

Alessandro Masat

DYNAMICAL AND COMPUTATIONAL  
PERSPECTIVES ON INTERPLANETARY  
ORBITAL UNCERTAINTY  
PROPAGATION







**POLITECNICO**  
MILANO 1863



DAER - Dipartimento di Scienze e Tecnologie Aerospaziali  
Politecnico di Milano

Doctoral Programme in Aerospace Engineering

**DYNAMICAL AND COMPUTATIONAL PERSPECTIVES  
ON INTERPLANETARY ORBITAL UNCERTAINTY  
PROPAGATION**

Doctoral dissertation of:  
**Alessandro Masat**

Supervisor:  
**Prof. Camilla Colombo**

Co-supervisor:  
**Dr. Arnaud Boutonnet**

Academic year 2022/2023 - Cycle XXXV

Copyright: © September 2023 by Alessandro Masat  
All rights reserved.

Doctoral issertation of:	<b>Alessandro Masat</b>
Supervisor:	<b>Prof. Camilla Colombo</b>
Co-supervisor:	<b>Dr. Arnaud Boutonnet</b>
Doctoral program coordinator:	<b>Prof. Pierangelo Masarati</b>
External reviewer:	<b>Prof. Giulio Baù</b>
External reviewer:	<b>Prof. Juan Félix San Juan Díaz</b>

Alessandro Masat, "*Dynamical and computational perspectives on inter-planetary orbital uncertainty propagation*", PhD Thesis, 2023, Politecnico di Milano, Supervisor: Prof. Camilla Colombo, Co-supervisor: Dr. Arnaud Boutonnet

---

## ABSTRACT

---

The analysis of modern space missions must not only fulfill scientific and operational tasks, but also assess the compliance thereof with a certain set of requirements. Different constraints exist for different objectives, aiming in general at safeguarding the space environment, to different extents. Space debris mitigation policies exist to prevent the further contamination of Earth orbits, already crowded with spacecrafts and fragments from past collisions, resulting from decades of unregulated use of space. Planetary protection guidelines apply instead in the interplanetary domain, addressing the need of preserving other planetary environments from the biological contamination with traces of life from Earth.

Despite the fundamentally different objectives, space debris mitigation policies and planetary protection analysis result in alike orbital dynamics studies. Both cases need to assess the effects of uncertainties, to model or avoid collisions with existing debris in the former, and to avoid impacts with celestial bodies of interest in the latter. Planetary protection's high accuracy requirements make the uncertainty propagation task a computationally intensive problem, truly becoming a limiting factor for most orbit-related analyses. Hence, the efficient and accurate simulation of orbital uncertainties has become an essential enabling feature in mission design, as well as the only mean to model and understand the evolution of Earth's debris environment.

This dissertation focuses on the propagation of interplanetary uncertainties, applying a multilateral approach to study some dynamical, statistical, and computational aspects. The proposed research is not limited to in-depth developments of the different fields, rather it highlights interconnections, mutual constraints, and limitations thereof.

Kustaanheimo-Stiefel variables are adopted to gain both in accuracy and efficiency of the single simulations, also assessing the influence that this formulation of the dynamics has in the uncertainty description. The effect of flyby events on the uncertainty is also reviewed, as a precursor for impacts in planetary protection, and for the scattering that it introduces in probability distributions. Aiming once more at increasing the efficiency of uncertainty propagation, the potential of the Picard-Chebyshev numerical scheme is assessed, as an alternative to traditional step-based integration methods. In combination with all the just mentioned aspects, parallel and GPU computing approaches are also discussed and devised.



---

## SOMMARIO

---

L'analisi di moderne missioni spaziali non deve assolvere esclusivamente a compiti scientifici e operazionali, ma anche garantire il rispetto di molti requisiti delle stesse. Diversi vincoli esistono in risposta a diversi obiettivi, con lo scopo generale di tutelare l'ambiente spaziale, in modi diversi. Le politiche di mitigazione di detriti spaziali esistono per prevenirne l'ulteriore accumulo in orbite terrestri, le quali sono già affollate da veicoli spaziali e frammenti originati da passate collisioni, frutto di decenni di utilizzo non regolato dello spazio. Le linee guida di protezione planetaria si applicano invece nell'ambiente interplanetario, rispondendo all'esigenza di preservare gli ecosistemi di altri pianeti dalla potenziale contaminazione biologica con forme di vita terrestri.

Nonostante gli obiettivi siano fundamentalmente diversi, le politiche di mitigazione di detriti spaziali e l'analisi di protezione planetaria convergono in simili studi relativi alla dinamica orbitale. In entrambi i casi, l'effetto di incertezze deve essere valutato, per modellare o evitare collisioni con i detriti spaziali nel primo, per evitare l'impatto con corpi celesti di interesse nel secondo. L'alta precisione richiesta nella protezione planetaria rende la propagazione dell'incertezza un problema ad alto costo computazionale, rendendola un vero fattore limitante per molte analisi connesse alla dinamica orbitale. Di conseguenza, l'efficiente e precisa simulazione di incertezze orbitali è diventata una parte essenziale e abilitante nel disegno di missioni spaziali, così come l'unico mezzo per modellare e comprendere l'evoluzione dei detriti spaziali attorno alla Terra.

Questa tesi si focalizza nella propagazione interplanetaria delle incertezze, applicando un approccio multidisciplinare per studiarne alcuni aspetti dinamici, statistici e computazionali. La ricerca proposta non si limita a sviluppi concentrati nei diversi aspetti, invece ne evidenzia le interconnessioni, i vincoli reciproci e le limitazioni.

La formulazione della dinamica di Kustaanheimo e Stiefel viene adottata per incrementare sia l'accuratezza che l'efficienza delle singole simulazioni, valutandone anche l'influenza nella descrizione dell'incertezza. Anche l'effetto sull'incertezza di passaggi ravvicinati con i pianeti viene rivisto, sia come precursore di impatti nell'analisi di protezione planetaria, che per la dispersione che introduce nelle distribuzioni di probabilità. Sempre con l'obiettivo di migliorare l'efficienza della propagazione dell'incertezza, il potenziale dello schema numerico di Picard e Chebyshev viene valutato, come alternativa ai tradizionali metodi di integrazione basati sugli step. Combinandoli

con tutti gli aspetti appena citati, anche approcci di calcolo parallelo e basato sulle GPU vengono discussi e proposti.

---

## ACKNOWLEDGMENTS

---

The research leading to these results has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme as part of project COMPASS (Grant agreement No 679086), [www.compass.polimi.it](http://www.compass.polimi.it), and from the European Space Agency (ESA) through the Open Space Innovation Platform (OSIP) co-funded research project "Robust trajectory design accounting for generic evolving uncertainties", Contract No. 4000135476/21/NL/GLC/my.

This research was supported by grants from NVIDIA® and utilized the NVIDIA RTX A6000 graphics card, awarded by NVIDIA®'s Applied Research Accelerator Programme to the project "GPU-accelerated algorithms for orbital uncertainty simulation applications".





---

## CONTENTS

---

Abstract	v
Sommario	vii
<b>1 Introduction</b>	<b>1</b>
1.1 Background	3
1.1.1 High-fidelity approaches for PP/SDM	5
1.1.2 Other Uncertainty Propagation techniques	6
1.1.3 A multidisciplinary problem	8
1.2 Motivation, objectives, and applications	9
1.3 Developed methods and implementation	10
1.4 Thesis organization	12
1.5 Contributions	13
<b>2 Kustaanheimo-Stiefel variables</b>	<b>17</b>
2.1 Regularization techniques	17
2.1.1 Overview of Regularization approaches	17
2.1.2 KS Regularization	21
2.2 KS variables for interplanetary propagation	26
2.2.1 Barycentric KS formulation	26
2.2.2 Adaptive non-dimensionalisation	28
2.2.3 Event interface, flyby detection and Center of integration switch	29
2.2.4 Fibration optimisation for numerical performance	30
2.3 KS – Cartesian uncertainty mapping	33
2.3.1 Fibration of the KS space and uncoupled dimension raising	34
2.3.2 Transformation methods	37
2.4 KS variables for uncertainty propagation	45
2.4.1 Stability of the KS equations of motion	46
2.4.2 Fictitious time synchronism on flyby propagation and post-encounter scattering mitigation	46
2.5 Applications	47
2.5.1 Single simulation accuracy and performance	47
2.5.2 Solar Orbiter’s upper stage of launcher MC PP analysis	52
2.5.3 Covariance transformation	55
2.5.4 KS variables for uncertainty propagation	60
2.6 Summary	66
2.6.1 KS variables for interplanetary simulations and PP/SDM	66
2.6.2 KS - Cartesian uncertainty mapping	69

2.6.3	KS variables and continuity of the propagated uncertainty . . . . .	70
3	Encounter characterization and keyhole maps . . . . .	71
3.1	Encounters and Sphere of Influence definitions . . . . .	72
3.1.1	Sphere of Influence and Hill's sphere . . . . .	74
3.1.2	Numerical investigations . . . . .	74
3.2	Analytical methods for flyby perturbations . . . . .	76
3.2.1	Third body perturbation . . . . .	77
3.2.2	Arbitrary order analytical procedure . . . . .	78
3.2.3	Hori's approach to the main problem . . . . .	79
3.2.4	Hori's approach for the third body perturbation: boundary conditions . . . . .	81
3.3	Dynamic flyby characterisation: Jacobian Eigenvalues . . . . .	91
3.3.1	Analytical spheroidal locus of points . . . . .	92
3.3.2	Dynamical analysis of the Jacobian eigenvalues . . . . .	93
3.4	Application: keyhole maps . . . . .	99
3.4.1	Detection of shallow encounters . . . . .	99
3.4.2	Encounter phasing in the Restricted Three Body Problem . . . . .	102
3.4.3	Keyhole maps and generalized orbital resonances . . . . .	107
3.5	Summary . . . . .	112
3.5.1	Analytical methods for flyby perturbations . . . . .	112
3.5.2	Dynamic flyby characterisation: Jacobian Eigenvalues . . . . .	113
3.5.3	Keyhole Map . . . . .	113
4	Picard-Chebyshev integration and augmentation . . . . .	117
4.1	PC numerical scheme . . . . .	118
4.1.1	Matrix form for vectorized and parallel computation . . . . .	119
4.1.2	Latest developments of the method . . . . .	121
4.2	Fixed-point high-precision resonant flyby optimisation . . . . .	121
4.2.1	B-plane search of optimal resonant flyby exit state . . . . .	123
4.2.2	B-plane for backward-recursive flyby design . . . . .	126
4.3	PC augmentation for large sets of initial conditions . . . . .	133
4.3.1	One-level augmentation . . . . .	133
4.3.2	Two-level and multi-level augmentation . . . . .	134
4.3.3	CPU and GPU implementation paradigms . . . . .	134
4.3.4	Sequential Augmented PC workflow . . . . .	135
4.3.5	OpenMP parallelized Augmented PC workflow . . . . .	135
4.3.6	CUDA Augmented PC workflow . . . . .	136
4.4	Applications . . . . .	144
4.4.1	Single trajectory propagation . . . . .	144
4.4.2	PC for Solar Orbiter-like flyby optimization . . . . .	146
4.4.3	CPU and GPU performance of the augmented PC method . . . . .	153

4.5	Summary . . . . .	164
4.5.1	Fixed-point high-precision resonant flyby optimisation . . . . .	164
4.5.2	PC augmentation for large sets of initial conditions	165
5	KS variables in CUDAjectory . . . . .	167
5.1	CUDAjectory general architecture . . . . .	167
5.2	Implementation of the KS dynamics . . . . .	168
5.2.1	High-level implementation . . . . .	169
5.2.2	Low-level code extension . . . . .	170
5.3	Application: JUICE debris mitigation analysis . . . . .	174
5.4	Summary . . . . .	177
6	Conclusion . . . . .	179
6.1	Summary and findings . . . . .	179
6.1.1	KS variables for uncertainty propagation . . . . .	179
6.1.2	Jacobian spheroids and the Keyhole Map . . . . .	180
6.1.3	Picard-Chebyshev for uncertainty propagation . . . . .	181
6.2	Limitations and applicability . . . . .	181
6.3	Remarks for future works . . . . .	182
A	Quaternion and tensor algebra . . . . .	185
A.1	Quaternion algebra . . . . .	185
A.2	Tensor statistics . . . . .	186
A.2.1	Tensor algebra . . . . .	186
A.2.2	Tensor description of statistical moments . . . . .	187
B	Perturbations by Lie Transforms . . . . .	189
B.1	Lie transforms summary . . . . .	189
B.2	Perturbations approach . . . . .	190
C	Parallel and GPU computing fundamentals . . . . .	193
C.1	Multi-CPU shared memory parallelism: OpenMP . . . . .	193
C.2	Parallel reduction . . . . .	194
C.3	GPU computing . . . . .	194
C.4	Main programming paradigms and the CUDA API . . . . .	196
C.5	Concurrency and advanced features . . . . .	197
	Bibliography . . . . .	199

