamiltonian remain bounded over time, which generally leads to slower state error growth and better preservation of qualitative system behavior compared to non-symplectic integration schemes [114, 89, 40]. Symplecticity is implied by the fact that the GLIRK method preserves quadratic first integrals of the state [25], which is a useful property in its own right when, for example, representing the attitude state using a quaternion and/or when propagating a system in which kinetic energy is conserved.

The difficulties associated with using a long-time-step implicit method like MCPI or BLC-IRK for 6DOF state prediction are enumerated in Section 1.2.2.5. In particular, long-time-step methods are generally implemented with a fixed step size, which may be problematic because of the multiple driving frequencies of the combined rotational and translational dynamics. On the other hand, a propagator with a mechanism for adapting the step size to meet a user-defined LTE tolerance avoids the difficulty of determining an appropriate fixed step size. Variable-step-size mechanisms exist for GLIRK, and such methods are the focus of this study [6, 113, 127].

In the current work, fixed-point iteration is used to solve the algebraic system of equations that arises at each propagation step (Eqs. (1.3)–(1.6)) because the method is parallelizable (unlike the Gauss-Seidel method) and avoids the computation of the Jacobian (unlike the Newton-Raphson method). It is noted that a Newton or quasi-Newton method (as opposed to a fixed-point method) must be used to retain the A stability property of a GLIRK solver and guarantee convergence of the the Runge-Kutta stage equations for large

step sizes [5]. In practice, however, convergence of the fixed-point iterations may be monitored numerically at each step, and the step size may be reduced if divergence is predicted. For reasonably high-precision 3DOF or 6DOF astrodynamics applications, step size reductions of this type are rare enough that any associated cost is significantly less than that of the Jacobian calculations required by Newton-based methods [8].

3.3.1 Variable-Fidelity Dynamics Models

Though the necessity of iterative methods to solve Eq. (1.4) is generally a disadvantage in terms of computational efficiency for IRK methods, problem-specific customizations of the iterative solver may greatly reduce the computational burden of the solution process. If applicable, variable-fidelity dynamics models (Section 1.2.2.3) allow for inexpensive, approximate computations at some iterations without compromising the accuracy of the converged solution of Eq. (1.4). This approach has been used successfully for 3DOF SO state propagation [33, 141, 113, 6, 5]. Here, a similar approach is taken for fully coupled 6DOF propagation, as described in Algorithm 1. The forces and torques considered in the low-fidelity model are specified in the discussion of each example simulation.

When using Algorithm 1, the high-fidelity dynamics model is evaluated at only two iterations per propagation step, regardless of the number of total iterations required for convergence. The three iteration convergence tolerances ϵ_1 , ϵ_2 , and ϵ_3 are user-defined parameters that may be used to control the precision of the solution. Though similar in spirit, this approach dif-

Algorithm 1 Iterative solution of IRK step (Eq. (1.4)) using variable-fidelity dynamics model.

Set convergence tolerances: $\epsilon_1 > \epsilon_2 > \epsilon_3 > 0$

$\mathbf{x}_i \leftarrow \mathbf{x}_{i,guess}, i = 1, \dots, s$ ▷ Set initial guess for \mathbf{x}_i

$j \leftarrow 1$ ▷ Start with first iteration tolerance

evaluation fidelity \leftarrow low ▷ Start with low-fidelity dynamics model

$\Delta \mathbf{f}_i \leftarrow \mathbf{0}, i = 1, \dots, s$

while $j \leq 3$ **do** ▷ Loop through iteration tolerances

if evaluation fidelity == low **then**

$\mathbf{f}_i \leftarrow \mathbf{f}_{low}(t_i, \mathbf{x}_i), i = 1, \dots, s$ ▷ Low-fidelity function evaluation

$\mathbf{f}_i \leftarrow \mathbf{f}_i + \Delta \mathbf{f}_i, i = 1, \dots, s$ ▷ Update full function approximation

else

$\mathbf{f}_i \leftarrow \mathbf{f}_{high}(t_i, \mathbf{x}_i), i = 1, \dots, s$ ▷ High-fidelity function evaluation

$\Delta \mathbf{f}_i \leftarrow \mathbf{f}_{low}(t_i, \mathbf{x}_i), i = 1, \dots, s$ ▷ Low-fidelity function evaluation

$\Delta \mathbf{f}_i \leftarrow \mathbf{f}_i - \Delta \mathbf{f}_i, i = 1, \dots, s$ ▷ High-fidelity contribution to model

end if

 Update $\mathbf{x}_i, i = 1, \dots, s$, using fixed-point iteration update equations

$\delta_x \leftarrow$ norm of change over all $\mathbf{x}_i, i = 1, \dots, s$

if $\delta_x < \epsilon_j$ **then** ▷ Converged to within current iteration tolerance

$j \leftarrow j + 1$ ▷ Proceed to next iteration tolerance

 evaluation fidelity \leftarrow high ▷ Perform high-fidelity function

 evaluation at next iteration

else

 evaluation fidelity \leftarrow low ▷ Perform low-fidelity function evaluation

 evaluation at next iteration

end if

end while

fers from previously published GLIRK variable-fidelity models used for 3DOF propagation. For example, both Jones [113] and Aristoff and Poore [5] utilize one or more low-fidelity dynamics model iterations before switching to the high-fidelity model at all subsequent iterations. Alternatively, as is done in Algorithm 1, Bradley et al. [33] use two high-fidelity iterations per step. However, in the Bradley algorithm, the two high-fidelity iterations are separated by a fixed number of low-fidelity iterations, and no subsequent low-fidelity iterations are performed following the second high-fidelity iteration. In the context of MCPI, Macomber [141] implements an approach similar to Algorithm 1, but the validity of the approach is only investigated for a 3DOF dynamics model consisting of the geopotential alone. Numerical examples presented in this chapter show that this variable-fidelity dynamics model approach is valid for the 6DOF propagation of an SO in the presence of realistic forces and torques – provided that the low-fidelity dynamics model is selected appropriately.

All tests presented in this chapter use $\epsilon_1 = 10^{-5}$, $\epsilon_2 = 10^{-12}$, and $\epsilon_3 = 10^{-15}$. Divergence of the iterative solver is handled by placing a maximum value on the number of iterations: If the solver does not converge to within ϵ_3 in 50 iterations, the step size is halved, and the step is rerun.

Initial guesses for the \mathbf{x}_i are provided using approximate analytical propagation methods: An elliptical orbit is assumed for the translational propagation [20], and constant-angular-velocity rotation is assumed for the rotational propagation [195]. Analytical solutions to more physically representative models, such as torque-free rotation [195], exist, but are more computationally demanding and are not found to improve overall efficiency when

combined with the variable-fidelity dynamics model strategy for the scenarios presented in this chapter. At early iterations, the fixed-point solver converges toward an approximate solution because only the inexpensive, low-fidelity dynamics model is evaluated. Therefore, the improvement in the initial guess between the constant-angular-velocity solution and the torque-free-rotation solution is unnecessary. If the full, high-fidelity dynamics model is evaluated at every iteration, a more accurate initial guess becomes more valuable.

3.3.2 Variable-Step-Size Propagation

Variable-step-size ODE solvers adaptively vary the step size of the independent variable by comparing the estimated LTE of the method to a user-prescribed tolerance. In this way, variable-step-size methods attempt to maximize efficiency by taking the minimum number of propagation steps needed to meet the user’s accuracy requirements. ERK methods commonly use embedded methods to estimate the LTE; that is, each propagation step is performed by two methods, each of unique order, that share the same \mathbf{A} and \mathbf{c} arrays but are defined by different \mathbf{b} arrays. This strategy allows for the efficient calculation of state estimates of different order. The LTE is then estimated based on the difference between the state estimates produced by the two methods and knowledge of the order of the LTE of each method [166].

For the GLIRK ODE solver, an embedded method of nearly the same order as the propagation method is difficult to achieve because an s -stage GLIRK method gives a solution of order $2s$, the highest order achievable by an RK method [6]. For this reason, previous authors have utilized alternative

methods for creating a variable-step-size GLIRK ODE solver for 3DOF SO propagation. For example, Jones [113] employs a method of van der Houwen and Sommeijer [193] that uses the differences between the solutions produced at consecutive iterations of the iterative solution process to approximate the LTE. Aristoff [4], on the other hand, uses a second, non-embedded IRK propagation of order near $2s$ (such as a Radau method) to produce a high-order LTE estimate. However, to the author’s knowledge, the precise methodology has not been specified in a published work [4, 5, 6, 7, 8]. The computational cost of the second IRK propagation is reduced by using the collocation polynomial produced by the original GLIRK propagation to obtain an initial guess (of order s) to Eq. (1.4) [88].

In the current work, a lower-order (order s), nearly embedded⁶ solution is used to estimate the LTE [127, 87]. A solution $\hat{\mathbf{x}}$ of order s is given by

$$\hat{\mathbf{x}}_{m+1} = \mathbf{x}_m + h \left(\gamma_0 \mathbf{f}(t_m, \mathbf{x}_m) + \sum_{j=1}^s \hat{b}_j \mathbf{k}_j \right). \quad (3.10)$$

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

⁶The embedded solution requires a single additional high-fidelity dynamics model evaluation at the initial time and state of the step.

$$\mathbf{V} = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ c_1 & c_2 & \cdots & c_s \\ \vdots & \vdots & \ddots & \vdots \\ c_1^{s-1} & c_2^{s-1} & \cdots & c_s^{s-1} \end{pmatrix}, \quad \mathbf{u} = \begin{pmatrix} 1 - \gamma_0 \\ \frac{1}{2} \\ \vdots \\ \frac{1}{j} \\ \vdots \\ \frac{1}{s} \end{pmatrix}. \quad (3.11)$$

The scalar γ_0 is a user-selected parameter. Following Kouya [127], the method implemented here uses $\gamma_0 = 1/8$. Once \mathbf{x}_{m+1} and $\hat{\mathbf{x}}_{m+1}$ are known, the procedure for adapting the step size proceeds identically to an ERK method:⁷

$$h_{new} = h_{old} \cdot \min \left[\Delta h_{max}, c \left(\frac{1}{\delta} \right)^p \right] \quad (3.12)$$

$$\delta = \sqrt{\frac{1}{n} \sum_{i=1}^n \left(\frac{\mathbf{x}_{i,m+1} - \hat{\mathbf{x}}_{i,m+1}}{\tau_i} \right)^2} \quad (3.13)$$

$$\tau_i = \text{atol} + \text{rtol} \cdot \max(|\mathbf{x}_{i,m}|, 1) \quad (3.14)$$

$$p = \frac{1}{s+1} \quad (3.15)$$

$$c = 0.8 \quad (3.16)$$

$$\Delta h_{max} = 2, \quad (3.17)$$

where subscript i indicates element i of the n -dimensional state vector, atol and rtol are user-defined absolute and relative tolerance parameters, respectively, and $0 < c < 1$ is set to increase the likelihood that the subsequent step is

⁷If both an adaptive step size and long-term, qualitatively correct system behavior are desirable, a more sophisticated step-size selection algorithm is required to rigorously ensure the symplectic property of the GLIRK method [89, 40].

accepted. In the event $\delta = 0$ to double precision, the step size is increased by a factor of 10. Note that if $h_{new} > h_{old}$, then the current step is accepted and h_{new} is used as the step size for the next time step, while, if $h_{new} < h_{old}$, the current step is rejected and retried with h_{new} as the step size.

The low computational cost of the LTE estimation procedure is offset by the low order of the nearly embedded solution (order s) relative to the GLIRK solution (order $2s$). In other words, for step size adaptation purposes, the GLIRK method is treated as an order- $(s + 1)$ method. The result is that the LTE estimate δ is likely to be larger than the actual LTE of the GLIRK method. Thus, h_{new} is likely to be underestimated, and some steps that should be acceptable may be rejected. One method for addressing large differences between the orders of the propagated and embedded methods is Jay’s “internal tolerance” scheme [112]. However, use of the internal tolerance is not found to reliably improve performance in the current application; an illustration of this finding is given in Appendix A. The test cases presented in this chapter use the more conservative step size selection scheme given by Eqs. (3.10)–(3.17).

A disadvantage of variable-step-size methods is the costly re-propagation of a step if the estimate of the LTE is found to be larger than the user-specified tolerance. In order to proactively seek out steps that are likely to fail, the current GLIRK implementation estimates the LTE following the first evaluation of the high-fidelity dynamics model (following convergence of the fixed-point iteration to within ϵ_1). If the estimate of the LTE is greater than the tolerance (i.e., $\delta > \tau$), the fixed-point iteration is aborted, and the step is rerun using the step size computed using Eq. (3.12). Thus, the second evaluation of the

high-fidelity dynamics model at each stage – which would have been wasted due to the rejection of the step – is avoided. While it is possible for false detections to cause unnecessary recomputations, empirical evidence suggests that false detections are rare relative to true detections because of the proximity of the partially converged solution to the fully converged solution.

An estimate for the step size to be used at the first propagation step is calculated using the initial value of the state and its time derivative, as described in Algorithm 2 [180].

Algorithm 2 Calculation of estimate for initial step size h_0 .

```

Set minimum and maximum step sizes  $h_{min}$  and  $h_{max}$ 
 $h_0 \leftarrow h_{max}$ 
 $\alpha \leftarrow \frac{atol}{\max(rtol, \epsilon_m)}$   $\triangleright \epsilon_m$  is machine epsilon
 $\beta \leftarrow \frac{\maxval(|\mathbf{f}(t_0, \mathbf{x}_0)|)}{\max(\maxval(|\mathbf{x}_0|), \alpha, \epsilon_m)}$ 
 $h_g \leftarrow \frac{\beta}{c \cdot rtol^p}$   $\triangleright p$  is defined in Eq. (3.15);  $c$  is defined in Eq. (3.16)
if  $h_g == 0$  then
     $h_0 \leftarrow h_{min}$ 
else if  $h_g \cdot h_0 > 1$  then
     $h_0 \leftarrow 1/h_g$ 
end if
 $h_0 \leftarrow \max(h_0, h_{min})$   $\triangleright$  Ensure  $h_0 \geq h_{min}$ 

```

3.3.3 Parallelization

Parallel computation within a single propagation is a trait of IRK methods not shared by ERK methods or linear multistep methods (Section 1.2.2.4).⁸ Such parallelization has the potential to yield strong benefits for both 3DOF

⁸Using any ODE solver, the propagation of STMs may be parallelized within a single propagation step over the dimension of the state.

and 6DOF SO propagation because the high-fidelity dynamics model evaluations required to compute the \mathbf{f}_j of RK methods generally dominate propagation runtime. While the possibility of parallelization of 3DOF GLIRK methods has been suggested previously [113, 6, 5, 33], analysis has been limited to the assumption of linear speedups – i.e., the use of s parallel threads reduces total runtime by a factor of s . However, this assumption provides only a best-case approximation of efficiency gains. Linear speedups are not typical for realistic implementations because the CPU time required to evaluate a typical dynamics model may not be large enough to hide the overhead of parallelization.

In the present work, the effect of parallelization on the efficiency of the GLIRK method applied to 6DOF SO propagation is studied using the OpenMP library with up to 23 threads [158]. Dynamics models of varying complexity are implemented to examine the consequences of parallelization overhead. At each iteration of the solution process of Eq. (1.4) at which high-fidelity dynamics model evaluations are required, the $s \mathbf{f}_j$ are evaluated in parallel using an `!$omp parallel do` loop. Static scheduling is used due to the very similar workload of each loop iteration. Parallelization of the evaluations of the low-fidelity dynamics model and the initial guesses for the \mathbf{x}_i is also possible. However, because of the short runtime of these routines, serial evaluation is sometimes preferred because of the overhead associated with parallelization. Further discussion is given for the two example scenarios described in the Section 3.4.

3.4 Results

The accuracy and efficiency of the GLIRK propagator are compared to a standard implementation of the explicit Runge-Kutta-Dormand-Prince (RKDP) 8(7) propagator and to the public-domain linear multistep (Adams-Moulton) propagator LSODE [166, 169].⁹ Results are presented for two scenarios: a rectangular prism SO in LEO and a HAMR SO in GEO. The 6DOF equations of motion are propagated with each ODE solver for three orbital periods unless otherwise specified. The quaternion unit norm constraint is enforced by renormalizing $\bar{\mathbf{q}}$ after each propagation step for all propagators. Accuracy is assessed via a “truth” model calculated using a quadruple-precision implementation of RKDP8(7) with a relative LTE tolerance of 10^{-25} . For GLIRK parallelization, the number of available threads is set to $(s + 1)$ so that each of the dynamics model evaluations – including the extra evaluation used in the estimation of the LTE – is provided a single thread. All code is written in Fortran and compiled using the Intel Visual Fortran Compiler XE 14.0.0.103 (64-bit) using the `-O2` optimization flag. All computations are performed on a 64-bit Windows 7 Enterprise workstation with two 12-core Intel Xeon E5-2680 v3 processors (clock speed 2.50 GHz) and 64 GB of RAM. Hyperthreading is disabled.

In the presentation of results, discrete data points (indicated by shapes in figures) are connected by lines. This convention is *not* intended to convey

⁹All input options for LSODE are set to default values except for the relative and absolute tolerance parameters: The absolute tolerance is set uniformly to machine epsilon, and the relative tolerance is varied.

Table 3.1: Initial state for SO in LEO.

(a) Translational state in classical orbital elements.	(b) Rotational state.
a 6,745.592 km	q_1 0
e 0.01	q_2 0
i 7.81 deg	q_3 0
Ω 100.21 deg	q_4 1
ω 152.83 deg	ω_1 0.573 deg/s
θ 0 deg	ω_2 0.573 deg/s
	ω_3 0.573 deg/s

information regarding trends between data points, but merely to aid the reader in differentiating between data sets.

3.4.1 Tumbling Object in Low Earth Orbit

The first test scenario is a tumbling rectangular prism SO in LEO. The initial state is given in Table 3.1, and the SO’s physical characteristics are described in Tables 3.2 and 3.3. The SO’s panels are given non-uniform reflectance properties to induce an SRP torque. Tables 3.4 and 3.5 give the high-fidelity and low-fidelity dynamics models, respectively. In all results given for this scenario, parallelization is limited to the high-fidelity dynamics model; parallelization of the low-fidelity model and initial guess generation is found not to improve efficiency for this scenario.

The differing frequencies of the translational and rotational motions are demonstrated in the state evolutions shown in Figure 3.1; for this scenario, the rotational dynamics evolve much more rapidly than the translational dynam-

Table 3.2: SO basic physical characteristics.

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

Table 3.3: SO panel reflectance properties.

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

Table 3.4: High-fidelity dynamics model for SO in LEO.

Source	Force Model	Torque Model
Gravity	70 \times 70 geopotential (spherical harmonics or interpolated [12]); Sun and Moon point mass (cubic spline ephemerides [10])	Earth gravity gradient
SRP	Direct (panel model)	Direct (panel model)
Drag	Doornbos model (panel-based) [55]; modified Harris-Priester atmospheric density [140]	Doornbos model (panel-based)

Table 3.5: Low-fidelity dynamics model for SO in LEO.

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

ics. The performance of the GLIRK propagator as a function of the number of

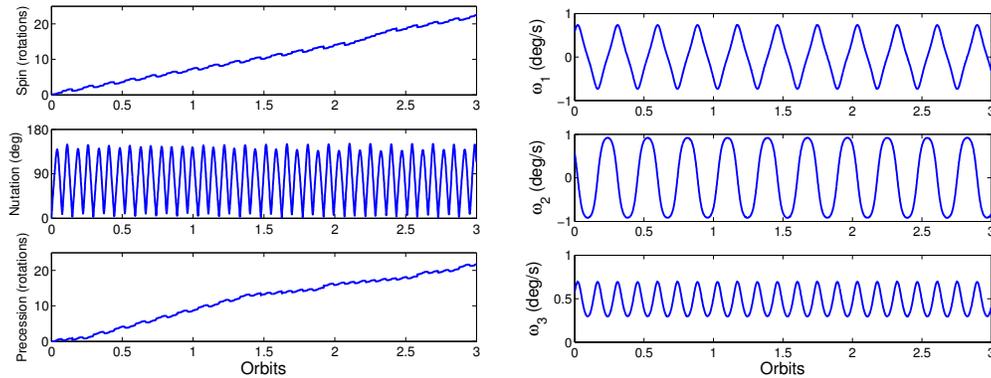
stages s is shown in Figure 3.2 for a three-orbit propagation.¹⁰ Figure 3.2a confirms that the use of the variable-fidelity models does not degrade propagation accuracy.¹¹ Figure 3.2b depicts the performance gains that may be realized by using variable-fidelity dynamics models in the fixed-point iteration. The optimal number of stages for a serial implementation for this scenario is seen to be in the range of 8–12, which is found to be typical for this application.¹² For a parallel implementation, the optimal number of stages depends on the dynamics model: As the CPU time required by the dynamics model increases, larger relative efficiency gains are available from parallelization using a high number of stages and threads. For this scenario, Figure 3.2b shows that efficiency gains using more than eight stages and nine threads are minimal for either the high-fidelity-only or variable-fidelity implementations.

Figure 3.2c shows the efficiency gains achieved by parallelizing the dynamics model evaluations at each propagation step. When the high-fidelity dynamics model is evaluated at every iteration, these evaluations consume a greater percentage of propagator runtime than when the variable-fidelity model is used. Thus, greater relative efficiency gains are observed for the high-fidelity-only propagator. As expected, the efficiency gains from parallelization lag the number of parallel threads employed. For the variable-fidelity propa-

¹⁰Each subfigure displays results corresponding to a single identical value of relative LTE tolerance, $\text{rtol} = 10^{-15}$. Unless otherwise specified, the number of threads made available for parallelization is equal to $(s + 1)$.

¹¹Note logarithmic scale of vertical axis.

¹²A similar number of stages per step has been found to perform well for 3DOF SO state propagation [8, 5]. However, other applications benefit from using a different number of stages per step [168].



(a) Body orientation described by 3–1–3 Euler angle sequence.

(b) Angular velocity in body-fixed reference frame.

Figure 3.1: Rotational state evolution for three-orbit LEO scenario.

gator, speedups between $3\times$ and $4\times$ compared to the serial performance are achieved for $s \geq 8$, while, for the high-fidelity-only propagator, $s \geq 16$ delivers relative speedups between $7\times$ and $8\times$. However, despite the superior *relative* efficiency gains of the high-fidelity-only propagator, greater *absolute* performance is still achieved via the variable-fidelity propagator for this scenario (Figure 3.2b).

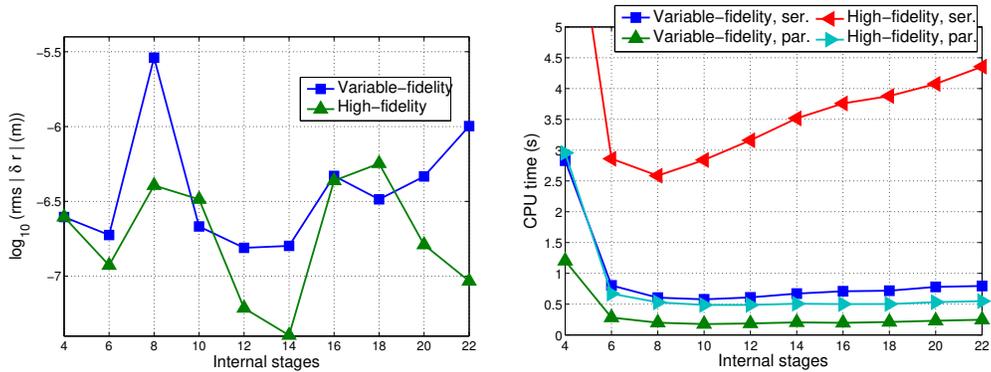
Figure 3.2d displays the relative efficiency improvements for the eight-stage GLIRK propagator as a function of the number of parallel threads available. Efficiency jumps correspond to threadcount increases that decrease the number of loop iterations required. For example, the high-fidelity-only propagator sees a large efficiency gain between seven and eight threads because $s = 8$. As expected, the availability of more than $(s + 1)$ threads provides no further efficiency gains.

The impact of the variable-fidelity dynamics model strategy is further underscored in Figure 3.3, which displays the relative CPU time requirements of each of the constituents of the high-fidelity dynamics model.¹³ The low-fidelity dynamics model evaluation time is also shown for comparison. (The low-fidelity model time is not included in the summation used to compute the total time.) The benefits of eliminating evaluations of the high-fidelity geopotential are clear – even though the runtime of the interpolated 70×70 field is approximately equivalent to that of a 15×15 field if the more common spherical harmonics formulation is used.

Figure 3.4 compares root-mean-square (RMS) accuracy and efficiency of the eight-stage GLIRK propagator to RKDP8(7) and LSODE for relative LTE tolerances ranging from 10^{-5} to 10^{-15} .¹⁴ Numeric subscripts in the legend represent the number of parallel threads used in computation. Meanwhile, “var.-fi.” indicates use of the variable-fidelity dynamics model, while “hi.-fi.” indicates use of the high-fidelity dynamics model only. Among serial implementations, the linear multistep solver LSODE is more efficient than either the IRK or ERK solvers, particularly for stringent LTE tolerances. This result supports the substantial heritage of linear multistep solvers for serial astrodynamics applications [148]. For example, the Gauss-Jackson method is used by United States Air Force Space Command to propagate the SO catalog using special perturbation techniques [23, 110, 152, 114]. In addition, the DIVA propagator used by the Jet Propulsion Laboratory is a linear multistep inte-

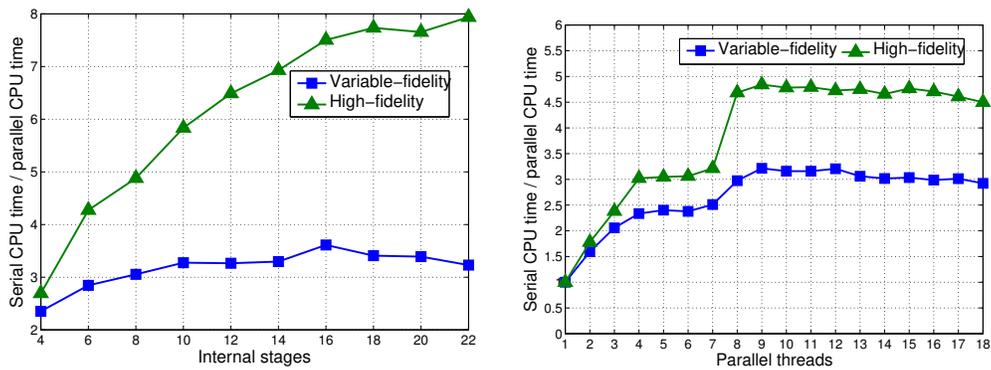
¹³Note logarithmic scale of vertical axis.

¹⁴Note logarithmic scale of vertical axes.



(a) Accuracy of position propagation.

(b) Absolute CPU time.



(c) Efficiency gain from parallelization.

(d) Efficiency gain as function of number of parallel threads for $s = 8$.

Figure 3.2: Performance of GLIRK propagator as a function of number of stages s using both variable-fidelity and high-fidelity-only dynamics models.

propagator [129], and LSODE is the default propagator in the Copernicus mission design software [155].

The poor performance of the serially implemented, high-fidelity-only GLIRK propagator demonstrates why the method has not been traditionally popular for astrodynamics applications. On the other hand, the variable-fidelity GLIRK propagator gives comparable or superior performance com-

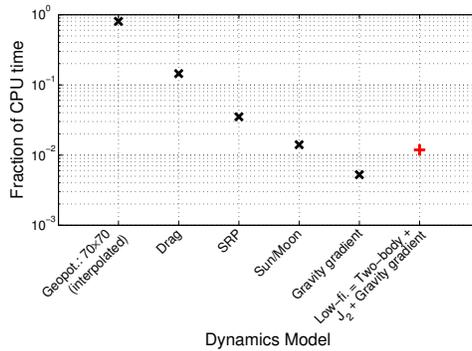
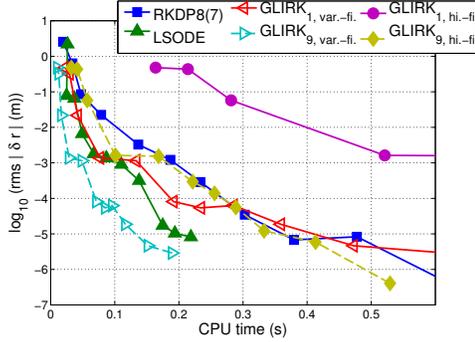


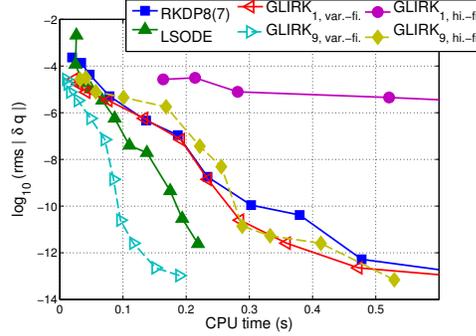
Figure 3.3: Fraction of total high-fidelity dynamics model CPU time spent evaluating each model constituent for three-orbit LEO scenario; alternative low-fidelity dynamics model time shown for comparison.

pared to the RKDP8(7) propagator, even when implemented serially. When parallelized, variable-fidelity GLIRK outperforms LSODE, showing improved efficiency for a given accuracy. For a 30-orbit propagation, the parallelized variable-fidelity GLIRK propagator achieves sub-centimeter-level accuracy in 31 percent less CPU time than LSODE and 61 percent less CPU time than RKDP8(7) (Figure 3.4c).

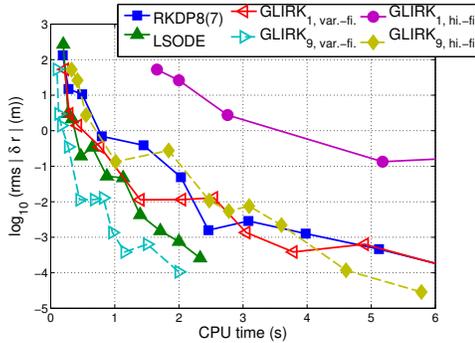
As the effects of parallelization and the variable-fidelity dynamics model increase with increasing dynamics model complexity, even further relative speed gains are possible. For example, higher-fidelity atmospheric density models can be an order of magnitude more computationally expensive than the modified Harris-Priester model implemented in this example (Chapter 2). Many more than six panels may be required to adequately represent an SO body. Additionally, as discussed in Section 1.2.3 and Chapter 4, the propagation of STMs via the variational equations is often important – and significantly increases the computational burden of the dynamics model. This



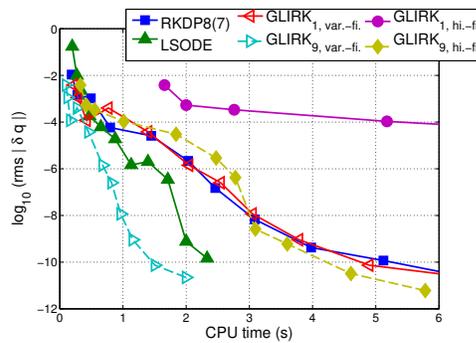
(a) Position accuracy, three orbits.



(b) Quaternion accuracy, three orbits.



(c) Position accuracy, 30 orbits.

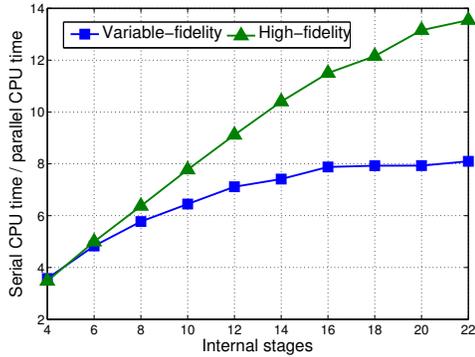


(d) Quaternion accuracy, 30 orbits.

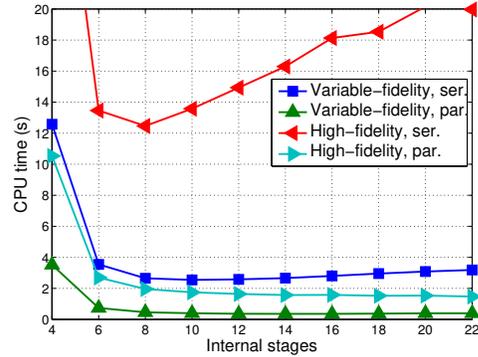
Figure 3.4: Accuracy vs. CPU time for propagation of tumbling SO in LEO; relative LTE tolerance of propagators is varied from 10^{-5} – 10^{-15} .

phenomenon is demonstrated in Figure 3.5, in which the 70×70 geopotential is calculated using spherical harmonics instead of the significantly faster interpolated gravity model.¹⁵ The peak relative effect of parallelization increases significantly for both the high-fidelity-only and variable-fidelity GLIRK propagators (Figure 3.2c vs. Figure 3.5a), and the parallelized variable-fidelity GLIRK is now four times faster than LSODE for sub-centimeter-level posi-

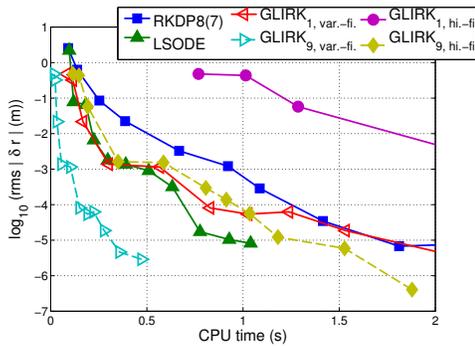
¹⁵Note logarithmic scale of vertical axes in Figures 3.5c and 3.5d.



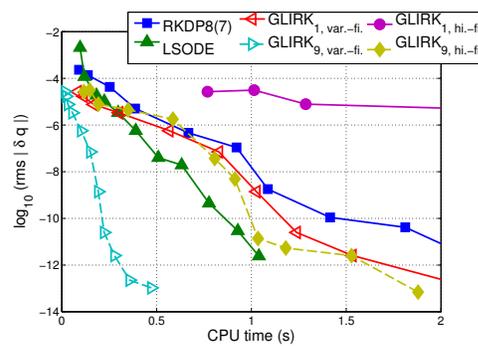
(a) Efficiency gain from parallelization.



(b) Absolute CPU time.



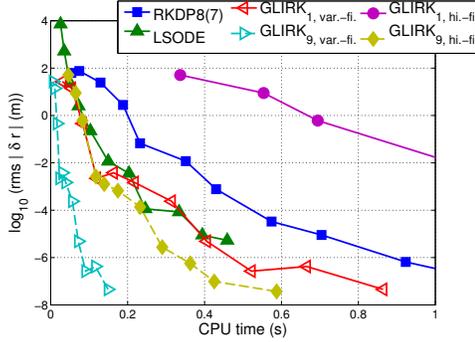
(c) Position accuracy, three orbits; relative LTE tolerance of propagators is varied from 10^{-5} – 10^{-15} .



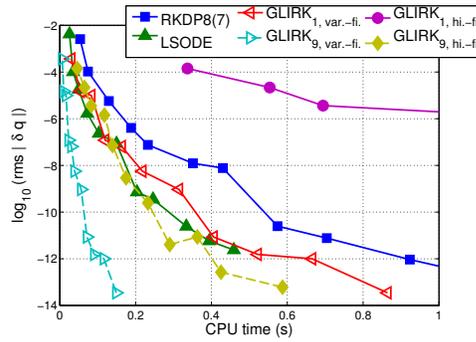
(d) Quaternion accuracy, three orbits; relative LTE tolerance of propagators is varied from 10^{-5} – 10^{-15} .

Figure 3.5: Effect of more computationally expensive dynamics function. Geopotential is calculated using spherical harmonics.

tion error after three orbital periods (Figure 3.5c). Comparing Figure 3.5b to Figure 3.2b, larger relative efficiency increases from parallelization allow for runtime improvements at larger s . However, gains achieved beyond 8–12 stages are minimal due to the inefficient serial performance of the high-order methods.



(a) Position accuracy, three orbits.



(b) Quaternion accuracy, three orbits.

Figure 3.6: Accuracy vs. CPU time for propagation of non-tumbling SO in LEO; relative LTE tolerance of propagators is varied from $10^{-5} - 10^{-15}$.

The spherical-harmonics-based LEO scenario is re-simulated assuming that the previously described SO is non-tumbling ($\boldsymbol{\omega}_0 = [0, 0, 0]^T$). Figures 3.5c and 3.5d are duplicated for this case in Figures 3.6a and 3.6b, respectively.¹⁶ For the non-tumbling case, the absolute CPU time required by all ODE solvers decreases because the attitude state changes more slowly, allowing for larger propagation step sizes. The performances of the ODE solvers relative to one another are similar to the tumbling case (Figure 3.5). In a serial environment, LSODE and the variable-fidelity GLIRK propagator display generally similar efficiency profiles, while the RKDP8(7) propagator requires more CPU time to achieve a given accuracy. When parallelization is used, the variable-fidelity GLIRK propagator once again provides the greatest efficiency among the options considered, here achieving an approximately $6\times$ speedup over LSODE for approximately sub-centimeter-level position error.

¹⁶Note logarithmic scale of vertical axes.

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

A second test scenario consists of a HAMR two-sided flat plate in GEO. An offset between the center of mass and center of pressure of the plate is assumed in order to induce an SRP torque. For the primary analysis of this example, the SO is not tumbling, and instead begins with zero angular velocity. The complete initial state and SO physical characteristics are given in Tables 3.6 and 3.7, respectively. In all results given for this scenario, both the low-fidelity and high-fidelity dynamics models are parallelized because of the relative expense of the low-fidelity model, as discussed below.

The dynamics models differ from the LEO example, as shown in Tables 3.8 and 3.9.¹⁷ Importantly, the low-fidelity torque model is identical to its high-fidelity counterpart. The reason is that the low-fidelity model is only useful if the approximation of the high-fidelity model is adequate: Referencing Eq. (3.5), for a HAMR SO with little or no angular velocity, the torque contribution to $\dot{\omega}$ dwarfs the inertia contribution. Further, in the current example scenario, SRP is the dominant source of torque, and therefore the largest driver of $\dot{\omega}$. Thus, failure to take into account SRP in the evaluation of the low-fidelity model results in inaccurate propagation. This contrasts with the 3DOF problem, in which the two-body gravitational acceleration term is orders of magnitude greater than all disturbing accelerations¹⁸, even for a HAMR SO in GEO.

¹⁷The spherical harmonics formulation of the geopotential is used because it is more efficient than the interpolation model for the low degree and order used for GEO propagation.

¹⁸Excepting possibly the effect of aerodynamic acceleration on an SO in an extremely low orbit.

Table 3.6: Initial state for HAMR SO in GEO.

(a) Translational state in classical orbital elements.	(b) Rotational state.
a 42,164 km	q_1 0
e 0.001	q_2 0
i 1 deg	q_3 0
Ω 145.92 deg	q_4 1
ω 266.13 deg	ω_1 0 deg/s
θ 0 deg	ω_2 0 deg/s
	ω_3 0 deg/s

Table 3.7: SO physical characteristics.

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

Table 3.8: High-fidelity dynamics model for SO in GEO.

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

Table 3.9: Low-fidelity dynamics model for SO in GEO.

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

The evolution of the rotational states is shown in Figure 3.7. Despite the small center-of-mass-center-of-pressure offset, the HAMR nature of the SO results in an SRP torque that significantly affects the attitude. Additionally, the small rotational angular momentum of the SO results in slower, less regular rotational motion than was observed in the tumbling LEO scenario, exemplified by the decreased number of complete rotations of the spin and precession angles.

The computational expense of the low-fidelity dynamics model relative to the high-fidelity model degrades the efficiency of the variable-fidelity GLIRK propagator, as shown in Figure 3.8 for a three-orbit propagation.¹⁹ This result is confirmed by Figure 3.9, as well.²⁰ The effect of parallelization is also diminished in this scenario because the high-fidelity dynamics model is significantly less expensive to evaluate than that of the LEO scenario due to the truncated geopotential and absence of aerodynamic forces. Thus, for the GEO scenario, a smaller fraction of the runtime is spent evaluating the dy-

¹⁹Unless otherwise specified, the number of threads made available for parallelization is equal to $(s + 1)$. Note logarithmic scale of vertical axis in Figure 3.8a.