---

LIST OF FIGURES

---

Figure 1.1	The aspects characterizing orbital uncertainty propagation. . . . .	9
Figure 2.1	Fibration visualization as function of $\varphi$ (color scale), for the first three KS vector components. Generator points refer to the values for which $\varphi = 0$ . . . . .	24
Figure 2.2	Time steps dependence on the fibration parameter $\varphi$ . . . . .	33
Figure 2.3	Non-dimensional magnitude of the minimum magnitude element (black), compared against the magnitude variation of all the KS state elements (grey). . . . .	33
Figure 2.4	Position error evolution for the different force models and formulations, with respect to JPL's data. . . . .	51
Figure 2.5	Relative position error evolution with higher precision relativistic simulation. . . . .	52
Figure 2.6	Superposition of Cartesian (blue) and KS (red) uncertainty distributions. . . . .	55
Figure 2.7	Superposition of Cartesian (blue) and KS (red) probability density estimations. . . . .	56
Figure 2.8	Accuracy of the UT-based technique against increasing number of samples for the MC-based conversion. . . . .	58
Figure 2.9	Accuracy of the EKF-based technique against UT-based conversion. . . . .	59
Figure 2.10	Physical time-synchronous evolution of the semi-major axis, for the flyby leg. . . . .	61
Figure 2.11	Random trajectory sample distance from the Moon (blue) and Earth (red). . . . .	62
Figure 2.12	Fictitious time-synchronous evolution of the semi-major axis, for the flyby leg. . . . .	62
Figure 2.13	Physical time-synchronous evolution of the Cartesian $x$ coordinate, for the flyby leg, in the Earth-Centered ECLIPJ2000[2] reference frame. . . . .	63
Figure 2.14	Fictitious time-synchronous evolution of the first KS coordinates, for the flyby leg. . . . .	63
Figure 2.15	Different fictitious time spans, for the post-flyby leg. . . . .	64
Figure 2.16	Fictitious time-synchronous evolution of the physical time, for the post-flyby leg. . . . .	65

Figure 2.17	Physical time-synchronous evolution of the Cartesian $x$ coordinate, for the post-flyby leg, in the SSB-Centered ECLIPJ2000[2] reference frame. . . . .	65
Figure 2.18	Fictitious time-synchronous evolution of one of the KS coordinates, for the post-flyby leg. . . . .	66
Figure 2.19	Initial window of the fictitious time-synchronous evolution of the first KS coordinates, for the post flyby leg. . . . .	67
Figure 2.20	Initial window of the fictitious time-synchronous evolution of the second KS coordinates, for the post flyby leg. . . . .	67
Figure 2.21	Speedup provided by the different formulations and implementations. . . . .	68
Figure 3.1	Distance from Earth and eigenvalue ratio evolution for the asteroid 2010RF <sub>12</sub> the Sun-Earth case. . . . .	75
Figure 3.2	Errors of Hori's solution [66] for the $J_2$ perturbation against the numerical simulations of the dynamics including Keplerian hyperbolic and $J_2$ effects. . . . .	81
Figure 3.3	Evolution of $\cos \psi(t)$ within Venus' SOI. . . . .	86
Figure 3.4	Numerical accuracy assessment of the Legendre polynomial expansion, integration starting from the entrance of the SOI. . . . .	87
Figure 3.5	Numerical accuracy assessment of the Legendre polynomial expansion, integration starting from the hyperbola pericenter and going forward in time. . . . .	87
Figure 3.6	Evolution of the position errors within Venus' SOI, setting $t_0$ as the entrance of the SOI. . . . .	88
Figure 3.7	Evolution of the position errors within Venus' SOI, setting $t_0$ as the hyperbola's pericenter. . . . .	89
Figure 3.8	Evolution of the position errors within Venus' SOI, setting $t_0 = t_1^*$ as the time when $\cos \psi(t_1^*) = -1/\sqrt{3}$ . . . . .	89
Figure 3.9	Zoom of Figure 3.8 nearby $t_0 = t_1^*$ . . . . .	90
Figure 3.10	Evolution of the position errors within Venus' SOI, setting $t_0 = t_2^*$ as the time when $\cos \psi(t_2^*) = 1/\sqrt{3}$ . . . . .	90
Figure 3.11	Zoom of Figure 3.10 nearby $t_0 = t_2^*$ . . . . .	91
Figure 3.12	Spheroidal loci of points and values of the parameter $\gamma$ for the Sun-Earth case. . . . .	94
Figure 3.13	Spheroidal loci of points and values of the parameter $\gamma$ for the Sun-Jupiter case. . . . .	95

Figure 3.14	Jupiter’s Jacobian percent error (color scale), compared against Hill’s surfaces (dotted) and SOI (dashed). . . . .	97
Figure 3.15	Jupiter’s Jacobian percent error (color scale), compared against Hill’s surfaces (dotted), SOI (dashed), and analytical loci of points (red). . .	97
Figure 3.16	Jupiter’s Jacobian percent error (color scale), compared against Hill’s surfaces (dotted), SOI (dashed), analytical loci of points (red), and zero-velocity surfaces (grey/shadowed). . . . .	98
Figure 3.17	Sample spheroids for $\gamma = 1, 10, 100$ and SOI, centered on Earth in the Sun-Earth synodic frame.	99
Figure 3.18	Orbital period, distance from Venus, and Jacobian eigenvalues evolution. Example from the PP analysis of Solar Orbiter. . . . .	100
Figure 3.19	Orbital period and Distance from Venus and Earth. Example from the PP analysis of Solar Orbiter. . . . .	101
Figure 3.20	Aggregate orbital period variations for MC simulations of, Apophis, 2010RF <sub>12</sub> , Solar Orbiter, and JUICE. . . . .	102
Figure 3.21	Visualization of phased trajectories with the $\gamma = 100$ spheroid, for $C_J = 2.95$ and the 3 : 2 resonance trajectory with Earth. . . . .	104
Figure 3.22	Visualization of phased trajectories (colored dots) with the $\gamma = 100$ spheroid, for $C_J = 2.95$ and the 3 : 2 resonance trajectory with Earth. Inner and outer spheroid angles marked with the black lines. . . . .	105
Figure 3.23	Link between the spheroid angle $\theta$ and the rotating longitude $l = \theta_{outer}$ in the case of Earth.	106
Figure 3.24	Topology of spheroid-phased trajectories. . . . .	107
Figure 3.25	Surrogate model for the impact probability density. . . . .	109
Figure 3.26	Sun-Earth Keyhole Map up to one encounter and 11 year time span. . . . .	110
Figure 3.27	Keyhole impacting trajectory prediction examples. . . . .	111
Figure 3.28	New outgoing trajectories after the encounter.	114
Figure 4.1	Graphical representation of the reference frame of analysis. Picture re-drawn based on original from [25]. . . . .	124

Figure 4.2	Visual accuracy improvement of the b-plane circle model. The analytical belts are bounded by the black circles, the yellow dots highlight the numerically detected resonances on the whole simulated cloud of initial conditions. . . . .	126
Figure 4.3	Block-scheme diagram of the unperturbed design algorithm developed in [95]. . . . .	128
Figure 4.4	Block-scheme diagram of features and steps embedded in the solution of the optimization problem of Equation (4.17). . . . .	131
Figure 4.5	Standard PC workflow. . . . .	135
Figure 4.6	Augmented PC workflow. . . . .	135
Figure 4.7	Augmented and OpenMP <sup>®</sup> parallelized PC workflow. . . . .	136
Figure 4.8	Augmented CUDA <sup>®</sup> PC workflow. . . . .	136
Figure 4.9	Dynamics kernel memory management. . . . .	141
Figure 4.10	PC and Runge-Kutta RK78 integration errors, as relative position difference with respect to JPL's data for the asteroid 2010RF <sub>12</sub> , for the pre-flyby (4.10a), flyby (4.10b) and post-flyby legs (4.10c), as well as globally for the whole integration span (4.10d). The color scale is the same for all the sub-figures, and reported in (4.10d). . . . .	146
Figure 4.11	PC serial execution runtimes for the asteroid 2010RF <sub>12</sub> , reported as relative runtime with respect to the maximum runtime obtained for the different number of Chebyshev nodes. . . . .	147
Figure 4.12	Visual representation of the b-plane pruning strategy. The pruning point corresponds to the blue dot, whereas the optimized point is depicted in dark orange. . . . .	152
Figure 4.13	Design difference with respect to relativistic simulation between V <sub>2</sub> and V <sub>3</sub> for the two node cases, and without manoeuvre. . . . .	153
Figure 4.14	Solar Orbiter's continuous first resonant phase with Venus. . . . .	153
Figure 4.15	Errors of C augmented and CUDA <sup>®</sup> programs, as the average of the error of all the states in the augmented systems. . . . .	155
Figure 4.16	Errors of C augmented and CUDA <sup>®</sup> programs, as the average of the error of all the states in the augmented systems. . . . .	156
Figure 4.17	Augmented system and independent integrations C code runtime comparison with OpenMP <sup>®</sup> parallelization and varying number of threads. . . . .	157

Figure 4.18	Augmented system and independent integrations C code speedup comparison with OpenMP <sup>®</sup> parallelization and varying number of threads. . . . .	157
Figure 4.19	Speedup comparison among C and CUDA <sup>®</sup> implementations. . . . .	159
Figure 5.1	CUDAjectory overall structure. Picture from [57].	168
Figure 5.2	ClusterSimulator structure in CUDAjectory. Picture from [57]. . . . .	169
Figure 5.3	CUDAjectory overall extension logic. . . . .	169
Figure 5.4	CUDAjectory Simulator extension logic. Picture modified from [57]. . . . .	170
Figure 5.5	Position error evolution example, KS and Cartesian coordinates in CUDAjectory. . . . .	175
Figure C.1	Parallel reduction graphical scheme. . . . .	195
Figure C.2	CPU vs GPU architecture difference graphical scheme. Picture from [112]. . . . .	195
Figure C.3	GPU program basic flow. . . . .	196

---

LIST OF TABLES

---

Table 1.1	Example of PP compliant impact probability thresholds for ESA missions. . . . .	4
Table 2.1	Simulation performance benchmark, AUJ-COWELL-FIXED, average of 200 runs. . . . .	48
Table 2.2	Simulation performances, EN-COWELL-FIXED, average of 200 runs. . . . .	49
Table 2.3	Simulation performances, EN-KS-SWITCH, average of 200 runs. . . . .	50
Table 2.4	Initial state of SolO's upper stage of launcher, J2000 reference frame [29]. . . . .	53
Table 2.5	Elements of the covariance matrix of SolO's upper stage of launcher, J2000 reference frame [29]. . . . .	53
Table 2.6	Simulation outcome of the uncertainty barycenter of SolO's upper stage of launcher. . . . .	54
Table 2.7	PP analysis of SolO's upper stage of launcher [29]	54
Table 3.1	Initial state of the test particle, ECLIPJ2000 reference frame centered on Venus [2]. . . . .	85
Table 4.1	Optimization levels for the full design of the arc $j$ to $j + 1$ . . . . .	132
Table 4.2	Flyby $V_3$ entrance state. . . . .	148



Table 4.3	Retrieved optimal b-plane coordinates $(\xi^*, \zeta^*)$ , exit time $t_{\text{out}}^{(j)}$ , and planetocentric velocity $\mathbf{U}^*$ . . . . .	148
Table 4.4	Maneuvering point, apocentre of unperturbed initial pruning solution. . . . .	149
Table 4.5	Optimization results, in terms of position difference residual $\Delta\mathbf{r}^*$ and correction effort $\Delta\mathbf{v}^*$ at the maneuvering time. . . . .	150
Table 4.6	Optimization results, in terms of initial position residual and $\Delta\mathbf{v}$ magnitude. . . . .	150
Table 4.7	Optimization results, in terms of initial optimal state $(\mathbf{r}_{\text{out}}^{(j)*}, \mathbf{v}_{\text{out}}^{(j)*})$ . . . . .	151
Table 4.8	Retrieved optimal b-plane coordinates $(\xi^*, \zeta^*)$ and planetocentric velocity $\mathbf{U}^*$ . . . . .	151
Table 4.9	Sequential runtimes for the independent runs and the augmented system executions. . . . .	156
Table 4.10	Runtimes for the independent runs and the augmented system executions. Average of 10 different runs each. . . . .	158
Table 4.11	13509-sized augmented state kernel profiling results, for the GTX 1050 GPU. . . . .	161
Table 4.12	1501-sized augmented state kernel profiling results, for the GTX 1050 GPU. . . . .	161
Table 4.13	1 and 10 streams CUDA <sup>®</sup> program executions. Average of 10 different runs each, for the GTX 1050 GPU. . . . .	162
Table 4.14	13509-sized augmented state kernel profiling results, for the RTX A6000 GPU. . . . .	162
Table 4.15	13509-sized augmented state kernel profiling results, for the RTX A6000 GPU. . . . .	163
Table 4.16	1 and 10 streams CUDA <sup>®</sup> program executions. Average of 10 different runs each, for the RTX A6000 GPU. . . . .	163
Table 5.1	CUDAjectory steps for Cartesian and KS cases. . . . .	176
Table 5.2	CUDAjectory runtimes for different configurations, with trajectory storage. Average of 5 different runs. . . . .	176
Table 5.3	CUDAjectory runtimes for different configurations, without trajectory storage. Average of 5 different runs. . . . .	176
Table 5.4	CUDAjectory outcome difference against a $10^{-14}$ relative tolerance evolve() simulation. . . . .	177

---

## LISTINGS

---

Listing 4.1	Summary of dynamics CUDA kernel. . . . .	139
Listing 4.2	Summary of error reduction CUDA kernels and device functions. . . . .	142
Listing 5.1	KS state class properties. . . . .	171
Listing 5.2	Cartesian and KS sample properties. . . . .	171
Listing 5.3	Absolute time retrieval methods. . . . .	172
Listing 5.4	KS dynamics function interface. . . . .	173

---

## ACRONYMS

---

BR <sub>3</sub> BP	-	Barycentric Restricted Three-Body Problem
COSPAR	-	Committee on Space Research
CPU	-	Central Processing Unit
CR <sub>3</sub> BP	-	Circular Restricted Three-Body Problem
EKF	-	Extended Kalman filter
ESA	-	European Space Agency
GPU	-	Graphics Processing Unit
JPL	-	Jet Propulsion Laboratory
KS	-	Kustaanheimo-Stiefel
MC	-	Monte Carlo
PC	-	Picard-Chebyshev
PP	-	Planetary protection
SDM	-	Space Debris Mitigation
SOI	-	Sphere Of Influence
SSB	-	Solar System's Barycenter
UP	-	Uncertainty Propagation
UT	-	Unscented Transform

---

## INTRODUCTION

---

The raised awareness on the environmental impact of space missions has made the preliminary analysis of trajectories increasingly more complex over the years, introducing constraints and requirements whose purpose is to preserve the space ecosystem. The most prominent example, tightly connected to the overall effort of mankind toward a more sustainable use of the world's resources, is the exponentially growing amount of debris objects orbiting around Earth. Pioneering missions, starting already back in the space race era, did not consider the environmental aspect at all, freely disposing human-crafted objects in the open space, for instance ascent modules or upper stages of launchers, or even complete spacecrafts at the end of their operational life. Such objects may be located in trajectories that do not re-entry Earth's atmosphere in the near future, thus remaining in potentially hazardous and uncontrolled orbits. Some of these disposed spacecrafts have collided with others and got disintegrated, polluting the environment with many "newly generated" debris, that potentially threat active spacecrafts.

Hence, how to properly dispose space mission objects has become part of the standard design procedures, both for Earth and interplanetary endeavors. Solar system missions must comply with space debris mitigation policies as well: the  $\Delta v$  required to reach the orbits of other planets makes, sometimes, the use of disposable upper stages necessary. Other mission scenarios use the launcher upper stage to get in an Earth Highly Elliptical Orbit, using then the on-board spacecraft propulsion to escape to the interplanetary space. In either case, the disposed launcher upper stage lies in an uncontrolled trajectory, actually becoming a debris: Space Debris Mitigation (SDM) policies come into play, for possibly threatened active spacecraft in Low or Geostationary Earth Orbits, with potential on-ground casualty risk, were the upper stage to re-enter Earth's atmosphere. For European Space Agency's (ESA) missions launched from French Guyana, policies that cover the latter follow the French Space Operations Act, developed by the french National Center for Space Studies (CNES) [87]. The former follow instead guidelines proposed by the United Nations, and implemented by ESA itself [42].

Additionally, interplanetary missions introduce a new type of risk that must be addressed: the contamination of other celestial bodies, or samples thereof, with traces of life from Earth. This aspect is what Planetary Protection (PP) policies focus on, with the utmost goal of protecting the future of space exploration from misleading scientific outcomes. In other words, if, for instance, the martian environment were contaminated by Earth bacteria remaining on the surface of a lander, life could indeed be found on Mars, although it would not be local, but directly introduced by the space mission. Similar requirements apply to some of Jupiter's and Saturn's moons, that may host local life in the liquid oceans under an upper icy crust. The Committee on Space Research (COSPAR) outlines and maintains PP policies [31], raising or lowering the required protection level for the different bodies based on the latest observations. For instance, Venus suddenly became of interest in 2021 after the detection of phosphine gas in its atmosphere [58], suggesting traces of aerial life and thereby requiring updated policies to protect possible future studies, to be then disputed only one year later [72].

The actions that can be taken to implement PP policies in space missions follow two main directions, depending on the function of each component of the payload or the launcher. Following the ESA technical requirements published by the former PP officer [78], objects such as the launcher lower stages that remain in Earth's atmosphere do not require any particular measure. On the other hand, landers, rovers, and ascent modules, which get in direct contact with the environment to be explored, must be perfectly sterilized, in any part. Differently, orbiting components need to be injected in a graveyard orbit at the end of their operational life, ensuring a maximum impact probability over a given time frame, with the body to be protected. Another component requiring a dedicated analysis is the launcher's upper stage: as it is in direct contact with Earth's atmosphere prior to and during the launch, any sterilization procedure would be ineffective. Additionally, after the interplanetary injection maneuver, the trajectory of upper stages is completely uncontrolled. Similarly, spacecrafts themselves may become non-PP-compliant in case of failure or maneuver execution errors, switching from active to uncontrolled, thus becoming of interest like upper stages. For objects of this kind, PP requirements become a per-celestial body threshold on the maximum allowed impact probability.

The only way to ensure that uncontrolled objects comply with PP and SDM policies is an accurate and robust mission analysis. Despite the definition of a reference trajectory, due to not completely exact measurements, dynamical uncertainties or, for instance, maneuver mechanization errors, the spacecraft may never actually follow the nominal orbit. The dispersion that originates from all these aspect introduces a relevant uncertainty to the analysis, contributing to the

build-up of impact probability levels of space missions. Therefore, the uncertainty needs to be propagated and statistically studied as efficiently as possible, guaranteeing the minimum prescribed accuracy. A possible impact probability mitigation strategy would be to avoid gravity assist maneuvers altogether, notably reducing the chaos that the flyby dynamics introduces to the problem at hand. Nonetheless, gravity assists are an enabling feature of space missions, allowing the achievement of otherwise unfeasible trajectory deflections, with artificial propulsion only. Hence, flybys are always a "risk" worth taking for the success of space endeavors, letting the analysts to reach the best trade-off solutions between propellant consumption, designed trajectory, and accumulated impact probability levels. For this reason, novel techniques and tools, that allow a more efficient and accurate propagation of the uncertainty, would considerably reduce the mission development time. At the same time, expanding the current knowledge on flybys, both per se and for their effects on the propagated uncertainty, would also provide mission analysts with better means to design robust trajectories.

This chapter gives an introduction to the overall research work, providing a first state-of-the-art and background analysis in Section 1.1. Section 1.1 aims at clarifying the work positioning, a more in-depth literature review is given in the introduction of each chapter, for all the covered topics. Then, Section 1.2 outlines the research motivation and objectives, followed by the developed methods and implementations in Section 1.3. The chapter closes presenting in Section 1.4 the global thesis organization, and the related publications in Section 1.5.

## 1.1 BACKGROUND

As already briefly introduced, the PP compliance of space missions must be assessed taking into account different aspects and uncertainty sources. In particular, PP analyses must consider the effects of [31, 78, 125]:

- the spacecraft/upper stage dispersion covariance;
- multiple close approaches with major bodies in the Solar System;
- the spacecraft reliability and possible failures;
- the spacecraft's position and velocity at any time during the mission;
- evidence of sufficient absorbed radiation to sterilize the spacecraft from possibly left terrestrial life forms;

resulting in five different mission classes in the COSPAR classification [31], ranging from the lowest (1) to the highest (5) PP-related risks:

1. missions to bodies without significant biological interest (e.g. Mercury);
2. missions to bodies with significant biological interest, but of low risk of compromising future research, in case of contamination (e.g. Venus);
3. flybys and orbit-only missions to bodies of significant biological interest and with high risk of compromising future research, in case of contamination (e.g. Mars, Europa, Enceladus);
4. missions that land on the surface of bodies of significant biological interest and with high risk of compromising future research, in case of contamination (e.g. Mars, Europa, Enceladus);
5. missions returning to Earth.

For objects that cannot be sterilized, e.g. launcher upper stages and full spacecrafts in case of failures, PP policies become a maximum allowed impact probability level. Table 1.1 summarizes some of the requirements used in the design of ESA missions [31, 78, 125]; cases without a specific time frame corresponds to missions whose contaminating effect should be prevented at all costs.

Table 1.1: Example of PP compliant impact probability thresholds for ESA missions.

MISSION	OBJECT	BIOSPHERE	IMPACT PROBABILITY	TIME FRAME
GENERAL				
Generic	Any	Any	$\leq 1 \times 10^{-3}$	50 years
MARS				
General	Upper stage	Mars	$\leq 1 \times 10^{-4}$	50 years
	Spacecraft	Mars	$\leq (1 - 5) \times 10^{-2}$	20-50 years
Sample return	Spacecraft	Earth	$\leq 1 \times 10^{-6}$	-
OUTER SOLAR SYSTEM				
General	Any	Subsurface ocean	$\leq 1 \times 10^{-4}$	-
Sample return	Spacecraft	Earth	$\leq 1 \times 10^{-6}$	-

In the mission analysis context, PP and SDM can be seen as nothing but a direct Uncertainty Propagation (UP) application in orbital

dynamics: given the final uncertainty state at the end of the time frame, assessing PP and SDM requirements becomes the computation of what part of the uncertainty has contributed to the overall impact probability, as well as whether this part breaches the given threshold or not.

Other than the numerous effects and situations to be modeled, Table 1.1 highlights two essential features of PP compliance analysis: very low impact probabilities should be estimated, often over long time intervals. While the statistical result is what ultimately matters, the robustness of the analysis has resulted in the implementation of simulators that feature complete dynamical models, so that all the force sources and effects can be accounted for. In addition, the non-linearity introduced by gravity assists has led to PP and SDM assessment tools exclusively built on Monte Carlo (MC) or MC-based approaches, preventing the adoption of more computationally efficient, yet less accurate because of the required model simplifications, UP techniques. The direct statistical influence of minor perturbing effects or of the chosen uncertainty description has not been explored yet, hence leaving the possibility to study the adoption of simplified models and alternative approaches.

#### 1.1.1 *High-fidelity approaches for PP/SDM*

Starting from the requirements introduced by Kminek et al. [78], the ESA SNAPPshot suite has been developed by Letizia et al. [88, 89] and Colombo et al. [29], to compute the impact probability of a given initial condition and uncertainty. A MC simulation is performed on a large set of trajectory propagations, computed in the Cartesian formulation of the full force dynamics, keeping a static, barycentric reference frame and non-dimensionalization quantities, throughout the integration. The impact probability of the orbiting object with all the encountered bodies is determined as the ratio between the number of impacting runs and the total number of generated samples. SNAPPshot also provides a b-plane [148] analysis of colliding trajectories, highlighting the orbital resonance patterns of most of the detected impacts.

Romano et al. [126] propose a subset simulation approach and a Line Sampling method for the computation of impact probabilities. Line Sampling consists of a Markov Chain MC-based approach, which bounds the uncertainty regions finding the lines perpendicular thereto, thus sampling the uncertainty along such directions. Line Sampling ensures a better impact probability estimate, especially for small absolute values, and is also more efficient, requiring around 35% less total runs than the standard MC. Another work from Romano [125] investigated the influence of numerical schemes in the performance and precision of the SNAPPshot [29] approach. An attempt has also been made to develop a covariance propagation technique in the

Cartesian formulation of the dynamics, based on Gaussian Mixture Models. Even before flyby events, the propagated continuous covariance has quickly degraded, because of the strong non-linearity and the Lyapunov instability of the Kepler problem, requiring to split the uncertainty initial distribution repeatedly and cumulatively [125]. The results obtained in [125], together with other updates, contributed to the development of an upgraded version of the SNAPPshot suite [30].