²⁰Note logarithmic scale of vertical axis. As in Figure 3.3, the low-fidelity model time is not included in the summation used to compute the total time

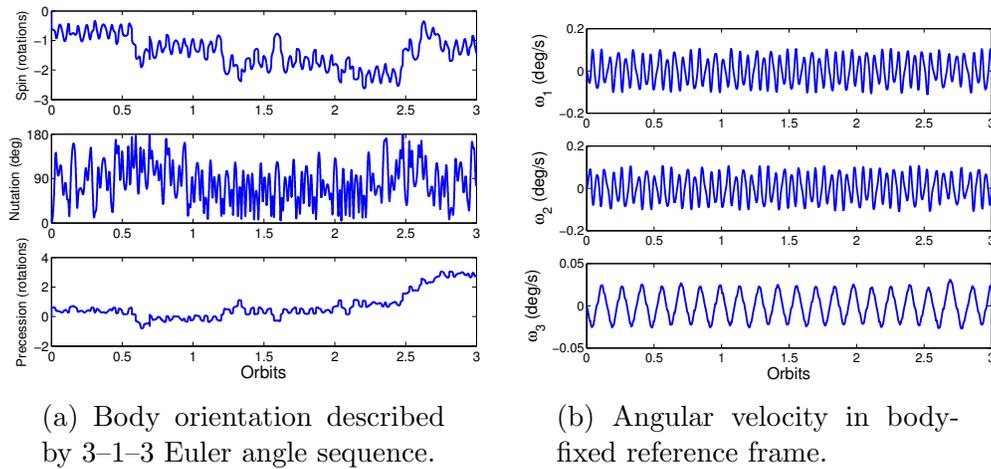
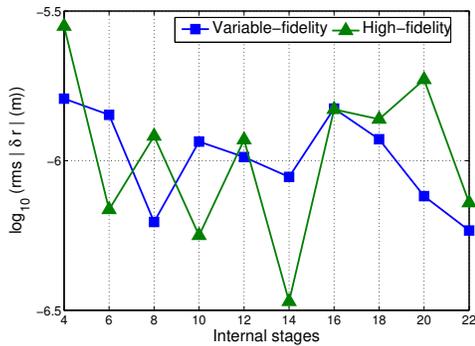


Figure 3.7: Rotational state evolution for three-orbit GEO scenario.

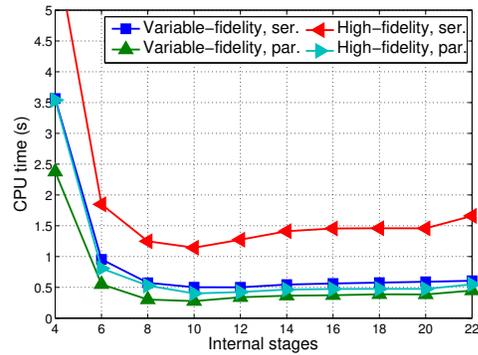
namics model, and the detrimental impact of parallelization overhead is more pronounced.

The consequences of the SO physical characteristics and dynamics model on the performance of the eight-stage GLIRK propagator relative to RKDP8(7) and LSODE are shown in Figure 3.10. Numeric subscripts in the legend represent the number of parallel threads used in computation. Meanwhile, “var.-fi.” indicates use of the variable-fidelity dynamics model, while “hi.-fi.” indicates use of the high-fidelity dynamics model only.²¹ Unlike in the LEO scenario, the variable-fidelity GLIRK propagator is only competitive with the other propagators for very high-accuracy applications, even when implemented in parallel. Also unlike the LEO scenario, RKDP8(7) is nearly as efficient as LSODE, due to two factors: (1) the ratio of the number of steps taken by RKDP8(7) to the number of steps taken by LSODE decreases from the LEO scenario to the

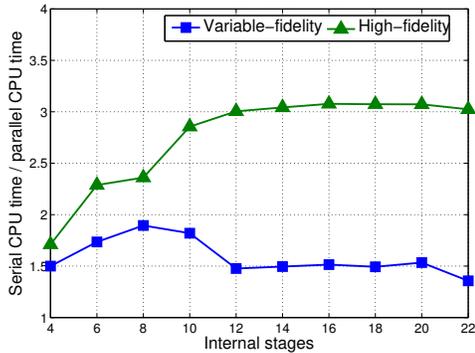
²¹Note logarithmic scale of vertical axes.



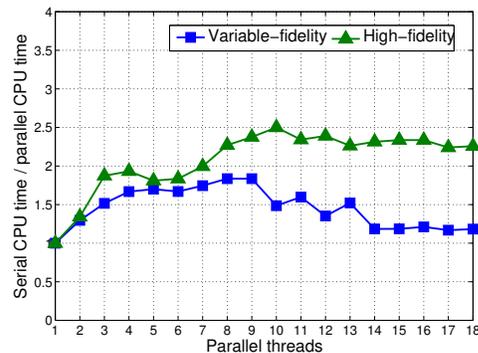
(a) Accuracy of position propagation.



(b) Absolute CPU time.



(c) Efficiency gain from parallelization.



(d) Efficiency gain as function of number of parallel threads for $s = 8$.

Figure 3.8: Performance of GLIRK propagator as a function of number of stages s using both variable-fidelity and high-fidelity-only dynamics models.

GEO scenario; and (2) a lower percentage of propagation steps are rejected for the GEO scenario, especially for RKDP8(7) at tight tolerances. Rejected steps are more costly for RK methods than for linear multistep methods when dynamics function evaluations dominate overall runtime because RK methods generally require significantly more dynamical function evaluations per step (in this case, 13 for RKDP8(7) vs. 1–2 for LSODE). The smaller percentage

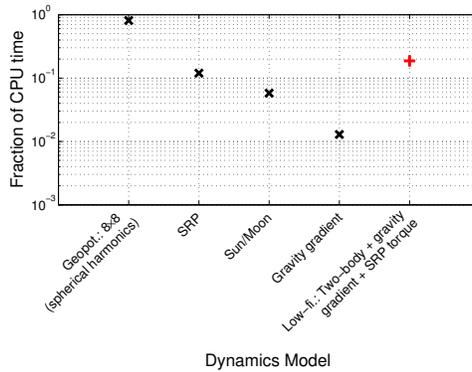


Figure 3.9: Fraction of total high-fidelity dynamics model CPU time spent evaluating each model constituent for three-orbit GEO scenario; alternative low-fidelity dynamics model time shown for comparison.

of rejected steps is likely due to the more uniformly changing dynamic environment of the slowly rotating GEO scenario. For example, there is no torque due to aerodynamic drag, and the torques caused by the gravity gradient and SRP change more slowly than in the LEO environment due to the slower rotation rate and increased orbital period of the SO. Thus, a step size h_{i+1} is more likely to differ significantly from the step size for the previous step h_i in the LEO propagation than in the GEO propagation. When a large difference between consecutive step sizes occurs, the predicted value for h_{i+1} may be inaccurate, resulting in either (1) a failed step (if the predicted h_{i+1} is too large) or (2) an inefficient step (if the predicted h_{i+1} is too small).

The scenario is also simulated assuming an initially tumbling plate, with $\boldsymbol{\omega}_0 = [0.573, \quad 0.573, \quad 0.573]^T$ deg/s; results are shown in Figure 3.11.²² As in the LEO scenario, propagation of a tumbling body requires more absolute CPU

²²Note logarithmic scale of vertical axes.

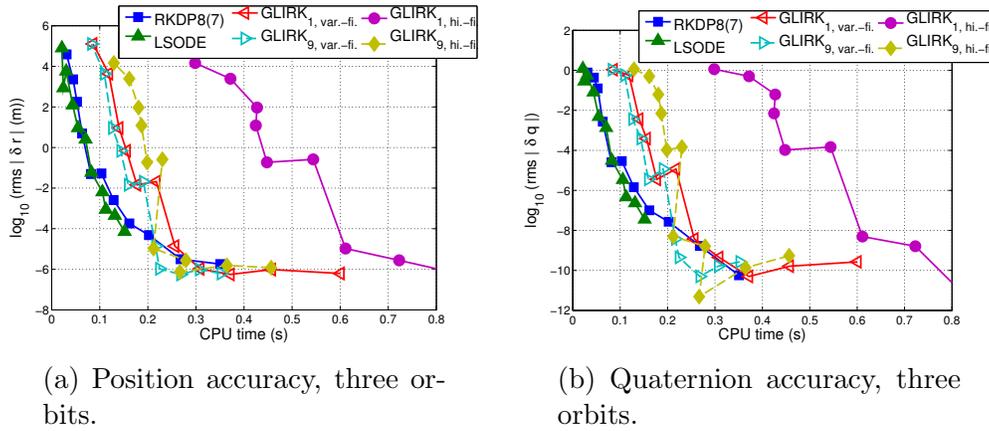


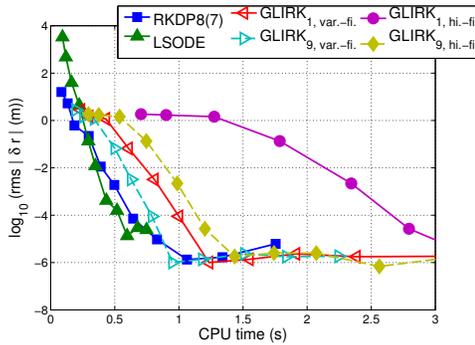
Figure 3.10: Accuracy vs. CPU time for propagation of HAMR SO in GEO; relative LTE tolerance of propagators is varied from $10^{-5} - 10^{-15}$.

time than an initially non-rotating body due to the higher frequencies of the rotational motion. The performance of the GLIRK propagator is improved in the tumbling scenario (compared to the non-tumbling GEO scenario) at loose LTE tolerances, though the efficiency still does not significantly exceed that of either LSODE or RKDP8(7) for a given RMS error. At more stringent tolerances, the relative performances of the propagators are similar to those observed in the non-tumbling case.

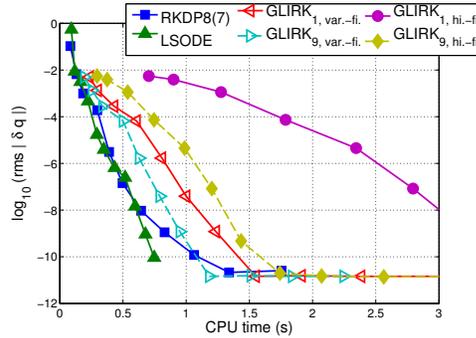
For any propagator, increasing the LTE tolerance beyond a limiting value²³ results in erratic behavior in an accuracy vs. CPU time plot.²⁴ This behavior can produce ranges of tolerance values in which the change in tolerance (and the resulting change in dynamics function evaluations) does not

²³This value is problem- and integrator-dependent.

²⁴Erratic behavior for an implicit method can also be caused by frequent divergence of the iterative procedure used to solve the RK update equations, but divergence does not occur over the range of tolerances displayed in the figures in this paper.



(a) Position accuracy, three orbits.



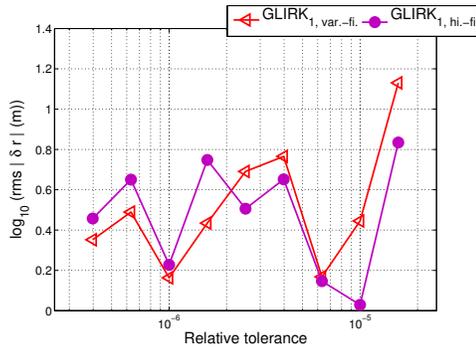
(b) Quaternion accuracy, three orbits.

Figure 3.11: Accuracy vs. CPU time for propagation of tumbling HAMR SO in GEO; relative LTE tolerance of propagators is varied from $10^{-5} - 10^{-15}$.

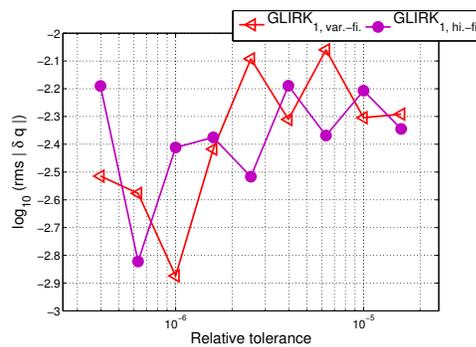
reliably correlate with a change in the accuracy of the propagation [188]. For this particular scenario, such a region exists for the GLIRK propagator for relative tolerances greater than approximately $10^{-6.2}$. As a result, the RMS state errors do not vary predictably at loose tolerances for GLIRK in Figure 3.11, and errors observed for a tolerance of 10^{-5} are similar to those observed for a tolerance of 10^{-7} – even though the former case requires fewer dynamics function evaluations. Figure 3.12 details this result by directly displaying RMS errors as functions of relative LTE tolerance for serial GLIRK propagations only, with the relative tolerance varied from $10^{-4.8} - 10^{-6.4}$.²⁵

Just as for the LEO scenario, the relative performance of the GLIRK propagator would improve if the dynamics were more strenuous: For example, SRP may be calculated using bidirectional reflectance distribution functions (BRDFs), SO self-shadowing effects may be relevant, and Earth-albedo and

²⁵Note logarithmic scale of horizontal and vertical axes.



(a) Position accuracy, three orbits.



(b) Quaternion accuracy, three orbits.

Figure 3.12: Accuracy vs. relative LTE tolerance for propagation of tumbling HAMR SO in GEO; relative LTE tolerance of propagators is varied from $10^{-4.8} - 10^{-6.4}$.

Earth-infrared radiation pressure and thermal radiation pressure may also be taken into account. The addition of any of these factors increases the performance gains available from parallelization. Also, because of the dominance of SRP torque on HAMR SOs in GEO, Earth and thermal radiation pressure torque may likely be ignored in the low-fidelity dynamics model²⁶, further improving efficiency.

3.5 Conclusions

A variable-step-size GLIRK ODE solver is applied to the fully coupled 6DOF propagation of an SO in Earth orbit. LTE is estimated using an inexpensive, nearly embedded lower-order method. The majority of dynamics model evaluations are replaced with fast, low-fidelity alternatives, decreasing

²⁶Unless the SO is in the Earth's shadow.

CPU runtime without compromising propagation accuracy. Additional unnecessary high-fidelity evaluations are eliminated by estimating the LTE prior to full convergence of the iterative solver at each propagation step. As these customizations suggest, GLIRK propagators – for both 3DOF and 6DOF astrodynamics applications – require careful tuning and problem-specific insight to maximize efficiency. However, the benefits of the GLIRK method are shown to be strong for certain common dynamical regimes, particularly given the growing ubiquity of parallel computing resources.

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

The use of variable-fidelity dynamics combined with the parallelizability of IRK methods means that the propagator is most efficient when the high-fidelity dynamics model is computationally intensive and a much less expensive, yet moderately accurate, low-fidelity model is available. For example, for a tumbling SO in LEO, the serially implemented, variable-fidelity, eight-stage GLIRK propagator is competitive with standard linear multistep and ERK propagators, though the linear multistep method is likely to be the most

efficient of the three tested options. However, when parallelized, the GLIRK method is shown to significantly outperform the serial alternatives.

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

Chapter 4

Decoupled Direct State Transition Matrix Calculation with Runge-Kutta Methods¹

4.1 Introduction

As discussed in Section 1.2.3, determining the evolution of a state governed by a system of ODEs is not sufficient for many applications; first-order, and sometimes higher-order, sensitivities of the state with respect to input parameters are often highly desirable or even essential. A vital element of sensitivity calculation is the computation of STMs, which makes the accurate and efficient calculation of STMs a necessity. In this chapter, the realization of this goal is examined using the DDM [58, 167] for the particular scenario in which the state is propagated using an RK ODE solver.

¹Work from this chapter has been accepted for presentation as:

- **Noble Hatten** and Ryan P. Russell. Decoupled direct state transition matrix calculation with Runge-Kutta methods. Paper 17-398. In *27th AAS/AIAA Space Flight Mechanics Meeting*, San Antonio, TX, February 2017. (accepted for presentation)

A manuscript based on the work in this chapter has been submitted for publication in a peer-reviewed journal:

- **Noble Hatten** and Ryan P. Russell. Decoupled direct state transition matrix calculation with Runge-Kutta methods. *SIAM J Sci Comput.* (under review; submitted November 2016)

In each instance, Ryan P. Russell supervised the work.

The DDM was originally introduced to calculate the first-order STM of a system propagated using the variable-order, variable-step Gear ODE solver [78]. As the name implies, the DDM decouples the solution of the STMs from the solution of the state at each propagation step: The state update is performed first, which results in an independent system of equations for the STM. More recently, in the context of IRK methods, a conceptually identical procedure was presented as the implicit function theorem (IFT) method [168].

Two implementations of the DDM are presented in the current work. One relies on the solution of a linear system to obtain the STM, while the other uses a necessarily iterative procedure. The linear-algebra-based alternative [168, 58, 207, 175, 176, 122] is derived for the computation of first- and second-order STMs and for use with either the single- or double-integrator form of the RK update equations. (The double-integrator form of the RK equations has been shown to improve efficiency vs. the single-integrator form, when applicable [6].) This method is valid for all RK solvers, and the implications for both the explicit and implicit classes are discussed. The iterative approach, previously described for first-order STM calculation in the context of MCPI [171], is also extended to the computation of second-order STMs for IRK methods.

In this chapter, both the linear-algebra and iterative versions of the DDM are shown to reduce the number of required Jacobian and Hessian evaluations compared to the CDM, which propagates the state and STMs together in an augmented state vector using the variational equations. Additionally, the linear-algebra-based DDM is shown to eliminate the need for initial guesses of

the STMs when using an IRK method (unlike the CDM and iterative DDM). The three direct STM calculation methods – the CDM and the two implementations of the DDM – are compared qualitatively and in a series of numerical examples of varying complexity for both ERK and IRK solvers. For the computationally intensive scenario of SO trajectory propagation, the impacts of variable-fidelity dynamics models and parallelization on IRK performance are analyzed. However, a detailed analysis of the relative merits of the indirect options (e.g., finite-difference methods) is beyond the scope of the current work.

4.2 Second-Order Runge-Kutta Methods

The RK equations for first-order systems of ODEs are presented in Eqs. (1.2)–(1.6). These equations may also be rewritten in a form specially applicable to a system of second-order ODEs, which is relevant to, for example, equations of motion based on Newton’s second law [6]. In this case, let $\mathbf{x} \triangleq [\mathbf{y}^T \quad \dot{\mathbf{y}}^T]^T$, where the dimensions of the vectors are $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{y}, \dot{\mathbf{y}} \in \mathbb{R}^{n'}$ (i.e., $n = 2n'$). Then,

$$\frac{d^2 \mathbf{y}}{dt^2} \triangleq \ddot{\mathbf{y}} = \mathbf{g}(t, \mathbf{y}, \dot{\mathbf{y}}), \quad \mathbf{y}(t_0) \triangleq \mathbf{y}_0, \quad \dot{\mathbf{y}}(t_0) \triangleq \dot{\mathbf{y}}_0 \quad (4.1)$$

$$\mathbf{y}_{m+1} = \mathbf{y}_m + h \dot{\mathbf{y}}_m + h^2 \sum_{j=1}^s \bar{b}_j \mathbf{g}_j \quad (4.2)$$

$$\dot{\mathbf{y}}_{m+1} = \dot{\mathbf{y}}_m + h \sum_{j=1}^s b_j \mathbf{g}_j \quad (4.3)$$

$$\mathbf{g}_i \triangleq \mathbf{g}(t_i, \mathbf{y}_i, \dot{\mathbf{y}}_i), \quad i = 1, \dots, s \quad (4.4)$$

$$t_i \triangleq t_m + hc_i \quad (4.5)$$

$$\mathbf{y}_i \triangleq \mathbf{y}_m + \bar{c}_i h \dot{\mathbf{y}}_m + h^2 \sum_{j=1}^s \bar{a}_{ij} \mathbf{g}_j \quad (4.6)$$

$$\dot{\mathbf{y}}_i \triangleq \dot{\mathbf{y}}_m + h \sum_{j=1}^s a_{ij} \mathbf{g}_j. \quad (4.7)$$

The \bar{a}_{ij} , \bar{b}_i , and \bar{c}_i are derived from the method-defining arrays:

$$\bar{a}_{ij} = \sum_{k=1}^s a_{ik} a_{kj}, \quad \bar{b}_i = \sum_{j=1}^s b_j a_{ji}, \quad \bar{c}_i = \sum_{j=1}^s a_{ij}. \quad (4.8)$$

When applicable, this formulation has been found to decrease the number of iterations required to obtain convergence at each step of an IRK method [6, 141].

4.3 STM Calculation with Runge-Kutta Methods

Both the CDM and DDM are applicable to all RK methods. However, the requirement of an iterative method to solve for the state update at each step of an IRK propagation allows for some variety of implementation of the DDM. In this section, the CDM is briefly described, and the two DDM alternatives are presented in more detail.

4.3.1 Coupled Direct Method

The evolution of STMs is governed by a set of ODEs known as the variational equations, which depend on the state. (The relevant equations are given in Appendix B.) As such, it is natural to combine the state and the

columns of the STMs into an augmented state vector, which may be propagated using any ODE solver. This approach is commonly used with both explicit and implicit solvers.

4.3.2 Decoupled Direct Method, Option A (DDMa): Multi-Iteration Approach

When using an IRK, the calculation of the STMs may be decoupled from the state update by using multiple iterative solution procedures at each step. In the first loop, the update equations are only used to solve for the true state (not including any STMs). Subsequently, a second loop is used to solve for the STMs. The advantage of this approach when used with an IRK is that the Jacobian \mathbf{J} (Eq. (B.8)) and Hessian \mathbf{H} (Eq. (B.9)) at each internal stage need only be evaluated once, after convergence of the state. This situation contrasts with that of the CDM, in which \mathbf{f} , \mathbf{J} , and \mathbf{H} are all evaluated at every iteration of the nonlinear solution process. Thus, the cost of each iteration used to solve for the STMs is significantly reduced. Additionally, convergence may be obtained in fewer total iterations when solving for STMs using pre-converged Jacobians and Hessians [171].

4.3.3 Decoupled Direct Method, Option B (DDMb): Linear-Algebra Approach

Another approach that uses a single evaluation of \mathbf{J} and \mathbf{H} per stage per step takes advantage of the linearity of the STM update equations. Like the multi-iteration method, the so-called linear-algebra method first iterates to solve for the state at each internal stage. However, instead of using the

variational equations to iteratively solve for the STMs, the fundamental RK equation is differentiated with respect to \mathbf{x}_m , which results in a system of *linear* algebraic equations for the first-order STM Φ^1 . Additionally, because the STM is the result of analytical differentiation of the RK update equation, over each step, the calculated STMs are the exact sensitivities of the approximated solution (subject to roundoff error). For a first-order system of ODEs, this method has been independently derived previously to calculate first-order sensitivities [168, 58, 175, 176, 122]. Here, the derivation of those equations is summarized, and the method is extended to include calculation of the second-order STM Φ^2 and modified to allow for use with the double-integrator form of the RK equations. A diagram that shows the relevant quantities for a single step of a two-stage RK method is given in Figure 4.1. The dashed box encloses step inputs, while the solid box encloses step outputs.

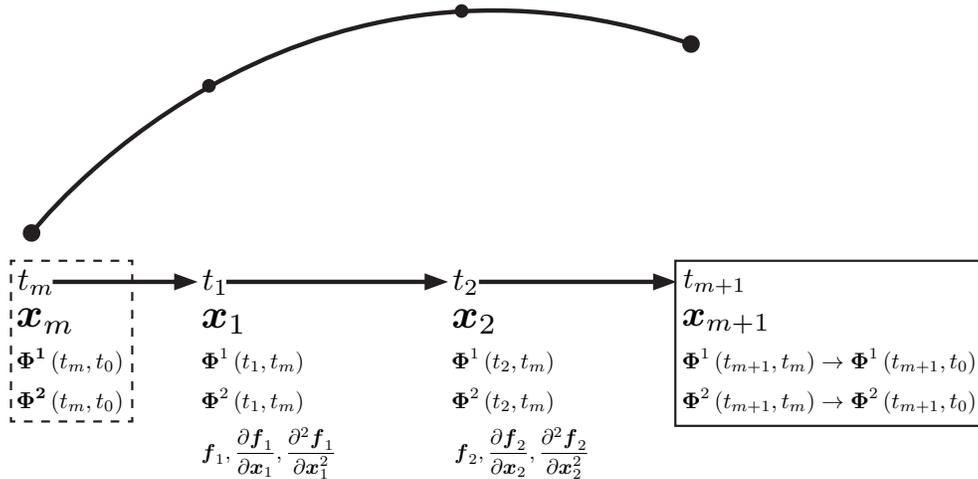


Figure 4.1: Conceptual diagram of one step of a two-stage RK method.

4.3.3.1 First-Order ODEs

For a system of first-order ODEs, the linear system is derived by differentiating Eq. (1.3) with respect to \mathbf{x}_m once or more and manipulating the resulting equations.

First-Order STM Differentiating Eq. (1.3) with respect to \mathbf{x}_m gives

$$\frac{\partial \mathbf{x}_{m+1}}{\partial \mathbf{x}_m} = \mathbf{I}_n + h \sum_{j=1}^s b_j \frac{\partial \mathbf{f}_j}{\partial \mathbf{x}_m} \quad (4.9)$$

$$= \mathbf{I}_n + h \sum_{j=1}^s b_j \mathbf{J}_j \frac{\partial \mathbf{x}_j}{\partial \mathbf{x}_m} \quad (4.10)$$

$$\mathbf{J}_i \triangleq \mathbf{J}(t_i, \mathbf{x}_i), \quad i = 1, \dots, s \quad (4.11)$$

$$\frac{\partial \mathbf{f}_i}{\partial \mathbf{x}_m} = \mathbf{J}_i \frac{\partial \mathbf{x}_i}{\partial \mathbf{x}_m}, \quad i = 1, \dots, s, \quad (4.12)$$

where \mathbf{I}_n is the $n \times n$ identity matrix. The $\partial \mathbf{x}_j / \partial \mathbf{x}_m$ are eliminated from Eq. (4.10) using a relation obtained from differentiating Eq. (1.6) with respect to \mathbf{x}_m :

$$\frac{\partial \mathbf{x}_i}{\partial \mathbf{x}_m} = \mathbf{I}_n + h \sum_{j=1}^s a_{ij} \frac{\partial \mathbf{f}_j}{\partial \mathbf{x}_m}, \quad i = 1, \dots, s. \quad (4.13)$$

Substituting Eq. (4.13) into Eq. (4.12) yields

$$\frac{\partial \mathbf{f}_i}{\partial \mathbf{x}_m} = \mathbf{J}_i \left(\mathbf{I}_n + h \sum_{j=1}^s a_{ij} \frac{\partial \mathbf{f}_j}{\partial \mathbf{x}_m} \right), \quad i = 1, \dots, s. \quad (4.14)$$

Rewriting Eq. (4.14) to isolate the unknown $\partial \mathbf{f}_j / \partial \mathbf{x}_m$ results in the linear system $\mathbf{D}_{ns \times ns} \boldsymbol{\xi}_{ns \times n} = \boldsymbol{\eta}_{ns \times n}$, where

$$\boldsymbol{\xi} = \begin{bmatrix} \frac{\partial f_1}{\partial \mathbf{x}_m} \\ \vdots \\ \frac{\partial f_s}{\partial \mathbf{x}_m} \end{bmatrix}, \quad \boldsymbol{\eta} = \begin{bmatrix} \mathbf{J}_1 \\ \vdots \\ \mathbf{J}_s \end{bmatrix} \quad (4.15)$$

$$\mathbf{D}_{ij} = \begin{cases} -\mathbf{J}_i h a_{ij} & \text{if } i \neq j \\ \mathbf{I} - \mathbf{J}_i h a_{ij} & \text{if } i = j \end{cases}. \quad (4.16)$$

\mathbf{D}_{ij} refers to the ij submatrix of \mathbf{D} , which is $n \times n$. \mathbf{D} is also the matrix whose factored form would be used to perform a full Newton-Raphson update if that scheme were used to solve for the elements of the state itself at the internal stages of an IRK.² The partial derivatives present in \mathbf{D} and $\boldsymbol{\eta}$ are simply the system Jacobians evaluated at the internal stages following the convergence of the state in the initial iterative procedure.

Second-Order STM The second-order STM is found by differentiating Eq. (4.9) with respect to \mathbf{x}_m :

$$\frac{\partial^2 \mathbf{x}_{m+1}}{\partial \mathbf{x}_m^2} = h \sum_{j=1}^s b_j \frac{\partial^2 \mathbf{f}_j}{\partial \mathbf{x}_m^2}. \quad (4.17)$$

The $\partial^2 \mathbf{f}_j / \partial \mathbf{x}_m^2$ are the $n^3 s$ elements to be determined through the solution of the linear system of equations. Eq. (4.13) is also differentiated with respect to \mathbf{x}_m to obtain

²Due to the expense of evaluating \mathbf{J} at every internal stage at every iteration, a full Newton-Raphson scheme is generally avoided in favor of an approximate Newton method, fixed-point method, or Gauss-Seidel method.

$$\frac{\partial^2 \mathbf{x}_i}{\partial \mathbf{x}_m^2} = h \sum_{j=1}^s a_{ij} \frac{\partial^2 \mathbf{f}_j}{\partial \mathbf{x}_m^2}, \quad i = 1, \dots, s. \quad (4.18)$$

Through the chain rule, \mathbf{J} and \mathbf{H} relate Φ^2 to the $\partial^2 \mathbf{f}_j / \partial \mathbf{x}_m^2$. After manipulation, the linear system matrix \mathbf{D} for Φ^2 is found to be the same as that for the first-order case – meaning that only a single factorization is required to obtain both Φ^1 and Φ^2 . However, due to the increased dimensionality of Φ^2 , the system must be solved for a larger $\boldsymbol{\eta}$. The system arrays are given element-wise by

$$\boldsymbol{\xi}_k = \begin{bmatrix} \frac{\partial}{\partial \mathbf{x}_m(k)} \left(\frac{\partial \mathbf{f}_1}{\partial \mathbf{x}_m} \right) \\ \vdots \\ \frac{\partial}{\partial \mathbf{x}_m(k)} \left(\frac{\partial \mathbf{f}_s}{\partial \mathbf{x}_m} \right) \end{bmatrix}_{ns \times n}, \quad \boldsymbol{\eta}_k = \begin{bmatrix} \boldsymbol{\eta}_1(1:n, 1:n, k) \\ \vdots \\ \boldsymbol{\eta}_s(1:n, 1:n, k) \end{bmatrix}_{ns \times n}, \quad k = 1, \dots, n \quad (4.19)$$

$$\begin{aligned} \boldsymbol{\eta}_l(i, j, k) = & \sum_{q=1}^n \left\{ \frac{\partial}{\partial \mathbf{x}_l(q)} \left(\frac{\partial \mathbf{f}_l(i)}{\partial \mathbf{x}_l(j)} \right) \frac{\partial \mathbf{x}_l(q)}{\partial \mathbf{x}_m(k)} \right\} + \\ & h \sum_{\alpha=1}^s a(l, \alpha) \sum_{p=1}^n \frac{\partial \mathbf{f}_\alpha(p)}{\partial \mathbf{x}_m(j)} \sum_{q=1}^n \left\{ \frac{\partial}{\partial \mathbf{x}_l(q)} \left(\frac{\partial \mathbf{f}_l(i)}{\partial \mathbf{x}_l(p)} \right) \frac{\partial \mathbf{x}_l(q)}{\partial \mathbf{x}_m(k)} \right\}. \end{aligned} \quad (4.20)$$

In addition to the Jacobians and Hessians at the internal stages, Eq. (4.20) relies on (1) the solutions of the linear system for the first-order STM (the $\partial \mathbf{f}_i / \partial \mathbf{x}_m$) and (2) the first-order STMs at the internal stages (the $\partial \mathbf{x}_i / \partial \mathbf{x}_m$). Calculation of the latter is not required to obtain the first-order STM at t_{m+1} , but may be found directly through evaluation of Eq. (4.13).

4.3.3.2 Double Integrator

For a system of second-order ODEs, Eqs. (4.2) and (4.3) are each differentiated with respect to both \mathbf{y}_m and $\dot{\mathbf{y}}_m$.

First-Order STM For Φ^1 , four expressions result from the differentiation:

$$\frac{\partial \mathbf{y}_{m+1}}{\partial \mathbf{y}_m} = \mathbf{I}_{n'} + h^2 \sum_{i=1}^s \bar{b}_i \frac{\partial \mathbf{g}_i}{\partial \mathbf{y}_m} \quad (4.21)$$

$$\frac{\partial \mathbf{y}_{m+1}}{\partial \dot{\mathbf{y}}_m} = h \mathbf{I}_{n'} + h^2 \sum_{i=1}^s \bar{b}_i \frac{\partial \mathbf{g}_i}{\partial \dot{\mathbf{y}}_m} \quad (4.22)$$

$$\frac{\partial \dot{\mathbf{y}}_{m+1}}{\partial \mathbf{y}_m} = h \sum_{i=1}^s b_i \frac{\partial \mathbf{g}_i}{\partial \mathbf{y}_m} \quad (4.23)$$

$$\frac{\partial \dot{\mathbf{y}}_{m+1}}{\partial \dot{\mathbf{y}}_m} = \mathbf{I}_{n'} + h \sum_{i=1}^s b_i \frac{\partial \mathbf{g}_i}{\partial \dot{\mathbf{y}}_m}. \quad (4.24)$$

Rearranging the expressions as before, it is found that the solutions of *two* linear systems are now required: one for the $\partial \mathbf{g}_i / \partial \mathbf{y}_m$ and another for the $\partial \mathbf{g}_i / \partial \dot{\mathbf{y}}_m$. However, the two systems share the same \mathbf{D} matrix. Thus, \mathbf{D} need only be factored once and the result reused to efficiently solve the second system.³ Further, the dimensions of \mathbf{D} are smaller if the double-integrator form is used instead of applying the single-order form to a second-order system: In the double-integrator case, $\mathbf{D} \in \mathbb{R}^{n's \times n's}$, while, in the single-integrator case, $\mathbf{D} \in \mathbb{R}^{ns \times ns}$. Thus, using the double-integrator form when appropriate

³In fact, the two systems may be solved as a single combined system with n righthand-side columns formed by $\boldsymbol{\eta}_i = [\boldsymbol{\eta}_i^1 \quad \boldsymbol{\eta}_i^2]$.

improves computational efficiency. Mimicking the notation of Eqs. (4.15) and (4.16), the relevant arrays are

$$\boldsymbol{\xi}^j = \begin{bmatrix} \boldsymbol{\xi}_1^j \\ \vdots \\ \boldsymbol{\xi}_s^j \end{bmatrix}_{n' \times n'}, \quad \boldsymbol{\eta}^j = \begin{bmatrix} \boldsymbol{\eta}_1^j \\ \vdots \\ \boldsymbol{\eta}_s^j \end{bmatrix}_{n' \times n'}, \quad j = 1, 2 \quad (4.25)$$

$$\boldsymbol{\xi}_i^1 = \frac{\partial \mathbf{g}_i}{\partial \mathbf{y}_m}, \quad \boldsymbol{\eta}_i^1 = \frac{\partial \mathbf{g}_i}{\partial \mathbf{y}_i} \quad (4.26)$$

$$\boldsymbol{\xi}_i^2 = \frac{\partial \mathbf{g}_i}{\partial \dot{\mathbf{y}}_m}, \quad \boldsymbol{\eta}_i^2 = \bar{c}_i h \frac{\partial \mathbf{g}_i}{\partial \mathbf{y}_i} + \frac{\partial \mathbf{g}_i}{\partial \dot{\mathbf{y}}_i} \quad (4.27)$$

$$\mathbf{D}_{ij} = \begin{cases} -\frac{\partial \mathbf{g}_i}{\partial \mathbf{y}_i} h^2 \bar{a}_{ij} - \frac{\partial \mathbf{g}_i}{\partial \dot{\mathbf{y}}_i} h a_{ij} & \text{if } i \neq j \\ \mathbf{I}_{n'} - \frac{\partial \mathbf{g}_i}{\partial \mathbf{y}_i} h^2 \bar{a}_{ij} - \frac{\partial \mathbf{g}_i}{\partial \dot{\mathbf{y}}_i} h a_{ij} & \text{if } i = j \end{cases}. \quad (4.28)$$

\mathbf{D}_{ij} is a submatrix of dimension $n' \times n'$. The complete STM may be reconstructed as

$$\boldsymbol{\Phi}^1 = \frac{\partial \mathbf{x}_{m+1}}{\partial \mathbf{x}_m} = \begin{bmatrix} \frac{\partial \mathbf{y}_{m+1}}{\partial \mathbf{y}_m} & \frac{\partial \mathbf{y}_{m+1}}{\partial \dot{\mathbf{y}}_m} \\ \frac{\partial \dot{\mathbf{y}}_{m+1}}{\partial \mathbf{y}_m} & \frac{\partial \dot{\mathbf{y}}_{m+1}}{\partial \dot{\mathbf{y}}_m} \end{bmatrix}. \quad (4.29)$$

Second-Order STM The second-order STM is found by differentiating Eqs. (4.21)–(4.24) with respect to \mathbf{y}_m and $\dot{\mathbf{y}}_m$, which results in eight equations. This number may be reduced if the nomenclature from the first-order RK equations is recalled. The expressions may be written as

$$\frac{\partial}{\partial \mathbf{x}_m} \left(\frac{\partial \mathbf{y}_{m+1}}{\partial \mathbf{x}_m} \right) = h^2 \sum_{i=1}^s \bar{b}_i \frac{\partial}{\partial \mathbf{x}_m} \left(\frac{\partial \mathbf{g}_i}{\partial \mathbf{x}_m} \right) \quad (4.30)$$

$$\frac{\partial}{\partial \mathbf{x}_m} \left(\frac{\partial \dot{\mathbf{y}}_{m+1}}{\partial \mathbf{x}_m} \right) = h \sum_{i=1}^s b_i \frac{\partial}{\partial \mathbf{x}_m} \left(\frac{\partial \mathbf{g}_i}{\partial \mathbf{x}_m} \right), \quad (4.31)$$

where each of the two arrays is dimensioned $n' \times n \times n$. Thus,

$$\mathbf{\Phi}^2 = \frac{\partial^2 \mathbf{x}_{m+1}}{\partial \mathbf{x}_m^2} = \begin{bmatrix} \frac{\partial}{\partial \mathbf{x}_m} \left(\frac{\partial \mathbf{y}_{m+1}}{\partial \mathbf{x}_m} \right) \\ \frac{\partial}{\partial \mathbf{x}_m} \left(\frac{\partial \dot{\mathbf{y}}_{m+1}}{\partial \mathbf{x}_m} \right) \end{bmatrix}. \quad (4.32)$$

The task, then, is to solve for the $\partial/\partial \mathbf{x}_m [\partial \mathbf{g}_i/\partial \mathbf{x}_m]$. The procedure follows that described for systems of first-order ODEs. The linear system matrix \mathbf{D} is, once again, the same as was required for calculation of $\mathbf{\Phi}^1$ (Eq. (4.28)). The factorization is therefore reused with new $\boldsymbol{\eta}$ arrays. The system may be written as

$$\boldsymbol{\xi}_k^j = \begin{bmatrix} \boldsymbol{\xi}_{k,1}^j \\ \vdots \\ \boldsymbol{\xi}_{k,s}^j \end{bmatrix}_{n's \times n}, \quad \boldsymbol{\eta}_k^j = \begin{bmatrix} \boldsymbol{\eta}_1^j(1:n', 1:n, k) \\ \vdots \\ \boldsymbol{\eta}_s^j(1:n', 1:n, k) \end{bmatrix}_{n's \times n}, \quad (4.33)$$

$$k = 1, \dots, n', \quad j = 1, \dots, 4$$

$$\boldsymbol{\xi}_{k,i}^1 = \frac{\partial}{\partial \mathbf{y}_m(k)} \left(\frac{\partial \mathbf{g}_i}{\partial \mathbf{y}_m} \right), \quad \boldsymbol{\xi}_{k,i}^2 = \frac{\partial}{\partial \mathbf{y}_m(k)} \left(\frac{\partial \mathbf{g}_i}{\partial \dot{\mathbf{y}}_m} \right) \quad (4.34)$$

$$\boldsymbol{\xi}_{k,i}^3 = \frac{\partial}{\partial \dot{\mathbf{y}}_m(k)} \left(\frac{\partial \mathbf{g}_i}{\partial \mathbf{y}_m} \right), \quad \boldsymbol{\xi}_{k,i}^4 = \frac{\partial}{\partial \dot{\mathbf{y}}_m(k)} \left(\frac{\partial \mathbf{g}_i}{\partial \dot{\mathbf{y}}_m} \right) \quad (4.35)$$

$$\begin{aligned} \boldsymbol{\eta}_l^1(i, j, k) &= \sum_{q=1}^n \left\{ \frac{\partial}{\partial \mathbf{x}_l(q)} \left(\frac{\partial \mathbf{g}_l(i)}{\partial \mathbf{y}_l(j)} \right) \frac{\partial \mathbf{x}_l(q)}{\partial \mathbf{y}_m(k)} \right\} + \\ &h^2 \sum_{\alpha=1}^s \bar{a}_{l\alpha} \sum_{p=1}^{n'} \frac{\partial \mathbf{g}_\alpha(p)}{\partial \mathbf{y}_m(j)} \sum_{q=1}^n \left\{ \frac{\partial}{\partial \mathbf{x}_l(q)} \left(\frac{\partial \mathbf{g}_l(i)}{\partial \mathbf{y}_l(p)} \right) \frac{\partial \mathbf{x}_l(q)}{\partial \mathbf{y}_m(k)} \right\} + \\ &h \sum_{\alpha=1}^s a_{l\alpha} \sum_{p=1}^{n'} \frac{\partial \mathbf{g}_\alpha(p)}{\partial \mathbf{y}_m(j)} \sum_{q=1}^n \left\{ \frac{\partial}{\partial \mathbf{x}_l(q)} \left(\frac{\partial \mathbf{g}_l(i)}{\partial \dot{\mathbf{y}}_l(p)} \right) \frac{\partial \mathbf{x}_l(q)}{\partial \mathbf{y}_m(k)} \right\} \end{aligned} \quad (4.36)$$

$$\begin{aligned}
\boldsymbol{\eta}_l^2(i, j, k) = & \bar{c}_l h \sum_{q=1}^n \left\{ \frac{\partial}{\partial \mathbf{x}_l(q)} \left(\frac{\partial \mathbf{g}_l(i)}{\partial \mathbf{y}_l(j)} \right) \frac{\partial \mathbf{x}_l(q)}{\partial \mathbf{y}_m(k)} \right\} + \\
& h^2 \sum_{\alpha=1}^s \bar{a}_{l\alpha} \sum_{p=1}^{n'} \frac{\partial \mathbf{g}_\alpha(p)}{\partial \dot{\mathbf{y}}_m(j)} \sum_{q=1}^n \left\{ \frac{\partial}{\partial \mathbf{x}_l(q)} \left(\frac{\partial \mathbf{g}_l(i)}{\partial \mathbf{y}_l(p)} \right) \frac{\partial \mathbf{x}_l(q)}{\partial \mathbf{y}_m(k)} \right\} + \\
& h \sum_{\alpha=1}^s a_{l\alpha} \sum_{p=1}^{n'} \frac{\partial \mathbf{g}_\alpha(p)}{\partial \dot{\mathbf{y}}_m(j)} \sum_{q=1}^n \left\{ \frac{\partial}{\partial \mathbf{x}_l(q)} \left(\frac{\partial \mathbf{g}_l(i)}{\partial \dot{\mathbf{y}}_l(p)} \right) \frac{\partial \mathbf{x}_l(q)}{\partial \mathbf{y}_m(k)} \right\} + \\
& \sum_{q=1}^n \left\{ \frac{\partial}{\partial \mathbf{x}_l(q)} \left(\frac{\partial \mathbf{g}_l(i)}{\partial \dot{\mathbf{y}}_l(j)} \right) \frac{\partial \mathbf{x}_l(q)}{\partial \mathbf{y}_m(k)} \right\}.
\end{aligned} \tag{4.37}$$

The expressions for $\boldsymbol{\eta}^3(i, j, k)$ and $\boldsymbol{\eta}^4(i, j, k)$ are obtained from Eqs. (4.36) and (4.37), respectively, by performing the expression replacement

$$\frac{\partial \mathbf{x}_l(q)}{\partial \mathbf{y}_m(k)} \rightarrow \frac{\partial \mathbf{x}_l(q)}{\partial \dot{\mathbf{y}}_m(k)} \tag{4.38}$$

As with the first-order STM case, taking advantage of the double-integrator form improves computational efficiency by decreasing the size of the \mathbf{D} matrix that must be factored and by decreasing the total number of unknowns in the linear system.