Still, the computational cost related to the just described approaches remained high, requiring the use of high-performance computing facilities and/or large servers to obtain the required estimates within days/weeks. Hence, ESA improved the state-of-the-art software pool by developing CUDAjectory [32, 57, 128]. CUDAjectory introduces the emerging Graphics Processing Unit (GPU) computing technology to the problem, resulting in a remarkable runtime reduction of PP and SDM analyses. CUDAjectory implements a propagator featuring SNAPPshot’s dynamical model, the runtime reduction is achieved by propagating a massive number of samples in parallel, exploiting the peculiar architecture of graphics cards.

#### 1.1.2 *Other Uncertainty Propagation techniques*

Non MC-reliant approaches all try to simplify the UP problem to a certain degree, with the various methods differing on the extent and the approximation approach. While few works simplify the required dynamical model, nearly all strategies do not deal with close approach-related discontinuities. Hence, these techniques have proved their potential in Earth-related problems, such as modeling the evolution of space debris clouds, yet their implementation to interplanetary applications requires a solution to the flyby problem first. With the goal of providing a clearer positioning of the research presented in this dissertation, this section presents a brief overview of the most commonly adopted UP approaches in orbital dynamics, regardless the application. As it is also stated in the following sections, the objective of this thesis is neither to devise a new UP approach, nor to directly improve the existing techniques. Rather, this work focuses on the analysis of what actually characterizes UP, with direct application to PP and SDM tasks.

In the most general sense, the evolution of an uncertainty cloud is governed by Fokker-Planck equations [56]. Nonetheless, the direct integration of these partial differential equations is in general avoided, given the high number of dimensions of the orbital state, often using linear models or fully non-linear MC techniques (such as the already presented approaches for PP and SDM) [91].

On a different extent, UP can be simplified modeling the generic uncertainty with the sum of weighted Gaussian distributions: the Gaussian Mixture Model approach has been introduced by Terejanu



et al. [142], and later improved with specific application to the space debris problem [34, 152]. A similar approach, although not specifically referred to the orbital UP problem, is given by the Unscented Transform (UT) [74, 75], also tested by Romano [125] for the PP problem. Gaussian uncertainty can be described by a carefully selected set of points, called the *sigma points*. The fully non-linear dynamics is used to propagate this set of points, whose weighted sum would yield the propagated uncertainty. UT and Gaussian Mixture Models suffer from the orbital problem being highly non-linear, resulting in a prominent distortion of initially Gaussian distributions.

Recently, Frey [47] applied Gaussian Mixture Models as surrogate framework to reconstruct the uncertainty arising from space debris fragmentation events, described as fully probabilistic continua. The probability density is propagated with a fluid dynamics-like diffusion equation, using the method of the characteristics, keeping a Keplerian formulation of the dynamics and averaging Earth's oblateness effects. Hence, the propagated characteristics are used for the consequent training of the surrogate model, that in turn provides a global estimate of the evolution of the newly created debris cloud. While the approach could introduce considerable gains in computational efficiency, the application in the interplanetary environment thereof is not straightforward. Were flyby events to occur, discontinuities would arise in the dynamics, directly impacting the uncertainty continuum in a chaotic way. Thus, prior to the definition of interplanetary continua, a method to isolate flybys must be devised, together with the necessary techniques to split and re-merge the original uncertainty, so that the flyby scattering effect can be properly modeled.

Polynomial Chaos Expansion methods work slightly differently from UT and Gaussian Mixture Models, approximating the uncertainty with a series expansion and providing the evolution of higher order statistical moments of the propagation [159]. While Polynomial Chaos expansion can in general retain the full model description [71], very high orders are required to reach PP and SDM-like accuracy levels.

Taylor-based methods have also been developed, resulting in the common concept of State Transition Matrix, at the first order. Higher order terms have also been employed, giving birth to State Transition Tensor-based techniques [115]. Recent developments introduced a directional definition of the State Transition Tensor, resulting in an overall increased accuracy for non-linear UP problems [16]. While high precision levels could be in principle achieved, the computation of the partial derivatives required to fill the tensor elements becomes increasingly complex, raising the expansion order. Additionally, approaches of this kind require the system dynamics to be continuous and differentiable, making Taylor-based methods particularly sensitive to flybys, as well as unsuitable to model maneuver failures.

A similar yet different technique uses Taylor algebra to replace operations between real numbers with operations between polynomials. Differential Algebra has been developed building on this concept, expanding the dynamics function to an arbitrarily high order with Taylor polynomials, while still preserving a high computational efficiency, because of the new algebra type [12]. This technique has found a successful application in the study of Apophis' close encounter with Earth [5], and has been extended to the propagation of large uncertainties upon combination with domain splitting techniques [162]. Unfortunately, as in the State Transition Tensor case, Differential Algebra approaches require a continuous and differentiable dynamics.

Between the UP and Uncertainty Sampling realms lies the Line of Variation approach, used by Milani et al. [109] to study the impact probability of near-Earth asteroids. Strictly speaking, the Line of Variation approach samples the uncertainty along a given differentiable curve in the orbital elements space [107, 109, 110]. Nonetheless, the description in terms of orbital elements allows to predict a string-like shape connected to the dynamics, along which most of the uncertainty variation will take place. However, this technique still provides "simply" an uncertainty sampling method, whose propagation is then deferred to other approaches (e.g., MC in [109]).

### 1.1.3 *A multidisciplinary problem*

UP, especially in the high-fidelity case required by PP and SDM applications, is an intrinsically multidisciplinary problem. Figure 1.1 schematically depicts the three major areas that heavily influence outcome and performance of every UP technique, regardless the environment and the applications. For instance, choosing MC approaches over other uncertainty models corresponds to selecting one particular statistical description, over some other. Similarly, Cartesian coordinates are, in general, just one of the possible ways of representing the formulation of the orbital dynamics. Finally, be it the propagation of a massive amount of trajectories or the training of a complex surrogate model, the chosen computing framework and technology plays an important role in the practical applicability of the any UP approach, even just in terms of providing an uncertainty estimate in a reasonable computational time.

As Figure 1.1 highlights, choices made on either area may have a significant impact on the others. For instance, while a complete dynamical model that includes all bodies in the solar system is what enables accurate analyses, it also limits the adoption of continuum-based statistical descriptions. In this sense, flyby events are a conceptual discontinuity that scatters the propagated uncertainty, that completely loses the properties that allowed its model to be continuous. Similarly, certain element-based formulations of the dynamics may be particularly sen-

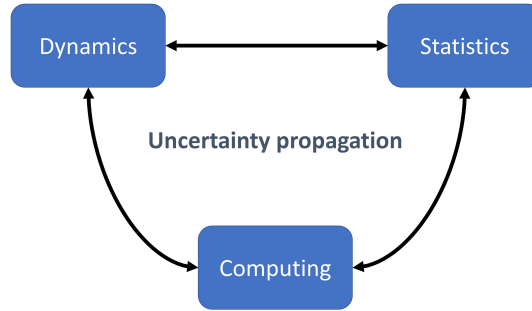


Figure 1.1: The aspects characterizing orbital uncertainty propagation.

sitive to flyby events, preventing their straightforward implementation in the interplanetary environment. Finally, while GPUs do represent today's best-performing computational framework, their architecture makes GPU programming difficult, resulting in rather simple algorithms only readily implementable in GPU code. Hence, other than advancements in any of the areas presented in Figure 1.1, research activities may also cover the interconnections among the different fields.

So far, UP has been studied focusing on each single research direction highlighted in Figure 1.1 exclusively, without considering cross-interactions. Be that different statistical descriptions of the uncertainty, these works have all kept a fixed formulation of the dynamics. Hence, whether the choice made for the equations of motion has an impact on the description of the uncertainty or not needs to be assessed. Similarly, choices made on the dynamics may have a significant impact on the computational aspects, as well as a GPU computing approach may be unsuitable for certain statistical methods or equations of motion.

## 1.2 MOTIVATION, OBJECTIVES, AND APPLICATIONS

Analyzing the compliance of space missions with PP and SDM requirements is a computationally intensive task. The small impact probabilities to be estimated make the use of complete force models necessary, as well as may require the propagation of hundreds of thousands of samples. Thus, any improvement in terms of computational efficiency of the single simulations represents an upgrade to state-of-the-art approaches for PP and SDM analysis.

The effect of close encounters in the accuracy of numerical trajectory propagation has not been understood completely, yet. Additionally, flybys introduce a chaotic scattering effect in the propagation of the uncertainty, with shallow and distant interactions that can make the difference between detecting or missing impacting trajectories. Therefore, the effect of encounters on the propagated uncertainty needs to be studied, both in terms of devising frameworks to mitigate the

uncertainty scattering effect, and attempting to model the long-range interactions that lead to otherwise inexplicable collisions.

Most numerical simulation techniques rely, in general, on step-based approaches: given an initial condition, differential equations are integrated computing the evolution of the trajectories, estimating step by step forward (or backward) in time. While the available techniques of this kind are some of the most robust and validated simulation methods, the intrinsically sequential nature thereof may have a performance limit dictated by the algorithm concept. For this reason, it may be worth exploring the potential, in terms of accuracy and computational performances, of iteration-based schemes, that start from a full trajectory guess and update it through iterations, fitting arbitrarily complex dynamical models.

Finally, as already mentioned in Section 1.1.3, the interplay of the different fields of UP should be studied as well. Certain formulations of the equations of motion may be the preparatory framework for fully probabilistic approaches to UP. At the same time, GPU computing has powered the development of Artificial Intelligence models, thus GPUs may be the enabling features for complex surrogate uncertainty and trajectory prediction models to effectively work.

### 1.3 DEVELOPED METHODS AND IMPLEMENTATION

This dissertation focuses and studies each of the highlighted aspect. Ideally, proposing a unified technique that combines all the developed techniques in a single UP and/or PP/SDM tool is the research goal to aim for. Nonetheless, the research activity made evident that considerable preparatory work was required, so that each of the three research field (dynamics, statistics, and GPU computing) could be ready to be connected with the other. For this reason, rather than attempting the development of a unified UP approach, this work proposes a set of analytical and numerical methods and techniques that all present improvement to the state of the art in UP and PP/SDM. At the same time, the results obtained in the development of these tools show that recombining the multidisciplinary effort of this research is possible, presenting possible development pathways as well as future extension and applications.

The high non-linearity of the Cartesian equations of the orbital motion represents a limit for the computational efficiency of high-fidelity numerical simulations. The implicitly straightforward interface with physical forces and perturbing accelerations has made the Cartesian formulation the most popular dynamics description in numerical simulations, allowing the accurate implementation of complex and high-precision force models. Yet, considering the trajectory propagation per se, Cartesian coordinates are far from being the best dynamics formulation, either for accuracy and computational efficiency. Element-based

formulations that possibly include the regularization of the equations of motion are widely considered the best strategies to compute orbits. The lack of simplicity and the complex physical interpretation behind these techniques has perhaps represented the major limitation in the adoption thereof. Hence, this work implements, interfaces, and studies the performance of the Kustaanheimo-Stiefel (KS) formulation of the dynamics, in direct application to UP, PP, and SDM tasks, via simulations employing established numerical integration schemes. KS variables are eventually implemented in the CUDAJectory software, making also a first application of the synergy between improved dynamical formulation and GPU computing technologies.

The work on KS variables is not limited to improving the simulation efficiency of PP and SDM. Rather, the influence of the dynamics formulation on the uncertainty shape is numerically assessed. Similarly, the role of physical time as independent integration variable of the simulation of the uncertainty is analyzed. This aspect leads to the development of an alternative numerical description of the propagated uncertainty distribution in KS coordinates, that uses a fictitious time as independent variable to synchronize upon. This approach results in a distribution whose behavior remains highly regular and predictable in all cases, flyby dynamics included, thereby suggesting the suitability of KS variables to continuum statistical descriptions of the distribution. Analytical and numerical uncertainty mapping techniques between Cartesian and KS variables are also devised.

Flybys are studied aiming at a generalization of the close approach concept, that includes shallow and weak interactions, where the presence of both the Sun and the planet flown by have a significant impact on the dynamics. An analytical description of a generalized SOI concept is proposed, based on the eigenvalues of the dynamics' Jacobian, starting from concepts used for the step adaptation of numerical integration schemes. Using the new SOI definition, a novel concept to approach the whole PP/SDM problem is presented: in the N-body perturbed dynamics, impacting trajectories are identified not only on keyholes belonging to resonant trajectories, but also in off-nominal conditions that require a three-body approach to be fully characterized. Hence, the Keyhole Map is constructed, a graphical tool that accurately predicts and locates keyholes in a generalized sense.

The Picard-Chebyshev (PC) integration scheme is studied, as an alternative to traditional step-based simulators, implementing PC-based propagator and exploiting the fixed-point approach to build an efficient trajectory optimization framework. An augmented version of the PC method is also proposed, improving the computational efficiency on parallel and GPU computing architectures.

#### 1.4 THESIS ORGANIZATION

Six chapters compose this dissertation, each covering one of the proposed research areas, with Chapter 1 being the current introduction. All the remaining chapters start by locating the topic in the research objectives, together with a review of state-of-the-art techniques and results thereof, except the conclusions drawn in Chapter 6. Then, all the theoretical developments are stated, divided in sections if more than one research area is explored. Finally, each chapter proposes application test cases, taken or inspired by real interplanetary mission scenarios.

Chapter 2 studies the adoption of KS variables in PP/SDM and orbital UP problems. After a detailed analysis of the benefits of KS variables on the single simulations in Section 2.2, Section 2.3 proposes some strategies to map uncertainty distributions between the KS and the Cartesian realms. Next, Section 2.4 studies the role of the KS regularization and of the synchronization on the fictitious time on the behavior of the propagated uncertainty. Finally, Chapter 2 ends with application test cases for all the proposed methodologies in Section 2.5.

Chapter 3 focuses on the understanding of close encounters. Section 3.2 assesses the effectiveness of perturbation approaches on hyperbolic trajectories to include the Sun effects during flybys. Section 3.3 discusses the dynamical meaning of the Jacobian eigenvalues and presents the new definition of SOI. Chapter 3 ends with a direct application of the new SOI concept, constructing the Keyhole Map and applying it to the space debris mitigation compliance analysis of ESA's JUICE mission.

Chapter 4 explores the potential and the applicability of the PC numerical scheme to the interplanetary trajectory design and UP problem, investigating first the applicability of the method to the optimization of resonant flybys and trajectories, in Section 4.2. Section 4.3 presents the development of the augmented version of the PC scheme. Application test cases based on the ESA mission Solar Orbiter are shown in Section 4.4.

Finally, Chapter 5 presents the implementation details of KS variables in the CUDAjectory software. After a brief introduction on the software structure in Section 5.1, the chapter describes the implementation details and concepts to include KS variables as a new formulation of the dynamics in Section 5.2. The chapter closes testing the performance of the KS formulation in CUDAjectory on the compliance analysis of ESA's JUICE mission with SDM policies, in Section 5.3.

The dissertation ends with the global conclusions drawn from the work, in Chapter 6. Other than providing a summary of the key findings and a discussion thereof, Chapter 6 highlights the connection between the different research directions explored in this work, devis-

ing a few recommendation for future works to connect the concepts in novel UP techniques.

## 1.5 CONTRIBUTIONS

The research results of this thesis have been presented in several international conferences, and published in international journals. The unpublished contents refer mainly to Chapters 3 (except the already published work on perturbation methods) and 5, together with Section 2.3. Section 2.4 covers instead an analysis that has only been presented at the 73<sup>rd</sup> International Astronautical Conference (IAC 2022). All these unpublished contents will be submitted for journal publications in the coming months.

### *Journal articles*

- [1] G. Campiti, A. Masat, and C. Colombo, “B-plane resonant flybys: theory extension to elliptical orbits and dynamic programming application,” *Submitted to Journal of Guidance, Control, and Dynamics*, 2023.
- [2] M. Lara, A. Masat, and C. Colombo, “A torsion-based solution to the hyperbolic regime of the  $J_2$  problem,” *Nonlinear Dynamics*, vol. 111, no. 10, pp. 9377–9393, May 2023, ISSN: 1573-269X. DOI: [10.1007/s11071-023-08325-w](https://doi.org/10.1007/s11071-023-08325-w).
- [3] A. Masat and C. Colombo, “B-plane and picard–chebyshev integration method: Surfing complex orbital perturbations in interplanetary multi-flyby trajectories,” *Acta Astronautica*, vol. 194, pp. 216–228, 2022, ISSN: 0094-5765. DOI: [10.1016/j.actaastro.2022.01.045](https://doi.org/10.1016/j.actaastro.2022.01.045).
- [4] A. Masat, C. Colombo, and A. Boutonnet, “GPU-based high-precision orbital propagation of large sets of initial conditions through Picard–Chebyshev augmentation,” *Acta Astronautica*, vol. 204, pp. 239–252, 2023, ISSN: 0094-5765. DOI: [10.1016/j.actaastro.2022.12.037](https://doi.org/10.1016/j.actaastro.2022.12.037).
- [5] A. Masat, C. Colombo, and A. Boutonnet, “Kuustanheimo–Stiefel - Cartesian Uncertainty Transformation,” *Submitted to Astronomy and Astrophysics*, 2023.
- [6] A. Masat, A. Rocchi, A. Boutonnet, and C. Colombo, “Jacobian Spheroids, shallow encounters, and the Keyhole Map,” *Submitted to Journal of Guidance, Control, and Dynamics*, 2023.
- [7] A. Masat, M. Romano, and C. Colombo, “Kustaanheimo–Stiefel Variables for Planetary Protection Compliance Analysis,” *Journal of Guidance, Control, and Dynamics*, vol. 45, no. 7, pp. 1286–1298, 2022. DOI: [10.2514/1.G006255](https://doi.org/10.2514/1.G006255).



*Conference proceedings*

- [1] G. Campiti, A. Masat, and C. Colombo, “Dynamic Programming Applied to Resonant Flyby Design,” in *2021 AAS/AIAA Astrodynamics Specialist Conference*, Big Sky, MT, USA, Aug. 2021. [Online]. Available: <https://hdl.handle.net/11311/1189604>.
- [2] M. Lara, A. Masat, and C. Colombo, “Tracking close flybys via intermediary orbits,” in *28<sup>th</sup> International Symposium on Space Flight Dynamics (ISSFD)*, Beijing, China, 2022. [Online]. Available: <https://hdl.handle.net/11311/1236593>.
- [3] A. Masat, C. Colombo, and A. Boutonnet, “Eccentric Anomaly Synchronism and Regularised Dynamics for Continuum Interplanetary Orbital Uncertainty Propagation,” in *73<sup>rd</sup> International Astronautical Congress (IAC 2022)*, Paris, France, Sep. 2022. [Online]. Available: <https://hdl.handle.net/11311/1222115>.
- [4] A. Masat, C. Colombo, and A. Boutonnet, “Surfing Chaotic Perturbations in Interplanetary Multi-Flyby Trajectories: Augmented Picard-Chebyshev Integration for Parallel and GPU Computing Architectures,” in *AIAA SCITECH 2022 Forum*, San Diego, CA, USA, 2022. DOI: [10.2514/6.2022-1275](https://doi.org/10.2514/6.2022-1275).
- [5] A. Masat, C. Colombo, and A. Boutonnet, “GPU Augmented trajectory propagation: orbital regularisation interface and NVIDIA CUDA Tensor Core performance,” in *9<sup>th</sup> International Conference on Astrodynamics Tools and Techniques, ICATT 2023*, Sopot, Poland, Jun. 2023.
- [6] A. Masat, M. Romano, and C. Colombo, “Combined B-plane and Picard-Chebyshev approach for the continuous design of perturbed interplanetary resonant trajectories,” in *31<sup>st</sup> AAS/AIAA Space Flight Mechanics Meeting*, vol. AAS-21-289, Charlotte, NC, USA, Feb. 2021. [Online]. Available: <https://hdl.handle.net/11311/1189603>.
- [7] A. Masat, M. Romano, and C. Colombo, “Kustaanheimo–Stiefel Variables to Halve the Cost of Planetary Protection Compliance Analysis,” in *8<sup>th</sup> International Conference on Astrodynamics Tools and Techniques, ICATT 2021*, Virtual Conference, Jun. 2021. [Online]. Available: <https://hdl.handle.net/11311/1183121>.

*Workshop presentations*

- [1] A. Masat and C. Colombo, “Planetary protection and robust interplanetary mission design,” in *1<sup>st</sup> AIDAA Aerospace PhD Days*, Virtual Conference, 2021.



- [2] A. Masat and C. Colombo, "Technical and programmatical challenges in planetary protection and planetary defence," in *United Nations/China 2<sup>nd</sup> Global Partnership Workshop on Space Exploration and Innovation*, Haikou, China, 2022.
- [3] A. Masat, C. Colombo, and A. Boutonnet, "Flyby dynamical characterisation with Jacobian eigenvalues," in *5th International Workshop on Key Topics in Orbit Propagation Applied to SSA*, Logroño, Spain, 2022. [Online]. Available: <https://hdl.handle.net/11311/1222116>.



# 2

---

## KUSTAAANHEIMO-STIEFEL VARIABLES

---

This chapter centers on the adoption of Kustaanheimo-Stiefel (KS) coordinates, aiming at improving the efficiency of PP/SDM compliance analysis acting on the performance of the single simulations from a dynamics viewpoint. The theoretical background of the KS formulation, together with the PP/SDM adaptation and optimization thereof, is presented in Section 2.1. As the adoption of KS variables inevitably affects the uncertainty description and propagation per se, this research direction is also explored in Sections 2.3 and 2.4, for the mapping between Cartesian and KS formulation of the uncertainty and its propagation, respectively. Finally, Section 2.5 proposes application test cases covering the full theoretical content of this chapter.

### 2.1 REGULARIZATION TECHNIQUES

Regularizing the dynamics is not necessarily attached to a new set of coordinates per se, in fact the KS regularization was born bringing together two concepts: the first, adding a fourth coordinate trying to describe the Kepler problem as a four dimensional harmonic oscillator, originally presented by Kustaanheimo [84], and the introduction of the time transformation of the Sundman type [140], operated later with the contribution of Stiefel [83]. The key concept is about converting the integration independent variable, switching from the physical time  $t$  to the fictitious time  $s$ . The generic Sundman transformation [140]:

$$\frac{ds}{dt} = \frac{\beta}{r} e^{\int K dt} \quad (2.1)$$

In general,  $\beta$  is an arbitrary constant coefficient and  $K$  is an arbitrary function of position, velocity and time. For instance, setting  $\beta = 1$  makes the fictitious time  $s$  scale like the eccentric anomaly, whereas  $\beta = 2$  introduces a true anomaly-like fictitious time evolution.

#### 2.1.1 Overview of Regularization approaches

All the later developments over the newly created KS formalism use to refer to the more comprehensive manuscript by Stiefel and Scheifele [139]. Bond proposed a variation of parameters approach that

generically accounts for perturbing effects starting from the original four-vector KS formulation [15].

In the work of Stiefel and Scheifele [139] the properties of quaternion algebra are mentioned, even though the original developers of the KS transformation did not pay much attention on this aspect. Quaternions in the KS formalism have been explored first by Velte [151] and better detailed by the successive works of Vivarelli [153–155], introducing the concept of anti-involute and quaternion cross product. The KS transformation was re-derived and refined in the quaternion formalism by Deprit et al. [39] where it was obtained by doubling from a Levi-Civita transformation. Deprit [37, 38, 40] also extensively studied the canonical Lissajous transformation, concept that has recently been re-proposed by Breiter and Langner [19, 20] starting from the KS formulation. A more recent work by Waldvogel [156] provided new insight on the algebraic properties of quaternions in the KS regularization and the related fibration of the KS space, re-defining Vivarelli's anti involute [154] as the quaternion star conjugate.

Saha [127] proposed a quaternion approach to the KS transformation more suitable to the analysis of Hamiltonian systems. Based on the work of Saha, Breiter and Langner [18] explored the role of the preferential direction chosen to perform the regularization, which they generalized to an arbitrary one of the original three-dimensional space. In fact, almost all the analyses already performed in the context of the KS formulation adhere to the original one proposed by Kustaanheimo and Stiefel [83]. Denoting with  $(x, y, z)$  the axes of a generic three-dimensional Cartesian space, not necessarily attached to any commonly used Solar System direction, the first coordinate  $x$  was selected as the reference direction for the KS transformation. Using the same notation as Breiter and Langner [18], choosing  $x$  as preferential direction the KS transformation is identified as  $KS_1$ , whereas for instance the  $KS_3$  version is presented by Saha [127] using the direction  $z$ . The work of Breiter and Langner [18] did not stop to this generalization, but provided new insight to the physical meanings of the KS coordinates themselves.