4.3.3.3 Variable-Step-Size Propagation

The linear-algebra-based methods developed in this section assume that the step size h is independent of the initial state of the step, \mathbf{x}_m . However, for a propagation scheme that adapts h at each step based on some function of the state, this assumption does not hold. Such is the case for a propagator that varies h in an attempt to meet a user-specified accuracy requirement based

on an estimate of the local truncation error at each step [56].⁴ In fact, this limitation also applies to methods based on propagating the variational equations, whether by an implicit or explicit ODE solver. Alternative sensitivity calculation methods, such as those based on a complex-step or finite-difference derivative approximation of an entire propagation, handle variable-step ODE solvers appropriately. At the same time, neither of these alternatives is itself perfect. Pellegrini and Russell [163] give a detailed analysis of sensitivity calculations using variable-step propagators; in the present work, all example scenarios use a fixed step size.

4.3.4 Variable-Fidelity Dynamics with IRK Methods

As introduced in Section 1.2.2, an IRK method requires the solution of a (generally nonlinear) system of algebraic equations at each propagation step, while an ERK method does not. Consequently, customizations of the nonlinear solver may be used to improve the efficiency of IRK methods. In particular, if the computational expense of the system dynamics dominates propagation runtime, then the variable-fidelity dynamics concepts described in Sections 1.2.2.3 and 3.3.1 may be applied to the calculation of STMs using either the CDM or the DDM.

It is cautioned that this strategy is not applicable for all systems; a suitably accurate and inexpensive low-fidelity approximation must exist for the variable-fidelity technique to be worthwhile (see Chapter 3). Additionally,

⁴The effects of the dependence of h on the state is most evident when loose tolerances are prescribed.

logic must be built into the nonlinear solver to determine whether to use the high-fidelity or low-fidelity model whenever a dynamics function evaluation is required. In this chapter, the impact of variable-fidelity dynamics models for both the CDM and DDM is examined using the algorithm presented in Chapter 3, which uses exactly two high-fidelity dynamics evaluations per stage per step. Other iterations of the nonlinear solver require evaluations of the low-fidelity model only. Implementation details for the CDM and the DDM are described below.

4.3.4.1 Coupled Direct Method

When using the CDM, the state and any STMs are collected in a single augmented state vector. Therefore, the variable-fidelity strategy requires low-fidelity versions of \mathbf{f} , \mathbf{J} , and \mathbf{H} to fully describe the low-fidelity dynamics of the augmented state. The variable-fidelity strategy is then applied to the entire augmented state vector.

4.3.4.2 Decoupled Direct Method

There are two options for implementing the variable-fidelity dynamics strategy using the DDM:

1. State-only propagation is performed using the variable-fidelity strategy, and the converged states at all internal stages are saved. Thus, two high-fidelity evaluations of \mathbf{f} per stage per step are performed. Subsequently, the high-fidelity \mathbf{J} and \mathbf{H} are evaluated once at each stage using the converged state values; no iteration is used in the calculation of \mathbf{J} and

\mathbf{H} . This method does not require low-fidelity versions of \mathbf{J} and \mathbf{H} because the variable-fidelity strategy is only used for propagation of the state \mathbf{x} , and not for the calculation of any STMs.

2. At the final (i.e., second in the current implementation) high-fidelity iteration of the nonlinear solution process for the state, the high-fidelity \mathbf{J} and \mathbf{H} are evaluated *alongside* the high-fidelity \mathbf{f} for each stage. At any subsequent iterations – all of which are low-fidelity – the low-fidelity \mathbf{f} , \mathbf{J} , and \mathbf{H} are all calculated, and the iterative process is terminated when the state \mathbf{x} at each stage has converged. As \mathbf{J} and \mathbf{H} are functions of t and \mathbf{x} only, convergence of \mathbf{x} implies convergence of \mathbf{J} and \mathbf{H} , as well. \mathbf{J} and \mathbf{H} are therefore *not* recalculated after the state-only nonlinear solver converges; the converged values of \mathbf{J} and \mathbf{H} are simply used in the construction of the STMs. Thus, like Option 1, Option 2 uses a single high-fidelity evaluation of \mathbf{J} and \mathbf{H} per stage per step. Unlike Option 1, Option 2 requires low-fidelity versions of \mathbf{J} and \mathbf{H} .

The motivation behind Option 2 is that evaluation of \mathbf{J} and \mathbf{H} often essentially requires evaluating \mathbf{f} , as well. Using Option 2 avoids the extra pseudo-evaluation of \mathbf{f} used by Option 1. The cost is that Option 2 requires some (usually small) number of low-fidelity \mathbf{J} and \mathbf{H} evaluations. Therefore, which option is more efficient depends on the relative computational costs of the low-fidelity and high-fidelity \mathbf{f} , \mathbf{J} , and \mathbf{H} for a specific system.

4.3.5 Comparison of the Three Direct Methods: CDM, DDM Linear Algebra, and DDM Multi-Iteration

Two primary criteria exist for comparing the three alternatives: ease of use and computational efficiency. From an implementation standpoint, the CDM method is the simplest option because the same ODE solution routine used to propagate the state alone is capable of propagating the augmented state vector. On the other hand, the DDMs require either (1) a customized ODE solver (which computes the STMs in conjunction with the state propagation) or (2) a post-processing routine (which computes the STMs after the state has been propagated). Either option (1) or (2) leads to increased bookkeeping relative to the CDM.

When comparing the computational efficiencies of the three methods, an immediate advantage of the linear-algebra method over both the CDM and multi-iteration method is that the linear system formulation eliminates the need for an initial guess for the elements of the STMs at each stage. In addition to removing a computational requirement, this feature makes the performance of the linear-algebra method more predictable than that of its competitors because there is no possibility for variations in the number of iterations necessary to achieve convergence of the STMs. Further efficiency comparisons of the linear-algebra method with the multi-iteration method are implementation-specific, as each of these two methods uses the same number of evaluations of the \mathbf{f} , \mathbf{J} , and \mathbf{H} equations. The most fundamental question is whether the construction and solution of the system of linear equations takes more or less time than the iterative solution approach used by the multi-

iteration method. The answer is dependent primarily on three factors: (1) the computational expense of the dynamics, (2) the size of the state vector (n)⁵, and (3) the number of internal stages per step (s).

First, as the cost of evaluating the dynamics increases, the CPU time spent doing so increasingly dominates the time spent solving for the STMs using any of the three methods. Therefore, computationally demanding dynamics are likely to result in nearly identical computation times for the multi-iteration and linear-algebra methods and comparatively poor performance from the CDM.

The second and third factors come into play because the matrix factored in the linear-algebra method is of size $ns \times ns$ (for the first-order form of the RK equations). Further, the operations required to create the “righthand-side” arrays are summations over either n or s . As the size of the state vector and/or the number of internal stages of the IRK method increase, the efficiency of the linear-algebra method is likely to decrease relative to that of the multi-iteration method. IRK methods that are intended to be used with a very large number of stages (e.g., 100), like BLC-IRK [33] or MCPI [16], may therefore obtain more efficient results using the multi-iteration method.

An additional factor that can affect the efficiencies of all three methods is the use of parallel processing. The independence of the dynamics function evaluations at each internal stage of an IRK method allows the evaluations

⁵Using the double-integrator form for a second-order system reduces the size of the linear system matrix.

to be performed in parallel at each iteration. This feature has the potential to provide greatest benefits for the CDM. First, the CDM evaluates the expensive Jacobians and Hessians more frequently than the other methods, and may therefore experience larger relative efficiency gains. Further, the CDM places the calculations related to solving for the state and all STMs within one iterative structure, while the other two methods “spread out” the computations. This organization works against the CDM in a serial computing environment. However, the less-demanding loops of the multi-iteration and linear-algebra methods cause parallelization overhead to become more apparent than the expensive loops of the CDM. For example, if using fixed-point iteration to solve for the first-order STM using the multi-iteration method, the parallelizable dynamics evaluations at each iteration of the *first* loop involve the calculation of \mathbf{f} at each internal stage. The impact of parallelization overhead on this loop is therefore directly related to the expense of calculating \mathbf{f} . Similarly, the cost of the dynamics also drives the parallelization efficiency of the post-convergence calculation of \mathbf{J} at each stage. On the other hand, the *second* loop (i.e., using the converged \mathbf{J} to calculate Φ^1 at each stage via Eq. (1.4)–(1.6)) consists only of a matrix multiplication, and parallelization may produce little or no benefit for a system of moderate size. No such issues exist for the CDM because the evaluations of \mathbf{J} and the construction of Φ^1 are performed in the same loop.

Each of the n linear solves required by the linear-algebra method to calculate the second-order STM are independent. Referencing Eq. (4.19), $\boldsymbol{\xi}_1 = \mathbf{D}\boldsymbol{\eta}_1$ is independent of $\boldsymbol{\xi}_2 = \mathbf{D}\boldsymbol{\eta}_2$, etc.. Thus, each of the $\boldsymbol{\eta}_k$ may be

assembled, and the corresponding systems solved, in parallel, which improves the parallelization properties of the method.

If a large number of threads is available, massive parallelization is realizable for the multi-iteration and linear-algebra methods. A trajectory segment spanning multiple steps may be propagated using the state-only dynamics. If the converged states at the internal stages for all steps in the segment are saved, the Jacobians and Hessians may be evaluated subsequently, completely separately from the original propagation of the state. Subsequently, either an iterative method or the linear-algebra method produces the STMs for each step, and Eqs. (B.12) and (B.13) are used to create the full trajectory STMs from the STMs for each individual step. In the limiting case, this strategy permits the simultaneous performance of *every* \mathbf{J} and \mathbf{H} evaluation, as well as the STM calculations at every step, for an entire trajectory. This concept has previously been implemented using an ERK ODE solver and graphics processing units (GPUs) [9]. However, while an explicit method's \mathbf{J} and \mathbf{H} evaluations are parallelizable across multiple steps, parallelization within a single step is not possible. Thus, use of an IRK allows for the number of parallel processes to be increased by a factor of s .

4.4 Results

To illustrate the relative merits of the three direct STM calculation methods, two scenarios of varying complexity are presented, and the CPU times required by the three methods are compared. All code is written in Fortran and compiled using the Intel Visual Fortran Compiler XE 14.0.0.103

(64-bit) using the `-O2` optimization flag. All computations are performed on a 64-bit Windows 7 Enterprise workstation with two 12-core Intel Xeon E5-2680 v3 processors (clock speed 2.50 GHz) and 64 GB of RAM. Hyperthreading is disabled. Algebraic linear systems are solved using the Intel Math Kernel Library (MKL) [105] implementation of the LAPACK [3] routine `dgesv`, which accepts a matrix righthand side and returns both the solution and the LU -factored system matrix. When performing subsequent solves that use the same system matrix – e.g., when solving for Φ^2 after Φ^1 has been obtained – the factorization may be reused. In these cases, the LAPACK routine `dgetrs` is used.

The example systems under consideration are nonstiff; therefore, at each step of the propagation, the system of nonlinear equations is solved using fixed-point iteration [88, 86]. The iterations are stopped when the root-mean-square of the relative changes in the states at the internal stages falls below a given threshold (10^{-15} is used for the presented results).

The DDM facilitates efficient use of an approximate Newton method as an alternative to fixed-point iteration. As the nonlinear solver for the DDM iterates only on the state \mathbf{x} , an approximate Newton method requires only an approximation of the Jacobian $\mathbf{J} = \partial \mathbf{f} / \partial \mathbf{x}$ at each internal stage of each propagation step. Such an approximation may be obtained from, for example, converged \mathbf{J} of the previous step, so that additional \mathbf{J} evaluations beyond those needed to calculate the STMs are not required [207]. On the other hand, the CDM iterates on the augmented state that includes \mathbf{x} and any calculated STMs. For the CDM, the Jacobian used by an approximate Newton method

requires the partial derivatives of any calculated STMs. For example, if \mathbf{x} and Φ^1 (and not Φ^2) are calculated using the CDM, an approximate Newton method would require not only \mathbf{f} and \mathbf{J} , but $\mathbf{H} = \partial\mathbf{J}/\partial\mathbf{x}$, as well. Thus, while use of an approximate Newton method is feasible for the CDM, the implementation is less efficient than for the DDM.

For consistency between the CDM and DDM implementations, fixed-point iteration is used for all examples presented in this chapter.

4.4.1 Simple Dynamics Example: Damped Oscillator

As a first example, the three methods are applied to a damped oscillator. The equations of motion for the system are

$$\ddot{x} + \nu\dot{x} + \omega_0^2x = \sin(x\dot{x}). \quad (4.39)$$

A nonlinearity is introduced in the forcing term so that the Hessian is nonzero. This system may be written in either first- or second-order form, with $n' = 1$ and $n = 2$. The system is propagated from $t_0 = 0$ to $t_f = 10$ for $\nu = 0.5$ and $\omega_0 = 2\pi$. The initial conditions are given by $x = 1$, $\dot{x} = 0$. A three-stage (sixth-order) GLIRK with a fixed step size of 0.05 is used to propagate the state and STMs using each permutation of (1) single-integrator or double-integrator ODE formulation, (2) CDM, multi-iteration, or linear-algebra STM calculation, and (3) Φ^1 calculation only or Φ^1 and Φ^2 calculation. The initial guesses for the states at the internal stages, needed by the fixed-point iteration scheme, are set to the initial value of the state at the current step. Variable-fidelity

dynamics and parallelization are not used due to the inexpensive dynamics and small size of the linear system.

The time evolutions of the state, Φ^1 , and Φ^2 are shown in Figure 4.2. The maximum relative difference between any two elements of $\Phi^1(t_f, t_0)$ or $\Phi^2(t_f, t_0)$ between any of the propagation methods is 1.58×10^{-13} . Figure 4.3 displays the averaged CPU times required by the propagations.⁶ For the propagation of Φ^1 only (Figure 4.3a), the CDM is the most efficient of the three approaches, followed by the linear-algebra method. The poor performances of the multi-iteration and linear-algebra methods result from the extremely inexpensive dynamics. The goal of both approaches is to decrease the number of \mathbf{f} , \mathbf{J} , and \mathbf{H} evaluations, which is achieved, as shown in Figure 4.4. If these evaluations are not a significant driver of CPU time, the additional overhead of the multi-iteration and linear-algebra methods causes inefficient propagation. For similar reasons, use of the double-integrator provides negligible benefit to the CDM and actually hurts the performance of the multi-iteration and linear-algebra methods despite a reduction in function evaluations.⁷

When calculating both Φ^1 and Φ^2 , the linear-algebra method is most efficient (Figure 4.3b). The relative improvement of the linear-algebra method compared to the Φ^1 -only case is due to the fact that the LU factorization performed to obtain Φ^1 is reused to calculate Φ^2 . The linear-algebra method also benefits from calculating \mathbf{H} less frequently than the CDM (Figure 4.4b).

⁶Note differing vertical axis scales.

⁷The number of \mathbf{J} evaluations is constant for the multi-iteration and linear-algebra approaches for the single- and double-integrators: one per stage per step.

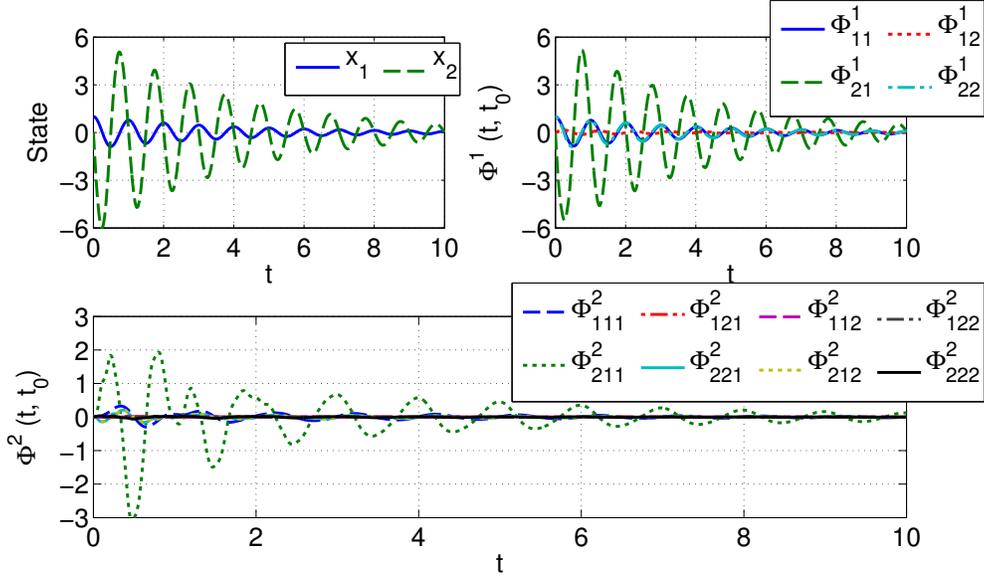


Figure 4.2: Time evolution of damped oscillator system.

This small increase in the expense of the dynamics accounts for the improved relative performance of the double-integrator compared to the first-order integrator. However, as seen in Figure 4.3b, the dynamics are still not nearly expensive enough to justify the use of the multi-iteration method.

4.4.2 Expensive Dynamics Examples: Space Object Trajectories

Examples are presented for problems with larger dimensions and more complicated dynamics. The techniques for STM calculation are applied to the trajectory of an SO in LEO, whose dynamics are governed by

$$\ddot{\mathbf{r}} = -\frac{\mu_*}{r^3}\mathbf{r} + \mathbf{a}_p. \quad (4.40)$$

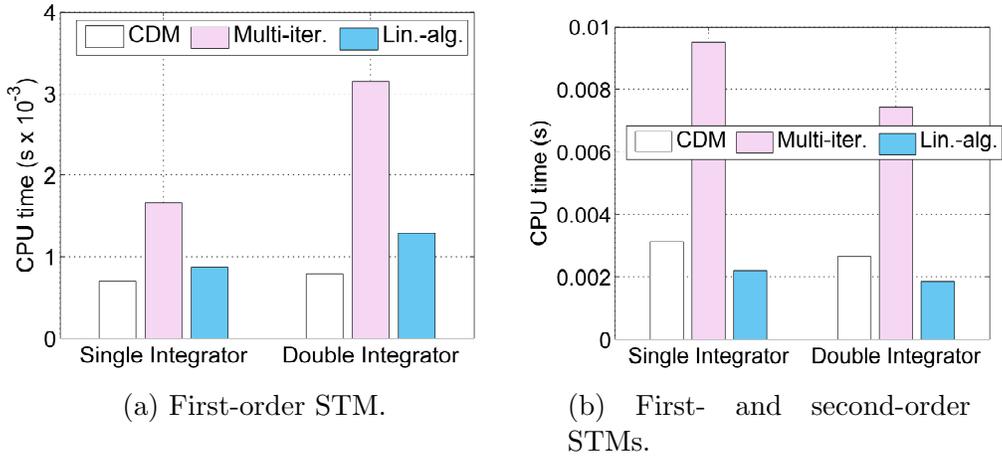


Figure 4.3: CPU time required to propagate state and calculate STMs for damped oscillator.

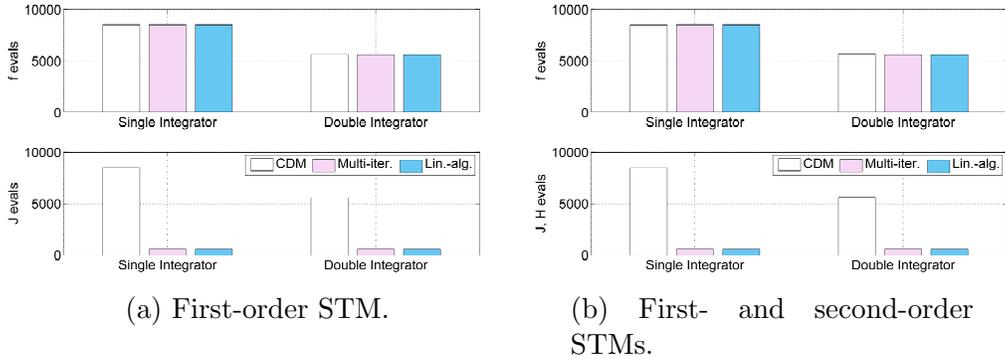


Figure 4.4: Number of function, Jacobian, and Hessian evaluations required to propagate state and calculate STMs for damped oscillator.

The Cowell formulation of the system is considered, in which $\mathbf{x} = [\mathbf{r}^T \quad \dot{\mathbf{r}}^T]^T$ [20]. For the double-integrator formulation, $\mathbf{y} = \mathbf{r}$. Two cases are considered: a low-fidelity case, in which \mathbf{a}_p consists only of the oblateness (i.e., J_2) term of the Earth’s geopotential, and a high-fidelity case, in which acceleration is calculated using the spherical harmonics implementation of a degree-and-order 70 geopotential model [183]. While a true “high-fidelity” model would include

other disturbing accelerations, the intent of this example is to demonstrate the performance difference between the various methods when applied to dynamics models of varying computational expense. For an SO in LEO, calculation of the acceleration caused by a high-fidelity geopotential represents a significantly time-consuming operation [11]. In addition, this scenario is used to highlight the capabilities of the variable-fidelity dynamics approach because the high-fidelity model is very well approximated by the much less computationally expensive low-fidelity model.

For each example, an eight-stage GLIRK is used to propagate the trajectory whose initial conditions are given in classical orbital elements [20] in Table 4.2.⁸ The trajectory is depicted in Figure 4.5, and the time evolutions of representative elements of the first- and second-order STMs are shown in Figure 4.6.⁹ The propagation duration is three orbital periods, and the fixed step size is two minutes. The undisturbed (i.e., two-body or Keplerian) dynamics are used to generate an initial guess for the states and, if necessary, first- and second-order STMs at the internal stages. (The solutions for position, velocity, and first- and second-order STMs are available analytically in the undisturbed case [165].) For the CDM and multi-iteration method, guesses are required

⁸Previous implementations of GLIRK for space object trajectory propagation have used 5–20 stages [113, 8, 5, 92, 93]. While increasing the number of stages increases the order of the ODE solver, stages are oversampled near the endpoints of a time step, and arbitrarily increasing s does not necessarily result in a more efficient propagation. (See Chapter 3.)

⁹In the notation of the legends in Figure 4.6, $\Phi_{1:3,1}^1(t, t_0) = \frac{\partial \mathbf{r}(t)}{\partial \mathbf{r}_1(t_0)}$, $\Phi_{1:3,1,1}^2(t, t_0) = \frac{\partial^2 \mathbf{r}(t)}{\partial \mathbf{r}_1(t_0)^2}$.

for all STMs, while the linear-algebra method requires guesses for only the position and velocity states.

Separate propagations are performed using (1) low-fidelity, (2) high-fidelity, and (3) variable-fidelity dynamics models. For the multi-iteration and linear-algebra methods, the variable-fidelity implementation is that which performs the high-fidelity evaluation of \mathbf{J} and \mathbf{H} within the fixed-point iteration loop rather than after full convergence of the state (i.e., option 2 for the DDM from Section 4.3.4). This strategy is found to be the more efficient of the two alternatives for this particular problem.

The propagations are performed both in serial and using OpenMP-based parallelization [158]. When parallelization is used, eight threads are employed, one for each stage of the GLIRK method. At each iteration of the nonlinear solver, the \mathbf{f} , \mathbf{J} and \mathbf{H} evaluations at each internal stage are evaluated in parallel. An exception to this rule is the case in which only the low-fidelity derivatives function and/or Jacobian – and *not* the low-fidelity Hessian – are evaluated (see Table 4.1). In this scenario, the overhead of parallelization outweighs the benefits of parallelizing the computations. For the linear-algebra-based calculation of Φ^2 , the construction and solution of the n (or n' for the double-integrator formulation) independent linear systems is performed in parallel. For the multi-iteration method, the fixed-point iteration updates for Φ^2 benefit from parallelization due to the expense of constructing Φ^2 even though \mathbf{J} and \mathbf{H} are not re-evaluated. Analysis of the previously-described massive parallelization technique is not performed.

Table 4.1: Parallelization of dynamics model evaluations.

	Low-fidelity			High-fidelity		
	f	J	H	f	J	H
Φ^1			N/A	✓	✓	N/A
Φ^1, Φ^2			✓	✓	✓	✓

Table 4.2: Initial state for space object in LEO.

Orbital element	Value
a	6745.592 km
e	0.01
i	7.81 deg
Ω	100.21 deg
ω	152.83 deg
θ	0 deg

Further details specific to calculation using each dynamics model are given in the following subsections. The high-fidelity and variable-fidelity cases are discussed together because the two methods are different strategies for propagating and calculating the STMs of the same dynamical system.

Low-Fidelity Dynamics The maximum relative difference between any two elements of $\Phi^1(t_f, t_0)$ or $\Phi^2(t_f, t_0)$ between any of the calculation methods is 1.19×10^{-11} . The averaged CPU times required by the different methods of calculating the STMs are shown in the topmost plots of Figures 4.7a and 4.7b.¹⁰

¹⁰Note differing vertical axis scales.

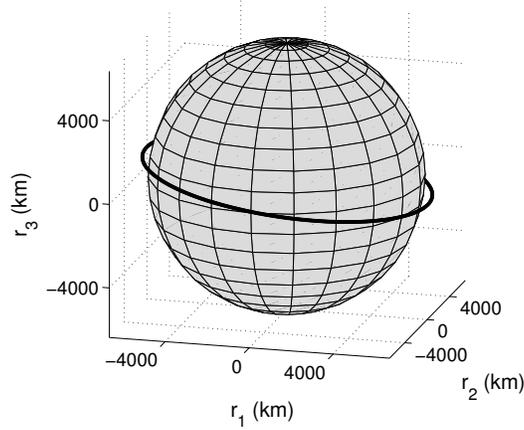


Figure 4.5: High-fidelity space object trajectory.

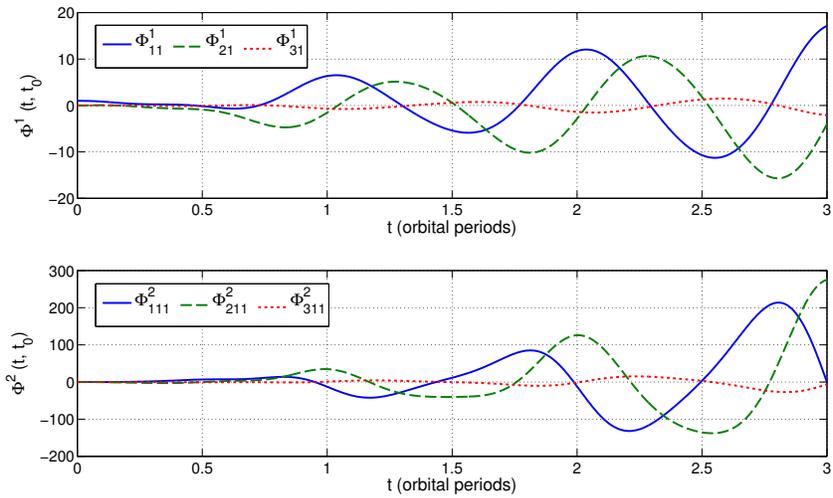


Figure 4.6: Time evolutions of representative elements of high-fidelity space object STMs.

For every subcase tested, the linear-algebra method is more efficient than either the CDM or multi-iteration method. The efficiency of the linear-algebra method is most pronounced when calculating both the first- and second-order STMs (Figure 4.7b). The reuse of the factored linear system matrix from the calculation of the first-order STM allows for speedups of two times or more

vs the other methods. This advantage holds regardless of whether the single- or double-integrator RK equations are used. None of the three methods benefits appreciably from parallelization due to the inexpensive dynamics. On the other hand, for both the first- and second-order STM calculation cases, the advantage of using the double-integrator form of the RK equations is clear.

High-Fidelity and Variable-Fidelity Dynamics The maximum relative difference between any two elements of $\Phi^1(t_f, t_0)$ or $\Phi^2(t_f, t_0)$ between any of the calculation methods is 4.12×10^{-11} . The similarity of this value to that of the low-fidelity dynamics case demonstrates that use of the variable-fidelity strategy does not adversely impact the precision of the state propagation or the STM calculation. The averaged CPU times required by the different methods of calculating the STMs are shown in the middle (high-fidelity) and lowermost (variable-fidelity) plots of Figures 4.7a and 4.7b. For this scenario, evaluations of the computationally expensive high-fidelity geopotential and its derivatives dominate propagation runtime. For all subcases, the linear-algebra and multi-iteration methods use the same number of \mathbf{f} , \mathbf{J} , and \mathbf{H} evaluations – which is fewer than the number used by the CDM. When calculating the first-order STM only (Figure 4.7a), the result is that the efficiencies of the linear-algebra method and multi-iteration method are nearly identical, and both significantly outperform the CDM. Use of variable-fidelity dynamics shrinks the efficiency gap because expensive dynamics are evaluated less frequently. When calculating both the first- and second-order STMs (Figure 4.7b), the linear-algebra method outperforms the multi-iteration method; the difference in CPU time is

more significant when variable-fidelity dynamics are used and/or when parallelization is used. The CDM takes greater advantage of parallelization than the multi-iteration method, to the extent that the runtimes of the two are similar for first- and second-order STM calculation with variable-fidelity dynamics. The linear-algebra method remains the most efficient option.

As parallelization has a significant impact on this test case, a more detailed look at the impact of the number of available threads is shown in Figure 4.8. As expected, the relative efficiency gains for all methods stagnate between four and seven threads for the eight-stage method because two loop iterations are still required to complete the function evaluations at all stages. For the same reason, the efficiency improves dramatically when the number of available threads is increased from seven to eight. When comparing the first-order STM calculation methods (Figure 4.8a), similar relative speedups are achieved by all methods when using a given number of threads. When the second-order STM is also calculated (Figure 4.8b), the efficiency gains exhibited by the multi-iteration method tend to lag those of the CDM and linear-algebra method. Two factors contribute to this result. First, the multiple iterative solves necessitate the creation of more individual parallel regions compared to the CDM, which results in more overhead. Second, even though the multi-iteration method uses the same number of parallel regions as the linear-algebra method, the multi-iteration method performs less work in parallel. Specifically, both methods identically use parallelization to calculate \mathbf{f} , \mathbf{J} , and \mathbf{H} ; however, the linear-algebra method parallelizes the entire operation

of calculating the second-order STM, while the multi-iteration method only parallelizes the STM update equations in the second fixed-point iteration loop.

4.4.3 Explicit Runge-Kutta Methods

While the primary emphasis of this chapter is IRK methods, the CDM and DDM are both applicable to ERK solvers, as well. Among the two DDM alternatives discussed in this study, only the linear-algebra method is relevant because the RK update equations are not solved iteratively for an explicit method. For the same reason, the primary benefit of the DDM for IRK solvers – the reduction of \mathbf{J} and \mathbf{H} evaluations – does not exist when using an ERK method. Thus, the CDM is generally more efficient than the DDM when calculating STMs with an ERK method.

The preferability of an ERK or IRK method is strongly problem-dependent. For nonstiff systems in which dynamics evaluations do not dominate runtime, a well-chosen ERK method is usually more efficient than an IRK method because of the iterative nature of the IRK solver [50]. However, an appropriate IRK method may be desirable due to attractive theoretical properties, such as A stability or symplecticity [88].¹¹ For more computationally intensive dynamical systems, IRK-specific customizations, such as variable-fidelity dynamics routines and parallelization, may also come into play. With these caveats noted, Figure 4.9 displays the results of applying an ERK method to the exam-

¹¹A Newton or approximate Newton scheme may be needed instead of fixed-point iteration to retain these properties (as well as computational efficiency) if the system is sufficiently stiff [88, 86].

Table 4.3: ERK implementation details.

	Oscillator	Orbit: J_2	Orbit: 70×70 sph. harm.
Method	Gill [50]	Dormand-Prince [166]	Dormand-Prince [166]
Stages	4	13	13
Order	4	8	8
Step size	0.015 time units	90 sec	40 sec

ple systems previously introduced for the serial calculation, single-integrator case. CPU times are normalized for each example system to produce a common scale. For the IRK orbit propagation results, the variable-fidelity CPU times are reported. ERK implementation details are given in Table 4.3. For all cases, the constant step size is selected such that the root-sum-square error between the test solution and a reference “truth” solution¹² is approximately equal for the ERK and IRK methods.

For the reasons discussed, the DDM performs less efficiently than the CDM for ERK methods, except for similar performance in the first-order STM calculation for the high-fidelity orbit propagation. Predictably, the efficiency of the IRK method improves relative to the ERK method when variable-fidelity dynamics models are used because the ERK method does not exploit this customization.

¹²The truth solution is generated by propagating the system using the IRK CDM with a step size 1/10th that of the test solution. Compensated summation is used in all cases to reduce roundoff error.

4.5 Conclusions

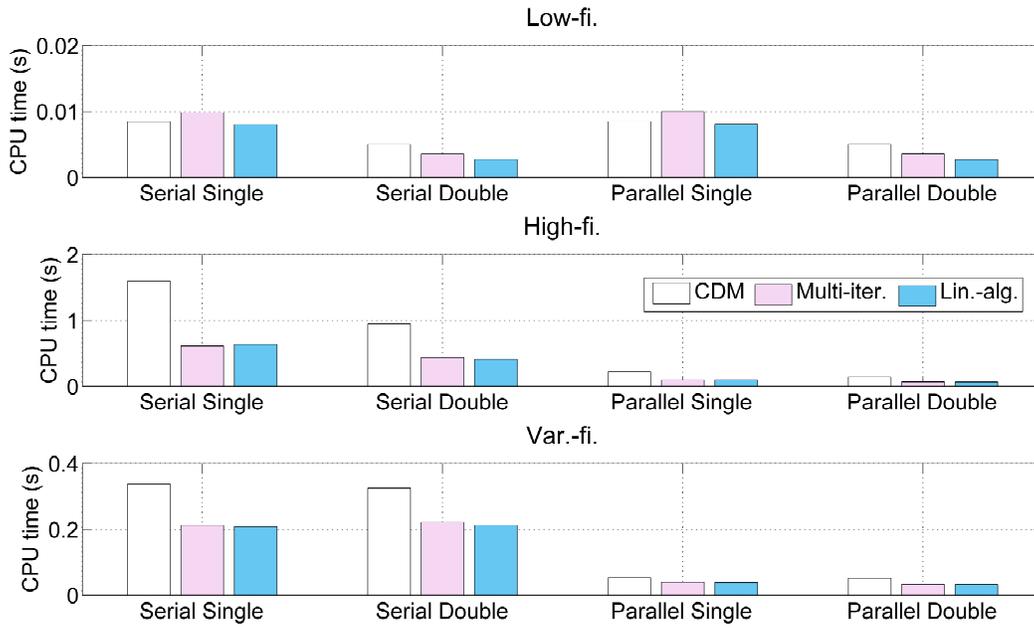
In this chapter, the decoupled direct method of calculating first- and second-order state transition matrices is derived for Runge-Kutta methods. Emphasis is placed on implicit Runge-Kutta methods because the decoupling of the state propagation and STM calculation reduces the number of Jacobian and Hessian evaluations compared to the common practice of propagating the variational equations using an augmented state vector (i.e., the coupled direct method). Two decoupling techniques are presented. The first, referenced as the “multi-iteration method,” solves for the state first, then uses an iterative technique to solve a separate system of equations for the STMs. The second, referenced as the “linear-algebra method,” directly differentiates the RK update equations to obtain a linear system for the STMs once the state is propagated. Previously introduced for calculation of the first-order STM, the linear-algebra method is extended here to accommodate calculation of the second-order STM. The system matrix whose factorization is required to solve for the second-order STM is shown to be identical to that used to calculate the first-order STM, which allows for reuse of the factored form. For both the first- and second-order STMs, the method is further extended for use with the double-integrator version of the RK equations, which, when applicable, produces a smaller linear system matrix for calculating STMs. Alone among the three methods discussed, the linear-algebra version of the DDM eliminates the need for an initial guess of the STMs at each internal stage of an IRK step because the resulting linear systems may be solved without iteration.