Partially quoting Breiter and Langner [18] and recalling the quaternion description of rotations, in the  $KS_1$  version "the normalized KS variables are the Euler-Rodrigues parameters of the rotation turning the  $x$  axis into the direction of the Cartesian position vector which generated the KS coordinates". From this sentence, the physical meaning of the fibration property of the transformation [18, 139, 156] also becomes more clear, recalling that such a rotation may happen in an infinite number of ways. Another work by Langner and Breiter [85] explores the properties of the quaternionic KS formulation in Hamiltonian systems, tackling also the rotating frame case. Some other work in a similar direction was done by Roa and Peláez [121], they make use of the Minkowskian geometry that was originally proposed by

Kustaanheimo and Stiefel [83, 84] to handle close approaches and their hyperbolic geometry. Still Roa et al. also worked on the KS formulation [124], showing that the Lyapunov stability does not apply to the whole KS fiber with respect to the physical time. Roa and Kasdin made use of quaternions also developing a new set of non-singular orbital elements that models the intermediary evolution of the orbital plane [120].

When dealing with regularized orbital dynamics, the DROMO formulation is among the latest state-of-the-art results for the perturbed two-body problem. The first implementation of DROMO elements was proposed by Peláez et al. [116], seeking for a numerically stable, regularized and efficient formulation, furthermore rendering the same accurate results for any type of primarily two-body trajectory (elliptical, parabolic, hyperbolic) [145], using a variation of parameters approach. *ideal frame*. The description of the motion is broken into three steps [145]: the evolution of the orbital plane, the evolution of the trajectory on the orbital plane (also called *ideal frame*, and first introduced by Deprit [36]) and finally the position change on the osculating orbit as a function of time. The independent variable is a fictitious time obtained by a Sundman transformation [140], which reduces to the true anomaly in case of Keplerian motion, instead of the eccentric anomaly of the KS case. The description of the ideal frame does not require time regularization per se, it was subsequently introduced by Palacios and Calvo [113].

The original DROMO formulation is singular for null angular momentum. Recent revisions worked to fix these sensitivity issues: Baù et al. [10] developed E-DROMO, a non-singular formulation of DROMO for any bound orbit, removing the singularities on null eccentricity and inclination. Roa et al. [123] analyzed the singularities posed by deep flybys and then proposed the re-formulation H-DROMO, not sensitive to a vanishing angular momentum. The latest updates involve the study of the evolution of an intermediate frame [11]. The interest on universal variables, capable of managing either closed, open or even rectilinear trajectories is introduced. Amato et al. [3] studied the reference frame switch for close approaches using the DROMO formulation, and numerically identified a region of "best switch distance" that does not correspond either to the Hill surfaces or the sphere of influence (SOI). The way all the DROMO formulations were built makes them sensitive to the task they are used for. In the case of flyby events, a frame switch procedure is required to enhance its robustness, being a variations of parameters approach. Moreover, if events need to be computed requiring to retrieve the propagated coordinates, the conversion burden is inevitably added to the integration runtime, although DROMO, and variations of parameters in general, remain more efficient in terms of accuracy and required function evaluations. Furthermore, the DROMO approach remains limited to the perturbed

two-body problem. KS coordinates are chosen because of the introduction of the barycentric version in this work, foreseeing a future validation of the technique to low-energy PP/SDM tasks, building a framework conceptually similar to the rotating frame case proposed by Langner and Breiter [85]. A new term appears in the KS Hamiltonian in [85], that solution may represent a suitable starting point to study a barycentric KS-like formulation in the context of the restricted three body problem, once the rotating frame assumption is removed. The version presented here regularizes the evolution of the physical time based on the distance from the barycenter, but keeps the inertial reference. The proposed barycentric formulation can be extended to a fully Hamiltonian system, if expressed as a doubled set of first order equations, and would be equivalent to the complete Hamiltonian set introduced by Breiter and Langner [18] upon extension to the conjugate momentum of the physical time  $t$ . In their derivation, Breiter and Langner exploit a zero-energy manifold for the full Hamiltonian to define the conjugate momentum  $U^*$ . The equations of motion are then derived from a new Hamiltonian obtained by applying the Sundman time transformation. The process to extend the barycentric equations of motion to a fully Hamiltonian formulation would be equivalent.

Other works are tailored to the exploration of the most numerical side of the formulation. An extensive analysis of this kind on the KS<sub>1</sub> case was proposed by Arakida and Fukushima [4]. Fukushima alone published a series of works that aim to improve the numerical performances of the integration in KS variables and regularized time [50–55]. Several aspects that all contributed to the enhancement of the pure numerical efficiency of the integration were touched: single and quadruple scaling techniques to each of the KS coordinates are proposed, and the latter was extended applying a linear transformation to obtain quasi-conserved quantities to monitor and adjust the scaling during the integration. A time element formulation that reduces the error growth was studied, and then extended to a complete set of variables through the variation of parameters approach. Still within the context of regularized formulation but without using the KS formulation, an orbital longitude integration and manifold correction methods were proposed.

Other works have developed a formalism stemming from the KS approach, in the attempt to regularize the complete N-body problem. Aarseth and Zare [1] started with the regularization of the three-body problem based on the KS formulation for a configuration considering the two primary binaries, extended shortly after by Heggie [62] for any configuration and having regularized collisions for any pairs. Palmer et al. [114] and Mikkola and Aarseth [100–104] extended Aarseth’s approach to a N-body chain regularization technique, together with different implementation solutions for how to switch the chain configuration in case of close encounters and analyses of the numerical

performance. Mikkola and Merrit [105, 106] extended the regularization chain algorithm also to include velocity-dependent perturbation sources, with specific mention to general relativity through the post-Newtonian approximation. Their work strictly involves the simulation of full N-body systems, whereas the proposed adaptation and implementation focuses on the motion of a small particle of negligible mass under the action of the gravitational forces of the N-bodies, as well as other arbitrary perturbations.

For PP/SDM applications, reducing the precision of the integration simplifying the physical model might lead to significantly different results because of the accumulated error, especially at the end of the integration span. This makes it necessary to deal with a force model that is as complete as possible, which requires the use of ephemerides data to avoid integrating the N-body problem for faster integrations. In fact, the computational burden is currently the major limitation to the extensive performance of PP/SDM analysis. Nowadays, the compliance with the requirements outlined for instance by Kminek et al. [78] is assessed with a MC based approach, for each point in the trajectory where the disposal of an object is required. Were a trajectory solution to be discarded, all the MC runs would need to be run anew, assessing whether the next candidate complies with the requirements or not, and so on. Improving the computational cost of the PP/SDM analysis turns therefore into a reduction of the overall trajectory design development time, as any solution whose compliance with the given requirements needs to be secured benefits from more efficient techniques.

On the application viewpoint, KS coordinates have already been used in a few applications. Hernandez and Akella [63] developed a Lyapunov-based guidance strategy using KS coordinates to target several types of orbit conditions, e.g. specified angular momentum vector, and applied it to the design of low thrust trajectories. Woolands et al. [166] developed a Lambert solver based on KS coordinates and used it to provide a good initial guess to the Picard-Chebyshev numerical integration of the perturbed two-body problem. Sellamuthu and Sharma analysed the  $J_2, J_3, J_4$  terms of Earth's oblateness and the third body luni-solar perturbation when approximated with a Legendre polynomial expansion with KS coordinates [130–132]. Using the equation they obtained, they then implemented an orbital propagator and studied the effects of such perturbations on resident space objects with high perigee and highly eccentric orbits.

### 2.1.2 *KS Regularization*

The Kustaanheimo-Stiefel (KS) formulation rewrites the two body problem as an isotropic<sup>1</sup>, four-dimensional harmonic oscillator. The

<sup>1</sup> All the four components oscillate with the same frequency and phase.

conservation of the orbital energy is also introduced, leading to a simple linear ordinary differential equation. The first formulation was proposed indeed by Kustaanheimo and Stiefel [83, 139], and extended the usual Cartesian position vector  $\underline{\mathbf{r}} = \{r_1, r_2, r_3\}^T$  into a four-vector, by adding the length  $r = \sqrt{\underline{\mathbf{r}} \cdot \underline{\mathbf{r}}}$  as fourth coordinate. The underline notation  $\underline{\mathbf{r}}$  is used to stress the difference between three-dimensional Cartesian vectors and the KS four vectors/quaternions, denoted in bold  $\mathbf{r}$ . The reader can find a summary of the quaternion algebra concepts necessary for the understanding of this chapter in Appendix A.1. Using the initial formulation proposed by Kustaanheimo, the physical coordinates are linked to the spinor regularized coordinates  $\mathbf{u} = \{u_1, u_2, u_3, u_4\}^T$  through:

$$\begin{aligned} r_1 &= u_1^2 - u_2^2 - u_3^2 + u_4^2 \\ r_2 &= 2(u_1 u_2 - u_3 u_4) \\ r_3 &= 2(u_1 u_3 + u_2 u_4) \\ r &= u_1^2 + u_2^2 + u_3^2 + u_4^2 \end{aligned} \quad (2.2)$$

later re-arranged in matrix-vector product as

$$\mathbf{r} = \mathbf{L}(\mathbf{u})\mathbf{u} \quad (2.3)$$

which gives  $\mathbf{r} = \{r_1, r_2, r_3, 0\}^T$ , with

$$\mathbf{L}(\mathbf{u}) = \begin{bmatrix} u_1 & -u_2 & -u_3 & u_4 \\ u_2 & u_1 & -u_4 & -u_3 \\ u_3 & u_4 & u_1 & u_2 \\ u_4 & -u_3 & u_2 & -u_1 \end{bmatrix} \quad (2.4)$$

Setting  $\beta = 1$  and  $K = 0$  in Equation (2.1) to operate the time transformation, replacing  $\mathbf{r} = \mathbf{L}(\mathbf{u})\mathbf{u}$  and its derivatives, the KS transformation converts the Kepler two-body problem

$$\ddot{\mathbf{r}} = -\frac{\mu}{r^3}\mathbf{r} \quad (2.5)$$

into

$$\mathbf{u}'' = \frac{\epsilon}{2}\mathbf{u} \quad (2.6)$$

where  $(\ddot{\cdot})$  and  $(\cdot)''$  stand for second  $t$  and  $s$  derivatives respectively, and  $\epsilon$  denotes the two-body orbital energy. Because the adopted Sundman regularization is of the first order, for unperturbed orbits the new independent variable  $s$  follows the evolution of the eccentric anomaly. The achieved behavior reflects a sort of slow motion movie for the near-pericenter part of the orbit, with smaller physical time steps the closer the trajectory gets to the attractor.

Vivarelli [154] first showed the connection with quaternions, introducing the definition  $\mathbf{u}^* = \{u_1, u_2, u_3, -u_4\}^T$  as the "anti-involute" of  $\mathbf{u}$ , later re-defined as "star conjugate" by Waldvogel [156]. For the sake of conciseness, the notation used by Waldvogel is proposed here.



### 2.1.2.1 KS regularized formulation using quaternion notation

The position vector  $\underline{\mathbf{r}}$  of the three-dimensional Cartesian space can be written as a quaternion  $\mathbf{r}$  with null  $i_3$  component:

$$\underline{\mathbf{r}} = \{r_1, r_2, r_3\}^T \longrightarrow \mathbf{r} = r_1 + i_1 r_2 + i_2 r_3 + i_3 \cdot 0 \quad (2.7)$$

In the following, quaternions are represented with the scalar part as the first element of the associated four-vector. It can be shown that [156], given a quaternion  $\mathbf{u} = u_1 + i_1 u_2 + i_2 u_3 + i_3 u_4$ , the mapping

$$\mathbf{r} = \mathbf{u}\mathbf{u}^* \quad (2.8)$$

produces a quaternion  $\mathbf{r}$  with vanishing  $i_3$  component, and with the other components as defined in Equation (2.2):

$$\begin{aligned} r_1 &= u_1^2 - u_2^2 - u_3^2 + u_4^2 \\ r_2 &= 2(u_1 u_2 - u_3 u_4) \\ r_3 &= 2(u_1 u_3 + u_2 u_4) \end{aligned} \quad (2.9)$$

The magnitude of  $\mathbf{r}$  can also be written in terms of quaternion operations:

$$r^2 = \|\underline{\mathbf{r}}\|^2 = |\mathbf{r}|^2 = \mathbf{u}\bar{\mathbf{u}} \quad (2.10)$$

with  $\bar{\mathbf{u}} = u_1 - i_1 u_2 - i_2 u_3 - i_3 u_4$  the standard definition of quaternion conjugate.

Following Waldvogel [156], inheriting the conformality properties of the Levi-Civita mapping can partially fix the degree of freedom left whenever mapping from  $\mathbb{R}^3$  to  $\mathbb{R}^4$ . A constraint is added for the definition of the components of  $\mathbf{u}$ , and particularly appears as the following differentiation rule for  $\mathbf{r}$ :

$$d\mathbf{r} = 2\mathbf{u} \, d\mathbf{u}^* \quad (2.11)$$

The fiber defining the mapping from  $\mathbb{R}^3$  to  $\mathbb{R}^4$  can be parametrized by the angle  $\varphi$ , through a two-step process. First, the unique quaternion with vanishing  $i_3$  component is found:

$$\mathbf{u}|_{\varphi=0} = \frac{\mathbf{r} + |\mathbf{r}|}{\sqrt{2(r_1 + |\mathbf{r}|)}} \quad (2.12)$$

Then, the whole KS fiber is found through  $\varphi$  as

$$\mathbf{u} = \mathbf{u}|_{\varphi=0} e^{i_3 \varphi} = \mathbf{u}|_{\varphi=0} (\cos \varphi + i_3 \sin \varphi) \quad (2.13)$$

which is proved as follows [156]:

$$\mathbf{u}\mathbf{u}^* = \mathbf{u}|_{\varphi=0} e^{i_3 \varphi} e^{-i_3 \varphi} \mathbf{u}^*|_{\varphi=0} = \mathbf{u}|_{\varphi=0} \mathbf{u}^*|_{\varphi=0} = \mathbf{r} \quad (2.14)$$

Figure 2.1 shows a visualization of the KS fibration property. All the variable-color ellipses refer to the common generator point, i.e. the

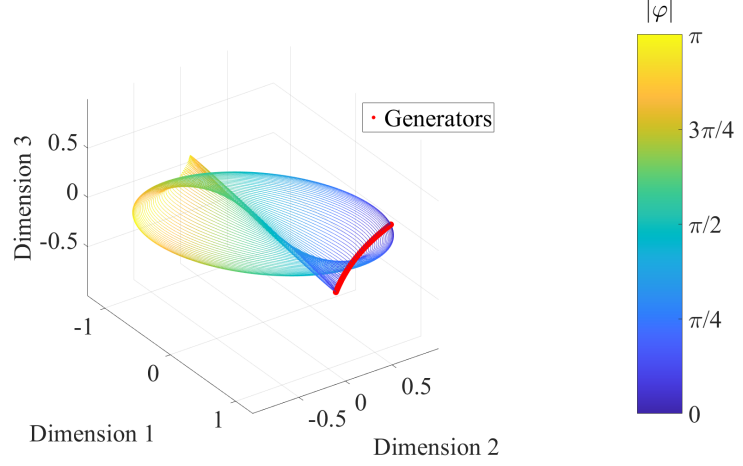


Figure 2.1: Fibration visualization as function of  $\varphi$  (color scale), for the first three KS vector components. Generator points refer to the values for which  $\varphi = 0$ .

KS state obtained by setting  $\varphi = 0$ , and all correspond to the same mapped Cartesian state. Analogous representations could be made for all the KS components triplets, all generating ellipse-like-shaped loci of points for different KS generators.

Equation (2.11) can be used to obtain a conversion formula to obtain the Cartesian velocity  $\dot{\mathbf{r}}$  from the KS velocity  $\mathbf{u}'$ . Introducing the chain rule and using Equation (2.1) with  $K = 0$  and  $\beta = 1$  gives

$$\dot{\mathbf{r}} = \frac{d\mathbf{r}}{dt} = 2\mathbf{u} \frac{d\mathbf{u}^*}{ds} \frac{ds}{dt} = \frac{2}{r} \mathbf{u} \mathbf{u}'^* \quad (2.15)$$

which for the Cartesian to KS conversion results in, exploiting the properties of quaternion algebra:

$$\begin{aligned} \mathbf{u}' &= \frac{r}{2} \left( \dot{\mathbf{r}} \frac{\bar{\mathbf{u}}}{|\mathbf{u}|^2} \right)^* \\ &= \frac{1}{2} (\dot{\mathbf{r}} \bar{\mathbf{u}})^* \end{aligned} \quad (2.16)$$

The KS velocity follows directly the choice made for the fibration parameter  $\varphi$ , and, analogously to the position case, it holds that

$$\mathbf{u}' = \mathbf{u}'|_{\varphi=0} e^{i_3 \varphi} \quad (2.17)$$

the only difference being, in general, the non-null  $i_3$  component of  $\mathbf{u}'|_{\varphi=0}$  because of the quaternion product taken in Equation (2.16).

Denoting with  $(\dot{\cdot})$  the derivative with respect to the physical time, the dynamics of the two-body problem can be equivalently written by the Cartesian three-dimensional coordinate  $\underline{\mathbf{r}}$  or the quaternion with vanishing  $i_3$  component  $\mathbf{r}$ :

$$\ddot{\underline{\mathbf{r}}} = -\frac{\mu}{r^3} \underline{\mathbf{r}} \iff \ddot{\mathbf{r}} = -\frac{\mu}{r^3} \mathbf{r} \quad (2.18)$$

and the two-body energy can be also written in terms of the quaternion  $\mathbf{r}$

$$\frac{1}{2}|\dot{\mathbf{r}}|^2 - \frac{\mu}{r} = \epsilon = \text{const} \quad (2.19)$$

The fictitious time  $s$  is now introduced by the Sundman transformation of Equation (2.1) [140], which becomes:

$$dt = r \, ds; \quad \frac{d(\cdot)}{ds} = (\cdot)' \quad (2.20)$$

The derivatives with respect to the physical time  $t$  become

$$(\dot{\cdot}) = \frac{1}{r}(\cdot)'; \quad (\ddot{\cdot}) = \left(\frac{1}{r}(\cdot)'\right)' = \frac{1}{r} \left(\frac{1}{r}(\cdot)'\right)' = -\frac{1}{r^3}r'(\cdot)' + \frac{1}{r^2}(\cdot)'' \quad (2.21)$$

with the dynamics and the energy equations rewritten as:

$$\begin{aligned} r\mathbf{r}'' - r'\mathbf{r}' + \mu\mathbf{r} &= 0 \\ \frac{1}{2r^2}|\mathbf{r}'|^2 - \frac{\mu}{r} &= \epsilon \end{aligned} \quad (2.22)$$

Using Equations (2.8) and (2.10), the differentiation rule presented in Equation (2.11) gives

$$\mathbf{r}' = 2\mathbf{u}\mathbf{u}^*; \quad \mathbf{r}'' = 2\mathbf{u}\mathbf{u}^{*''} + 2\mathbf{u}'\mathbf{u}^{*'}; \quad \mathbf{r}' = \mathbf{u}'\bar{\mathbf{u}} + \mathbf{u}\bar{\mathbf{u}}' \quad (2.23)$$

Replacing the definitions of  $\mathbf{r}, \mathbf{r}', \mathbf{r}'', \mathbf{r}'$  in the dynamics and energy equations and exploiting the properties of quaternion algebra, Equation (2.23) becomes:

$$\begin{aligned} 2r\mathbf{u}^{*''} + (\mu - 2|\mathbf{u}'|^2)\mathbf{u}^* &= 0 \\ \mu - 2|\mathbf{u}'|^2 &= -r\epsilon \end{aligned} \quad (2.24)$$

Finally, the expression of the orbital energy appears in the regularized dynamics equation, leading to the simple four-dimensional harmonic oscillator [156]:

$$\mathbf{u}^{*''} - \frac{\epsilon}{2}\mathbf{u}^* = 0 \iff \mathbf{u}'' - \frac{\epsilon}{2}\mathbf{u} = 0 \quad (2.25)$$

### 2.1.2.2 Perturbed problem

Perturbing physical accelerations  $\mathbf{f}$ , if written again as a quaternion with vanishing  $i_3$  component, are accounted for simply by an additional non-null term on the right hand side of the equations of motion. The generic perturbed two-body problem becomes

$$\ddot{\mathbf{r}} = -\frac{\mu}{r^3}\mathbf{r} + \mathbf{f}(\mathbf{r}, t) \quad (2.26)$$

The derivation remains equal to the unperturbed case. Introducing the derivative with respect to the KS time gives

$$r\mathbf{r}'' - r'\mathbf{r}' + \mu\mathbf{r} = r^3\mathbf{f}(\mathbf{r}, t) \quad (2.27)$$

and using  $r = \mathbf{u}\bar{\mathbf{u}}$ , after multiplying both sides by  $\mathbf{u}^{-1}$  a similar expression to Equation (2.24) is obtained [156]

$$\begin{aligned} 2r\mathbf{u}^{*\prime\prime} + (\mu - 2|\mathbf{u}'|^2)\mathbf{u}^* &= r^2\bar{\mathbf{u}}\mathbf{f}(\mathbf{r}, t) \\ \mu - 2|\mathbf{u}'|^2 &= -r\epsilon \end{aligned} \quad (2.28)$$

Note that the energy  $\epsilon$  is not necessarily constant in the perturbed case. Replacing the energy expression in the equations of motion and dividing by  $2r$  leads to

$$\mathbf{u}^{*\prime\prime} - \frac{\epsilon}{2}\mathbf{u}^* = \frac{r}{2}\bar{\mathbf{u}}\mathbf{f}(\mathbf{r}, t) \quad (2.29)$$

The final expression is obtained by taking the star conjugate of both sides:

$$\mathbf{u}'' - \frac{\epsilon}{2}\mathbf{u} = \frac{r}{2}\mathbf{f}(\mathbf{r}, t)\bar{\mathbf{u}}^* \quad (2.30)$$

Since the physical time  $t$  can appear in  $\mathbf{f}(\mathbf{r}, t)$  either implicitly or explicitly, it should still be tracked, even though its removal was necessary to reach a simpler form of the equations of motion. Closed form expressions for  $t = t(s)$  cannot be found for the perturbed case, whereas  $s$  evolves like the eccentric anomaly [156], up to a constant, in the unperturbed two-body problem. However, from Equation (2.20) the relation  $dt/ds = r$  can be used to add the physical time as another state element for numerical integrations accounting for perturbations. The two-body energy  $\epsilon$  can either be computed at each time step with

$$\epsilon = -\frac{1}{r}(\mu - 2|\mathbf{u}'|^2) \quad (2.31)$$

or be added as another state element as well, and its derivatives are defined as:

$$\begin{aligned} \epsilon' &= \underline{\mathbf{r}}' \cdot \underline{\mathbf{f}}(\underline{\mathbf{r}}, t) \\ \dot{\epsilon} &= \dot{\underline{\mathbf{r}}} \cdot \underline{\mathbf{f}}(\underline{\mathbf{r}}, t) \end{aligned} \quad (2.32)$$

## 2.2 KS VARIABLES FOR INTERPLANETARY PROPAGATION

### 2.2.1 Barycentric KS formulation

The original KS formulation of the orbital dynamics requires the reference frame to be centered in one body, that serves both as regularization point and primary attractor for the computation of the orbital energy. However, a more complex dynamics that does not necessarily follow a dominantly two-body trajectory could not benefit from the KS regularization if it was kept in its standard form. Moreover, as it will be shown in Section 2.5.1, the barycentric formulation of the dynamics builds a more efficient simulation already in the Cartesian form, since tidal terms are not present. Such an efficiency improvement is even more relevant when accounting for relativistic effects, since the

N-bodies barycenter is the origin of the reference frame where the Post-Newtonian Einstein-Infeld-Hoffmann equations are derived [160]. Because of this reason, with the purpose of building a simulation setup that remains as general as possible in the cases it can efficiently tackle although still featuring the core benefits of the KS formulation, a barycentric formulation of the KS equations of motion is derived. For the sake of conciseness only the perturbing effects of the N bodies are presented. Other perturbing sources, such as relativistic effects and solar radiation pressure, are not explicitly written, as they would follow the same process to be included in the KS formulation of the dynamics.

Before dealing with the barycentric case and to highlight the differences with respect to the perturbing effects of the N-bodies on the traditional KS formulation, keeping the frame centered on one of the N-bodies brings:

$$\mathbf{u}'' - \frac{\epsilon}{2}\mathbf{u} = -\frac{r}{2} \sum_{\substack{i=1 \\ i \neq i_p}}^N \left( \frac{\mu_i(\mathbf{r} - \mathbf{r}_i)}{|\mathbf{r} - \mathbf{r}_i|^3} + \frac{\mu_i \mathbf{r}_i}{|\mathbf{r}_i|^3} \right) \bar{\mathbf{u}}^* \quad (2.33)$$

with  $i_p$  identifying the primary body, included in the definition of the two-body energy  $\epsilon$ ,  $\mathbf{r}_i$  and  $\mu_i$  position vector with respect to the primary and gravitational parameter of the  $i$ -th body, respectively.