Owing to the reduction in Jacobian and Hessian evaluations, the performances of both decoupled methods improve relative to the CDM as the computational cost of the dynamics increases. For moderately expensive, non-stiff dynamics, and for a low-to-moderate number of internal stages per step (up to eight were used in this study), the linear-algebra method is likely to outperform both the CDM and multi-iteration method when fixed-point iteration is used to solve the RK update equations. Further, the comparative efficiency of the linear-algebra method improves if both the first- and second-order STMs are calculated. The DDM also facilitates a more efficient implementation of an approximate Newton method as an alternative to fixed-point iteration compared with the CDM.

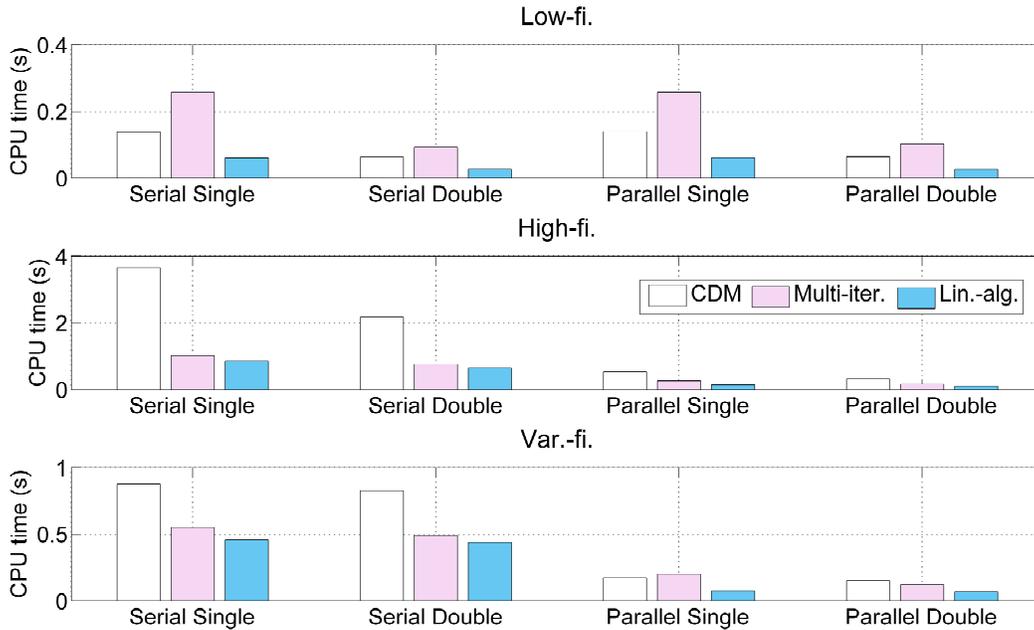
The numerical examples presented in this chapter confirm the utility of variable-fidelity dynamics models in the context of STM calculation using IRK methods for the artificial satellite problem: Significant increases in efficiency are achieved without sacrificing precision. This strategy is not strictly limited to SO propagation; however, it is cautioned that careful testing should accompany application to other systems.

The impact of parallelization at the internal-stage-evaluation level is examined for all three direct STM calculation methods using OpenMP. Overall, in a LEO trajectory propagation example – either using an inexpensive J_2 geopotential or an expensive degree/order-70 spherical harmonics geopotential – the linear-algebra method is found to provide the most efficient performance for nearly all permutations of the computational options tested (i.e., serial vs. parallel, single-integrator vs. double-integrator, and low-fidelity vs. variable-

fidelity vs. high-fidelity dynamics). On the other hand, it is shown that the linear-algebra method does not outperform the CDM when used with an ERK method because the number of Jacobian and Hessian evaluations is not reduced. Numerical results confirm the superiority of ERK methods for inexpensive, nonstiff problems, while the utility of IRK methods is shown for computationally expensive systems for which variable-fidelity dynamics and/or parallelization may be used effectively.

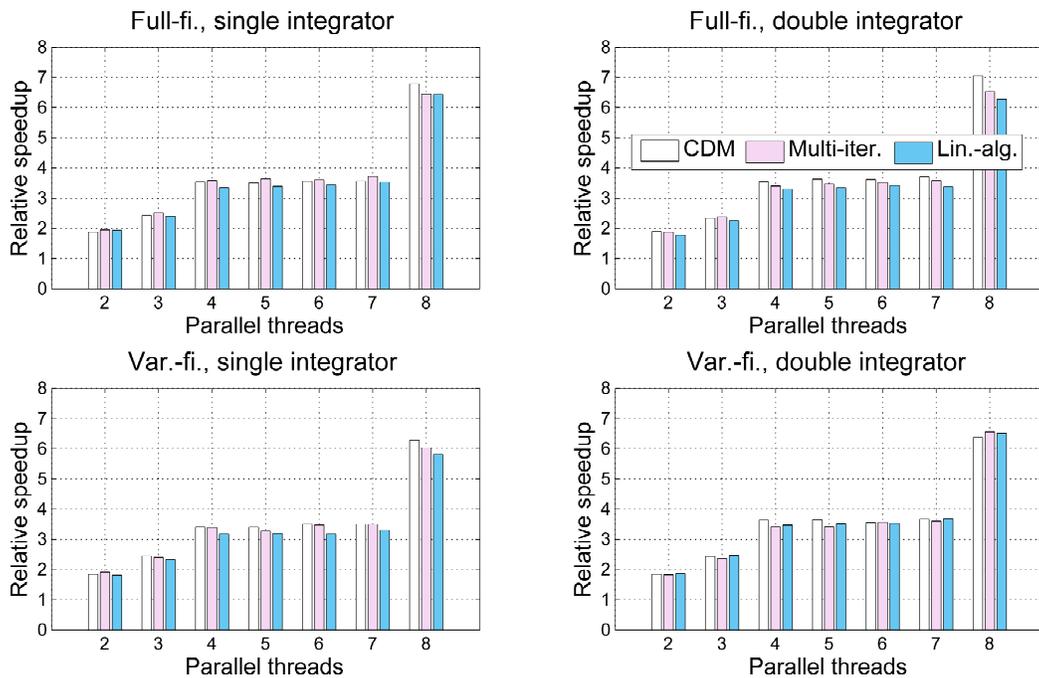


(a) First-order STM.

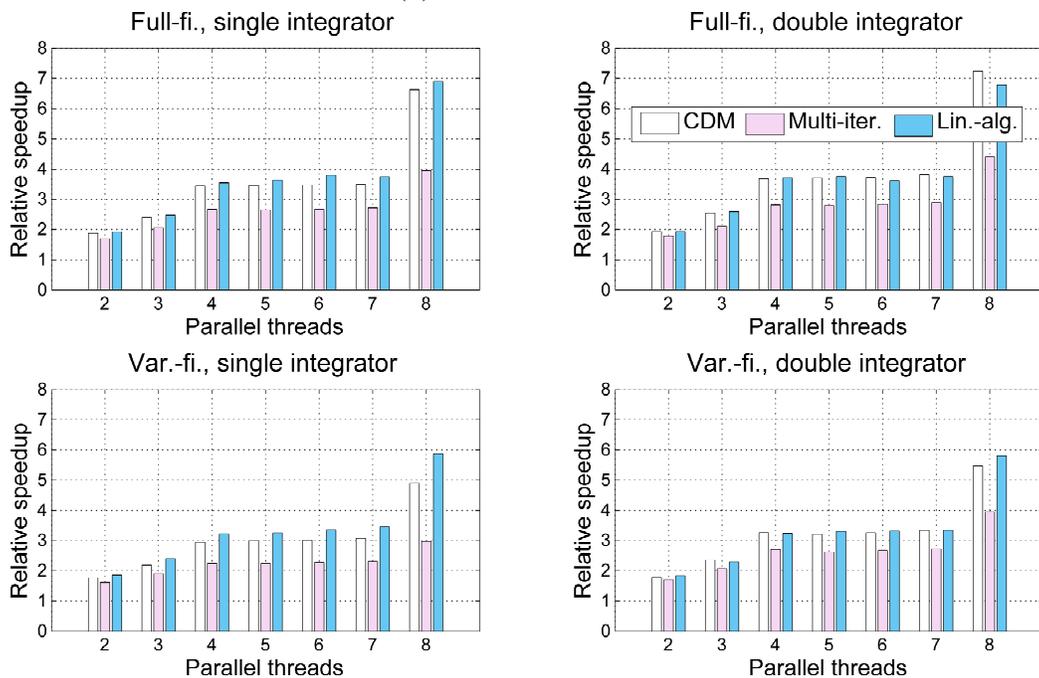


(b) First- and second-order STMs.

Figure 4.7: CPU time required to propagate state and calculate STMs for space object in LEO.

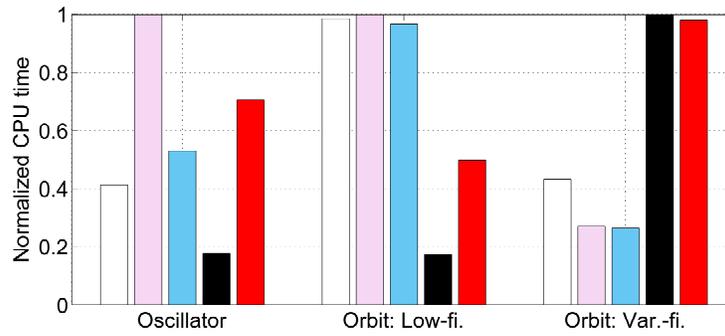


(a) First-order STM.

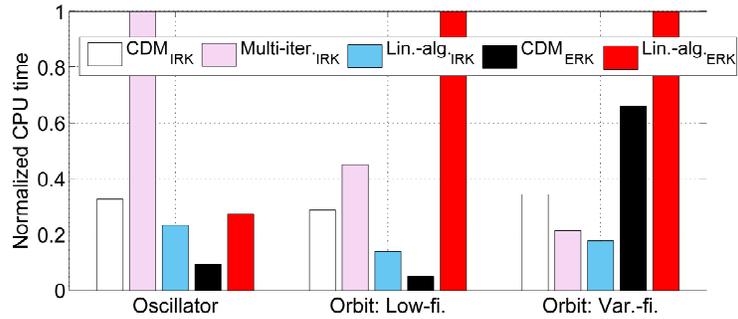


(b) First- and second-order STMs.

Figure 4.8: Multiplicative speed increase vs. serial calculation as function of number of available parallel threads. 158



(a) First-order STM.



(b) First- and second-order STMs.

Figure 4.9: Normalized CPU time required to propagate state and calculate STMs for example scenarios using the single-integrator formulation of the RK equations and serial computation. Normalization is performed individually for each scenario.

Chapter 5

The Eccentric Case of a Fast-Rotating, Gravity-Gradient-Perturbed Space Object Attitude Solution¹

5.1 Introduction

In this chapter, a closed-form perturbation solution for the attitude evolution of an SO under certain assumptions is developed. In part, this work serves as a prelude to Chapter 6, in which the attitude solution is used in conjunction with a hybrid SP/GP algorithm for 6DOF SO state prediction (Section 1.4). At the same time, the closed-form rotational solution may be used even outside the hybrid framework. For example, as the number of SOs – particularly in LEO – continues to grow, active debris removal is increasingly viewed as an important component of any long-term strategy for avoiding the unchecked proliferation of undesirable SOs [43, 139]. Architectures for removal missions typically focus on de-orbiting large SOs, such as derelict rocket bodies, because a collision involving a large SO would result in the creation of more

¹Work from this chapter has been accepted for presentation as:

- **Noble Hatten** and Ryan P. Russell. The eccentric case of a fast-rotating, gravity-gradient-perturbed satellite attitude solution. Paper 17-373. In *27th AAS/AIAA Space Flight Mechanics Meeting*, San Antonio, TX, February 2017. (accepted for presentation)

Ryan P. Russell supervised the work.

new debris than a collision involving only smaller SOs. While specific removal tactics vary, any generic methodology would likely benefit from knowledge of a target SO's attitude [177, 157, 203]. However, the computational expense of 6DOF state prediction complicates both the estimation and prediction of SO attitude [65, 202, 138]. Perturbation techniques offer an alternative that can improve efficiency at the cost of (ideally) small losses in fidelity [26]. While a number of such methods have been produced (Section 1.2.4.4), the work in this chapter focuses on extending and analyzing a specific solution procedure that produces promising results for large tumbling bodies in LEO.

In a series of works, Lara and Ferrer used the Lie-Deprit method [52] to produce an approximate, closed-form solution for the attitude evolution of a fast-rotating, triaxial SO subject to gravity-gradient torque [134, 133, 132]. The Lara-Ferrer solution procedure is simplified by restricting the SO to a circular orbit; here, the solution is rederived for an elliptical orbit. The resulting explicit time dependence of the Hamiltonian is eliminated through use of the extended phase space [59], and no expansions in powers of the eccentricity are required. The numerical example originally presented by Lara and Ferrer [134] is also modified for the case of eccentric orbits to verify the new equations and analyze the behavior of the solution. Further, several additional numerical studies are performed to provide practitioners with a more realistic idea of the types of scenarios in which the solution may be most fruitfully applied:

- The small parameter of the perturbation procedure is varied to determine solution degradation as the parameter grows (and thus begins to violate the assumptions of the solution). As the parameter is a function of (1) the SO orbit, (2) the SO's rotational angular momentum, and (3) the SO's physical properties, the analysis establishes guidelines for the physical and dynamic regimes in which the solution is appropriate.
- The closed-form solution is compared to a more realistic gravity-gradient-based reference solution than was used in the original Lara-Ferrer publications. This new comparison reveals additional secular error sources not addressed in the published numerical example that may, in fact, dominate the errors introduced by the Lie-Deprit transform procedure.
- The elliptical orbit solution is applied to a scenario based on the physical and dynamic properties of a specific tumbling rocket body in LEO. This example is representative of a class of SOs whose attitude evolution is of interest and, as is demonstrated in the example, whose characteristics meet the assumptions of the closed-form solution. As part of the presentation, the explicit transformation equations required to apply the closed-form solution to an SO spinning about its axis of minimum inertia are presented. (The original presentation for circular orbits considers only spin about the axis of maximum inertia [134].)

To maintain consistency, the nomenclature of Lara and Ferrer [134] is adopted when possible. Equations that are repetitions of those presented in Lara and Ferrer [134] are given in Appendix D in the interest of brevity in

the main text. It is acknowledged that gravity-gradient torque is generally not the only relevant torque acting on an SO. However, depending on orbit and physical properties, the gravity-gradient torque may be the dominant perturbation to torque-free motion [134]. In the Earth-orbit regime, solar radiation pressure, aerodynamic drag, geomagnetic, and other torques may also be important. Investigation of the effect of these torques on the accuracy of the gravity-gradient-based, closed-form solution or the inclusion of additional torques in the perturbation solution is beyond the scope of the current study.

5.2 Solution Procedure

The original perturbation method uses the torque-free motion of a triaxial rigid body (i.e., the Euler-Poinsot problem) as the reference solution. The unperturbed Hamiltonian, though conveniently written in Andoyer variables [94, 45], is canonically transformed to action-angle variables (the so-called Lara-Ferrer solution variables), in which the unperturbed Hamiltonian is a function of momenta only. The disturbing potential of the gravity-gradient torque is added to the Hamiltonian, and it is assumed that the torque is adequately described by the point-mass gravity term of the primary body. For the purposes of translational motion, both the primary body and the SO are assumed to act as point masses, which allows for the elimination of the translational term from the Hamiltonian. The SO is assumed to be in a circular orbit about the primary. The Hamiltonian is simplified via the “fast-rotating” SO assumption, which requires the rate of variation of the Andoyer angle μ to be suitably faster than the frequencies of both the Andoyer angle ν and the orbital

motion of the SO. (The Andoyer angles are depicted in Figure 5.1.²) The perturbed Hamiltonian is reduced to a function of momenta only, and the secular rates of the angles are obtained. The reduction is accomplished through two successive canonical transformations, each obtained via the Lie-Deprit procedure [52], which eliminate angles from the Hamiltonian through averaging [26]. The equations of the Lie-Deprit method are given in Appendix C.

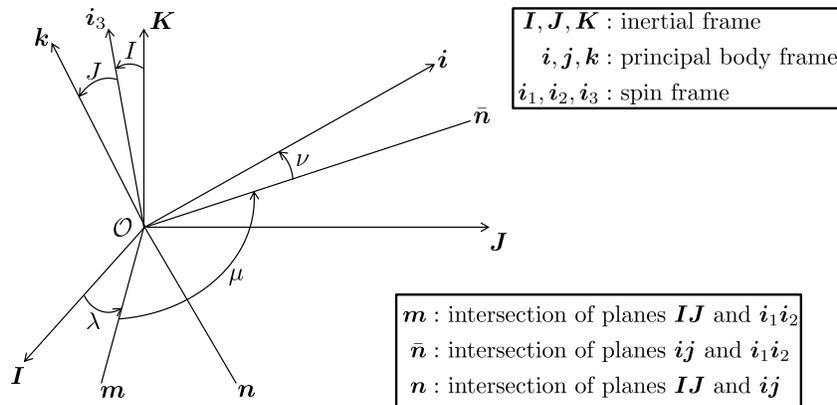


Figure 5.1: Andoyer variables; adapted from Celletti [45].

5.2.1 Extension to Elliptical Orbits

The Hamiltonian is a function of r , which, in the circular orbit case, is constant. To handle the elliptic case, the problem is moved to the extended phase space [59]. A timelike variable τ and its conjugate momentum T are added to the variable set, and the Hamiltonian is augmented such that

²The angle μ defines the orientation of the i_1i_2 plane. The angle ν defines the orientation of the ij plane. Note that i_3 is parallel to the rotational angular momentum.

$$\mathcal{H} = \Phi + U + T \tag{5.1}$$

$$\frac{d\tau}{dt} = \frac{\partial \mathcal{H}}{\partial T} = 1. \tag{5.2}$$

When necessary, r is expressed as a function of τ rather than t . Additionally, the variable $\phi = h - \theta$ is replaced by h itself³, and the problem is not moved to a rotating frame, as is done in the original formulation. The potential U is the potential for the circular case (Eq. (D.9)) scaled by $(a/r)^3$.

The Lie-Deprit procedure commences by assuming that T is of first order and U is of second order relative to Φ . As is demonstrated most clearly using Andoyer variables (Eq. (D.7)), this assumption is based on $U = \frac{1}{2}\epsilon^2\hat{U}$, where $\epsilon = n/(M/C)$ and \hat{U} is $\mathcal{O}(1)$ relative to Φ for the circular orbit case. (Here, n is the orbital mean motion.) The impact of scaling the circular gravity-gradient potential by $(a/r)^3$ for the elliptical case is discussed after the transformation equations are presented.

The first canonical transformation, which averages over the solution angle l , is nearly identical to that of the circular case because l is related to the orientation of the body and is not affected by the eccentricity of the orbit. The $(a/r)^3$ scaling is applied to the averaged term $\mathcal{H}_{0,2}$, the generating function term W_2 , and the transformation expressions (Eqs. (D.12), (D.14), and (D.17)–(D.22), respectively). The timelike variable τ is unaffected by the

³The conjugate momenta of ϕ and h are identical.

transformation because the generating function is independent of T . The first averaging results in the variable transformation $\xi \rightarrow \xi'$.

The second transformation ($\xi' \rightarrow \xi''$), which averages over one period in τ , is described in more detail. When appropriate, references are made to corresponding equations in Appendices C and D. The singly averaged Hamiltonian \mathcal{K} is given by (Eq. (C.1))

$$\mathcal{K} = \mathcal{K}_{0,0} + \epsilon \mathcal{K}_{1,0} + \frac{\epsilon^2}{2} \mathcal{K}_{2,0}, \quad (5.3)$$

where

$$\mathcal{K}_{0,0} = \frac{G^2}{2A} \left(1 - \frac{C-A}{C} \frac{f}{f+m} \right) \quad (5.4)$$

$$\mathcal{K}_{1,0} = T \quad (5.5)$$

$$\mathcal{K}_{2,0} = 2 \left(\frac{a}{r} \right)^3 \frac{n^2}{8} \kappa [1 - 3s_I^2 \sin^2(h - \theta)]. \quad (5.6)$$

$\mathcal{K}_{0,0}$ and $\mathcal{K}_{2,0}$ are based on Eqs. (D.4) and (D.12), respectively, while $\mathcal{K}_{1,0}$ arises due to the use of the extended phase space. The procedure outlined in Appendix C is followed to derive the transformation. The first homological equation is (Eq. (C.16))

$$\mathcal{K}_{0,1} = \mathcal{K}_{1,0} - \frac{\mathcal{D}V_1}{\mathcal{D}t}, \quad (5.7)$$

where (Eqs. (C.5)–(C.8))

$$\frac{\mathcal{D}V_1}{\mathcal{D}t} = \frac{\partial V_1}{\partial t} - L'_1 \mathcal{K}_{0,0}. \quad (5.8)$$

The use of the extended phase space removes any explicit dependency of the system on t , so $\frac{\partial V_1}{\partial t} = 0$. In addition, $\mathcal{K}_{0,0}$ is in a fully reduced form and is independent of all coordinates. Further, because h and τ are the only coordinates in \mathcal{K} , it may be assumed that the only coordinates on which V depends are h and τ , as well. Combining these observations with the fact that $\mathcal{K}_{0,0}$ is independent of the momenta conjugate to h and τ , it is seen that $L'_1 \mathcal{K}_{0,0}$ reduces to zero. Thus, $\mathcal{K}_{0,1} = \mathcal{K}_{1,0}$.

The second homological equation is (Eq. (C.17))

$$\mathcal{K}_{0,2} = \mathcal{K}_{2,0} + L'_1 \mathcal{K}_{1,0} + L'_1 \mathcal{K}_{0,1} - \frac{\mathcal{D}V_2}{\mathcal{D}t}. \quad (5.9)$$

Taking into account $\mathcal{K}_{0,1} = \mathcal{K}_{1,0}$, which is only a function of T , and setting $V_2 = 0$, Eq. (5.9) reduces to⁴

$$\mathcal{K}_{0,2} = \mathcal{K}_{2,0} - 2 \frac{\partial V_1}{\partial \tau}. \quad (5.10)$$

$\mathcal{K}_{0,2}$ is chosen as the average of $\mathcal{K}_{2,0}$ over one period in τ , which is equivalent to averaging over one period in the mean anomaly M_0 :

⁴In the development of the equations of the second averaging, all solution variables are doubly transformed. Following the convention of Lara and Ferrer [134], the double-prime notation is not used explicitly to improve readability.

$$\mathcal{K}_{0,2} = \langle \mathcal{K}_{2,0} \rangle_\tau = \frac{1}{T_p} \int_0^{T_p} \mathcal{K}_{2,0} d\tau = \frac{1}{2\pi n} \int_0^{2\pi} \mathcal{K}_{2,0} dM_0 \quad (5.11)$$

$$= \frac{n^2 \kappa (3c_I^2 - 1)}{4(1 - e^2)^{3/2}}, \quad (5.12)$$

where κ is defined in Eq. (D.13) and is a function of momenta and body properties only. The closed-form integration for $\mathcal{K}_{2,0}$ is facilitated by the relation $a^2 \sqrt{1 - e^2} dM_0 = r^2 d\theta$, where θ is the true anomaly. Solving Eq. (5.9) then gives the generating function

$$V_1 = \frac{n\kappa}{16(1 - e^2)^{3/2}} \left\{ (4 - 6s_I^2) (\theta - M_0 + e \sin \theta) + s_I^2 [3 \sin(2\theta - 2h) + 3e \sin(\theta - 2h) + e \sin(3\theta - 2h)] \right\}, \quad (5.13)$$

where, once again, the differential relation between M_0 and θ is used to perform the quadrature. Eq. (5.13) is similar in form to the generating function term for the third canonical transformation of Lara et al. [137]. It is also emphasized that M_0 and θ are functions of τ . The transformation equations of the second averaging are obtained via Eqs. (C.9)–(C.21), and result in $\boldsymbol{\xi}' = \boldsymbol{\xi}'' + \delta\boldsymbol{\xi}$, with the individual transformations given by⁵

⁵In addition to accounting for orbit eccentricity, Eq. (5.14) corrects a typo in Eq. (86) of Lara and Ferrer [134].

$$\delta l = -\frac{1}{(1-e^2)^{3/2}} \frac{3}{16} \frac{n}{G}. \quad (5.14)$$

$$\left\{ \frac{(C-B)^2 A}{BC} \frac{\pi}{2K(m)} \frac{\sqrt{f(1+f)}}{m\sqrt{f+m}} \right. \\ \left. \left[\frac{f+m}{1-m} \frac{E^2(m)}{K^2(m)} - 2f \frac{E(m)}{K(m)} + f + \frac{C+B}{C-B} m \right] \right\} \\ \{ (4-6s_I^2) (\theta - M_0 + e \sin \theta) + \\ s_I^2 [3 \sin(2\theta - 2h) + 3e \sin(\theta - 2h) + e \sin(3\theta - 2h)] \} \\ \delta h = \frac{1}{(1-e^2)^{3/2}} \frac{3}{8} \frac{n}{G} \frac{H}{G} \kappa. \quad (5.15)$$

$$\left[2(\theta - M_0 + e \sin \theta) - \sin(2\theta - 2h) - e \sin(\theta - 2h) - \frac{1}{3} e \sin(3\theta - 2h) \right] \\ \delta g = -\frac{H}{G} \delta h - \frac{L}{G} \delta l \quad (5.16)$$

$$\delta \tau = 0 \quad (5.17)$$

$$\delta L = 0 \quad (5.18)$$

$$\delta G = 0 \quad (5.19)$$

$$\delta H = \frac{1}{(1-e^2)^{3/2}} \frac{n}{8} \kappa s_I^2 [3 \cos(2\theta - 2h) + 3e \cos(\theta - 2h) + e \cos(3\theta - 2h)]. \quad (5.20)$$

The variable f is defined in Eq. (D.1), while m is defined in Eqs. (D.2) and (D.6); both are functions of momenta and body properties only. The inverse transformations are obtained by applying Eqs. (C.22)–(C.27). The doubly averaged Hamiltonian is (Eq. (C.2))

$$S = \Phi + T - \frac{n^2 \kappa}{8(1-e^2)^{3/2}} \left(1 - \frac{H^2}{G^2}\right). \quad (5.21)$$

The second averaging eliminates both τ and h from S , so S is a function of momenta only, and no additional transformations are required to obtain a fully reduced Hamiltonian. The secular rates of the coordinates in doubly transformed variables are

$$\begin{aligned} \frac{dl}{dt} = & \frac{G^2}{2A} \left(\frac{C-A}{C}\right) \frac{f}{(f+m)^2} \frac{\partial m}{\partial L} - \\ & \frac{3n^2}{8(1-e^2)^{3/2}} (B-A) \left(1 - 3\frac{H^2}{G^2}\right) \frac{1+f}{(f+m)^2} \\ & \left\{ 1 + \frac{C-B}{B} \left[\frac{f+m}{2m} - \frac{f}{m} \frac{E(m)}{K(m)} + \frac{1}{2m} \frac{f+m}{1-m} \frac{E^2(m)}{K^2(m)} \right] \right\} \frac{\partial m}{\partial L} \end{aligned} \quad (5.22)$$

$$\frac{dh}{dt} = \frac{H}{(1-e^2)^{3/2}} \frac{3n^2}{4G^2} \kappa \quad (5.23)$$

$$\frac{dg}{dt} = \frac{2\Phi}{G} - \frac{H}{G} \frac{dh}{dt} - \frac{L}{G} \frac{dl}{dt}, \quad (5.24)$$

where $\partial m/\partial L$ is given by Eq. (D.5). All the elliptic equations reduce to the circular versions if $e = 0$. The expressions are valid for all $e < 1$, though numerical instability is possible for $e \approx 1$ due to small divisors of the form $(1-e^2)^{3/2}$. At large eccentricities, the growth of this divisor can also cause a violation of the assumption that the gravity-gradient torque term in Eq. (5.21) is second-order small compared to Φ . However, the likelihood of this occurrence is diminished by the presence of $n^2 = \mu_*/a^3$ in the numerator of the gravity-gradient torque term. For an SO in Earth orbit, eccentricity cannot

be arbitrarily large without a suitably large semimajor axis (and a suitably small mean motion) if the SO is to avoid intercepting the Earth.

Due to the scaling of the gravity-gradient potential, the second-order small parameter for the elliptical case is effectively

$$\epsilon_{ell}^2 = \epsilon^2 \left(\frac{a}{r}\right)^3 \quad (5.25)$$

$$= \left(\frac{n}{M/C}\right)^2 \left(\frac{a}{r}\right)^3 \quad (5.26)$$

$$= \left(\frac{1}{M/C}\right)^2 \frac{\mu_*}{r^3}. \quad (5.27)$$

Thus, ϵ_{ell}^2 is largest when the SO is near periapsis, and errors related to the averaging of the orbital motion are largest near periapsis, as well. The largest value taken by ϵ_{ell}^2 is the same as ϵ^2 for a circular orbit whose semimajor axis is equal to the periapsis distance of the elliptical orbit. Intuitively, this result corresponds to the fact that any gravitational effect dissipates as the distance between the SO and the central body increases.

5.3 Numerical Studies

A set of scenarios is simulated numerically to assess the closed-form solution. In all scenarios, the SO is subject to the central-body gravitational force of the Earth and gravity-gradient torque only. The impact of the extension to elliptic orbits is explored by performing a simulation over a range of eccentricities. Next, the small parameter ϵ is varied to determine the range of values over which the closed-form solution is effective. For each of these

first two scenarios, the parameter values and comparison “truth” solution are intentionally made as close as possible to those presented in Lara and Ferrer [134] to directly compare with previously presented results. In a third scenario, the closed-form solution is compared to a truth solution that is freed from the fast-rotating SO assumptions used by Lara and Ferrer [134]. Finally, the closed-form solution is applied to a tumbling rocket body in LEO, demonstrating the relevance to active space debris removal architectures.

5.3.1 Parameter Sweep: Eccentricity

To demonstrate the effects of eccentricity on the evolution of the analytical solution, the numerical example of Lara and Ferrer [134] is run several times. The periapsis distance is held constant ($r_p \approx 6995$ km), while e is varied from 0 to 0.8. All simulations begin at periapsis and encompass seven orbital periods. The initial rotational state and SO physical properties are the same for all simulations.

Figure 5.2 demonstrates how ϵ_{ell}^2 varies as an SO progresses in its orbit for a moderately eccentric orbit (Figure 5.2a) and a highly eccentric orbit (Figure 5.2b). Figure 5.3 displays the maximum and minimum values of ϵ_{ell}^2 for a range of eccentricities.⁶ These plots confirm the discussion surrounding Eq. (5.27).

The rotational state evolutions are compared with a reference state generated by numerically integrating the equations of motion that arise from

⁶Note logarithmic scale of vertical axis of Figure 5.3.

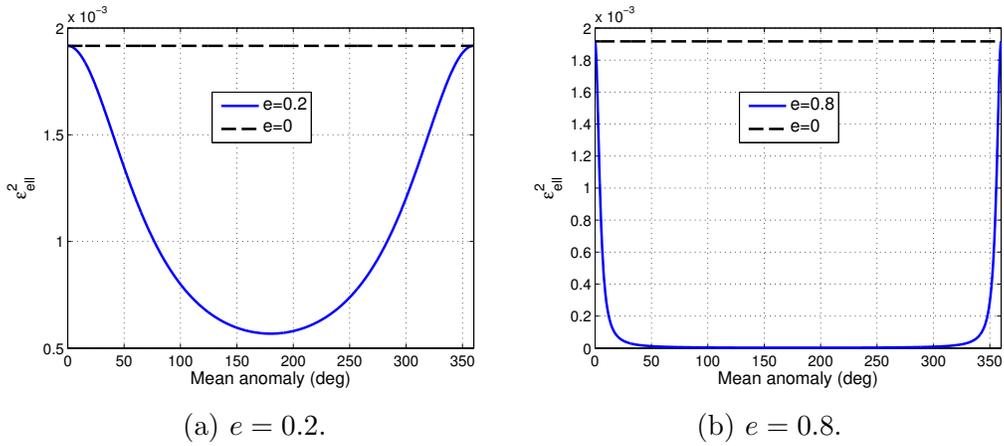


Figure 5.2: Square of small parameter of perturbation procedure for elliptical orbit as function of mean anomaly.

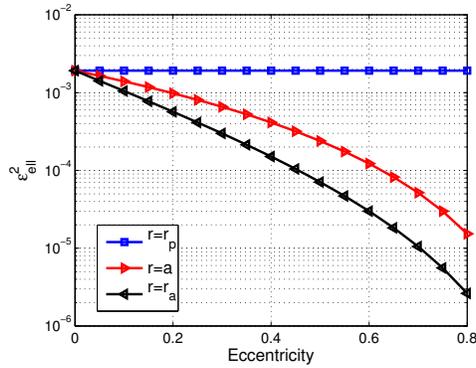


Figure 5.3: Square of small parameter of perturbation procedure for elliptical orbit evaluated at $r = r_p$, $r = a$, and $r = r_a$ as function of eccentricity.

the Hamiltonian specific to a fast-rotating SO (Eq. (D.11)) in Lara and Ferrer [134], modified to handle elliptic orbits). This truth state reflects perfect applicability of the fast-rotating SO assumptions, and therefore is not equivalent to a high-fidelity propagation of the rotational motion (even if torques other than gravity-gradient are ignored) [161]. Nevertheless, this truth state is used to preserve compatibility with the original results presented by Lara and Fer-

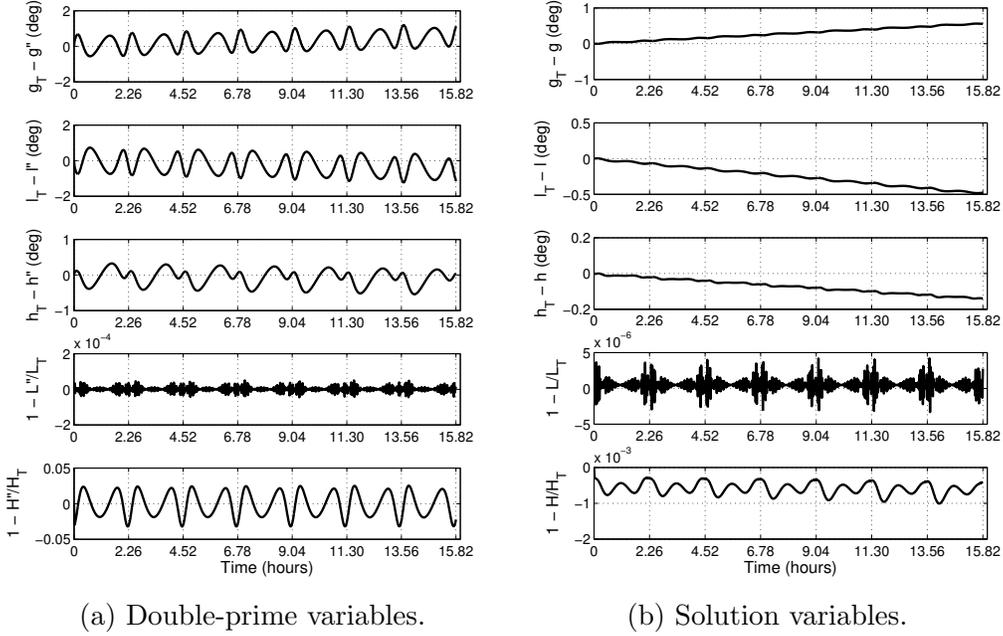


Figure 5.4: Differences between analytically obtained perturbation solution and numerically integrated fast-rotating truth solution (subscript T denotes truth); $e = 0.2$.

rer [134, 133, 132]. Similarly, in Figures 5.4 and 5.5, differences between the averaged solutions and the truth solution are presented in terms of Lara-Ferrer solution variables, which facilitates direct comparison with Figures 3 and 4 in Lara and Ferrer [134].⁷ It is noted that each horizontal axis tick in the time evolution plots in this chapter corresponds to one orbital period.

⁷The results presented in the current work are obtained using explicit perturbation transformations for both forward and backward transformations. For example, $\xi' = \xi'' + \delta\xi$, with $\delta\xi$ expressed in terms of double-prime variables, and $\xi'' = \xi' - \delta\xi$, with $\delta\xi$ expressed in terms of single-prime variables. (See Appendix C for more details on Lie-Deprit transformation equations.) The results presented by Lara and Ferrer [134] were obtained using only the $\xi'' \rightarrow \xi'$ and $\xi' \rightarrow \xi$ transformation equations. In those works, the inverse transformations were performed by solving the forward transformation equations implicitly. The differences between the explicit and implicit solutions are insignificant at the scale of Figures 5.4 and 5.5.

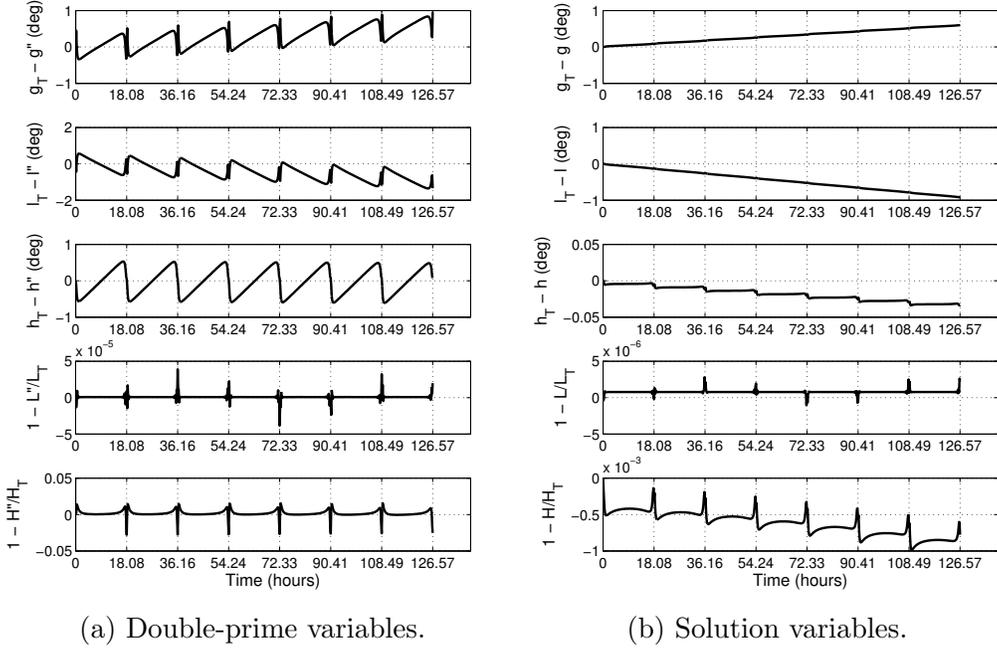


Figure 5.5: Differences between analytically obtained perturbation solution and numerically integrated fast-rotating truth solution (subscript T denotes truth); $e = 0.8$.

Unlike the circular case, in which the frequencies of periodic errors related to the averaging of the orbital motion are constant throughout the orbit, eccentric orbits are characterized by rapid changes in the elements of ξ'' near periapsis (Figures 5.4a and 5.5a). This trend is especially visible for the highly eccentric orbit presented in Figure 5.5. As in the circular-orbit case, transforming from ξ'' to ξ eliminates much of the periodicity in the errors, and improves the accuracy of the momenta L and G by one or more orders of magnitude (Figures 5.4b and 5.5b). Despite the changes in frequencies, the secular growth rates of the errors are similar to those of the circular case, as are the amplitudes of periodic errors. This result is quantified in Figure 5.6, which

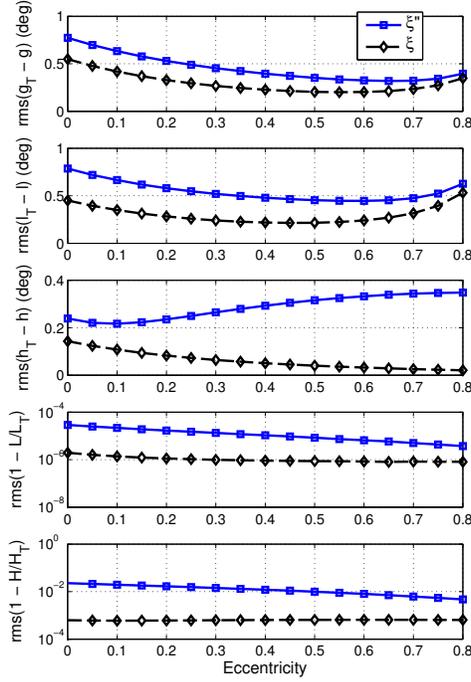


Figure 5.6: Root-mean-square of differences between analytically obtained perturbation solutions and numerically integrated fast-rotating solutions.

gives the RMS errors of the elements of ξ and ξ'' over seven orbital periods.⁸ The modest decreases in the RMS errors as e increases are attributable to the decrease in ϵ_{ell}^2 near apoapsis for increasingly eccentric orbits.

5.3.2 Parameter Sweep: Nominal ϵ

Regardless of the eccentricity of the orbit, the parameter ϵ must be suitably small for the perturbation method to produce a reasonable solution. As ϵ is directly proportional to n , the effects of varying ϵ are explored by executing the example scenario of Lara and Ferrer [134] over a range of n ,

⁸Note logarithmic scale of vertical axes for L and H plots.

which is equivalent to varying semimajor axis ($n = \sqrt{\mu_*/a^3}$).⁹ Figure 5.7 shows the variation of a as ϵ^2 progresses from 10^{-5} to 10^{-1} .¹⁰ (The nominal value used in the original example is $\epsilon^2 = 1.917 \times 10^{-3}$, and is represented by a star in Figure 5.7.) Of course, values of a less than the radius of the Earth are not feasible; in practice, such values of ϵ^2 would occur due to different SO rotational and/or physical properties.

Figure 5.8 displays the RMS differences between the closed-form solutions and the numerically integrated fast-rotating solutions over a seven-orbit time span.¹¹ The improvement in agreement between the closed-form and reference solutions as ϵ decreases is clear. For $\epsilon^2 = 10^{-1}$, for example, RMS differences over seven orbits approach 100 deg in the angles g and l , while, for $\epsilon^2 \leq 10^{-3}$, RMS differences are less than 1 deg. The differences in the RMS values achieved by the double-prime variables and the fully transformed variables increase (in a logarithmic scale) as ϵ decreases because, for large values of ϵ , the differences between both versions of the closed-form solution and the reference solution overshadow the differences between the two versions of the closed-form solution. Overall, the RMS error of the closed-form solution in each variable increases approximately exponentially with ϵ^2 (i.e., $\text{RMS}(\xi_i) \approx c_i(\epsilon^2)^{k_i}$). The fully transformed solution exhibits a lower RMS floor (smaller c_i) and faster RMS growth rate (larger k_i) than the double-prime solution, until the fully transformed and double-prime values nearly converge at large ϵ .

⁹Eccentricity is zero in these scenarios.

¹⁰Note logarithmic scale of horizontal axis.

¹¹Note logarithmic scales of horizontal and vertical axes.

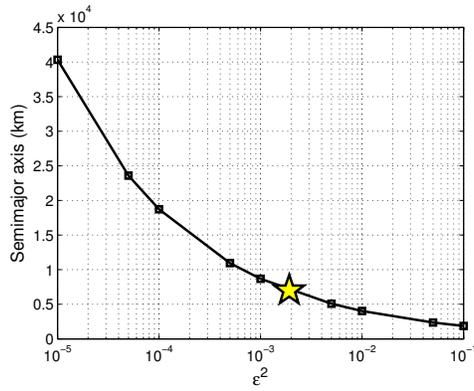


Figure 5.7: Semimajor axis as function of ϵ^2 , assuming constant SO rotational and physical properties.

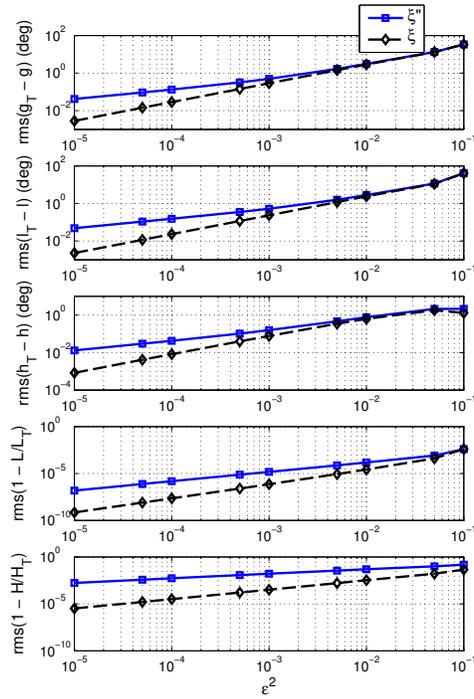


Figure 5.8: Root-mean-square of differences between analytically obtained perturbation solutions and numerically integrated solutions as functions of ϵ^2 .

5.3.3 Comparison with Unaveraged Gravity-Gradient Torque

To maintain consistency with the results presented in Lara and Ferrer [134], the comparisons presented in Sections 5.3.1 and 5.3.2 use, as truth, Hamilton’s equations of motion derived from Eq. (D.7). While this methodology is appropriate for assessing the impact of the two Lie transformations, the applicability of the fast-rotating SO assumption to a given problem is not addressed. To give an indication of the effect of the assumptions of the fast-rotating model, the system is propagated again using the Eulerian equations of motion with unaveraged gravity-gradient torque [49]. Using a standard linearization based on the assumption that the dimensions of the SO are much smaller than the distance between the centers of mass of the SO and the central body, the torque may be written as [49]

$$T_{GG} = \frac{3\mu_*}{r^5} \mathbf{r}_B \times (\mathbf{J}_0 \mathbf{r}_B). \quad (5.28)$$

The equations of motion are formulated by expressing the attitude of the SO relative to the inertial orbital frame with a quaternion and the angular velocity vector, expressed in a body-fixed frame [74]. (See Section 3.2.)

Figures 5.9a and 5.9b duplicate Figures 3 and 4 of Lara and Ferrer [134], respectively, except for the difference in the reference solution. Figure 5.10 does likewise for Figure 5.4 of the present work. The consequences of the initial averaging over the Andoyer angle μ – performed prior to the Lie transformations – are shown clearly. Errors in the angle h and its conjugate momentum H are similar in magnitude to those observed when the

fast-rotating Hamiltonian is used to generate the reference model. However, the angles g and l experience much faster secular error growth rates, and the amplitude of errors in the momentum L is larger by multiple orders of magnitude (though still accurate to better than 1 percent over seven orbital periods). Through comparison of Figures 5.9 and 5.10, it may be observed that the secular error growth rate is not dependent only on the orbital period, as was the case when the fast-rotating Hamiltonian was used to generate the reference model. Instead, the rotational frequencies of the SO body are important, as well. Consequently, the errors in g and l for the eccentric orbit are greater than the corresponding errors in the circular orbit after each has completed seven orbital periods because the orbital period of the eccentric orbit is more than 30 minutes longer than that of the circular orbit. Thus, more SO body revolutions occur for the eccentric case than for the circular case over the time frames considered. This trend is displayed in Figure 5.11, which duplicates Figure 5.6 using the truth model based on Eq. (5.28). The differences between the double-prime and solution variables may be difficult to distinguish in the scale of the plot. The more important result is that the differences created by the use of the fast-rotating Hamiltonian dwarf those created by the Lie transformations. Thus, for the closed-form solution to reasonably approximate the motion, care must be taken to ensure that the assumptions of the fast-rotating starting model are adequately met: The frequency of the Andoyer variable μ must be much greater than both the orbital frequency and the frequency of the Andoyer variable ν .

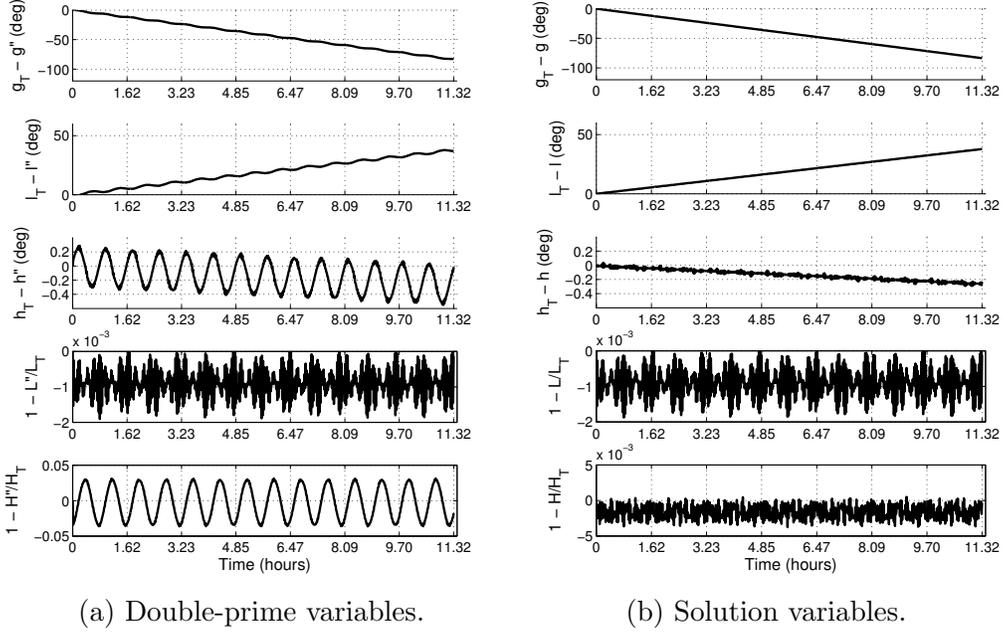


Figure 5.9: Differences between analytically obtained perturbation solution and numerically integrated truth solution (subscript T denotes truth); $e = 0$.

To quote Celletti [45], “the angle $[\mu]$ provides the motion of the equatorial axis [of the body frame (i.e., the $\mathbf{i}\mathbf{j}$ axis in Figure 5.1)] with respect to the inertial frame ... [and] the angle $[\nu]$ gives the motion of the angular momentum with respect to the body frame.” Thus, the body must be spinning at a significantly faster rate about its 3 axis than the angular momentum vector is precessing relative to the body. In the example of Lara and Ferrer [134], the initial conditions correspond to $\boldsymbol{\omega} = [0.7909, 0.1570, 1.3909]^T$ deg/s. While ω_3 is the largest component of $\boldsymbol{\omega}$, it is not particularly dominant: $|\omega_3| < 2|\omega_1|$.

In Figure 5.12, RMS error results are presented for the circular-orbit case in which the initial ω_3 is varied from a minimum, equal to the nominal

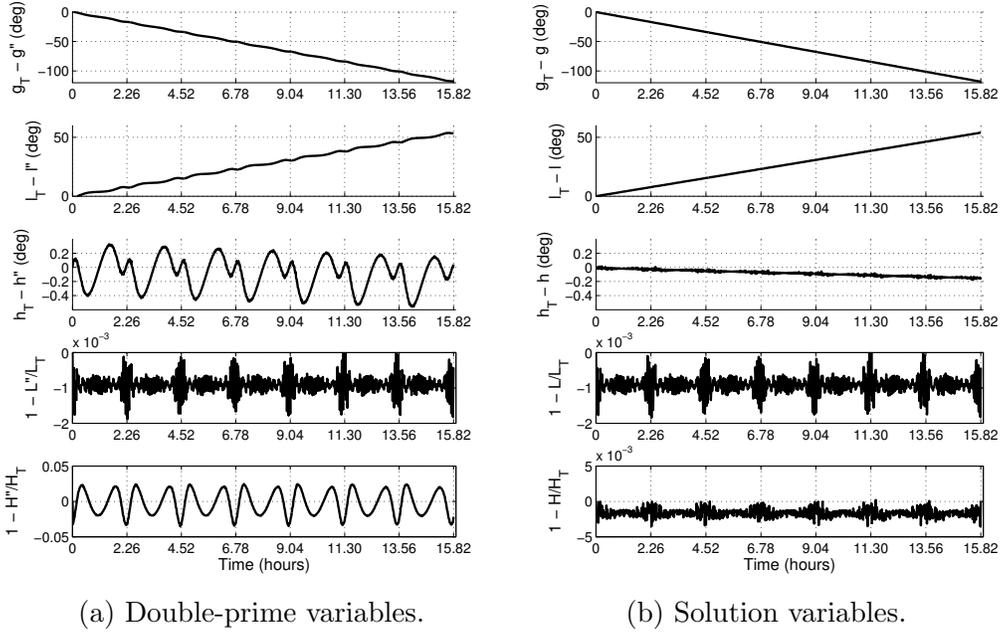


Figure 5.10: Differences between analytically obtained perturbation solution and numerically integrated truth solution (subscript T denotes truth); $e = 0.2$.

value, to a maximum, equal to 10 times the nominal value. The RMS values decrease in an approximately inverse exponential manner as ω_3 increases. While it is difficult to prescribe a precise rule of thumb, significantly better performance is obtained when the rate of rotation about the primary spin axis is an order of magnitude larger than the rotation rates about the other axes when compared to the case $|\omega_3| \approx 1.76|\omega_1|$. However, the body inertia properties must be taken into account when determining whether the assumption of fast rotation is valid due to the nature of ν .

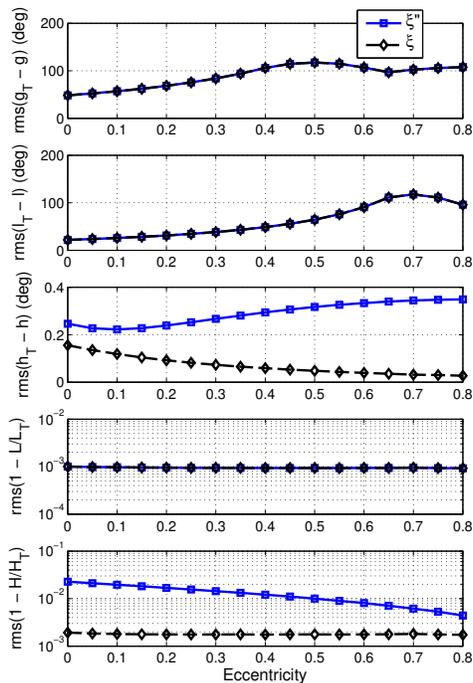


Figure 5.11: Root-mean-square of differences between analytically obtained perturbation solutions and numerically integrated solutions.

5.3.4 Application to Tumbling Rocket Body

One important scenario that may meet the assumptions of fast rotation well is a tumbling rocket body in LEO. As an example, the solution is used to predict the attitude evolution of an SO based on the third-stage engine of the Dnepr-1 launch vehicle that achieved orbit on 29 July 2009 [151]. The rocket body is modeled as resembling a hollow cylinder 1 m in height and 3 m in diameter, with a mass of 2356 kg [106]. Slight perturbations to the principal moments of inertia are introduced to demonstrate the method's ability to handle triaxial SOs. Full physical and dynamic initial conditions for the simulation are given in Tables 5.1 and 5.2, respectively. The tumbling

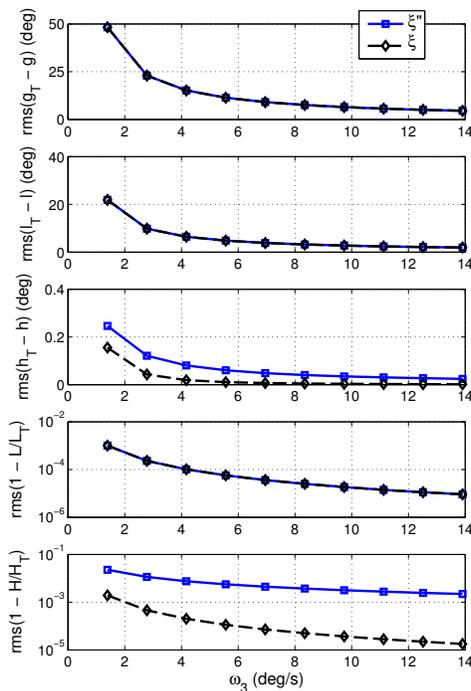


Figure 5.12: Root-mean-square of differences between analytically obtained perturbation solutions and numerically integrated solutions; $e = 0$.

rate about the minor axis (“flat spin” [177]) is based on a flash period¹² of 11.6 seconds, which was estimated on 23 August 2009 [51, 146]. The top and bottom of the rocket body are assumed to have similar reflectance properties, and the flash period is taken to be half of the rotation period of the body of its minor axis. While the flash period is by no means a perfect indicator of angular rate [157], and angular rates of the rocket body are likely to be significantly different in the present day, the value used is representative of

¹²The brightness of an SO, as measured by an observer on Earth, may change over time, in large part due to the presentation of different surfaces with unique reflectance properties to the observer and to the Sun. A flash period is the time between consecutive maxima in the observed brightness of an SO [51].

Table 5.1: Rocket body inertia tensor elements along principal axes in body-fixed frame.

A	2083.01 kg · m ²
B	2318.70 kg · m ²
C	3537.93 kg · m ²

a relatively recently launched rocket body [177, 146, 198]. The body is also given small angular rates about its other axes to model imprecise tumbling. The initial attitude corresponds to a state in which the sides, top, and bottom of the cylinder become visible to the Earth as the body rotates, allowing for the measurement of the flash period.

The SO is assumed to be tumbling about its axis of minimum inertia due to the oblate nature of the body. The derivations of Lara and Ferrer [134] are limited to the case of rotation about the axis of maximum inertia, but the transformations required to handle rotation about the axis of minimum inertia are presented in Appendix E.

Figure 5.13 displays the differences in attitude between the perturbation solution and the numerically integrated equations of motion over seven orbital periods. The truth solution is generated using Eq. (5.28) (i.e., *not* the solution generated by the Hamiltonian of fast rotation). As this example is not intended to directly align with the example presented in Lara and Ferrer [134], the results are presented in Andoyer variables, which permit a more intuitive interpretation than the Lara-Ferrer solution variables. The Andoyer variable

Table 5.2: Initial rocket body state.

(a) Orbital state. ^a	(b) Rotational state.
a 7370 km	q_1 0.0257
e 0.043	q_2 0.0236
θ 0 deg	q_3 0.7128
	q_4 0.7005
	ω_1 15.5172 deg/s
	ω_2 0.1146 deg/s
	ω_3 1.1459 deg/s

^aOrbit orientation elements (e.g., inclination) are unimportant because only point-mass gravity is considered.

values are obtained from the perturbation solution in two ways: First, by a direct transformation from the double-prime variables (Figure 5.13a), and, second, by successively transforming from double-prime to prime variables, from prime variables to solution variables, and from solution variables to Andoyer variables (Figure 5.13b).

Figure 5.14 displays a similar representation for the Andoyer momenta. In this case, Figure 5.14a shows the time evolution of the truth momenta values to give a sense of scale, while Figures 5.14b and 5.14c mimic Figures 5.13a and 5.13b, respectively.¹³ In both Figures 5.13 and 5.14, solid dark areas exist due to the very high rotational frequencies of the SO.

Both the angles and the momenta of the perturbation solution agree well with the truth solution over the plotted time span, though, as expected, secular errors do exist that will inevitably degrade the perturbation solution

¹³Absolute differences are used rather than relative differences because such a plot would be distorted by large relative differences when N passes through zero.

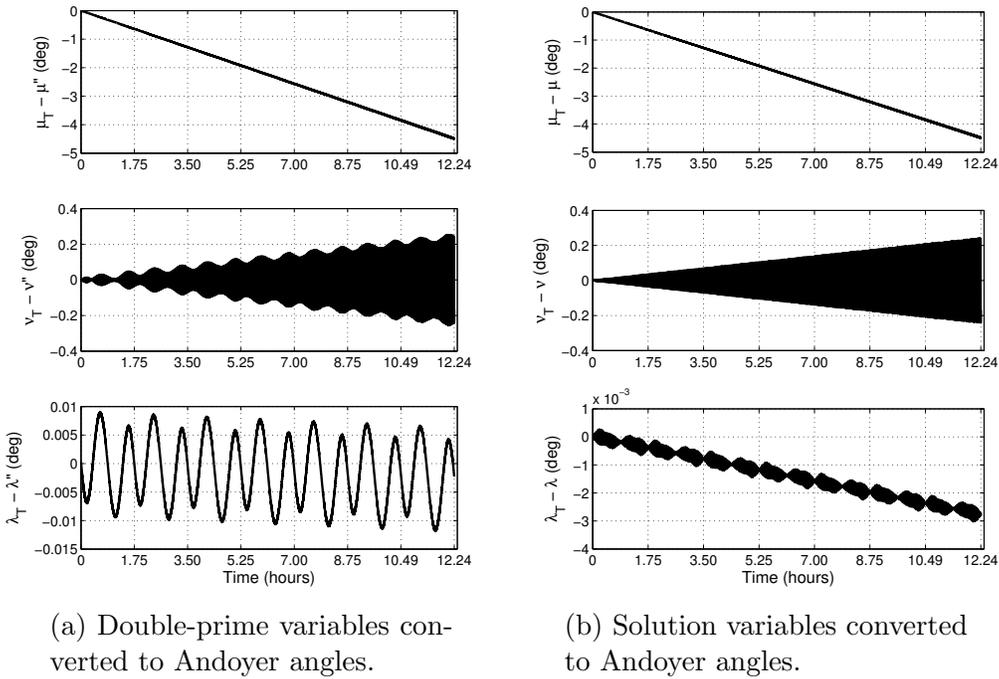
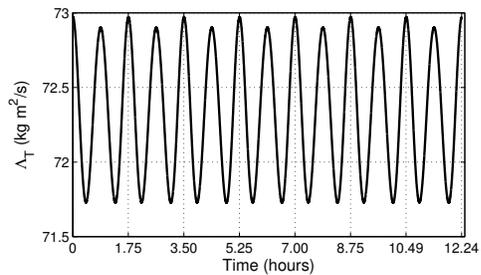
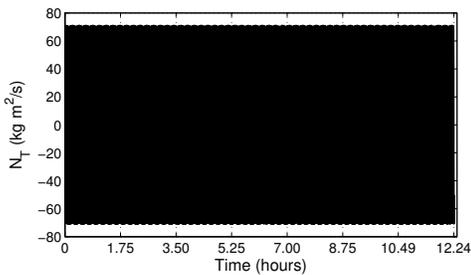


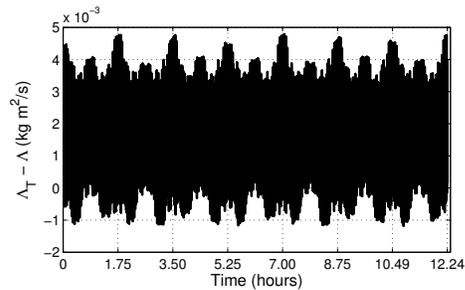
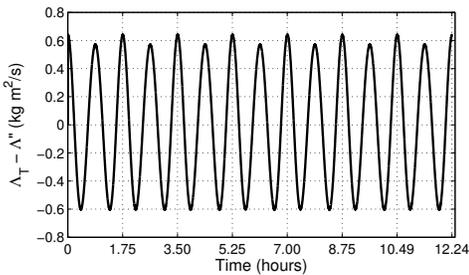
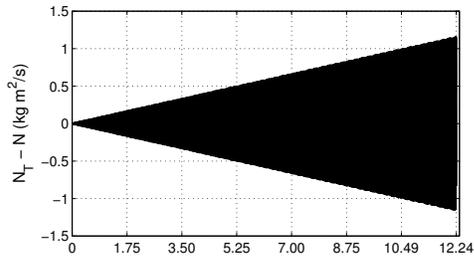
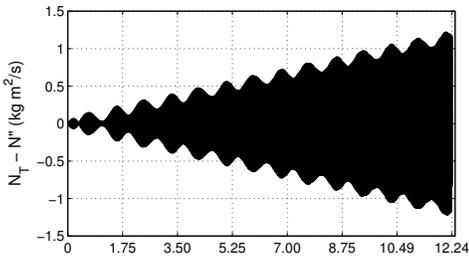
Figure 5.13: Differences between analytically obtained perturbation solution and numerically integrated truth solution (subscript T denotes truth) for tumbling rocket body.

over time. The improved performance compared to the example application presented in Figures 3 and 4 of Lara and Ferrer [134] is caused primarily by the greatly increased rotation rate of the SO. (The initial primary spin rate of the rocket body is approximately 15.5 deg/s, while the initial primary spin rate of the example of Lara and Ferrer [134] is approximately 1.4 deg/s.) The fast rotation rate results in a small value of ϵ , which reduces errors in the perturbation solution relative to the fast-rotating model.¹⁴ Crucially, the fast rotation rate also decreases errors caused by the fast-rotating SO assumption

¹⁴The nominal value of the small parameter of the perturbation procedure for the rocket body example is $\epsilon^2 = 3.855 \times 10^{-5}$.



(a) Andoyer momenta generated by numerically integrated truth solution for tumbling rocket body.



(b) Differences between truth solution and double-prime variables converted to Andoyer momenta.

(c) Differences between truth solution and solution variables converted to Andoyer momenta.

Figure 5.14: Differences between analytically obtained perturbation solution and numerically integrated truth solution (subscript T denotes truth) for tumbling rocket body.

(Figures 5.13 and 5.14 vs. Figure 5.9). Finally, even though the comparisons are not given in the Lara-Ferrer solution variables, the effects of transforming from double-prime to solution variables prior to moving to Andoyer variables

are still apparent. In particular, the periodicity of errors is significantly reduced for the angle λ and its conjugate momentum Λ . This result is examined further in the context of the hybrid SP/GP state prediction method in Chapter 6.

5.4 Conclusions

Lara and Ferrer's closed-form perturbation solution for a fast-rotating triaxial SO subject to gravity-gradient torque is extended from circular to elliptical orbits. The Hamiltonian is rewritten to accommodate nonzero eccentricity, and the resulting explicit time dependence is eliminated by moving to the extended phase space. The new transformation equations depend on orbital true anomaly, meaning that a Kepler solve is required to evaluate the transformations as functions of time. However, both the transformation and secular-rate equations remain fully analytical.

The numerical example presented by Lara and Ferrer [134] is applied to orbits over a wide range of eccentricities. The frequencies of the periodic errors are not constant, as in the circular case, but are instead highest near periapsis, where the gravity-gradient torque is largest and the SO moves most rapidly. Nevertheless, it is demonstrated that the amplitudes of the periodic errors and their secular growth rates as functions of orbital revolutions are comparable to the corresponding values for the circular case.

Additional numerical studies are performed that provide a more complete picture of the applicability of the closed-form solution. The small pa-

parameter of the perturbation procedure is varied to capture changes in the accuracy of the solution as the viability of the Hamiltonian ordering assumption changes. Root-mean-square errors relative to the reference solution generated by the Hamiltonian of a fast-rotating SO, measured in Lara-Ferrer solution variables, increase approximately exponentially as functions of the square of the small parameter, ϵ^2 . For a representative body, RMS errors in all angles are less than 1 deg over seven orbital periods for $\epsilon^2 \leq 10^{-3}$.

The impact of the assumption of fast rotation is examined by varying the SO's angular velocity and comparing against a reference solution that does not assume fast rotation. The errors introduced by using the fast-rotating Hamiltonian to derive the perturbation solution may be significantly greater than those of the Lie transforms procedure if the assumptions of fast rotation are not adequately met. However, there exists a significant number of relevant SOs for which the assumption of fast rotation holds very well, including, importantly, some used rocket bodies. This claim is supported by a numerical example based on a Dnepr third-stage rocket body: Over seven orbital periods, the maximum difference (in Andoyer angles) between the closed-form solution and a reference solution that does not assume fast rotation is less than 5 deg. This capability represents a promising result for reducing the computational burden of estimating and predicting the attitude of potential target bodies for active debris removal missions. In addition, use of this closed-form attitude solution in combination with an analytical, semianalytical, or numerical solution for a SO's translational state has the potential to improve the efficiency of full 6DOF state prediction. This possibility is explored in Chapter 6.

Chapter 6

A Semianalytical Technique for Six-Degree-of-Freedom Space Object Propagation¹

6.1 Introduction

Chapter 1 introduces the concepts of SP, GP, and semianalytical methods for translational, rotational, and 6DOF SO state prediction. In this chapter, the 6DOF problem is addressed through the development of what may be called a *hybrid* SP/GP technique. Specifically, SO attitude is calculated using a GP method, and this approximation is used to inform an SP propagation of the 3DOF state. The development of this approach is motivated chiefly by two factors:

1. 6DOF SP propagation is notoriously slow due to factors including (1) the increased size of the state vector, (2) the use of potentially complicated SO shape models for the calculation of body forces and torques, and (3) differences between the characteristic time scales of the translational and

¹Work from this chapter has been accepted for presentation as:

- **Noble Hatten** and Ryan P. Russell. A semianalytical technique for six-degree-of-freedom space object propagation. Paper 17-376. In *27th AAS/AIAA Space Flight Mechanics Meeting*, San Antonio, TX, February 2017. (accepted for presentation)

Ryan P. Russell supervised the work.

rotational dynamics [65]. Using GP to approximate SO attitude allows for the removal of the rotational state from the numerically propagated state vector. This change significantly increases the efficiency of the overall state prediction because the numerical propagator is less influenced by the fast time scales of the rotational dynamics. The hybrid method may thus also be considered a semianalytical technique [44].

2. In many cases, greater accuracy is desired in the prediction of the 3DOF state than in the rotational state (e.g., catalog maintenance, conjunction analysis, etc.). However, body forces like those caused by aerodynamic drag and SRP depend on SO attitude. Thus, even though the GP attitude prediction is not high-fidelity, the 3DOF state prediction may be improved by using a better approximation of the attitude than would be available from the common “cannonball” assumption. Approximate knowledge of SO attitude can be valuable for applications such as active debris removal (Chapter 5) [177, 157, 203]. The hybrid method therefore serves as a middle ground between 3DOF and 6DOF SP methods.

The overall hybrid method is independent of the closed-form attitude solution, which allows a practitioner to select from available solutions a method that meets their needs. Here, the solution procedure presented in Chapter 5 is used to numerically demonstrate the capabilities of the hybrid method. Algorithmic elements unique to this attitude solution are discussed. The hybrid method is compared against SP 3DOF-only and fully coupled 6DOF propagations for (1) a tumbling rocket body in LEO and (2) a HAMR plate in a

geostationary transfer orbit (GTO). These two scenarios are chosen because of their practical importance, sensitivity to attitude variations, and the assumptions of the specific closed-form attitude solution. First, the prevalence and large size of disposed rocket bodies in Earth orbit make them primary targets for active debris removal. Any generic active debris removal strategy would benefit from the approximate knowledge of a target SO's attitude that the hybrid propagation method provides, as discussed in Chapter 5 [177, 157, 203]. Second, HAMR objects are significantly disturbed by body forces and torques. If the SO surface area changes over time, the invalidity of the cannonball assumption of a 3DOF propagation is therefore more detrimental to the accuracy of state prediction for a HAMR SO than for other types of SOs [65].

6.2 Hybrid SP/GP Method for 6DOF SO State Prediction

A step-by-step description of the hybrid algorithm is given below; the process is summarized in Figure 6.1.²

1. Given an initial time, 6DOF state, and constants (e.g., SO inertia tensor), values are calculated for the expressions needed by the closed-form rotational state prediction procedure.
2. A numerical ODE solution routine is initialized. The state vector of propagation contains only the 3DOF SO state.

²The subscript “tr” indicates translational state, while the subscript “rot” indicates rotational state.

3. Criteria (if any) for reinitializing the closed-form rotational state prediction are defined. Reinitialization of the closed-form attitude solution algorithm may be desirable if the SO state evolves such that the values calculated in the most recent execution of Step 1 no longer adequately represent the scenario.
4. At each propagation step, the ODE solver calls a routine to evaluate the translational state equations \mathbf{f}_{tr} one or more times. Each evaluation requires the calculation of the external forces acting on the SO at the current time and state. While the 3DOF state is obtained from the propagation of the solution of the system of ODEs, the rotational state is obtained by evaluating the analytical attitude solution at the given time. This strategy requires that the values calculated in Step 1 be available to the \mathbf{f}_{tr} evaluation routine in some form.
5. If necessary, Step 1 is repeated (see Step 3).
6. If desired, the values of the rotational state at each propagation step may be output and saved.

The specific method used to predict the rotational state is irrelevant to the algorithm, as long as the prediction is a closed-form function of time and initial conditions only. The method may therefore be selected by the practitioner based on considerations such as (1) applicability to physical/dynamical conditions, (2) ease of implementation, and (3) computational efficiency. As an illustration of the hybrid method, SO attitude is calculated in the examples of this chapter using the solution presented in Chapter 5.

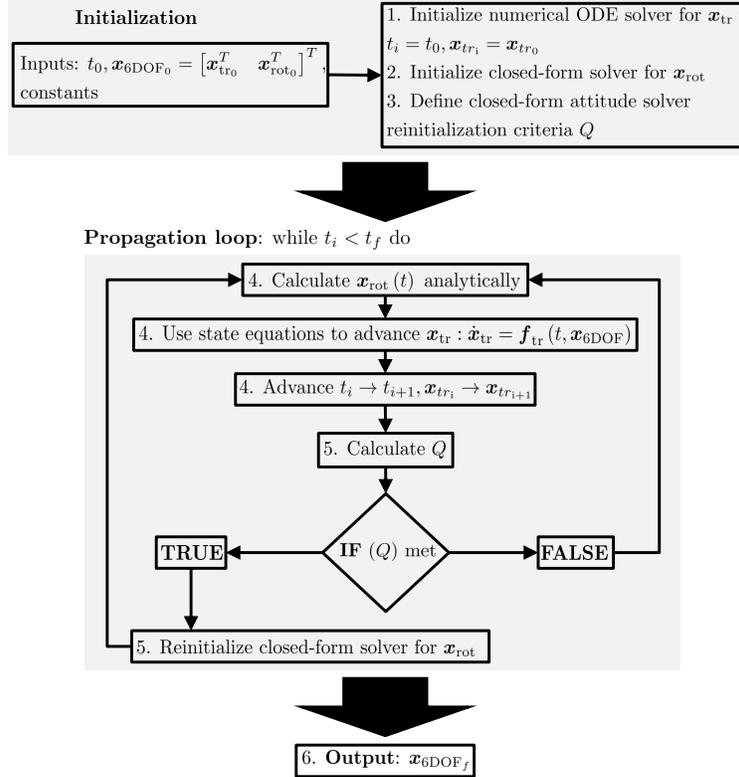


Figure 6.1: Flow chart of hybrid method.

Once a rotational GP solution is selected, any points specific to that theory must be addressed. For the present theory, such considerations include:

- The Hamiltonian is initially formulated using Andoyer variables [94, 45], so transformations between the Andoyer set and any other desired attitude representations (e.g., a transformation matrix) must be implemented. Additionally, the Andoyer variables are singular for certain SO configurations, so a transformation to an alternative set of variables must be performed if it is expected that a singularity condition may be encountered [73].

- Averaging over an angle implicitly assumes that the angle circulates; this criterion must be met by all angles over which averaging is performed.
- The perturbation solution is found by performing the Lie-Deprit transformation procedure twice in succession, each time to eliminate the terms in the Hamiltonian that are periodic in a different variable. As a result, there are actually three possibilities for calculating a solution in the untransformed variables once the solution is obtained in the transformed, “doubly averaged” variables:
 1. transform directly from the doubly averaged variables to Andoyer variables;
 2. transform from doubly averaged to singly averaged variables prior to transforming to Andoyer variables;
 3. transform from doubly averaged to singly averaged variables, and from singly averaged to unaveraged variables prior to transforming to Andoyer variables.

As the number of transformations increases, the amount of periodicity recovered in the solution increases, as well. However, computational effort also increases. The impact of which transformation strategy is chosen is investigated in the numerical examples in Section 6.3.

- The transformation equations of the perturbation theory rely on elliptic integrals and Jacobi elliptic functions; a satisfactory implementation of these functions is therefore required [72, 68, 69, 70, 71].

Regardless of which GP method is selected, the accuracy of the hybrid method is strongly dependent on how well the assumptions of the analytical attitude solution are met. In the present case, the most important assumptions are:

1. The SO is rotating quickly, in the sense that the assumptions of the averaging procedure are valid [134].
2. The parameter of the perturbation procedure, ϵ , is small.³
3. Gravity-gradient torque is the only torque acting on the SO.
4. The SO is in a Keplerian, elliptical orbit.

Assumptions (1), (2), and (4) are functions of the initial conditions and mass properties of the SO. For an Earth-orbiting SO, assumption (3) is never completely valid due to the presence of torques caused by aerodynamic drag, SRP, the Earth's magnetic field, etc. Thus, the validity of the assumption must be assessed on a case-by-case basis, taking into account such factors as SO mass properties, reflectance properties, and electromagnetic properties, as well as the orbit regime.

Assumption (4) is not strictly valid for any force model other than a simple two-body model. However, this issue may be circumvented to some extent because orbital perturbations generally cause slow changes to an initial osculating Keplerian orbit. Closed-form solution values calculated using an

³Assumptions (1) and (2) are related, but not identical.

initial reference orbit may therefore be used to predict rotational states without significantly compromising accuracy as long as the osculating orbit at the time of prediction does not deviate overmuch from the reference orbit. If the deviation becomes too large, the reference orbit may be reselected based on the current state, and the expressions used to calculate the rotational state as a function of time may be reinitialized. (The initial time used by the closed-form solution must be updated, as well.)

While it is feasible to reinitialize the reference orbit at *every* time step, frequent updates can lead to undesirable side effects. First, the efficiency of the overall propagation is worsened due to the increased number of computations. Second, if a new reference orbit is selected too frequently, the quality of the attitude solution actually degrades, even though the validity of assumption (4) improves. The reason is that the Lie-Deprit transformation procedure used to obtain the perturbation solution is truncated at first order. Therefore, when the transformed variables generated by the solution procedure (the ξ'') are converted back to untransformed variables (the ξ), some information is lost. The problem is exacerbated if these new untransformed variables are then, in turn, used to perform Step 1 of the hybrid algorithm. It is thus advisable to change the reference orbit prudently.

In the examples presented in this chapter, the decision to reinitialize the reference orbit is based on the time duration since the most recent initialization, measured in units of the orbital period. The time between reinitializations is set prior to propagation and held constant. More sophisticated criteria are possible, such as a metric that measures a “distance” between the

current osculating orbit and the most recent reference orbit. It is important that the selected criterion may be computed quickly to preserve the efficiency of the hybrid method.

Another factor that may be important for estimation applications is the handling of unknown events, such as maneuvers, collisions, or space weather changes that can cause unpredicted state changes. Event detection and post-event state estimation are challenging tasks when tracking only the translational state of an SO, and numerous methods have been put forth to address this problem [125]. The difficulty increases when the full 6DOF state is under consideration, particularly if attitude measurement data is not provided directly by the SO under consideration (e.g., via an inertial measurement unit), as would be the case for a debris SO. If attitude state uncertainties are estimated to be large following an event detection, it may be prudent to revert to 3DOF-based translational state prediction until additional measurements that reduce attitude uncertainty are accumulated. Otherwise, the benefit of 6DOF state prediction for the translational state – improved calculation of body forces via instantaneous attitude information – may become a detriment because of possibly inaccurate attitude estimates. While this issue is relevant to both full 6DOF and hybrid prediction techniques, the hybrid method presents an additional complication: The validity of the assumptions of the closed-form attitude solution may be unknown for some time following event detection. For example, referencing the theory presented in Chapter 5, if rotational state uncertainty is high, it may be unknown whether the angular

rates of the SO justify the assumption of fast rotation used in the averaging procedure.

Even in the absence of an unknown event, significant attitude uncertainty may produce a poor hybrid-based state prediction due to inaccurate body force estimates and invalid solution assumptions. Use of the hybrid method is therefore most appropriate in scenarios in which at least a moderately accurate initial 6DOF state estimate and SO physical model is available. In such situations, the hybrid method may provide a more accurate translational state prediction than a 3DOF-only propagation, while also providing a reasonably accurate rotational state estimate. At the same time, the hybrid prediction may require significantly less computation time than a full 6DOF propagation. These claims are examined quantitatively in Section 6.3.

6.3 Results

Numerical results are presented to demonstrate the utility of the hybrid method. First, the GP solution for rotation is isolated to assess its computational cost. Then, two case studies are presented to show that (1) the assumptions of this hybrid method implementation are met for realistic and important scenarios and (2) the hybrid method provides a meaningful bridge between 3DOF and 6DOF SP methods for these scenarios.

All code is written in Fortran and compiled using the Intel Visual Fortran Compiler XE 14.0.0.103 (64-bit) using the `-O2` optimization flag. All computations are performed on a 64-bit Windows 7 Enterprise workstation with

two 12-core Intel Xeon E5-2680 v3 processors (clock speed 2.50 GHz) and 64 GB of RAM. Hyperthreading is disabled. The numerical solution of systems of ODEs is calculated using the LSODE package, which provides an open-source, variable-step-size, variable-order, linear-multistep (Adams method) ODE solver [169]. The IRK methods discussed in Chapters 3 and 4 are avoided here in an effort to isolate the effects of the hybrid method when using a generic ODE solver. A reference “truth” solution is generated using a quadruple-precision implementation of the full 6DOF system. For the hybrid method, a new reference orbit is generated once per orbit period; in testing for both case studies, this frequency was found to be a reasonable implementation that balances the accuracy of the assumptions of the closed-form solution and computational effort and inaccuracies caused by overly frequent coordinate transformations.

6.3.1 Computational Speed of Closed-Form Attitude Solution

The computation times required to perform the various transformations of the closed-form attitude theory are depicted in Figure 6.2 to help practitioners calibrate the CPU effort.⁴ Times are normalized by the compute time of the 8×8 gravitational acceleration ($1.29 \mu\text{sec}$ using a spherical harmonics implementation), and the meanings of the abbreviations used in the legend of Figure 6.2 are given in Table 6.1. Note that $\bar{\mathbf{q}}$ and $\boldsymbol{\omega}$ are the quaternion and angular velocity vector representations of the attitude and attitude rates,

⁴All CPU times are averaged over a number of trials such that the total compute time for each case is greater than 1 sec.

respectively, needed for evaluation of attitude-dependent 3DOF acceleration terms. Additionally, ξ , ξ' , and ξ'' are the solution variables, the singly transformed variables, and the doubly transformed variables of the closed-form theory, respectively; a dot represents a time derivative. The data points shown in Figure 6.2 represent the various transformation levels of the closed-form theory described in the Section 6.2.

Predictably, as the number of transformations increases, so does the compute time: The three transformation levels (0, 1, and 2) are approximately equivalent to spherical harmonics acceleration calculations of degree and order 11, 13, and 16, respectively, while the initialization procedure is roughly equivalent to a degree and order 14 calculation. It is therefore important to perform only those transformations that result in significant improvements in solution accuracy. Additionally, the overall computational cost of the dynamics model must be taken into account when deciding whether or not to use the hybrid method. A very inexpensive dynamics model may see little or no speed increase from the hybrid method compared to a full 6DOF propagation.

However, faster analytical solutions may be feasible depending on the specific scenario. For the current closed-form theory, a significant driver of compute time is the calculation of elliptic functions and integrals, which appear due to the triaxiality of the body. If the body is axisymmetric, alternative theories that rely only on trigonometric functions are available [135]. For a nearly axisymmetric body, a small triaxiality may be treated as a perturbation, and again a closed-form theory in terms of trigonometric functions may be

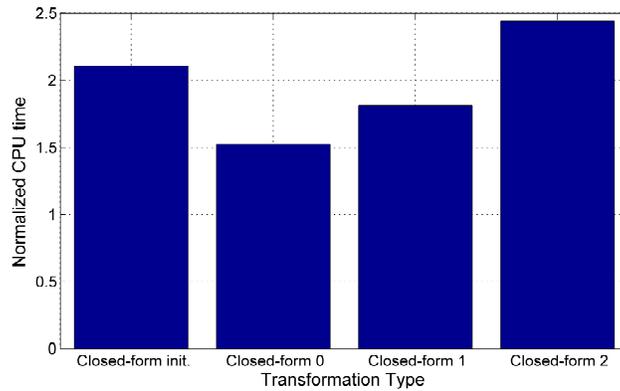


Figure 6.2: Relative CPU times for closed-form attitude theory operations.

derived [137]. Such simplifications, when applicable, improve the efficiency of the closed-form solution [70].

6.3.2 Tumbling Rocket Body in LEO

The first example application simulates 15 orbits of a rocket body SO in LEO. The physical properties of the SO are based on a Centaur upper stage, approximately modeled as a cylinder with length 12.68 m, diameter 3.05 m, and inert mass 2243 kg [186]. In order to induce a larger aerodynamic drag torque than would otherwise be generated, the center of mass of the SO is moved away from the geometrical center by 0.634 m along the long axis. This modification is made so that the simulation yields a more conservative demonstration of the hybrid method’s capabilities: The closed-form attitude solution takes into account only gravity-gradient torque, so aerodynamic torque creates attitude modeling inaccuracies for the hybrid method. A small asymmetry in the radial plane of the SO is introduced, as shown in Table 6.2, in order to the triaxiality assumption of the closed-form attitude theory.

Table 6.1: Summary of coordinate transformations.

Figure 6.2 label	Transformation	Description
Closed-form init.	$\bar{\mathbf{q}}, \boldsymbol{\omega} \rightarrow \dot{\boldsymbol{\xi}}''$	Initialization transformation: 3DOF attitude acceleration variables \rightarrow closed-form attitude solution variables
Closed-form 0	$\dot{\boldsymbol{\xi}}'' \rightarrow \boldsymbol{\xi}'' \rightarrow \bar{\mathbf{q}}, \boldsymbol{\omega}$	Solution variables \rightarrow 3DOF acceleration variables (direct)
Closed-form 1	$\dot{\boldsymbol{\xi}}'' \rightarrow \boldsymbol{\xi}'' \rightarrow \boldsymbol{\xi}' \rightarrow \bar{\mathbf{q}}, \boldsymbol{\omega}$	Solution variables \rightarrow 3DOF acceleration variables (one intermediate transformation)
Closed-form 2	$\dot{\boldsymbol{\xi}}'' \rightarrow \boldsymbol{\xi}'' \rightarrow \boldsymbol{\xi}' \rightarrow \boldsymbol{\xi} \rightarrow \bar{\mathbf{q}}, \boldsymbol{\omega}$	Solution variables \rightarrow 3DOF acceleration variables (two intermediate transformations)

The SO orbit is based on that of a Centaur SO launched in 1972 and still in LEO today (NORAD ID 6155); Table 6.3a gives the initial orbital state used in this example [151]. The initial rotational state assumes a primary tumbling motion about the axis of greatest inertia, with a rotational period of approximately 32 sec. This rotation rate is not necessarily that of the modeled SO, but rotation rates of similar magnitudes have been estimated for disposed rocket bodies using light curves [177, 51, 146]. The initial rotational state of the body-fixed frame with respect to the ECI frame is given in Table 6.3b.

The 3DOF force model consists of a 70×70 spherical harmonics implementation of the geopotential, aerodynamic drag, and point-mass lunisolar gravitational forces. Atmospheric density is calculated using the cubic Harris-Priester model discussed in Chapter 2 with $\bar{F}_{10.7} = 150$ sfu (moderate solar activity). For 6DOF and hybrid propagations, torques caused by the gravity gradient and aerodynamic drag are considered. For the purposes of calculating body forces and torques, the SO is modeled using six flat, rigid panels arranged as a rectangular prism, with side lengths 12.68 m, 3.0 m, and 3.1 m. For 6DOF and hybrid propagations, the algorithm of Doornbos is used to calculate individual lift and drag coefficients for each panel; for 3DOF propagations, the Doornbos method is integrated in closed form assuming a uniform, spherical SO to obtain a single drag coefficient [55] (i.e., cannonball-based drag). This method produces a calculated drag coefficient rather than relying on a user-specified value. While the use of more panels could better approximate a cylinder (or a shape closer to a true rocket body, including nozzles, etc.), this simple model is sufficient for a demonstration of the hybrid methodology. This physical model is also representative of the level of modeling that might be used if limited SO body information is known.

If a significantly higher-fidelity physical model (i.e., one made up of many more individual panels) is available, the practitioner has multiple options when using the hybrid method. First, the hybrid may employ the high-fidelity shape model. In this case, while the efficiency advantage of the hybrid method over fully numerical 6DOF propagation remains, the computational expense of *either* relative to 3DOF propagation increases due to the calculation of body

forces on a large number of surfaces. An alternative is to use a lower-fidelity, approximate shape model with the hybrid method. This strategy is also an option for obtaining faster, less-accurate state predictions using full 6DOF propagation, but is particularly synergistic with the hybrid method because the latter is based on efficiency-improving approximations.

A key consideration in comparing a 3DOF propagation of the system to either a 6DOF or hybrid propagation is the surface area value used to calculate aerodynamic drag. For the 6DOF or hybrid model, the area of each panel exposed to the atmosphere is calculated dynamically as a function of the attitude, but the 3DOF model has no such information. Therefore, a constant area is assumed. In the results presented in this study, two constant area values are used. First, a quick, rough value is obtained by averaging the areas of the sides of the rectangular prism model (28.88 m^2). Second, the area value that produces the smallest RMS error relative to the 6DOF reference solution over the course of the propagation is determined numerically using a grid search; this value is found to be 41.85 m^2 . This second method is not feasible in a practical scenario, but is included to account for the possibility of a more intelligently determined area value than a simple average.

Simulation results are summarized in Figure 6.3, which shows position state error relative to the 6DOF reference propagation as a function of CPU time for a range of relative tolerances for the ODE solver. The 0, 1, and 2 subscripts of “Hybrid” in the legend have the same meanings as the 0, 1, and 2 labels for “Closed-form” given in Table 6.1 for the closed-form attitude solution. The 3DOF legend entry with no subscript indicates the averaged area

Table 6.2: Rocket body inertia tensor elements along principal axes in body-fixed frame.

A	6593.76 kg · m ²
B	40492.81 kg · m ²
C	40658.00 kg · m ²

Table 6.3: Initial rocket body state.

(a) Orbital state.	(b) Rotational state.
a 7042.35 km	q_1 0.0596
e 0.00374	q_2 0.0132
i 35.0 deg	q_3 0.2667
Ω 122.0 deg	q_4 0.9618
ω 237.0 deg	ω_1 0.1146 deg/s
θ 0.0 deg	ω_2 1.1459 deg/s
	ω_3 11.2511 deg/s

is used, while the “Best A” subscript indicates the numerically determined lowest-error area is used. Finally, results for two full 6DOF propagations are presented. The subscript “All” indicates that the same relative tolerance is used for all states in the ODE solver. Meanwhile, the subscript “Low” means that the relative tolerance on the rotational states is held fixed at 10^{-3} while the tolerance on the translational states is varied. The latter option is included for comparison because it mimics the ideology of the hybrid method by producing a lower-fidelity attitude solution alongside a more precise translational

solution. Different data points for a single prediction type are obtained by varying the relative local truncation error tolerance of the ODE solver.

The two styles of 3DOF propagation behave similarly except for an improvement in accuracy when the optimized drag area is used. Each is inexpensive relative to the 6DOF and, to a lesser extent, hybrid propagations. However, the 3DOF solutions reach a relatively high “accuracy floor”, at which point reducing the tolerance of the ODE solver no longer produces a more accurate solution. For the averaged-area propagation, the minimum RMS error is approximately 66 m, while, for the best-area propagation, the minimum RMS error is approximately 20 m.

For the fully numerical 6DOF propagation, the accuracy of the “Low” version increases without a corresponding increase in CPU time over the majority of the translational state integration tolerance range. The reason is that the tolerance on the rotational state, though held at a loose value, nevertheless drives the step size of the propagation until the translational relative tolerance is less than or equal to 10^{-13} . Alternatively, the CPU time for the “All” version increases with each decrease in tolerance. As a result, the accuracy floor of the “All” propagation is more than two orders of magnitude smaller than that of the “Low” propagation, but a CPU time of about 14 sec is required to reach the smaller floor.

The utility of the hybrid method is visible in the lower left portion of Figure 6.3, in which the hybrid method offers a solution that is more accurate than the best-area 3DOF propagation and faster than the “Low” 6DOF

propagation. The hybrid method (regardless of transformation level) achieves approximately meter-level accuracy – more than an order of magnitude below the accuracy floor of the best-area 3DOF solution – using about three and six times less CPU time than the “Low” and “All” 6DOF methods, respectively.

At the same time, notes of caution regarding use of the hybrid method must be mentioned. First, appropriate integration tolerance (or step size) selection for the ODE solver used to propagate the translational equations of motion is important. Excessively loose tolerances (or large step sizes) produce results no more accurate than a 3DOF propagation, while excessively tight tolerances (or small step sizes) result in less accuracy for a given CPU time than a full 6DOF propagation. This outcome is related to the validity of the assumptions of the closed-form attitude solution: the better the analytical solution matches the true rotational dynamics, the more accurate the computation of body forces, and the lower the accuracy floor of the numerically propagated translational state. Improvements in the validity of the assumptions are therefore likely to decrease the integrator tolerance values at which use of the hybrid method is advantageous. For this example, the relative tolerance “sweet spot” is approximately $10^{-10} - 10^{-12}$.

Secondly, the choice of transformation type used by the attitude solution (0, 1, or 2) impacts the accuracy and efficiency of the propagation, though the speed variation is not large for the current example due to the expense of the translational dynamics model. Over the range of tolerances in which the hybrid method is beneficial for this example, there is little difference in accuracy between the transformation types, as well. However, as shown in Fig-

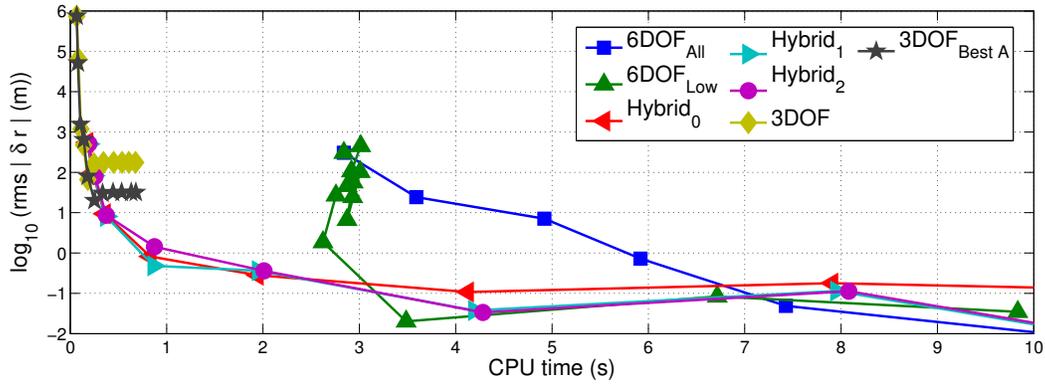


Figure 6.3: RMS position error as a function of CPU time for 15-orbit propagation of a rocket body in LEO.

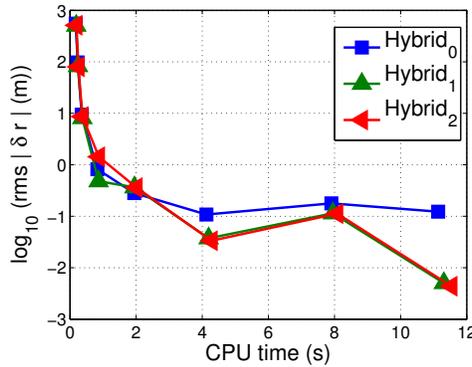


Figure 6.4: Simplification of Figure 6.3, featuring only the hybrid solutions and extending the CPU time axis to show all data points.

Figure 6.4, differences do exist: Primarily, transformation types 1 and 2 tend to provide similar accuracies, while less accuracy is obtained from transformation type 0. This result is examined in further detail in Section 6.3.3.

6.3.2.1 Impact of SO Rotation Rate

The rocket body scenario is re-simulated for integer multiples of the initial primary spin rate (ω_3) ranging from 1–10 (see the leftmost column of

Table 6.4). Each propagation scheme is run for a range of ODE solver relative tolerance values: $\text{tol} = 10^x$, with x varied in increments of 0.2 (i.e., $\dots, 10^{-9}, 10^{-9.2}, 10^{-9.4}, \dots$). For the hybrid and 6DOF variations, the loosest ODE solver tolerance that results in an RMS position error smaller than 7 m is identified. This cutoff value is chosen because it corresponds to an improvement of approximately one order of magnitude over the best-case accuracy obtained by the 3DOF propagation using an average area value (see Figure 6.3). The selected tolerances for each case are given in Table 6.4. For these specific runs, the speedups of the 6DOF “Low,” Hybrid₀, Hybrid₁, and Hybrid₂ propagations relative to the 6DOF “All” propagation are shown in Figure 6.5.

This parameter sweep demonstrates how the hybrid method becomes more effective as the rotation rate of the SO increases: The relative speedups of the hybrid method increase from approximately 10× to greater than 80× over the range of initial spin rates plotted in Figure 6.5. Meanwhile, the relative speedups achieved by the “Low” 6DOF option remain relatively constant between 1.5× and 1.8× as the initial spin rate increases.

A faster rotation rate means that a fully numerical 6DOF propagation must take smaller time steps to capture the high-frequency changes in the rotational state. The hybrid method is less affected by such changes because the rotational state is calculated analytically. The increased spin rate is therefore only important to the step sizes taken by the hybrid method inasmuch as the time evolution of body forces acting on the SO changes.

Table 6.4: Common logarithm (i.e., \log_{10}) of ODE solver relative tolerances corresponding to data points in Figure 6.5.

Initial ω_3 (deg/s)	ODE solver tolerance				
	Hybrid ₀	Hybrid ₁	Hybrid ₂	6DOF _{All}	6DOF _{Low}
11.2511	-10.2	-10.2	-10.2	-4.8	-11.6
22.5021	-10	-10	-10	-5	-12.6
33.7532	-10	-10.2	-10	-4.2	-10
45.0042	-10.2	-10	-10.2	-4.2	-10.2
56.2553	-10	-10.2	-10	-4.2	-9.2
67.5063	-10	-10.2	-10	-4.2	-8.8
78.7574	-10.2	-10.4	-10	-4.2	-9
90.0084	-10	-10.4	-10	-4	-9
101.2595	-10	-10	-10	-4	-9
112.5105	-10.2	-10	-10.2	-4	-9.2

The accuracy of the rotational state predictions of the hybrid method improves as the spin rate increases because the closed-form attitude solution assumes fast rotation. However, even the slowest spin rate considered in this case study is fast enough that the effect on the overall accuracy of the predicted position state is minimal over the 15-orbit prediction time span.

6.3.3 HAMR Plate in GTO

The second example application is a three-orbit propagation of a nearly flat, HAMR plate in GTO. Like the rocket body SO, the HAMR SO is modeled as a rectangular prism, but with side lengths 0.5 m, 0.25 m, and 0.001 m. The mass of the SO is 0.00475 kg, which gives the SO an area-to-mass ratio similar to that of Kapton, a substance commonly used to thermally insulate satellites

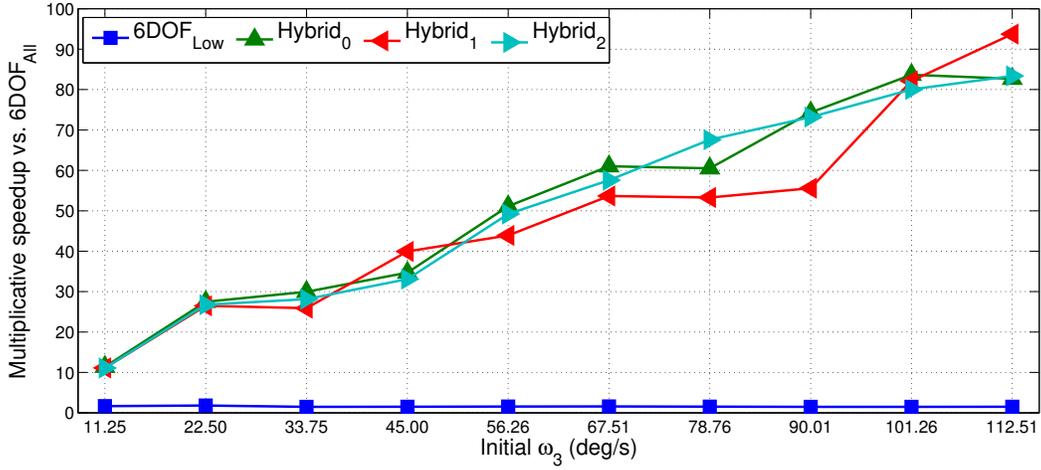


Figure 6.5: Multiplicative speedups of hybrid and “Low” 6DOF propagations compared to “All” 6DOF propagation.

[66]; the inertia parameters of the SO are given in Table 6.5. No center-of-mass offset is assumed in the SO, and the coefficients of specular and diffuse reflection are taken to be 0.60 and 0.26, respectively, for all surfaces; these values are consistent with Kapton [66].

The force model consists of a 33×33 interpolated implementation of the geopotential, aerodynamic drag, SRP, and point-mass lunisolar gravitational forces [12]. For the 6DOF and hybrid models, the SRP force is calculated for each panel of the SO as described by Früh et al. [66]. As with aerodynamic drag, the SRP force for the 3DOF-only model is calculated using a single, constant area value based on an assumption of a spherical body [66].

The highly eccentric GTO necessitates a high-degree/order geopotential near periapsis, but a lower-degree/order field is acceptable near apoapsis. The interpolated gravity model runs significantly more quickly than the spherical

Table 6.5: HAMR SO inertia tensor elements along principal axes in body-fixed frame.

<i>A</i>	$2.47 \times 10^{-5} \text{ kg} \cdot \text{m}^2$
<i>B</i>	$9.90 \times 10^{-5} \text{ kg} \cdot \text{m}^2$
<i>C</i>	$1.23 \times 10^{-4} \text{ kg} \cdot \text{m}^2$

harmonics formulation, eliminating the need for a variable-degree/order model to obtain an efficient propagation. For 6DOF and hybrid propagations, torques caused by the gravity gradient and aerodynamic drag are considered. Torques due to SRP are neglected because, for a rectangular prism body with uniform reflectance properties, the net SRP torque is always zero under the current SRP model [199]. In the case of an asymmetrical SO, SRP torque is likely to be a significant driver of rotational motion due to the HAMR nature of the SO (see Section 3.4.2) [66]. Selection of an alternative closed-form attitude solution that approximates the effects of SRP torques may be appropriate (e.g., Zanardi and Vilhena de Moraes [205]).

The SO initial conditions are given in Table 6.6. The rotational state corresponds to a 1 revolution-per-minute rotation rate about the minimum inertia axis.

Simulation results are summarized in Figure 6.6; the caption meanings are the same as for Figure 6.3. Expectedly, the 3DOF propagations are again the most efficient, but the RMS error floors for the averaged area and the best-

Table 6.6: Initial HAMR SO state.

(a) Orbital state.		(b) Rotational state.	
a	24009.05 km	q_1	0.0727
e	0.713	q_2	0.0481
i	20.6 deg	q_3	0.5926
Ω	198.0 deg	q_4	0.8008
ω	109.0 deg	ω_1	6.0000 deg/s
θ	0.0 deg	ω_2	0.1146 deg/s
		ω_3	0.2865 deg/s

case area solutions are both greater than 20 km.⁵ The “Low” 6DOF propagation achieves an RMS error of approximately 1.45 km before the required CPU time begins to increase due to decreases in integration tolerance for the translational states. As in the rocket body example, a region is found in which the hybrid method provides improved accuracy compared to the 3DOF propagation and improved efficiency compared to the “Low” 6DOF propagation. The Hybrid₁ solution achieves an RMS error less than 10 km approximately three times faster than the “Low” 6DOF propagator and reaches sub-kilometer-level RMS position errors approximately 2.5 times faster than the “All” 6DOF solution. (The “All” solution is faster than the “Low” solution at this – and all tighter – accuracy levels.)

This simulation starkly shows the differences that can arise in the Hybrid₁ and Hybrid₂ solutions vs. the Hybrid₀ solution. The additional variable transformations that differentiate Hybrid₁ from Hybrid₀ recover periodic

⁵The averaged area value is 0.04192 m², and the best-case area is 0.07258 m².

solution terms related to averaging over the orbital period. Meanwhile, moving from Hybrid₁ to Hybrid₂ recovers periodic terms related to averaging over a body orientation angle. The amplitudes of the averaged variations related to the orbital period are generally the larger of the two by a significant margin, resulting in the observed trends (Chapter 5) [134]. It is therefore advisable to retain at least the Hybrid₁ solution, though, for the cases tested, any additional accuracy gained by using the Hybrid₂ solution is likely to be offset by the corresponding increase in CPU time.

For this example, the relative integration tolerance sweet spot for the hybrid method is approximately $10^{-6} - 10^{-9}$, which is not as tight as that for the rocket body example. This result is due to (1) the increased importance of accurate attitude information for the accurate translational state prediction of the HAMR SO and (2) the decreased validity of the closed-form solution's assumption that gravity-gradient torque is the only torque acting on the SO. The approximations of the hybrid method are more detrimental for the HAMR plate than for the rocket body due to the rocket body's higher density and its spin axis. (Rotation about the maximum-inertia axis is likely to be more stable than rotation about the minimum-inertia axis because of energy dissipation effects [120].)

More generally, from the two numerical examples, it is seen that the most favorable tolerance level for the hybrid method corresponds to the "knee" in the error vs. CPU time curve: the point at which the magnitude of the slope of the curve created by successively decreasing the integration tolerance begins

to decrease.⁶ Unfortunately, the tolerance value for this point is dependent on not only the physical and dynamic properties of the simulation, but the ODE solver and the step size selection algorithm, as well. Further, in practice, the speed advantages of the hybrid method may be lost if a high-fidelity 6DOF reference solution must be generated repeatedly to determine the optimal tolerance for the hybrid method. It is therefore likely infeasible to pronounce a specific integration tolerance value range applicable to all scenarios.

This type of deficiency is not unique to the hybrid method. The loosest integration tolerance that reaches the accuracy floor of the 3DOF propagation is unlikely to be known *a priori*, potentially leading to significant computational waste if an overly tight tolerance is selected. Additionally, while the “Low” 6DOF method can provide improved efficiency compared to a full 6DOF propagation over a range of integration tolerances for the translational state, this range is also problem-dependent. Even if full 6DOF propagation is used, there is no single integration tolerance that is optimal across all applications [188].

6.4 Conclusions

The limiting assumptions – and, thus, limited accuracy – of 3DOF SO state prediction and the high computational burden of full 6DOF propagation leave room for intermediate techniques. In this chapter, one such option is developed by exploiting the rich history of analytical and semiana-

⁶This discussion assumes a variable-step ODE solver. In the case of a fixed-step method, the role of the integration tolerance is played by the step size.

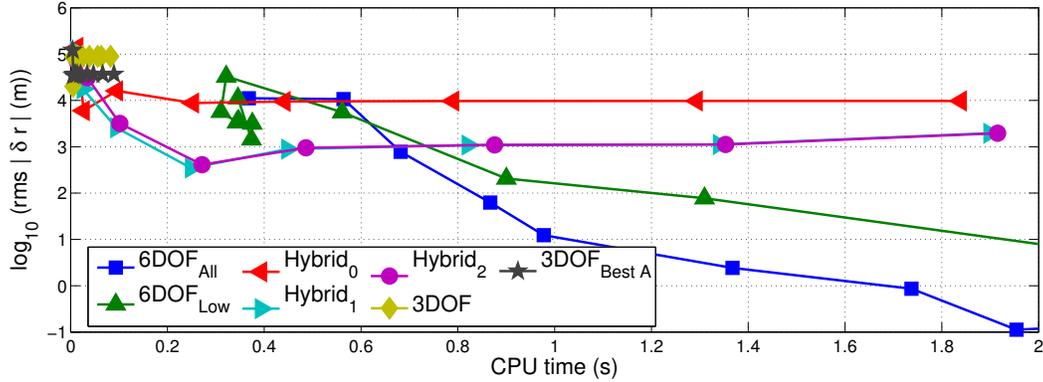


Figure 6.6: RMS position error as a function of CPU time for three-orbit propagation of a HAMR plate in GTO.

lytical approximate solutions for SO translational and rotational motion. The result is a semianalytical, hybrid special/general perturbation algorithm in which the translational state is propagated numerically, informed by closed-form approximations of the rotational state. The hybrid method calculates body translational forces like aerodynamic drag and solar radiation pressure using a dynamically updated attitude and a user-defined SO physical model. This force model contrasts with the typical assumption of a spherical body used by 3DOF propagations. At the same time, the hybrid method may take larger step sizes than a full 6DOF propagation because the attitude is not part of the numerically integrated state.

The hybrid method is agnostic to the choice of closed-form attitude solution, allowing a practitioner to select the algorithm that best fits their needs (e.g., SO physical and dynamical properties, computational resources). The method is illustrated using a perturbation solution, derived using the Lie-Deprit method, that is applicable to a fast-rotating, triaxial, rigid body in an

elliptical orbit subject to gravity-gradient torque (Chapter 5). The capabilities of this hybrid method implementation are demonstrated in two example simulations of practical importance: a rocket body in LEO and a HAMR plate in GTO. The translational equations of motion for both simulations use dynamics models outside the assumptions of the attitude perturbation theory (e.g., a high-fidelity geopotential, aerodynamic drag, SRP, etc.). The deviations from Keplerian motion are addressed by reinitializing the attitude solution procedure once per orbit revolution. The hybrid method is shown to produce position predictions more than an order of magnitude more accurate than a best-case 3DOF propagation at approximately one-third of the computational cost of a customized fully numerical 6DOF propagation. It is also demonstrated that efficiency gains realizable via the hybrid method increase as the rotation rate of the object increases because the numerical ODE solver does not have to directly propagate the fast-changing attitude states. However, as with any technique involving numerical integration, care must be taken in the selection of the integration tolerance (for a variable-step method) or step size (for a fixed-step method): If the tolerance is too loose, minimal improvement over a 3DOF propagation may be seen, while, if the tolerance is too tight, a full 6DOF propagation may produce more accurate results with greater efficiency. Improved applicability of the assumptions of the closed-form attitude solution decreases the optimal integration tolerance values of the hybrid method because greater accuracy in the rotational state makes possible greater accuracy for the numerically propagated translational state.

At near-optimal integration tolerances, the hybrid method offers a bridge between 3DOF and fully numerical 6DOF propagations. The improved accuracy attained for SO position state prediction vs. 3DOF cannonball propagation has the potential to, for example, decrease uncertainties in conjunction assessments. The approximate knowledge of SO attitude produced by the hybrid method may be important for applications such as active debris removal. At the same time, the speed increase vs. full 6DOF propagation makes the hybrid method more amenable to large-catalog applications. Due to these qualities, the hybrid method may also be an attractive choice for SOs whose physical characteristics are only approximately known or whose attitude state is moderately uncertain. In these situations, the expense of a fully numerical 6DOF propagation may not be justified because of the uncertainties caused by the lack of available information. The hybrid method, combined with a rough SO physical model, can achieve a reasonable facsimile of the full 6DOF propagation while using significantly less computational resources.

Chapter 7

Conclusions

The tongue-in-cheek Wirth's Law and its less-famous sibling state "Software is getting slower more rapidly than hardware becomes faster" and "Software expands to fill the available memory," respectively [200]. These adages were originally intended to lament the bloat of word processors, operating systems, and the like as computer hardware grows more powerful. However, a similar claim may be made that is specifically applicable to scientific computational modeling: The demands of technical software grow to meet or exceed every hardware advancement.

This statement is not intended to be as harshly critical as Wirth's Law. On the contrary, the exploitation of each new hardware innovation by the scientific community has led to the continuous development of heretofore infeasible techniques. In astrodynamics, rapid improvements in hardware capabilities have resulted in Monte Carlo uncertainty propagation, large-dimension parameter optimization, and stochastic optimization methods, to name but a few examples. Yet, as software takes on more and more challenging tasks, it is worth remembering that computational power is still finite, and the capabilities of lower-level routines – which may be called millions of times in the execution of a single high-level application – remain as important as ever.

With these ideas in mind, this dissertation presents a set of new techniques, each designed to improve one or more of the fidelity, efficiency, and utility of SO state prediction and sensitivity calculation. The necessity and ubiquity of these tools for vital and demanding applications like space situational awareness and space mission design strongly motivate the continued development of analytical and numerical methods tailored to the SO problem. In this final chapter, the major conclusions of the work are summarized and directions for future work are discussed. A publication history of the material is given in Appendix F.

7.1 Dissertation Summary and Primary Contributions

The first area addressed is thermospheric modeling, a key element of state prediction for SOs in LEO due to the importance of aerodynamic forces. In Chapter 2, the so-called cubic Harris-Priester atmospheric density model is presented. CHP improves upon the classic modified HP model by using interpolation techniques to ensure smooth changes in density, in particular as a function of geodetic altitude. CHP adds a polynomial functional dependence on the $\bar{F}_{10.7}$ solar flux proxy by performing a least-squares fit of “measurements” produced by a higher-fidelity density model. Continuous partial derivatives of the density with respect to SO state, time, and $\bar{F}_{10.7}$ are derived. These contributions improve the accuracy of the HP model, especially in extreme solar conditions, while also increasing robustness and utility for sensitivity calculation. At the same time, CHP retains strong efficiency char-

acteristics in the calculation of density and associated sensitivities relative to other models.

In Chapter 3, a methodology is proposed to improve the efficiency of the historically slow SP propagation of the coupled SO orbit/attitude (i.e., 6DOF) state. A variable-step-size GLIRK ODE solver is described, and customizations including variable-fidelity dynamics models and pre-convergence LTE estimation are shown to improve propagator efficiency. In a serial environment, performance is found to peak when using 8–12 stages. The GLIRK solver is parallelized using OpenMP. Significant efficiency gains when using more than 8–12 stages (and a corresponding number of threads) are only observed for very expensive dynamics models due to parallelization overhead. This result suggests that the approximation of linear speedups used in some previous works may not always be appropriate. Nevertheless, for scenarios in which an appropriate low-fidelity dynamics model may be created, the parallelized GLIRK propagator is found to outperform high-order, variable-step-size linear-multistep and ERK ODE solvers.

The necessity of calculating STMs in addition to state predictions for many common applications is addressed in Chapter 4. The DDM of STM calculation takes advantage of the speed and accuracy attainable through use of analytical Jacobians and Hessians. Unlike the CDM, the calculation of STMs is separated from the propagation of the state. In one form, STMs are obtained through direct differentiation of the ODE solver update equations (rather than via propagation of the variational equations). STMs obtained in this manner are therefore precise sensitivities of the specific equations used to propagate

the state. In Chapter 4, the DDM equations for second-order STM calculation are derived for RK ODE solvers, and both first- and second-order DDM equations are derived for the double-integrator form of the RK update equations. While applicable to ERK, as well, the DDM for IRK is shown to greatly reduce the number of required Jacobian and Hessian evaluations relative to the CDM, thereby reducing computation time. Efficiency for IRK methods is further improved by effectively applying variable-fidelity dynamics modeling and parallelization techniques (discussed in Chapter 3) to STM calculation.

In Chapters 5 and 6, a semianalytical hybrid SP/GP method for the state prediction of nonspherical SOs is introduced. The method combines a closed-form perturbation solution for the SO rotational state with the numerical propagation of the translational state to produce a prediction that is both more accurate than a 3DOF cannonball-based SP propagation and more efficient than a fully coupled 6DOF SP propagation. Groundwork for the hybrid method is laid in Chapter 5, in which a perturbation solution for a fast-rotating triaxial SO subject to gravity-gradient torque is extended to the elliptical orbit case. The fidelity of the closed-form solution is demonstrated for scenarios of varying eccentricity and values of the small parameter of the perturbation method. In addition, the use of the closed-form solution in applications other than the hybrid propagator is discussed. Fast attitude prediction for rocket bodies targeted for active debris removal is identified as an important case that may meet the assumptions of the perturbation solution.

In Chapter 6, the hybrid method itself is presented in detail. Two scenarios of a nonspherical SO subject to body forces are simulated: a rocket

body in LEO and a HAMR plate in GTO. In each case, the attitude prediction capabilities of the hybrid model, using the closed-form solution described in Chapter 5, facilitate position state predictions one or more orders of magnitude more accurate than a 3DOF cannonball propagation. At the same time, the hybrid predictions require significantly less runtime than a fully coupled 6DOF propagation.

The material presented in the dissertation spans a significant portion of the ever-important accuracy vs. efficiency spectrum for SO state prediction and is therefore applicable to a variety of applications. Beginning at the lowest fidelity level, the analytical GP methods discussed for translational (Section 1.2.4.2) and rotational (Section 1.2.4.4, Chapter 5) motion are most appropriate when moderate accuracy is sufficient. Applications for the translational routines include routine SO catalog maintenance and preliminary conjunction analysis, as well as preliminary mission design work, such as trade studies in which a large number of possible trajectories must be evaluated quickly. The rotational theories may be applied to, for example, the prediction and estimation of the attitude of a derelict rocket body targeted for active debris removal.

Semianalytic 3DOF techniques (Section 1.2.4.3) produce predictions that, if properly customized, may provide sufficient accuracy to reliably predict SO conjunction events more rapidly than a 3DOF SP propagation [179, 64]. Nevertheless, 3DOF SP (Section 1.2.2.1) is necessary for precision orbit determination of near-spherical SOs (e.g., for science missions with tight orbit

accuracy requirements) and SOs for which attitude measurements are unavailable (e.g., for decreasing the uncertainty in conjunction analyses).

6DOF prediction (Section 1.2.2.6) provides the greatest accuracy of the methods presented, but is the most computationally intensive. Observability of the rotational state is also required. The hybrid method (Chapter 6), which sacrifices rotational state accuracy for speed, may be appropriate for catalog maintenance of HAMR SOs whose translational states are not well predicted by 3DOF-only propagation. Fully numerical 6DOF propagation (Chapter 3), though slower, may be used as an alternative in situations in which no suitable closed-form attitude theory is available, and may be necessary for an SO with precise pointing requirements.

7.2 Future Work

The cubic Harris-Priester atmospheric density model presented in Chapter 2 uses a global functional relationship to include dependencies on the 81-day centered $\bar{F}_{10.7}$ solar flux proxy. The initial calibration of the CHP model via repeated higher-fidelity density model evaluations requires on the order of minutes of computation time, with exact needs dependent on the particular high-fidelity model used. It is therefore feasible to recalculate the CHP coefficients often to take into account the most recent measurement data. Future work could evaluate the benefits of frequent recalibrations.

Additional environmental dependencies – like daily $F_{10.7}$ values and geomagnetic indices – could be included in the CHP model. Determination of

an appropriate functional form would be required, and the likely appearance of coupling among the independent variables would necessitate use of a nonlinear least-squares procedure to calculate the modeling coefficients. Further work could also focus on improving the accuracy of the latitudinal dependency of the Harris-Priester model, which is currently based on orbital inclination rather than geodetic latitude.

The GLIRK ODE solver for 6DOF propagation (Chapter 3) could be improved by use of a better step size selection mechanism. The current implementation produces overly conservative step sizes due to the low order of the nearly embedded comparison method relative to the propagated method. Development of a more accurate method of estimating the local truncation error while retaining efficiency is challenging due to the superconvergence of the GLIRK solver. Along these lines, creation of an efficient step size adaptation algorithm for long-time-step implicit ODE solvers like MCPI or BLC-IRK could improve the viability of such methods for 6DOF applications. Accomplishment of this goal could further increase speed gains due to parallelization because of the increased number of stages per step of long-time-step methods.¹

Implicit 6DOF propagation, particularly for HAMR SOs, would benefit from development of improved low-fidelity dynamics models. The current ODE solver underperforms for HAMR SOs relative to other classes of SOs because of the relatively small difference between the computational burdens of the low-fidelity and high-fidelity dynamics models. A less expensive low-

¹Efficient variable-step implementations of MCPI and BLC-IRK would benefit 3DOF propagation, as well.

fidelity model that provides an adequate representation of the true dynamics would significantly improve efficiency. At the same time, the development of automated algorithms for selecting an appropriate low-fidelity dynamics model without user input would expand the possible user base for implicit propagation methods.

Further testing of the DDM of STM calculation for RK methods presented in Chapter 4 would provide a more complete picture of the potential benefits of the technique. An efficiency comparison against STMs produced by algorithmic differentiation algorithms would quantify the speed differences between the DDM and indirect methods of comparable accuracy. Also, modern computing architectures could be more fully exploited by parallelizing not only the evaluations of Jacobians/Hessians at each time step but the STM calculations at all time steps, as well. Nested parallelization of this kind is an ideal arena for massive parallelization, e.g., using GPUs. The NVIDIA CUDA standard for GPU computing supports nested parallelization, and development of a GPU-based version of the STM calculation methods discussed in Chapter 4 could provide further speed enhancements [154].

The results of Chapters 3 and 4 show that parallelization overhead when using multicore CPUs with implicit ODE solvers is nontrivial, even when the dynamics model is relatively expensive. Additional speed increases may be possible with further investigation and implementation of overhead mitigation techniques.

The hybrid SP/GP 6DOF propagation method discussed in Chapters 5 and 6 can be matured in several ways. First, an automated procedure for determining an appropriate time to calculate a new reference orbit for rotational GP methods that rely on low-fidelity orbit models (e.g., an elliptical orbit) would improve the robustness of the hybrid method. The selection of a proper integration tolerance (or fixed step size) is another important part of maximizing the benefits of the hybrid method. The development of an algorithm for efficiently optimizing the integration tolerance with minimal user input would reduce the guesswork currently associated with the hybrid method.

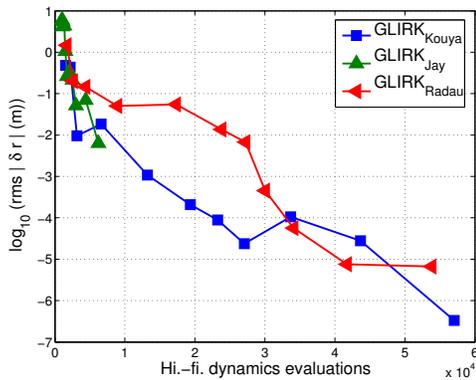
Additionally, the chosen closed-form attitude solution is one of many possibilities. If, for example, gravity-gradient torque is not dominant or the SO of interest does not rotate rapidly, an alternative theory may be substituted to improve results. If the SO is axisymmetric, spherical, or nearly so, computation-saving simplifications are possible. In the long run, a comprehensive algorithm that automatically and intelligently selects an appropriate closed-form attitude solution for a given scenario would greatly expand the utility of the hybrid method. Even further, automating the entire procedure of propagator class selection from the wealth of possibilities – 3DOF GP, 3DOF semianalytic, 3DOF SP, 6DOF SP, or 6DOF hybrid – for a given application could have a profound effect on ensuring that a near-optimal amount of computational resources is assigned to achieve a desired state prediction fidelity.

Appendices

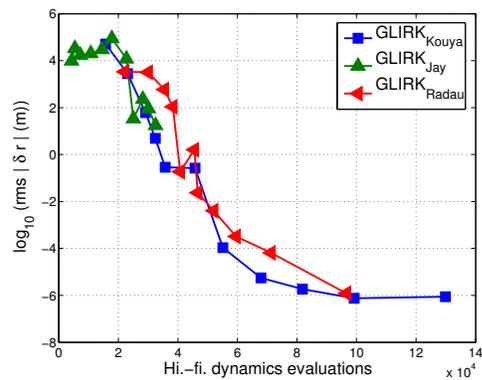
Appendix A

GLIRK Local Truncation Error Estimation

As mentioned in Chapter 3, multiple methods exist for calculating the “comparison solution” used to estimate the LTE of a GLIRK ODE solver. Position accuracy as a function of high-fidelity dynamics function evaluations (a proxy for CPU time) for three such methods is shown in Figure A.1. The figure displays results for three-orbit propagations of the tumbling LEO scenario (Figure A.1a) and non-tumbling GEO scenario (Figure A.1b) presented in Chapter 3. The variable-fidelity dynamics strategy is used, and the GLIRK solution uses eight stages. The “Kouya” method, which is based on a nearly embedded solution of order s , is the strategy used to produce the results given in Chapter 3 [127]. The “Jay” method uses an internal tolerance parameter to generate a less conservative estimate of the LTE based on the order- s comparison solution [112]. The “Radau” method produces a comparison solution using a non-embedded s -stage Radau-IA propagation, which produces a solution of order $(2s - 1)$ [88]. For the results presented in Figure A.1, it is assumed that the initial guess for the Radau-IA solution is accurate enough that only one high-fidelity dynamics function evaluation per stage is required to achieve convergence.



(a) LEO scenario (tumbling) position accuracy, three orbits.



(b) GEO scenario (non-tumbling) position accuracy, three orbits.

Figure A.1: Position accuracy as a function of high-fidelity dynamics function evaluations for three LTE calculation methods; relative LTE tolerance of propagators is varied from $10^{-5} - 10^{-15}$.

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

Appendix B

State Transition Matrix Calculation Via the Direct Method

For the system described by Eq. (1.2), the STMs describe the evolution of deviations in the initial state [185]:

$$\delta \mathbf{x}(t) = \Phi^1(t, t_0) \delta \mathbf{x}(t_0) + \frac{1}{2} \delta \mathbf{x}(t_0)^T \bullet_2 \Phi^2(t, t_0) \delta \mathbf{x}(t_0) + \mathcal{O}(\delta \mathbf{x}(t_0)^3), \quad (\text{B.1})$$

where, for any $\mathbf{v}_{n \times 1}$ and $\mathbf{T}_{n \times n \times n}$ [162],

$$[\mathbf{T} \bullet_2 \mathbf{v}](i, j) = [\mathbf{v}^T \bullet_2 \mathbf{T}](i, j) \triangleq \sum_{p=1}^n \mathbf{T}(i, p, j) \mathbf{v}(p). \quad (\text{B.2})$$

The first- and second-order STMs satisfy

$$\Phi^1(t, t_0) = \frac{\partial \mathbf{x}(t)}{\partial \mathbf{x}(t_0)} \quad (\text{B.3})$$

$$\Phi^2(t, t_0) = \frac{\partial^2 \mathbf{x}(t)}{\partial \mathbf{x}(t_0)^2}. \quad (\text{B.4})$$

In these equations, Φ^2 is organized such that

$$\Phi^2(t, t_0)(i, j, k) = \frac{\partial^2 \mathbf{x}(t)(i)}{\partial \mathbf{x}(t_0)(j) \partial \mathbf{x}(t_0)(k)}. \quad (\text{B.5})$$

Φ^2 contains symmetries due to the commutativity of mixed partial derivatives; that is, $\Phi^2(i, j, k) = \Phi^2(i, k, j)$.

The STMs for the system under consideration evolve subject to ODEs known as the variational equations:

$$\dot{\Phi}^1(t, t_0) = \mathbf{J}(t, \mathbf{x}) \Phi^1(t, t_0) \quad (\text{B.6})$$

$$\dot{\Phi}^2(t, t_0) = \mathbf{J}(t, \mathbf{x}) \bullet_1 \Phi^2(t, t_0) + \Phi^{1T}(t, t_0) \bullet \mathbf{H}(t, \mathbf{x}) \bullet \Phi^1(t, t_0) \quad (\text{B.7})$$

$$\mathbf{J} \triangleq \frac{\partial \mathbf{f}(t, \mathbf{x})}{\partial \mathbf{x}(t)} \quad (\text{B.8})$$

$$\mathbf{H} \triangleq \frac{\partial^2 \mathbf{f}(t, \mathbf{x})}{\partial \mathbf{x}(t)^2}, \quad (\text{B.9})$$

where, for any $M_{n \times n}$ and $T_{n \times n \times n}$,

$$[\mathbf{T} \bullet_1 \mathbf{M}](i, j, k) = [\mathbf{M} \bullet_1 \mathbf{T}](i, j, k) = \sum_{p=1}^n \mathbf{T}(p, j, k) \mathbf{M}(i, p) \quad (\text{B.10})$$

$$[\mathbf{M}^T \bullet \mathbf{T} \bullet \mathbf{M}](i, j, k) = \sum_{p=1}^n \sum_{q=1}^n \mathbf{T}(i, p, q) \mathbf{M}(p, j) \mathbf{M}(q, k). \quad (\text{B.11})$$

The array \mathbf{J} is known as the Jacobian, while \mathbf{H} is called the Hessian. Initial conditions for STM propagation are $\Phi^1(t_0, t_0) = \mathbf{I}_n$ and $\Phi^2(t_0, t_0) = \mathbf{0}_{n \times n \times n}$, where \mathbf{I}_n is the $n \times n$ identity matrix and $\mathbf{0}_{n \times n \times n}$ is an $n \times n \times n$ array of zeros.

Two STMs may be “linked” in the independent variable to calculate the STM over a different span:

$$\Phi^1(t_2, t_0) = \Phi^1(t_2, t_1) \Phi^1(t_1, t_0) \quad (\text{B.12})$$

$$\Phi^2(t_2, t_0) = \Phi^1(t_2, t_1) \bullet_1 \Phi^2(t_1, t_0) + \Phi^{1T}(t_1, t_0) \bullet \Phi^2(t_2, t_1) \bullet \Phi^1(t_1, t_0). \quad (\text{B.13})$$

B.1 Second-Order ODEs

For a system of second-order ODEs (Eq. (4.1)), the second-order RK form given by Eqs. (4.2)–(4.4) may be used to propagate the variational equations [183]. For the first-order STM, Φ^1 is rewritten as

$$\Phi_{n \times n}^1 = \begin{bmatrix} \Phi_{1_{n' \times n}}^1 \\ \Phi_{2_{n' \times n}}^1 \end{bmatrix}, \quad (\text{B.14})$$

where Φ_1^1 and Φ_2^1 satisfy

$$\Phi_1^1(t, t_0) = \frac{\partial \mathbf{y}(t)}{\partial \mathbf{x}(t_0)} \quad (\text{B.15})$$

$$\Phi_2^1(t, t_0) = \frac{\partial \dot{\mathbf{y}}(t)}{\partial \mathbf{x}(t_0)} = \dot{\Phi}_1^1(t, t_0) \quad (\text{B.16})$$

$$\ddot{\Phi}_1^1(t, t_0) = \frac{\partial \mathbf{g}(t, \mathbf{y}, \dot{\mathbf{y}})}{\partial \mathbf{y}(t)} \Phi_1^1(t, t_0) + \frac{\partial \mathbf{g}(t, \mathbf{y}, \dot{\mathbf{y}})}{\partial \dot{\mathbf{y}}(t)} \dot{\Phi}_1^1(t, t_0). \quad (\text{B.17})$$

Thus, Eq. (B.17) is the second-order ODE to be propagated. The initial conditions are given by

$$\Phi_1^1(t_0, t_0) = [\mathbf{I}_{n'} \quad \mathbf{0}_{n' \times n'}] \quad (\text{B.18})$$

$$\dot{\Phi}_1^1(t_0, t_0) = [\mathbf{0}_{n' \times n'} \quad \mathbf{I}_{n'}]. \quad (\text{B.19})$$

Similarly, for the second-order STM, Φ^2 is rewritten as

$$\Phi_{n \times n \times n}^2 = \begin{bmatrix} \Phi_{1, n' \times n \times n}^2 \\ \Phi_{2, n' \times n \times n}^2 \end{bmatrix}, \quad (\text{B.20})$$

where Φ_1^2 and Φ_2^2 satisfy

$$\Phi_1^2(t, t_0) = \frac{\partial \Phi_1^1(t, t_0)}{\partial \mathbf{x}(t_0)} \quad (\text{B.21})$$

$$\Phi_2^2(t, t_0) = \frac{\partial \dot{\Phi}_1^1(t, t_0)}{\partial \mathbf{x}(t_0)} = \dot{\Phi}_1^2 \quad (\text{B.22})$$

$$\ddot{\Phi}_1^2(t, t_0) = \ddot{\Phi}^2(t, t_0) (n' + 1 : n, 1 : n, 1 : n). \quad (\text{B.23})$$

Due to the increased complexity of Eq. (B.7) compared to Eq. (B.6), Eq. (B.23) is not simplified in the same manner as Eq. (B.17). The required subarray of Eq. (B.7), when substituted into Eq. (B.23), must be written in terms of \mathbf{y} and $\dot{\mathbf{y}}$ rather than in terms of \mathbf{x} . The initial conditions for both Φ_1^2 and $\dot{\Phi}_1^2$ are simply appropriately sized arrays of zeros.

Appendix C

The Lie-Deprit Method

The Lie-Deprit method is a perturbation-based algorithm for removing short-period terms from a Hamiltonian system to obtain a simpler system – one that depends only on long-period and/or secular terms. Here, the method is described with emphasis on the computational procedure; further theoretical details are available in, e.g., Boccaletti and Pucacco [26], Deprit [52], Kamel [118, 119].

The Hamiltonian of the original system is assumed to be dependent on a small parameter ϵ such that

$$\mathcal{H}(\mathbf{q}, \mathbf{p}; t; \epsilon) = \sum_{n=0}^{\infty} \frac{1}{n!} \epsilon^n \mathcal{H}_{n,0}(\mathbf{q}, \mathbf{p}; t), \quad (\text{C.1})$$

where \mathbf{q} and \mathbf{p} are vectors of the coordinates and generalized momenta, respectively, of the system and t is the independent variable. The Lie-Deprit method seeks a transformed Hamiltonian \mathcal{K} such that

$$\mathcal{K}(\mathbf{Q}, \mathbf{P}; t; \epsilon) = \sum_{n=0}^{\infty} \frac{1}{n!} \epsilon^n \mathcal{H}_{0,n}(\mathbf{Q}, \mathbf{P}; t) = \sum_{n=0}^{\infty} \frac{1}{n!} \epsilon^n \mathcal{K}_{n,0}(\mathbf{Q}, \mathbf{P}; t), \quad (\text{C.2})$$

where \mathbf{Q} and \mathbf{P} are vectors of the coordinates and generalized momenta, respectively, of the new system.¹ The transformation is achieved via a generating function, defined by

$$W(\mathbf{Q}, \mathbf{P}; t; \epsilon) = \sum_{n=0}^{\infty} \frac{1}{n!} \epsilon^n W_{n+1}(\mathbf{Q}, \mathbf{P}; t). \quad (\text{C.3})$$

The generating function is obtained by solving a partial differential equation known as the homological equation. This equation is solved repeatedly, once for each nonzero coefficient of ϵ^n in Eq. (C.1), and is given by

$$\begin{aligned} \mathcal{K}_{n,0} = \mathcal{H}_{n,0}(\mathbf{Q}, \mathbf{P}; t) + \\ \sum_{j=1}^{n-1} \left[\binom{n-1}{j-1} L'_j \mathcal{H}_{n-j,0} + \binom{n-1}{j} \mathcal{K}_{j,n-j} \right] - \frac{\mathcal{D}W_n}{\mathcal{D}t}, \end{aligned} \quad (\text{C.4})$$

where

$$L'_j f = \frac{\partial f}{\partial \mathbf{Q}} \cdot \frac{\partial W_j}{\partial \mathbf{P}} - \frac{\partial f}{\partial \mathbf{P}} \cdot \frac{\partial W_j}{\partial \mathbf{Q}} \quad (\text{C.5})$$

$$\frac{\mathcal{D}W_n}{\mathcal{D}t} = \frac{\partial W_n}{\partial t} - L'_n \mathcal{H}_{0,0} \quad (\text{C.6})$$

$$\mathcal{K}_{j,i} = L'_j \mathcal{K}_i - \sum_{m=1}^{j-1} \binom{j-1}{m-1} L'_j \mathcal{K}_{j-m,i} \quad (\text{C.7})$$

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}. \quad (\text{C.8})$$

¹The $\mathcal{H}_{n,0}$ are the terms of the original Hamiltonian, while the $\mathcal{H}_{0,n}$ are the terms of the transformed Hamiltonian.

Eq. (C.5) is the Lie operator and may be written as the Poisson bracket $\{f; W_j\}$.

The variable transformations are obtained from

$$\mathbf{q} = \mathbf{Q} + \sum_{n=1}^{\infty} \frac{1}{n!} \epsilon^n \mathbf{Q}^{(n)}(\mathbf{Q}, \mathbf{P}; t) \quad (\text{C.9})$$

$$\mathbf{p} = \mathbf{P} + \sum_{n=1}^{\infty} \frac{1}{n!} \epsilon^n \mathbf{P}^{(n)}(\mathbf{Q}, \mathbf{P}; t), \quad (\text{C.10})$$

where

$$\mathbf{Q}^{(n)} = \frac{\partial W_n}{\partial \mathbf{P}} + \sum_{j=1}^{n-1} \binom{n-1}{j} \mathbf{Q}_{j, n-j} \quad (\text{C.11})$$

$$\mathbf{P}^{(n)} = -\frac{\partial W_n}{\partial \mathbf{Q}} + \sum_{j=1}^{n-1} \binom{n-1}{j} \mathbf{P}_{j, n-j} \quad (\text{C.12})$$

$$\mathbf{Q}_{j,i} = L'_j \mathbf{Q}^{(i)} - \sum_{m=1}^{j-1} \binom{j-1}{m-1} L'_m \mathbf{Q}_{j-m,i} \quad (\text{C.13})$$

$$\mathbf{P}_{j,i} = L'_j \mathbf{P}^{(i)} - \sum_{m=1}^{j-1} \binom{j-1}{m-1} L'_m \mathbf{P}_{j-m,i}. \quad (\text{C.14})$$

To second order (the maximum order used in the present work), the relevant equations are:

$$\mathcal{K}_{0,0} = \mathcal{H}_{0,0} \quad (\text{C.15})$$

$$\mathcal{K}_{1,0} = \mathcal{H}_{1,0} - \frac{\mathcal{D}W_1}{\mathcal{D}t} \quad (\text{C.16})$$

$$\mathcal{K}_{2,0} = \mathcal{H}_{2,0} + L'_1 \mathcal{H}_{1,0} + L'_1 \mathcal{K}_{1,0} - \frac{\mathcal{D}W_2}{\mathcal{D}t} \quad (\text{C.17})$$

$$\mathbf{Q}^{(1)} = \frac{\partial W_1}{\partial \mathbf{P}} \quad (\text{C.18})$$

$$\mathbf{Q}^{(2)} = \frac{\partial W_2}{\partial \mathbf{P}} + L'_1 \mathbf{Q}^{(1)} \quad (\text{C.19})$$

$$\mathbf{P}^{(1)} = -\frac{\partial W_1}{\partial \mathbf{Q}} \quad (\text{C.20})$$

$$\mathbf{P}^{(2)} = -\frac{\partial W_2}{\partial \mathbf{Q}} + L'_1 \mathbf{P}^{(1)}. \quad (\text{C.21})$$

The inverse coordinate transformations – useful for transforming initial conditions from physical variables to solution variables – are given by

$$\mathbf{Q} = \mathbf{q} + \sum_{n=1}^{\infty} \frac{1}{n!} \epsilon^n \mathbf{q}^{(n)}(\mathbf{q}, \mathbf{p}; t) \quad (\text{C.22})$$

$$\mathbf{P} = \mathbf{p} + \sum_{n=1}^{\infty} \frac{1}{n!} \epsilon^n \mathbf{p}^{(n)}(\mathbf{q}, \mathbf{p}; t), \quad (\text{C.23})$$

where, to second order [52],

$$\mathbf{q}^{(1)} = -\frac{\partial W_1}{\partial \mathbf{p}} \quad (\text{C.24})$$

$$\mathbf{q}^{(2)} = -\frac{\partial W_2}{\partial \mathbf{p}} + \left\{ \frac{\partial W_1}{\partial \mathbf{p}}; W_1 \right\} \quad (\text{C.25})$$

$$\mathbf{p}^{(1)} = \frac{\partial W_1}{\partial \mathbf{q}} \quad (\text{C.26})$$

$$\mathbf{p}^{(2)} = \frac{\partial W_2}{\partial \mathbf{q}} + \left\{ -\frac{\partial W_1}{\partial \mathbf{q}}; W_1 \right\}. \quad (\text{C.27})$$

Unlike in the previous equations, the generating function terms W_j are assumed to be expressed in the original variables \mathbf{q} and \mathbf{p} in Eqs. (C.24)–(C.27).

With the machinery in place, the task becomes the selection of the $\mathcal{H}_{0,n}$ ($= \mathcal{K}_{n,0}$) so that Eq. (C.4) may be solved for the W_n . Frequently, $\mathcal{H}_{0,n}$ is selected to be the average value of the short-periodic terms in Eq. (C.4) to produce a W_n that is free of secular terms, and, simultaneously, eliminate one or more coordinates from $\mathcal{K}_{n,0}$. The system described by the Hamiltonian \mathcal{K} is thus, hopefully, easier to solve than the original system described by \mathcal{H} .

C.1 Example

As an example of the application of the Lie-Deprit method, a single-degree-of-freedom system with a first-order perturbation is considered.² The system is described by the Hamiltonian

$$\mathcal{H} = \omega_0 p - \frac{\epsilon}{6} p^2 \sin^4 q, \quad (\text{C.28})$$

where q and p are the scalar generalized coordinate and momentum, respectively, ω_0 is a constant, and ϵ is the small perturbation parameter. Following Eq. (C.1), the terms of \mathcal{H} are organized as

²This example is adapted from Boccaletti and Pucacco [26], Section 8.3.2.

$$\mathcal{H}_{0,0} = \omega_0 p \quad (\text{C.29})$$

$$\mathcal{H}_{1,0} = -\frac{1}{6} p^2 \sin^4 q = -\frac{p^2}{48} [3 - 4 \cos(2q) + \cos(4q)]. \quad (\text{C.30})$$

From Eq. (C.15), it is immediately seen that

$$\mathcal{K}_{0,0} = \omega_0 P, \quad (\text{C.31})$$

where Q and P are the transformed scalar generalized coordinate and momentum, respectively. The generating function W_1 is obtained by solving Eq. (C.16), which, for this scenario, is

$$\mathcal{K}_{1,0} = \underbrace{-\frac{P^2}{48} [3 - 4 \cos(2Q) + \cos(4Q)]}_{\mathcal{H}_{1,0}} - \underbrace{\omega_0 \frac{\partial W_1}{\partial Q}}_{\frac{\mathcal{D}W_1}{\mathcal{D}t}}. \quad (\text{C.32})$$

$\mathcal{K}_{1,0}$ may be selected as the average value of $\mathcal{H}_{1,0}$ over one period of Q (2π):

$$\mathcal{K}_{1,0} = \frac{1}{2\pi} \int_0^{2\pi} -\frac{P^2}{48} [3 - 4 \cos(2Q) + \cos(4Q)] dQ \quad (\text{C.33})$$

$$= -\frac{P^2}{16}. \quad (\text{C.34})$$

Substituting Eq. (C.34) into Eq. (C.32) yields

$$\underbrace{-\frac{P^2}{16}}_{\mathcal{K}_{1,0}} = -\underbrace{\frac{P^2}{48} [3 - 4 \cos(2Q) + \cos(4Q)]}_{\mathcal{H}_{1,0}} - \underbrace{\omega_0 \frac{\partial W_1}{\partial Q}}_{\frac{\mathcal{D}W_1}{\mathcal{D}t}}, \quad (\text{C.35})$$

which may be integrated in Q to solve for W_1 :

$$W_1 = -\frac{P^2}{192\omega_0} [\sin(4Q) - 8 \sin(2Q)]. \quad (\text{C.36})$$

With the generating function available, Eqs. (C.18) and (C.20) give the forward transformations ($q \rightarrow Q$ and $p \rightarrow P$) to first order, while Eqs. (C.24) and (C.26) give the inverse transformations ($Q \rightarrow q$ and $P \rightarrow p$) to first order:

$$q = Q - \frac{\epsilon P}{96\omega_0} [\sin(4Q) - 8 \sin(2Q)] \quad (\text{C.37})$$

$$p = P - \frac{\epsilon P^2}{48\omega_0} [\cos(4Q) - 4 \cos(2Q)] \quad (\text{C.38})$$

$$Q = q + \frac{\epsilon p}{96\omega_0} [\sin(4q) - 8 \sin(2q)] \quad (\text{C.39})$$

$$P = p + \frac{\epsilon p^2}{48\omega_0} [\cos(4q) - 4 \cos(2q)]. \quad (\text{C.40})$$

The dynamics of the transformed system are obtained via Hamilton's equations of motion:

$$\mathcal{K} = \underbrace{\omega_0 P}_{\mathcal{K}_{0,0}} - \underbrace{\frac{\epsilon}{16} P^2}_{\epsilon \mathcal{K}_{1,0}} \quad (\text{C.41})$$

$$\frac{dQ}{dt} = \frac{\partial \mathcal{K}}{\partial P} = \omega_0 - \frac{\epsilon}{8} P \quad (\text{C.42})$$

$$\frac{dP}{dt} = -\frac{\partial \mathcal{K}}{\partial Q} = 0. \quad (\text{C.43})$$

Integrating in time gives closed-form solutions for P and Q as functions of time.

$$Q(t) = Q(t_0) + \left[\omega_0 - \frac{\epsilon}{8} P(t_0) \right] (t - t_0) \quad (\text{C.44})$$

$$P(t) = P(t_0). \quad (\text{C.45})$$

Thus, the average solution may be calculated in closed-form and analytically transformed from $\{Q, P\}$ to $\{q, p\}$ to recover periodic terms in the solution.

Hamilton's equations may also be used to obtain differential equations for the original system, Eq. (C.28):

$$\frac{dq}{dt} = \omega_0 - \frac{\epsilon}{3} p \sin^4 q \quad (\text{C.46})$$

$$\frac{dp}{dt} = -\frac{2}{3} \epsilon p^2 \sin^3 q \cos q. \quad (\text{C.47})$$

With the expressions presented, three solution possibilities exist. The first is to use a numerical ODE solution algorithm to approximate the solutions

of Eqs. (C.46) and (C.47). The second is to use Eqs. (C.44) and (C.45) to analytically obtain solutions of the averaged system in transformed variables (Q and P). Finally, periodic terms of the averaged solution may be recovered by performing the inverse transformation (Eqs. (C.39) and (C.40)) to obtain q and p from the closed-form solutions for Q and P . Figure C.1 compares the time evolutions of the three solutions for $t \in [0, 20]$, $q(t_0) = 0$, $p(t_0) = 1$, $\omega_0 = 1$, $\epsilon = 0.1$. For both q and p , the impact of performing the inverse transformation after obtaining the closed-form solution in Q and P is clear. The “Avg. transformed” solution tracks the numerically integrated solution very well over the time span plotted, though secular errors due to the truncated perturbation theory are visible, especially for q .

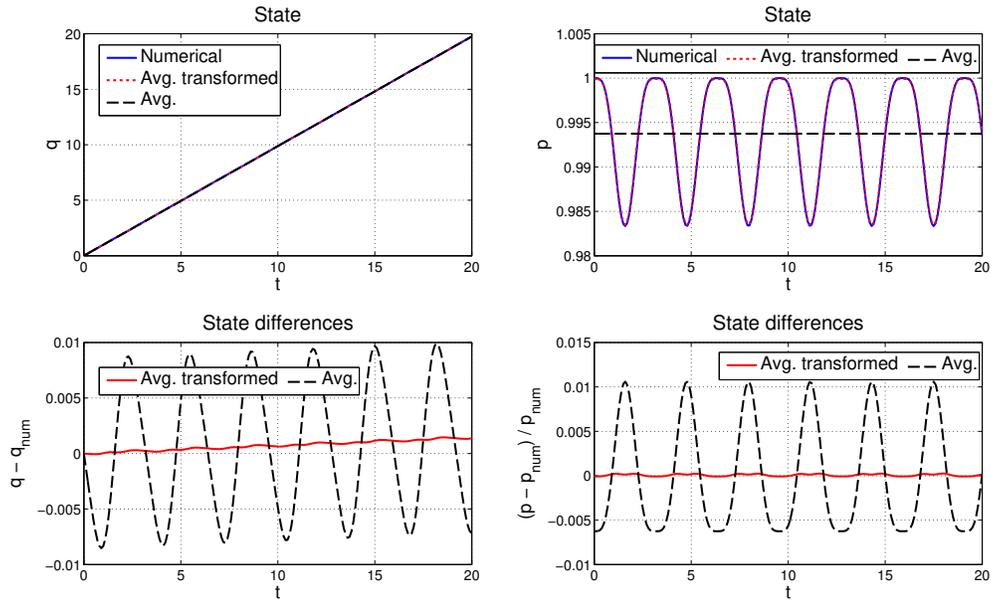


Figure C.1: Time evolutions of states for system (C.28), calculated in three different manners. “Numerical” indicates numerical solution of unaveraged ODEs; “Avg.” indicates closed-form solution of averaged dynamics *without* recovery of periodic terms; “Avg. transformed” indicates closed-form solution of averaged dynamics *with* recovery of periodic terms.

Appendix D

Lara-Ferrer Circular Orbit Equations

In Chapter 5, the equations of the perturbation solution for the elliptical orbit case that are the same or only slightly modified from the original circular case are omitted in the interest of space. The relevant equations are given here, and the corresponding equations in the original text [134] are referenced when appropriate. It is emphasized that these equations are the original equations, applicable to circular orbits only. The revisions described in Section 5.2.1 must be performed in order to achieve validity for eccentric orbits.

The auxiliary variables f (Eq. (11)), m (Eq. (12)), and Δ (Eq. (5)) are given by

$$f = \frac{C(B-A)}{(C-B)A} \quad (\text{D.1})$$

$$m = \frac{(C-\Delta)(B-A)}{(C-B)(\Delta-A)} \quad (\text{D.2})$$

$$\Delta = \frac{G^2}{2\Phi}. \quad (\text{D.3})$$

Eq. (18) gives the torque-free Hamiltonian in solution variables:

$$\Phi = \frac{G^2}{2A} \left(1 - \frac{C-A}{C} \frac{f}{f+m} \right). \quad (\text{D.4})$$

Eqs. (27) and (30) give variable relationships necessary for performing calculations:

$$L \frac{\partial m}{\partial L} = -\frac{L}{G} \frac{\pi}{K(m)} \frac{(f+m)^{3/2}}{\sqrt{f(1+f)}} \quad (\text{D.5})$$

$$\frac{L}{G} = \frac{2}{\pi} \sqrt{1+f} \sqrt{\frac{f}{f+m}} \left[\Pi(-f|m) - \frac{m}{f+m} K(m) \right]. \quad (\text{D.6})$$

Eq. (49) gives the Hamiltonian for gravity-gradient-perturbed rotation, simplified by the assumption of fast rotation, in Andoyer variables:

$$\mathcal{H} = \frac{M^2}{2C} \left\{ \left(\frac{\sin^2 \nu}{A/C} + \frac{\cos^2 \nu}{B/C} \right) s_J^2 + c_J^2 + \frac{1}{4} \left(\frac{n}{M/C} \right)^2 \right. \quad (\text{D.7})$$

$$\left. \left(1 - 3s_I^2 \sin^2 \phi \right) \left[\left(2 - \frac{B}{C} - \frac{A}{C} \right) (1 - 3c_J^2) - \left(\frac{B}{C} - \frac{A}{C} \right) (3 - 3c_J^2) \cos(2\nu) \right] \right\}$$

$$\phi \triangleq h - \theta. \quad (\text{D.8})$$

Meanwhile, the disturbing potential in solution variables is given by Eq. (59), with u defined in Eq. (76):

$$U = \frac{n^2}{8} (1 - 3s_I^2 \sin^2 \phi) \left\{ (2C - B - A) \cdot \right. \quad (\text{D.9})$$

$$\left[1 - 3 \frac{f}{f+m} \text{dn}^2(u|m) \right] + 3(B - A) \cdot$$

$$\left[1 - \frac{f}{f+m} \text{dn}^2(u|m) \right] \left[1 - 2 \frac{(1+f) \text{sn}^2(u|m)}{1+f \text{sn}^2(u|m)} \right] \left. \right\}$$

$$u = \frac{2K(m)}{\pi}l. \quad (\text{D.10})$$

The full fast-rotating Hamiltonian in solution variables is (Eq. 60):

$$\mathcal{H} = \Phi - nH + U, \quad (\text{D.11})$$

where Φ is given by Eq. (D.4), n is the orbital mean motion, H is the conjugate momentum of h , and U is given by Eq. (D.9).

The $H_{0,2}$ contribution to the averaged Hamiltonian of the first Lie-Deprit transformation is given by Eq. (65), with the auxiliary variable κ defined in Eq. (66):

$$\mathcal{H}_{0,2} = \frac{n^2}{4}\kappa (1 - 3s_I^2 \sin^2 \phi) \quad (\text{D.12})$$

$$\kappa = (B - A) \left\{ \frac{C - A}{B - A} + 1 - 3 \frac{1 + f}{m + f} \left[1 + \frac{C - B}{B} \frac{E(m)}{K(m)} \right] \right\}. \quad (\text{D.13})$$

The W_2 term of the generating function of the first Lie-Deprit transformation is given by Eq. (67). Calculations are facilitated by Eqs. (68) and (35), which give $Z(\Psi|m)$ and Ψ , respectively.

$$W_2 = -\frac{3n^2}{2G} (C - B) A \sqrt{f \frac{1+f}{f+m}} Z(\Psi|m) (1 - 3s_I^2 \sin^2 \phi) \quad (\text{D.14})$$

$$Z(\Psi|m) = E(\Psi|m) - \frac{E(m)}{K(m)} F(\Psi|m) \quad (\text{D.15})$$

$$\Psi = \text{am} \left(-\frac{2K(m)}{\pi} l|m \right). \quad (\text{D.16})$$

Eqs. (77)–(82) give the variable transformations for the first Lie-Deprit transformation, which averages over the angle l :

$$\Delta l = \frac{3 n^2}{4 G^2} (C - B) A \frac{\pi}{2K(m)}. \quad (\text{D.17})$$

$$\left[Z(\Psi|m) - 2(f+m) \frac{dZ(\Psi|m)}{dm} \right] (1 - 3s_I^2 \sin^2 \phi)$$

$$\Delta h = \frac{3 n^2}{4 G^2} (C - B) A \sqrt{f \frac{1+f}{m+f}} Z(\Psi|m) 6c_I \sin^2 \phi \quad (\text{D.18})$$

$$\Delta g = -\frac{3 n^2}{4 G^2} (C - B) A \sqrt{f \frac{1+f}{m+f}} Z(\Psi|m). \quad (\text{D.19})$$

$$(1 - 3s_I^2 \sin^2 \phi) - \frac{L}{G} \Delta l - \frac{H}{G} \Delta h$$

$$\Delta L = \frac{3 n^2}{4 G^2} (C - B). \quad (\text{D.20})$$

$$A \sqrt{f \frac{1+f}{m+f} \frac{2K(m)}{\pi}} \left[\frac{E(m)}{K(m)} - \text{dn}^2(u|m) \right] (1 - 3s_I^2 \sin^2 \phi)$$

$$\Delta G = 0 \quad (\text{D.21})$$

$$\Delta H = \frac{3 n^2}{4 G^2} (C - B) A \sqrt{f \frac{1+f}{m+f}} Z(\Psi|m) 3s_I^2 \sin(2\phi), \quad (\text{D.22})$$

where (Eq. (75))

$$\frac{dZ(\Psi|m)}{dm} = \frac{\text{cn}(u|m)}{2(1-m)}. \quad (\text{D.23})$$

Appendix E

Lara-Ferrer Fast-Rotating Satellite Solution: Spin about Axis of Minimum Inertia

As presented in Lara and Ferrer [134], the transformation from Andoyer variables to fully reduced action-angle variables (“solution variables”) is only applicable to an object spinning about its axis of maximum inertia. Rotation about the axis of minimum inertia may be considered by revising the transformation procedure as follows. (Transformed variables are denoted by a subscript T .)

The problem is reorganized such that the axis of minimum inertia is aligned most closely with the body 3 axis. Under the assumption $A \leq B \leq C$,

$$A_T = C, \quad B_T = B, \quad C_T = A \quad (\text{E.1})$$

$$\omega_{1,T} = \omega_3, \quad \omega_{2,T} = -\omega_2, \quad \omega_{3,T} = \omega_1 \quad (\text{E.2})$$

$$\mathbf{R}_T(1, 1 : 3) = \mathbf{R}(3, 1 : 3) \quad (\text{E.3})$$

$$\mathbf{R}_T(2, 1 : 3) = -\mathbf{R}(2, 1 : 3) \quad (\text{E.4})$$

$$\mathbf{R}_T(3, 1 : 3) = \mathbf{R}(1, 1 : 3). \quad (\text{E.5})$$

The transformed values are converted to Andoyer variables [45], and transformation to action-angle variables proceeds identically to the case of spin about

the maximum-inertia axis. When evaluating the analytical solution, the transformed moments of inertia must be used. Further, it must be remembered that the resulting attitude solutions do not describe the orientation of the original body-fixed frame with respect to the inertial frame. Once the result is calculated in (transformed) action-angle variables and converted to a (transformed) angular velocity vector and rotation matrix, Eqs. (E.1)–(E.3) must be applied to obtain the $\boldsymbol{\omega}$ and \mathbf{R} of the original body-fixed frame with respect to the original inertial frame.

Appendix F

Publications

Publications that contain material presented in this dissertation are listed below. In each instance, Ryan P. Russell supervised the work. Additionally, publications completed while the author was in the graduate program at the University of Texas at Austin, but which are not presented in the dissertation, are listed as “other” work.

Submitted to Refereed Journals

Chapter 2

- **Noble Hatten** and Ryan P. Russell. A smooth and robust Harris-Priester atmospheric density model for low Earth orbit applications. *Adv Space Res.* doi: 10.1016/j.asr.2016.10.015. (online first November 2016)

Chapter 3

- **Noble Hatten** and Ryan P. Russell. Parallel implicit Runge-Kutta methods applied to coupled orbit/attitude propagation. *J Astronaut Sci.* doi: 10.1007/s40295-016-0103-3. (accepted October 2016)

Chapter 4

- **Noble Hatten** and Ryan P. Russell. Decoupled direct state transition matrix calculation with Runge-Kutta methods. *SIAM J Sci Comput.* (under review; submitted November 2016)

Other Refereed Journal Publications

- **Noble Hatten** and Ryan P. Russell. Comparison of three Stark problem solution techniques for the bounded case. *Celest Mech Dyn Astr*, 121:39–60, 2015. doi 10.1007/s10569-014-9586-z.

Conference Proceedings

Chapter 2

- **Noble Hatten** and Ryan P. Russell. A smooth and robust Harris-Priester atmospheric density model. Paper AAS 16-406. In *26th AAS/AIAA Space Flight Mechanics Meeting*, Napa, CA, February 2016.

Chapter 3

- **Noble Hatten** and Ryan P. Russell. Parallel implicit Runge-Kutta methods applied to coupled orbit/attitude propagation. Paper AAS 16-395. In *26th AAS/AIAA Space Flight Mechanics Meeting*, Napa, CA, February 2016.

Chapter 4

- **Noble Hatten** and Ryan P. Russell. Decoupled direct state transition matrix calculation with Runge-Kutta methods. Paper 17-398. In *27th AAS/AIAA Space Flight Mechanics Meeting*, San Antonio, TX, February 2017. (accepted for presentation)

Chapter 5

- **Noble Hatten** and Ryan P. Russell. The eccentric case of a fast-rotating, gravity-gradient-perturbed satellite attitude solution. Paper 17-373. In *27th AAS/AIAA Space Flight Mechanics Meeting*, San Antonio, TX, February 2017. (accepted for presentation)

Chapter 6

- **Noble Hatten** and Ryan P. Russell. A semianalytical technique for six-degree-of-freedom space object propagation. Paper 17-376. In *27th AAS/AIAA Space Flight Mechanics Meeting*, San Antonio, TX, February 2017. (accepted for presentation)

Other Conference Proceedings

- Brandon A. Jones, **Noble Hatten**, Nicholas Ravago, and Ryan P. Russell. Ground-based tracking of geosynchronous space objects with a GM-CPHD filter. In *Advanced Maui Optical and Space Surveillance Technologies (AMOS) Conference*, Maui, HI, September 2016.

- Raymond G. Merrill, Nathan Strange, Min Qu, and **Noble Hatten**. Mars conjunction crewed missions with a reusable hybrid architecture. Paper 8.0104. In *IEEE Aerospace Conference*, Big Sky, MT, March 2015. (peer reviewed)
- **Noble Hatten** and Ryan P. Russell. Application of the Stark problem to space trajectories with time-varying perturbations. Paper AAS 14-230. In *24th AAS/AIAA Space Flight Mechanics Meeting*, Santa Fe, NM, January 2014.

Bibliography

- [1] Space-track. Web site. URL <https://www.space-track.org/auth/login>. Last accessed July 2016.
- [2] Charles H. Acton. Ancillary data services of NASA's Navigation and Ancillary Information Facility. *Planet Space Sci*, 44(1):65–70, 1996.
- [3] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, third edition, 1999. doi: 10.1137/1.9780898719604. URL <http://epubs.siam.org/doi/abs/10.1137/1.9780898719604>.
- [4] Jeffrey M. Aristoff. Method and system for propagating the state of an object and its uncertainty, Dec 2014. US Patent 8,909,588.
- [5] Jeffrey M. Aristoff and Aubrey B. Poore. Implicit Runge-Kutta methods for orbit propagation. In *AAS/AIAA Astrodynamics Specialist Conference*, Minneapolis, MN, August 2012.
- [6] Jeffrey M. Aristoff, Joshua T. Horwood, and Aubrey B. Poore. Orbit and uncertainty propagation: A comparison of Gauss-Legendre-, Dormand-Prince-, and Chebyshev-Picard-based approaches. *Celest Mech Dyn Astr*, 118(1):13–28, 2014.

- [7] Jeffrey M. Aristoff, Joshua T. Horwood, Navraj Singh, and Aubrey B. Poore. Error estimation and control for efficient and reliable orbit (and uncertainty) propagation. In *AAS/AIAA Space Flight Mechanics Meeting*, Santa Fe, NM, January 2014.
- [8] Jeffrey M. Aristoff, Joshua T. Horwood, and Aubrey B. Poore. Implicit-Runge-Kutta-based methods for fast, precise, and scalable uncertainty propagation. *Celest Mech Dyn Astr*, 122(2):169–182, 2015.
- [9] Nitin Arora. *High Performance Algorithms to Improve the Runtime Computation of Spacecraft Trajectories*. Ph.D. Dissertation, Georgia Institute of Technology, August 2013.
- [10] Nitin Arora and Ryan P. Russell. A fast, accurate, and smooth planetary ephemeris retrieval system. *Celest Mech Dyn Astr*, 108(2):107–124, 2010.
- [11] Nitin Arora and Ryan P. Russell. Efficient interpolation of high-fidelity geopotentials. *J Guid Control Dynam*, 39(1):128–143, 2016.
- [12] Nitin Arora, Vivek Vittaldev, and Ryan P. Russell. Parallel computation of trajectories using graphics processing units and interpolated gravity models. *J Guid Control Dynam*, 38(8):1345–1355, 2015.
- [13] Michael Athans, P. Wishner, Richard, and Anthony Bertolini. Suboptimal state estimation for continuous-time nonlinear systems from noisy measurements. *IEEE T Automat Contr*, 13(5):504–514, 1968.
- [14] Armand Awad, Anshu Narang-Siddarth, and Ryan Weisman. The method of multiple scales for orbit propagation with atmospheric drag.

In *AIAA Guidance, Navigation, and Control Conference*, San Diego, CA, January 2016.

- [15] Xiaoli Bai. *Modified Chebyshev-Picard Iteration Methods for Solution of Initial Value and Boundary Value Problems*. Ph.D. Dissertation, Texas A & M University, August 2010.
- [16] Xiaoli Bai and John L Junkins. Modified Chebyshev-Picard iteration methods for orbit propagation. *J Astronaut Sci*, 58(4):583–613, 2011.
- [17] Yaakov Bar-Shalom, X. Rong Li, and Thiagalingam Kirubarajan, editors. *Estimation with Applications to Tracking and Navigation*. John Wiley and Sons, Inc., New York, 2001. pp. 371–420.
- [18] F. Barlier, C. Berger, J. L. Falin, G. Kockarts, and G. Thuillier. A thermospheric model based on satellite drag data. *Ann Geophys*, 34(1): 9–24, 1978.
- [19] Roberto Barrio and Jesús Palacián. High-order averaging of eccentric artificial satellites perturbed by the earth’s potential and air-drag terms. *Proc R Soc A*, 459:1517–1534, 2003.
- [20] Roger R. Bate, Donald D. Mueller, and Jerry E. White. *Fundamentals of Astrodynamics*. Dover Publications, Inc., New York, NY, 1971. pp. 177–212.
- [21] Richard H. Battin. *An Introduction to the Mathematics and Methods of Astrodynamics*. American Institute of Aeronautics and Astronautics, Inc., Reston, VA, revised edition, 1999. pp. 447–450.

- [22] C. Berger, R. Biancale, M. Ill, and F. Barlier. Improvement of the empirical thermospheric model DTM: DTM-94 – a comparative review of various temporal variations and prospects in space geodesy applications. *J Geodesy*, 72(3):161–178, 1998.
- [23] Matthew Berry and Liam M. Healy. Implementation of Gauss-Jackson integration for orbit propagation. *J Astronaut Sci*, 52(3):331–357, 2004.
- [24] Matthew M. Berry. *A Variable-Step Double-Integration Multi-Step Integrator*. Ph.D. Dissertation, Virginia Polytechnic Institute and State University, April 2004.
- [25] Sergio Blanes and Fernando Casas. *A Concise Introduction to Geometric Numerical Integration*. Chapman & Hall/CRC Monographs and Research Notes in Mathematics. CRC Press, Boca Raton, FL, 2016. pp. 51–52.
- [26] Dino Boccaletti and Giuseppe Pucacco. *Theory of Orbits 2: Perturbative and Geometrical Methods*. Springer-Verlag, New York, NY, 1999. pp. 2, 28–36, 40–44, 69–80, 125–178.
- [27] Eric Bois. First-order theory of satellite attitude motion application to hipparcos. *Celestial Mech*, 39:309–327, 1986.
- [28] Eric Bois. Second-order theory of the rotation of an artificial satellite. *Celestial Mech*, 42:141–168, 1988.
- [29] Eric Bois and J. Kovalevsky. Analytical model of the rotation of an artificial satellite. *J Guid Control Dynam*, 13(4):638–643, 1990.

- [30] Bruce R. Bowman, W. Kent Tobiska, Frank A. Marcos, Cheryl Y. Huang, Chin S. Lin, and William J. Burke. A new empirical thermospheric density model JB2008 using new solar and geomagnetic indices. In *AAS/AIAA Astrodynamics Specialist Conference*, Honolulu, HI, August 2008.
- [31] Bruce R. Bowman, W. Kent Tobiska, Frank A. Marcos, and Cesar Valadares. The JB2006 empirical thermospheric density model. *J Atmos Sol-Terr Phy*, 70(5):774–793, 2008.
- [32] Ben K. Bradley, Brandon A Jones, Gregory Beylkin, and Penina Axelrad. A new numerical integration technique in astrodynamics. In *22nd Annual AAS/AIAA Space Flight Mechanics Meeting*, Charleston, SC, January–February 2012.
- [33] Ben K. Bradley, Brandon A. Jones, Gregory Beylkin, Kristian Sandberg, and Penina Axelrad. Bandlimited implicit Runge-Kutta integration for astrodynamics. *Celest Mech Dyn Astr*, 119(2):143–168, 2014.
- [34] Dirk Brouwer. Solution of the problem of artificial satellite theory without drag. *Astron J*, 64(1274):378–396, 1959.
- [35] Dirk Brouwer and Gen-Ichiro Hori. Theoretical evaluation of atmospheric drag effects in the motion of an artificial satellite. *Astron J*, 66(5):193–225, 1961.
- [36] S. Bruinsma. The DTM-2013 thermosphere model. *J Space Weather Space Clim*, A1, 2015.

- [37] S. Bruinsma, G. Thuillier, and F. Barlier. The DTM-2000 empirical thermosphere model with new data assimilation and constraints at lower boundary: Accuracy and properties. *J Atmos Sol-Terr Phy*, 65:1053–1070, 2003.
- [38] S. Bruinsma, N. Sanchez-Ortiz, E. Olmedo, and N. Guijaro. Evaluation of the DTM-2009 thermosphere model for benchmarking purposes. *J Space Weather Space Clim*, 2(A04), 2012.
- [39] H. Martin Bücker and Andre Vehreschild. Sensitivities of the MSIS-86 thermosphere model. In *Proceedings of the 20th European Conference on Modelling and Simulation*, 2006.
- [40] M. P. Calvo, M. A. López-Marcos, and J. M. Sanz-Serna. Variable step implementation of geometric integrators. *Appl Numer Math*, 28:1–16, 1998.
- [41] Stephen J. Casali and William N. Barker. Dynamic calibration atmosphere (DCA) for the high accuracy satellite drag model (HASDM). In *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, Monterey, CA, August 2002.
- [42] M. C. Castro, R. C. Vieira, and E. C. Biscaia, Jr. Automatic differentiation tools in the dynamic simulation of chemical engineering processes. *Braz J Chem Eng*, 17(4–7), 2000.
- [43] Marco M. Castronuovo. Active space debris removal—a preliminary mission analysis and design. *Acta Astron*, 69:848–859, 2011.

- [44] Paul J. Cefola, A. C. Long, and G. Holloway, Jr. The long-term prediction of artificial satellite orbits. In *AIAA 12th Aerospace Sciences Meeting*, Washington, D. C., 1974.
- [45] Alessandra Celletti. *Stability and Chaos in Celestial Mechanics*. Springer-Praxis, Berlin, 2006. pp. 83–88.
- [46] John E. Cochran. Effects of gravity-gradient torque on the rotational motion of a triaxial satellite in a precessing elliptic orbit. *Celestial Mech*, 6:127–150, 1972.
- [47] Bruce A. Conway, editor. *Spacecraft Trajectory Optimization*. Cambridge University Press, New York, 2010. pp. 79–111.
- [48] Jack W. Crenshaw and Philip M. Fitzpatrick. Gravity effects on the rotational motion of a uniaxial artificial satellite. *AIAA J*, 6(11):2140–2145, 1968.
- [49] Howard Curtis. *Orbital Mechanics for Engineering Students*. Elsevier Butterworth-Heinemann, Burlington, MA, 2005. pp. 435–437, 530–533.
- [50] M. E. Davis. *Numerical Methods and Modeling for Chemical Engineers*. John Wiley & Sons, Inc., New York, 1984. pp. 11–21.
- [51] B. De Pontieu. Database of photometric periods of artificial satellites. *Adv Space Res*, 19(2):229–232, 1997.
- [52] Andre Deprit. Canonical transformations depending on a small parameter. *Celestial Mech*, 1:12–30, 1969.

- [53] Andre Deprit and Arnold Rom. The main problem of artificial satellite theory for small and moderate eccentricities. *Celestial Mech*, 2:166–206, 1970.
- [54] Eelco Doornbos. Air density models derived from multi-satellite drag observations. Web site. URL <http://thermosphere.tudelft.nl/acceldrag/index.php>. Last accessed September 2015.
- [55] Eelco Doornbos. *Thermospheric Density and Wind Determination from Satellite Dynamics*. Ph.D. Dissertation, TU Delft, 2011.
- [56] J. R. Dormand and P. J. Prince. Practical Runge-Kutta processes. *SIAM J Sci Stat Comp*, 10(5):977–989, 1989. doi: 10.1137/0910057.
- [57] Douglas W. Dowd and Byron D. Tapley. Density models for the upper atmosphere. *Celestial Mech*, 20:271–295, 1979.
- [58] Alan M. Dunker. The decoupled direct method for calculating sensitivity coefficients in chemical kinetics. *J Chem Phys*, 81(5):2385–2393, 1984.
- [59] Dominic G. B. Edelen. Construction of autonomous canonical systems when the Hamiltonian depends explicitly on time. *Int J Eng Sci*, 26(6): 605–608, 1988.
- [60] E. Fehlberg. Classical fifth-, sixth-, seventh-, and eighth-order Runge-Kutta formulas with stepsize control. Technical Report TR R-287, NASA, 1968.

- [61] J. M. Ferrandiz and M. E. Sansaturio. Elimination of the nodes when the satellite is a non spherical rigid body. *Celest Mech Dyn Astron*, 46:307–320, 1989.
- [62] J. M. Ferrandiz, M. E. Sansaturio, and R. Caballero. On the roto-translatory motion of a satellite of an oblate primary. *Celest Mech Dyn Astron*, 57:189–202, 1993.
- [63] Sebastin Ferrer and Martin Lara. On roto-translatory motion, reductions and radial intermediaries. *J Astronaut Sci*, 139:21–39, 2011.
- [64] Daniel J. Fonte, Beny Neta, Chris Sabol, and D. A. Danielson. Comparison of orbit propagators in the research and development Goddard Trajectory Determination System (R & D GTDS). part I: Simulated data. In *AAS/AIAA Astrodynamics Specialist Conference*, Halifax, Nova Scotia, August 1995.
- [65] Carolin Früh and Moriba K. Jah. Attitude and orbit propagation of high area-to-mass ratio (HAMR) objects using a semi-coupled approach. *J Astronaut Sci*, 60(1):32–50, 2014.
- [66] Carolin Früh, Thomas M. Kececy, and Moriba K. Jah. Coupled orbit-attitude dynamics of high area-to-mass ratio (HAMR) objects: Influence of solar radiation pressure, Earth’s shadow and the visibility in light curves. *Celest Mech Dyn Astron*, 117(1):385–404, 2013.

- [67] K. Fujimoto, D. J. Scheeres, and K. T. Alfriend. Analytical non-linear propagation of uncertainty in the two-body problem. *J Guid Control Dynam*, 35(2):1439–1457, 2011.
- [68] T. Fukushima. Precise and fast computation of a general incomplete elliptic integral of third kind by half and double argument transformations. *J. Comp. Appl. Math.*, 236:1961–1975, 2012.
- [69] T. Fukushima. Fast computation of a general complete elliptic integral of third kind by half and double argument transformations. *J. Comp. Appl. Math.*, 253:142–157, 2013.
- [70] T. Fukushima. Precise and fast computation of Jacobian elliptic functions by conditional duplication. *Numer. Math.*, 2013. doi: 10.1007/s00211-012-0498-0.
- [71] T. Fukushima. Personal ResearchGate web page. Web site, 2014. URL https://www.researchgate.net/profile/Toshio_Fukushima. Last accessed August 2014.
- [72] T. Fukushima and H. Ishizaki. Numerical computation of incomplete elliptic integrals of a general form. *Celest. Mech. Dyn. Astron.*, 59:237–251, 1994.
- [73] Toshio Fukushima. New canonical variables for orbital and rotational motions. *Celest Mech Dynam Astron*, 60(1):57–68, 1994.
- [74] Toshio Fukushima. Simple, regular, and efficient numerical integration of rotational motion. *The Astron J*, 135(6):2298–2322, 2008.

- [75] David E. Gaylor, R. Page, and T. Berthold. Java Astrodynamics Toolkit. Web site. URL <http://jat.sourceforge.net>. Last accessed August 2015.
- [76] David E. Gaylor, Tobias Berthold, and Noriko Takada. JAVA Astrodynamics Toolkit (JAT). In *28th Annual AAS Guidance and Control Conference*, Breckenridge, CO, February 2005.
- [77] David E. Gaylor, Richard C. Page, and Kathryn V. Bradley. Testing of the Java Astrodynamics Toolkit propagator. In *AIAA/AAS Astrodynamics Specialist Conference*, Keystone, CO, August 2006.
- [78] C. W. Gear. *Numerical Initial Value Problems in Ordinary Differential Equations*. Prentice-Hall, Englewood Cliffs, NJ, 1971.
- [79] Giorgio E. O. Giacaglia. Notes on von Zeipel's method. Technical report, Goddard Space Flight Center, Greenbelt, MD, 1964.
- [80] Eberhard Gill. Smooth bi-polynomial interpolation of Jacchia 1971 atmospheric densities for efficient satellite drag computation. Technical Report DLR-GSOC IB 96-1, German Aerospace Center (DLR), 1996.
- [81] Philip E. Gill, Walter Murray, and M. H. Wright. *Practical Optimization*. Elsevier Academic Press, 1981.
- [82] Philip E. Gill, Walter Murray, and Michael A. Saunders. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM J Optimiz*, 12(4):979–1006, 2002.

- [83] D. R. Glandorf. Gravity gradient torque for an arbitrary potential function. *J Guid Control Dynam*, 9(1):122–125, 1986.
- [84] D. E. Goldberg and J. Richardson. Genetic algorithms with sharing for multimodal function optimization. In *Genetic Algorithms and their Applications: Proceedings of the Second International Conference on Genetic Algorithms*, Cambridge, MA, 1987. pp. 41–49.
- [85] Andreas Griewank, David Juedes, and Jean Utke. Algorithm 755: ADOL-C: A package for the automatic differentiation of algorithms written in C/C++. *ACM T Math Software*, 22(2):131–167, 1996.
- [86] Kjell Gustafsson and Gustaf Söderlind. Control strategies for the iterative solution of nonlinear equations in ODE solvers. *SIAM J Sci Comput*, 18(1):23–40, 1997. doi: 10.1137/S1064827595287109.
- [87] Ernst Hairer and Gerhard Wanner. Stiff differential equations solved by Radau methods. *J Comput Appl Math*, 111:93–111, 1999.
- [88] Ernst Hairer, Christian Lubich, and Gerhard Wanner. *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*. Number 31 in Springer Series in Computational Mathematics. Springer-Verlag, Berlin, second edition, 2006. pp. 30–35, 101, 147, 330–332.
- [89] Ernst Hairer, S. P. Norsett, and Gerhard Wanner. *Solving Ordinary Differential Equations 1: Nonstiff Problems*. Number 8 in Springer Series

in Computational Mathematics. Springer-Verlag, Berlin, third edition, 2008. pp. 196–200, 319–325.

- [90] I. Harris and W. Priester. Time-dependent structure of the upper atmosphere. Technical Report TN D-1443, NASA, Goddard Space Flight Center, MD, 1962.
- [91] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applicaitons. *Biometrika*, 57(1):97–109, 1970.
- [92] Noble Hatten and Ryan P. Russell. Parallel implicit Runge-Kutta methods applied to coupled orbit/attitude propagation. In *26th AAS/AIAA Space Flight Mechanics Meeting*, Napa, CA, February 2016.
- [93] Noble Hatten and Ryan P. Russell. Parallel implicit Runge-Kutta methods applied to coupled orbit/attitude propagation. *J Astronaut Sci*, 2016. doi: 10.1007/s40295-016-0103-3.
- [94] William Bailey Heard. *Rigid Body Mechanics: Mathematics, Physics and Applications*. Wiley-VCH, 2006. pp. 88–145.
- [95] A. E. Hedin. A revised thermospheric model based on mass spectrometer and incoherent scatter data – MSIS-83. *J Geophys Res*, 88:10170–10188, 1983.
- [96] A. E. Hedin. MSIS-86 thermospheric model. *J Geophys Res*, 92(A5): 4649–4662, 1987.

- [97] A. E. Hedin, J. E. Salah, J. V. Evans, C. A. Reber, G. P. Newton, N. W. Spencer, D. C. Kayser, D. Alcayde, P. Bauer, L. Cogger, and J. P. McLure. A global thermospheric model based on mass spectrometer and incoherent scatter data, MSIS 1, N2 density and temperature. *J Geophys Res*, 82:2139–2147, 1977.
- [98] Jonathan F. C. Herman, Brandon A. Jones, George H. Born, and Jeffrey S. Parker. A comparison of implicit integration methods for astrodynamics. In *2013 AAS/AIAA Astrodynamics Specialist Conference*, Hilton Head, SC, August 2013.
- [99] Donald L. Hitzl and John V. Breakwell. Resonant and non-resonant gravity-gradient perturbations of a tumbling tri-axial satellite. *Celestial Mech*, 3:346–383, 1971.
- [100] Hani M. Hohmmed, Mostafa K. Ahmed, Ashraf Owis, and Hany Dwidar. Analytical solution of the perturbed orbit-attitude motion of a charged spacecraft in the geomagnetic field. *International Journal of Advanced Computer Science and Applications*, 4(3):272–286, 2013.
- [101] Robert L. Holland and Hans J. Sperling. A first-order theory for the rotational motion of a triaxial rigid body orbiting an oblate primary. *Astron J*, 74(3):490–496, 1969.
- [102] Felix R. Hoots and Ronald L. Roehrich. Spacetrack Report No. 3: Models for propagation of NORAD element sets. Technical report, U. S. Air Force Aerospace Defense Command, Colorado Springs, CO, 1980.

- [103] Felix R. Hoots, Paul W. Schumacher Jr., and Robert A. Glover. History of analytical orbit modeling in the u. s. space surveillance system. *J Guid Control Dynam*, 27(2):174–185, 2004.
- [104] David G. Hull and Walton E. Williamson. Numerical derivatives for parameter optimization. *J Guid Control Dynam*, 2(2):158–160, 1979.
- [105] *Intel Math Kernel Library for Windows OS Developer Guide*. Intel Corporation, 2016. Revision: 049.
- [106] ISC Kosmotras. *Dnepr User’s Guide*. ISC Kosmotras, Moscow, issue 2 edition, Nov. 2001.
- [107] L. G. Jacchia. Static diffusion models of the upper atmosphere with empirical temperature profiles. *Smithsonian Contributions to Astrophysics*, 8:215–257, 1965.
- [108] L. G. Jacchia. Revised static models of the thermosphere and exosphere with empirical temperature profiles. Technical Report Special Report 332, SAO, Cambridge, 1971.
- [109] L. G. Jacchia. Thermospheric temperature, density and composition: New models. *Smithsonian Astrophysical Observatory Special Report*, 375, 1977.
- [110] J. Jackson. Note on the numerical integration of $d^2x/dt^2 = f(x, t)$. *Mon Not R Astron Soc*, 84:602–606, 1924.

- [111] James R. Jancaitis and John L. Junkins. Modeling in n dimensions using a weighting function approach. *J Geophys Res*, 79(23):3361–3366, 1974.
- [112] Laurent O Jay. Structure preservation for constrained dynamics with super partitioned additive Runge-Kutta methods. *SIAM J Sci Comput*, 20(2):416–446, 1998.
- [113] Brandon A. Jones. Orbit propagation using Gauss-Legendre collocation. In *AAS/AIAA Astrodynamics Specialist Conference*, Minneapolis, MN, August 2012.
- [114] Brandon A. Jones and Rodney L. Anderson. A survey of symplectic and collocation integration methods for orbit propagation. In *22nd Annual AIAA/AAS Space Flight Mechanics Meeting*, Charleston, SC, January–February 2012.
- [115] Brandon A. Jones, George H. Born, and Gregory Beylkin. Comparisons of the cubed-sphere gravity model with the spherical harmonics. *J Guid Control Dynam*, 33(2):415–425, 2010.
- [116] J. L. Junkins, M. R. Akella, and K. T. Alfriend. Non-Gaussian error propagation in orbital mechanics. *J Astronaut Sci*, 44(4):541–563, 1996.
- [117] John L. Junkins. Investigation of finite-element representations of the geopotential. *AIAA J*, 14(6):803–808, 1976.
- [118] Ahmed Aly Kamel. Expansion formulae in canonical transformations depending on a small parameter. *Celestial Mech*, 1:190–199, 1969.

- [119] Ahmed Aly Kamel. Lie transforms and the Hamiltonization of non-Hamiltonian systems. *Celestial Mech*, 4:397–405, 1971.
- [120] Marshall H. Kaplan. *Modern Spacecraft Dynamics and Control*. John Wiley and Sons, Inc., New York, 1976. pp. 61–64.
- [121] William M. Kaula. *Theory of Satellite Geodesy*. Dover Publications, Inc., Mineola, NY, 2000 dover edition, 1966. pp. 1–11.
- [122] Michael Octavis Keeve. *Study and Implementation of Gauss Runge-Kutta Schemes and Application to Riccati Equations*. Ph.D. Dissertation, Georgia Institute of Technology, September 1997.
- [123] T. S. Kelso and S. Alfano. Satellite orbital conjunction reports assessing threatening encounters in space. In *AAS/AIAA Space Flight Mechanics Conference*, Copper Mountain, CO, 2005.
- [124] D. Kim, John L. Junkins, James D. Turner, and A. Bani-Younes. Multi-segment adaptive modified Chebyshev Picard iteration method. In *AAS/AIAA Space Flight Mechanics Meeting*, Santa Fe, NM, January 2014.
- [125] Hyun Chul Ko and Daniel J. Scheeres. Maneuver detection with event representation using thrust Fourier coefficients. *J Guid Control Dynam*, 39(5):1080–1091, 2016.
- [126] Darin C. Koblick. Parallel high-precision orbit propagation using the modified Picard-Chebyshev method. Master’s Thesis, California State University, Long Beach, 2012.

- [127] Tomonori Kouya. Practical implementation of high-order multiple precision fully implicit Runge-Kutta methods with step size control using embedded formula. *Int J Numer Meth Applic*, 9(2):85–108, 2013.
- [128] Yoshihide Kozai. The motion of a close Earth satellite. *Astron J*, 64 (1274):367–377, 1959.
- [129] Fred T. Krogh. On testing a subroutine for the numerical integration of ordinary differential equations. *J ACM*, 20(4):545–562, 1973.
- [130] Gregory Lantoine and Ryan P. Russell. A fast second-order algorithm for preliminary design of low-thrust trajectories. In *59th International Astronautical Congress*, Glasgow, UK, September-October 2008.
- [131] Gregory Lantoine, Ryan P. Russell, and Thierry Dargent. Using multi-complex variables for automatic computation of high-order derivatives. *ACM T Math Software*, 38(3):16:1–16:21, 2012.
- [132] Martin Lara and S. Ferrer. Closed form integration of the Hitzl-Breakwell problem in action-angle variables. *Adv Astronaut Sci*, 145:27–39, 2012.
- [133] Martin Lara and S. Ferrer. Complete closed form solution of a tumbling triaxial satellite under gravity-gradient torque. *Adv Astronaut Sci*, 143: 255–274, 2012.
- [134] Martin Lara and S. Ferrer. Closed form perturbation solution of a fast rotating triaxial satellite under gravity-gradient torque. *Cosm Res*, 51 (4):289–303, 2013.

- [135] Martin Lara, Toshio Fukushima, and Sebastin Ferrer. First-order rotation solution of an oblate rigid body under the torque of a perturber in circular orbit. *Astron Astrophys*, 519:1–10, 2010.
- [136] Martin Lara, Juan F. San-Juan, and Luis M. Lopez-Ochoa. Delaunay variables approach to the elimination of the perigee in artificial satellite theory. *Celest Mech Dyn Astron*, 120:39–56, 2014.
- [137] Martn Lara, Toshio Fukushima, and Sebastin Ferrer. Ceres’ rotation solution under the gravitational torque of the Sun. *Mon Not R Astron Soc*, 415:461–469, 2011.
- [138] Clemence Le Fevre, Vincent Morand, Michel Delpech, Clement Gazzino, and Yannick Henriquel. Integration of coupled orbit and attitude dynamics and impact on orbital evolution of space debris. In *AAS/AIAA Space Flight Mechanics Meeting*, Williamsburg, VA, January 2015.
- [139] Eugene Levin, Jerome Pearson, and Joseph Carroll. Wholesale debris removal from LEO. *Acta Astron*, 73:100–108, 2012.
- [140] A. C. Long, J. O. Cappellari, Jr., C. E. Velez, and A. J. Fuchs. Goddard Trajectory Determination System (GTDS) mathematical theory revision 1. Technical Report FDD/552-89/0001, NASA, Goddard Space Flight Center, MD, 1989.
- [141] Brent David Macomber. *Enhancements to the Chebyshev-Picard Iteration Efficiency for Generally Perturbed Orbits and Constrained Dynamical Systems*. Ph.D. Dissertation, Texas A&M University, August 2015.

- [142] F. L. Markley. *Spacecraft Attitude Determination and Control*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1978. pp. 510–522.
- [143] Joaquim R. R. A. Martins, Peter Sturdza, and Juan J. Alonso. The connection between the complex-step derivative approximation and algorithmic differentiation. In *39th Aerospace Sciences Meeting and Exhibit*, Reno, NV, January 2001.
- [144] Joaquim R. R. A. Martins, Peter Sturdza, and Juan J. Alonso. The complex-step derivative approximation. *ACM T Math Software*, 29(3): 245–262, 2003.
- [145] Ravishankar Mathur. *An Analytical Approach to Computing Step Sizes for Finite-Difference Derivatives*. Ph.D. Dissertation, The University of Texas at Austin, 2012.
- [146] Mike McCants. Mike McCants’ BWGS PPAS page. Web site. <https://www.prismnet.com/mmccants/bwgs/index.html>. 29 June 2016.
- [147] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *J Chem Phys*, 21(6):1087–1092, 1953.
- [148] Oliver Montenbruck. Numerical integration methods for orbital motion. *Celest Mech Dyn Astron*, 53:59–69, 1992.
- [149] Oliver Montenbruck and Eberhard Gill. *Satellite Orbits: Models, Methods, and Applications*. Springer-Verlag, Berlin, corrected 2nd printing edition, 2001.

- [150] Paul Nacozy. The intermediate anomaly. *Celest Mech Dyn Astron*, 16: 309–313, 1977.
- [151] NASA. NASA space science data coordinated archive. Web site. URL <http://nssdc.gsfc.nasa.gov/nmc/masterCatalog.do?sc=2009-041B>. Last accessed June 2016.
- [152] National Research Council. *Continuing Kepler's Quest: Assessing Air Force Space Command's Astrodynamics Standards*. The National Academies Press, Washington, D. C., 2012. pp. 19–43.
- [153] Ali H. Nayfeh. *Perturbation Methods*. Wiley-VCH, Weinheim, Germany, 2000. pp. 1–2, 228–307.
- [154] NVIDIA Corporation. CUDA C programming guide. Technical Report PG-02829-001_v7.5, NVIDIA Corporation, Santa Clara, CA, 2015.
- [155] Cesar A. Ocampo. An architecture for a generalized spacecraft trajectory design and optimization system. In *International Conference on Libration Point Missions and Applications*, Girona, Spain, 2002.
- [156] Cesar A. Ocampo and Jean-Philippe Munoz. Variational equations for a generalized spacecraft trajectory model. *J Guid Control Dynam*, 33(5): 1615–1622, 2010.
- [157] Gregory W. Ojakangas, P. Anz-Meador, and H. Cowardin. Probable rotation states of rocket bodies in low Earth orbit. In *13th Annual Advanced Maui Optical and Space Conference*, Maui, HI, 2012.

- [158] OpenMP Architecture Review Board. *OpenMP Application Program Interface, Version 4.0*. OpenMP Architecture Review Board, July 2013.
- [159] Ryan S. Park and Daniel J. Scheeres. Nonlinear mapping of Gaussian statistics: Theory and applications to spacecraft trajectory design. *J Guid Control Dynam*, 29(6):1367–1375, 2006.
- [160] Ryan S. Park and Daniel J. Scheeres. Nonlinear semi-analytic methods for trajectory estimation. *J Guid Control Dynam*, 30(6):1668–1676, 2007.
- [161] I. Pelivan and S. Theil. Higher accuracy modelling of gravity-gradient induced forces and torques. In *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, Keystone, CO, August 2006.
- [162] Etienne Pellegrini and Ryan P. Russell. On the accuracy of trajectory state-transition matrices. In *AAS/AIAA Astrodynamics Specialist Conference*, Vail, CO, August 2015.
- [163] Etienne Pellegrini and Ryan P. Russell. On the accuracy of trajectory state-transition matrices. *J Guid Control Dynam*, to appear.
- [164] J. M. Picone, A. E. Hedin, D. P. Drob, and A. C. Aikin. NRLMSISE-00 empirical model of the atmosphere: Statistical comparisons and scientific issues. *J Geophys Res*, 107(A12):1–16, 2002.
- [165] Edward T. Pitkin. Second transition partial derivatives via universal variables. *J Astronaut Sci*, 13:204–207, 1966.

- [166] P J Prince and J. R. Dormand. High order embedded Runge-Kutta formulae. *J Comput Appl Math*, 7(1):67–75, 1981.
- [167] George J Prokopakis. A decoupled direct method for the solution of ordinary boundary value problems. *Appl Math Model*, 17:499–503, 1993.
- [168] Rien Quiryne. Automatic code generation of implicit Runge-Kutta integrators with continuous output for fast embedded optimization. Master’s Thesis, Katholieke Universiteit Leuven, 2012.
- [169] Krishnan Radhakrishnan and A.C. Hindmarsh. Description and use of LSODE, the Livermore Solver for Ordinary Differential Equations. Technical Report UCRL-ID-113855, Lawrence Livermore National Laboratory, Livermore, CA, 1993.
- [170] J. L. Read, A. Bani Younes, and J. L. Junkins. Efficient orbit propagation of orbital elements using modified Chebyshev Picard iteration method. In *International Conference on Computational & Experimental Engineering and Sciences*, Reno, NV, August 2015.
- [171] Julie L. Read, Ahmad Bani Younes, Brent Macomber, James Turner, and John L. Junkins. State transition matrix for perturbed orbital motion using modified Chebyshev Picard iteration. *J Astronaut Sci*, 2015. doi: 10.1007/s40295-015-0051-3.
- [172] Ricardo Leon Restrepo Gomez. Automatic algorithm for accurate numerical gradient calculation in general and complex spacecraft trajectories. Master’s Report, University of Texas at Austin, 2010.

- [173] Charles E. Roberts. An analytical model for upper atmosphere densities based upon Jacchia's 1970 models. *Celestial Mech*, 4:368–377, 1971.
- [174] Juan F. San-Juan, Luis M. Lopez, and Rosario Lopez. Higher-order analytical attitude propagation of an oblate rigid body under gravity-gradient torque. *Math Probl Eng*, 2012:1–15, 2012.
- [175] Adrian Sandu and Philipp Miehe. Forward, tangent linear, and adjoint Runge-Kutta methods for stiff chemical kinetic simulations. *Int J Comput Math*, 87(11):2458–2479, 2010.
- [176] Adrian Sandu, D. Daescu, and G. R. Carmichael. Direct and adjoint sensitivity analysis of chemical kinetic systems with KPP: 1 – theory and software tools. Technical report, Michigan Technological University, April 2002.
- [177] Fabio Santoni, Emiliano Cordelli, and Fabrizio Piergentili. Determination of disposed-upper-stage attitude motion by ground-based optical observations. *J Spacecr Rockets*, 50(3):701–708, 2013.
- [178] L. Sehnal. Model of the upper total density distribution TD88. Technical Report 16, Department of Astronomy – Beograd, 1988.
- [179] Srinivas J. Setty, Paul J. Cefola, Oliver Montenbruck, Hauke Fiedler, and Martin Lara. Investigating the suitability of analytical and semi-analytical satellite theories for space object catalogue maintenance in geosynchronous regime. In *AAS/AIAA Astrodynamics Specialist Conference*, Hilton Head, SC, 2013.

- [180] L. F. Shampine and M. W. Reichelt. The MATLAB ODE suite. *SIAM J Sci Comput*, 18(1):1–22, 1997.
- [181] Mark F. Storz, Bruce R. Bowman, and James I. Branson. High accuracy satellite drag model (HASDM). In *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, Monterey, CA, August 2002.
- [182] B. Surlan and S. Segan. TD-88Up: Upgraded neutral Earth’s thermosphere total density TD-88 model. *Serbian Astronomical Journal*, 178: 57–63, 2009.
- [183] Byron D. Tapley, Bob E. Schutz, and George H. Born. *Statistical Orbit Determination*. Elsevier Academic Press, Amsterdam, 2004. pp. 485–486, 511–515.
- [184] Brian Tolman, R. Benjamin Harris, Tom Gaussiran, David Munton, Jon Little, Richard Mach, Scot Nelsen, and Brent Renfro. The GPS Toolkit: Open Source GPS Software. In *Proceedings of the 16th International Technical Meeting of the Satellite Division of the Institute of Navigation*, Long Beach, California, September 2004.
- [185] James D. Turner, Manoranjan Majji, and John L. Junkins. High-order state and parameter transition tensor calculations. In *AIAA/AAS Astrodynamics Specialist Conference*, Honolulu, HI, August 2008.
- [186] United Launch Alliance. *Atlas V Launch Services User’s Guide*. United Launch Alliance, Centennial, CO, March 2010.

- [187] United States Naval Observatory. Approximate solar coordinates. Website, 2012. URL <http://aa.usno.navy.mil/faq/docs/SunApprox.php>. Last accessed July 2015.
- [188] Hodei Urrutxua, Claudio Bombardelli, Javier Roa, and Juan Luis Gonzalo. Quantification of the performance of numerical orbit propagators. In *AAS/AIAA Space Flight Mechanics Conference*, Napa, CA, February 2016.
- [189] David Vallado and Paul Crawford. SGP4 orbit determination. In *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, Honolulu, HI, August 2008.
- [190] David Vallado, Paul Crawford, Richard Hujsak, and T. S. Kelso. Revisiting Spacetrack Report #3: Rev 2. In *AIAA/AAS Astrodynamics Specialist Conference*, Keystone, CO, August 2006.
- [191] David A. Vallado and David Finkleman. A critical assessment of satellite drag and atmospheric density modeling. *Acta Astronaut*, 95:141–165, 2014.
- [192] Jozef C. Van der Ha. Perturbation solution of attitude motion under body-fixed torques. *Acta Astron*, 12(10):861–869, 1985.
- [193] P. J. van der Houwen and D. P. Sommeijer. Parallel iteration of high-order Runge-Kutta methods with stepsize control. *J Comput Appl Math*, 29(1):111–127, 1990.

- [194] Rudolph van der Merwe. *Sigma-point Kalman Filters for Probabilistic Inference in Dynamic State-space Models*. Ph.D. Dissertation, Oregon Heath & Science University, 2004.
- [195] Ramses van Zon and Jeremy Schofield. Numerical implementation of the exact dynamics of free rigid bodies. *J Comput Phys*, 225:145–164, 2007.
- [196] Vivek Vittaldev. *Uncertainty Propagation and Conjunction Assessment for Resident Space Objects*. Ph.D. Dissertation, The University of Texas at Austin, 2015.
- [197] A. Wächter and L. T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math Program*, 106(1):25–57, 2006.
- [198] Charles J. Wetterer and Moriba K. Jah. Attitude determination from light curves. *J Guid Control Dynam*, 32(5):1648–1651, 2009.
- [199] Charles J. Wetterer, John L. Crassidis, Richard Linares, Thomas M. Kelecy, Marek K. Ziebart, Moriba K. Jah, and Paul J. Cefola. Refining space object radiation pressure modeling with bidirectional reflectance distribution functions. *J Guid Control Dynam*, 37(1):185–196, 2014.
- [200] Niklaus Wirth. A plea for lean software. *Computer*, 28(2):64–68, 1995.
- [201] Wolfram. Wolfram language & system documentation center: ImplicitRungeKutta method for NDSolve. Web site, 2015. URL <https://reference.wolfram.com/language/tutorial/NDSolveImplicitRungeKutta.html>. Last accessed November 2015.

- [202] James Woodburn and Sergei Tanygin. Efficient numerical integration of coupled orbit and attitude trajectories using an Encke type correction algorithm. In *AAS/AIAA Astrodynamics Specialist Conference*, Quebec City, 2001.
- [203] T. Yanagisawa and H. Kurosaki. Shape and motion estimate of LEO debris using light curves. *Adv Space Res*, 50(1):136–145, 2012.
- [204] M. C. Zanardi and L. Simal Moreira. Analytical attitude propagation with non-singular variables and gravity gradient torque for spin stabilized satellite. *Adv Space Res*, 40:11–17, 2007.
- [205] M. C. Zanardi and R. Vilhena de Moraes. Analytical and semi-analytical analysis of an artificial satellite’s rotational motion. *Celest Mech Dyn Astron*, 75:227–250, 2000.
- [206] M. C. Zanardi, I. M P. Quirelli, and H. K. Kuga. Analytical attitude prediction of spin stabilized spacecrafts perturbed by magnetic residual torque. *Adv Space Res*, 36:460–465, 2005.
- [207] Hong Zhang and Adrian Sandu. FATODE: A library for forward, adjoint, and tangent linear integration of ODEs. *SIAM J Sci Comput*, 36(5):504–523, 2014. doi: 10.1137/130912335.

Vita

Noble Hatten was born in Austin, Texas and has lived there his entire life. He earned a B.S. with highest honors in Aerospace Engineering from the University of Texas at Austin in 2011; his M.S. in the same field was obtained from UT Austin the following year, under the supervision of Dr. Cesar Ocampo. In 2013, Noble began work toward his Ph.D. at UT Austin, supervised by Dr. Ryan P. Russell. During his time as a student, Noble worked as a research assistant, teaching assistant, and assistant instructor, and interned at NASA Langley Research Center and the Institute for Computational Engineering and Sciences.

Permanent address: noble.hatten@gmail.com

This dissertation was typeset with L^AT_EX[†] by the author.

[†]L^AT_EX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's T_EX Program.