A few modifications are required to write the dynamics centered in the barycenter of the N bodies involved. In general, through the regularization, smaller physical time steps are implicitly taken in the proximity of the center of the reference frame. Its correspondence with the main attractor in the Keplerian problem is not necessary, but becomes convenient when combined with the expression for the orbital energy  $\epsilon$ . In the barycentric case, the state  $\mathbf{r}$  does not identify the position with respect to the primary, therefore every single gravitational contribution must be included among the right hand side terms. Equation (2.22) for the barycentric state becomes:

$$\begin{aligned} r\mathbf{r}'' - r'\mathbf{r}' &= -\sum_{i=1}^N \frac{\mu_i(\mathbf{r} - \mathbf{r}_i)}{|\mathbf{r} - \mathbf{r}_i|^3} \\ \frac{1}{2r^2}|\mathbf{r}'|^2 - \sum_{i=1}^N \frac{\mu_i}{|\mathbf{r} - \mathbf{r}_i|} &= \epsilon_0 \end{aligned} \quad (2.34)$$

and introducing the KS variables  $\mathbf{u}$

$$\begin{aligned} \mathbf{u}'' &= \frac{|\mathbf{u}'|^2}{r}\mathbf{u} - \frac{r}{2} \sum_{i=1}^N \frac{\mu_i(\mathbf{r} - \mathbf{r}_i)}{|\mathbf{r} - \mathbf{r}_i|^3} \bar{\mathbf{u}}^* \\ \frac{2}{r}|\mathbf{u}'|^2 - \sum_{i=1}^N \frac{\mu_i}{|\mathbf{r} - \mathbf{r}_i|} &= \epsilon_0 \end{aligned} \quad (2.35)$$

The energy equation may be included in the dynamics only if accounted as the total energy  $\epsilon_0$  of the system. The presented formulation is suitable for fully numerical simulations and any other perturbing effect could be included as done for the effects of the  $N$  bodies. The numerical performances are presented in Section 2.5.1.

To extend the presented barycentric formulation to a complete Hamiltonian set of equations, the process proposed by Breiter and Langner [18] can be followed, with the only difference that the zero-energy manifold should be set on the kinetic contribution alone. This follows the way gravitational forces are treated in the force-based derivation outlined in this section, since they are all included in the right-hand side term. On the contrary, preserving a body-centric reference frame allows to keep the definition of main and perturbing gravitational potential, leading exactly to the formulation proposed by Breiter and Langner [18].

### 2.2.2 Adaptive non-dimensionalisation

When numerically simulating any dynamic phenomenon, well posed reference quantities allow the states' magnitude to remain as close as possible to the unity along the trajectory, which may boost the numerical performances of the simulator as the time steps taken can be the largest.

Traditional strategies implement static reference quantities, such as the "AU-year" based non-dimensionalization for interplanetary tasks, which takes the astronomical unit and the year as typical length and time scales,  $l_{\text{ref}}$  and  $t_{\text{ref}}$ , for the orbital phenomena. As only two out of all the four quantities involved are independent, the reference velocity  $v_{\text{ref}}$  and gravitational parameter  $\mu_{\text{ref}}$  are derived from them:

$$\begin{aligned} l_{\text{ref}} &= \text{AU} \\ t_{\text{ref}} &= \text{Year} \\ v_{\text{ref}} &= \frac{l_{\text{ref}}}{t_{\text{ref}}} \\ \mu_{\text{ref}} &= \frac{l_{\text{ref}}^3}{t_{\text{ref}}^2} \end{aligned} \tag{2.36}$$

Although simple, this reference choice is not optimal for several reasons. First, the state is well referenced only for near-Earth objects. Secondly, the gravitational parameters are never close to the unity, consequently the primary acceleration will not be close to 1. Only the Sun is represented in an acceptable way, with its gravitational parameter  $\mu$  equal to  $2\pi$  because of the relation between the year and Earth's orbital period. Lastly, were any flybys to happen, their characterizing fast dynamics would be excessively different from the non-dimensional time and length scales.

It may be reasonable to introduce another non-dimensionalization strategy, that is not sensitive to either the geometry of the possible unperturbed solution (ellipses or hyperbolas in the weakly perturbed two-body problem) and the primary attractor (the Sun for open interplanetary space or a planet for temporary flybys). The just made observations lead to the following general non-dimensionalization algorithm, valid for both interplanetary and planetary systems, starting from the dimensional state magnitudes expressed with respect to a known primary (i.e. knowing if that initial state is within or without the SOI of a planet).

The new reference dimensions are defined through a four-step process.  $\mu_{\text{ref}}$  is set equal to either the primary or the equivalent gravitational parameter, for dominantly two-body or multi-body cases respectively. Keeping a general notation to include both the equivalent and primary cases, a reference two-body energy is computed as

$$\epsilon_{\text{ref}} = -\frac{\mu_{\text{ref}}}{r} + \frac{1}{2}v^2 \quad (2.37)$$

where  $r$  and  $v$  refer to position and velocity magnitude in the current reference frame (Sun-centric, planetocentric, or barycentric).  $\epsilon_{\text{ref}}$  is used to set the reference length  $l_{\text{ref}}$  as the absolute value of the semi-major axis of the fictitious orbit that would arise from  $\mu_{\text{ref}}$  and position and velocity in the current reference frame:

$$l_{\text{ref}} = \frac{\mu_{\text{ref}}}{2|\epsilon_{\text{ref}}|} \quad (2.38)$$

Finally, the reference time  $t_{\text{ref}}$  and the reference velocity  $v_{\text{ref}}$  are obtained from the values  $\mu_{\text{ref}}$  and  $l_{\text{ref}}$ :

$$\begin{aligned} t_{\text{ref}} &= \sqrt{\frac{l_{\text{ref}}^3}{\mu_{\text{ref}}}} \\ v_{\text{ref}} &= \frac{l_{\text{ref}}}{v_{\text{ref}}} \end{aligned} \quad (2.39)$$

Note that for closed two-body orbits non-dimensionalized in the presented way the orbital period becomes equal to  $2\pi$ . The choice should be constrained to the sole principal attractor for flyby events whose dynamics is primarily two-body, so that the typical time and length scales can be properly identified.

### 2.2.3 Event interface, flyby detection and Center of integration switch

In interplanetary dominantly two-body trajectories featuring a close approach, properly catching both heliocentric cruise and flyby events with a single reference choice is impossible, in terms of non-dimensionalization. Additionally, introducing the regularization makes the dynamics sensitive to close encounters, as the integration takes implicitly smaller steps only closer to the regularization point. These issues

may be overcome by dynamically switching the center of integration in case of flyby events, following the approach presented by Amato et al. [3]. The purpose of the proposed work is only to validate the concept of dynamic center switch, without trying to identify an optimal switch distance as in [3]. The results shown in Section 2.5.1 prove that, at least for long term application, the simple definition of SOI does not affect the integration accuracy.

In particular, the following steps can be taken each time the center of integration is switched, either for initialization or flyby event. First, the state at the entrance/exit of the SOI is converted into its Cartesian and dimensional representation. Secondly, the center of the reference frame is moved by simple state vector summation. Third and last, the state is converted back in the original formulation and is made non-dimensional according to the newly updated reference length, time, velocity and gravitational parameter. The same logic can be followed for any event or function that requires the retrieval of the Cartesian coordinates. For force functions that depend on position and velocity, Equation (2.30) provides the necessary interface to map the Cartesian acceleration to its KS counterpart.

#### 2.2.4 *Fibration optimisation for numerical performance*

The right-hand sides of Equations (2.33) and (2.35) are all multiplied by their corresponding element of the KS position vector. Consequently, in the unperturbed case a null element of the vector  $\mathbf{u}$  implies a constant value for the corresponding element of  $\mathbf{u}'$  throughout the integration. Thus the same element, initially null, evolves linearly, in contrast to the sinusoidal or hyperbolic behavior of the remaining elements, depending on the bound or unbound trajectory case. As a consequence, having a null element results in an increased numerical stiffness of the system, because of the significantly different state variations.

Despite this effect remains small even in the case of null element, the degree of freedom left by choosing the fibration parameter can be exploited to optimize the simulation, maximizing the numerical stability to achieve minimized integration steps. Other than restoring a common variation trend among the state elements, the best performance can be achieved by also minimizing the difference among the variation magnitudes. A suitable choice of fibration parameter  $\varphi$  would maximize the minimum magnitude element: all the magnitudes of the remaining elements will be bounded from below and, at the same time, their magnitude will be reduced with respect to the original null-element case, to accommodate for the position constraint of Equation (2.8).



### 2.2.4.1 Necessary condition for the optimal selection of $\varphi$

First, the following lemma is formulated and proved:

**Lemma.** *Maximizing (minimizing) the value of any of the elements of both the initial and averaged KS states implies having another element with null magnitude.*

*Proof.* For both the initial KS vectors  $\mathbf{u}_0(\varphi)$  and  $\mathbf{u}'_0(\varphi)$ , denoting both the cases with  $\mathbf{p}(\varphi)$  and the four components with  $p_l(\varphi)$ ,  $l = 1, \dots, 4$  one finds

$$\frac{d\mathbf{p}(\varphi)}{d\varphi} = \{-p_4(\varphi), p_3(\varphi), -p_2(\varphi), p_1(\varphi)\}^T$$

where  $dp_l/d\varphi = 0$  is necessary for  $p_l(\varphi)$  to be locally maximized or minimized, which implies that, at the same time, another element of  $\mathbf{p}(\varphi)$  vanishes.

Since the pairs of linked elements correspond, in vector position, for both  $\mathbf{u}_0(\varphi)$  and  $\mathbf{u}'_0(\varphi)$ , the lemma holds for initial, period-average, and the interval-averaged KS states, for both closed and open trajectories, since being all linear combinations of  $\mathbf{u}_0(\varphi)$  and  $\mathbf{u}'_0(\varphi)$ .  $\square$

Finally, the necessary optimality condition is found:

**Necessary optimality condition.** *If the magnitude of the minimum magnitude element of both the initial and the averaged KS state vectors is maximized, then that specific  $\varphi^*$  makes that magnitude equal to the magnitude of another of the elements of the same vector.*

*Proof.* Any element of both  $\mathbf{u}_0(\varphi)$  and  $\mathbf{u}'_0(\varphi)$  is a continuous, periodic and bounded function of  $\varphi$ , therefore also the elements of the generic averaged KS state  $\tilde{\mathbf{p}}(\varphi)$  are so (because all the average states are linear combination of  $\mathbf{u}_0(\varphi)$  and  $\mathbf{u}'_0(\varphi)$ ).

Suppose that  $\tilde{p}_l(\varphi)$  is the local minimum magnitude element of  $\tilde{\mathbf{p}}(\varphi)$ . Then, spanning the range  $0 \leq \varphi \leq 2\pi$ , if for  $\varphi = \varphi^*$  it holds that

$$|\tilde{p}_l(\varphi^*)| = |\tilde{p}_m(\varphi^*)|$$

then if

$$|\tilde{p}_l(\varphi^* + \delta)| > |\tilde{p}_m(\varphi^* + \delta)|$$

$\tilde{p}_m(\varphi^* + \delta)$  is the new minimum magnitude element at  $\varphi^* + \delta$ , with finite  $\delta$  and  $l, m = 1, \dots, 4$ .

Because of the above stated lemma, there cannot exist any  $\varphi^*$  such that the minimum magnitude element satisfies

$$|\tilde{p}_l(\varphi^*)| = \max_{\varphi} |\tilde{p}_l(\varphi)|$$

because another element would vanish, e.g.  $|\tilde{p}_m(\varphi^*)| = 0$ , with  $m \neq l$ . Consequently,  $|\tilde{p}_l(\varphi^*)| = |\tilde{p}_m(\varphi^*)|$  at least for one pair  $(l, m)$  is necessary for  $\varphi^*$  to identify the maximized minimum magnitude element.  $\square$

The presented necessary optimality condition can be used to evaluate all the possible candidates of  $\varphi$  that maximize the minimum magnitude element of the averaged state vector, and then select the actual maximizer. Within a specified  $\varphi$  interval, at most 28 evaluations of the values of  $\varphi$  making two elements equal in magnitude are needed to check all the candidate points, for the eight-dimensional KS state including position and velocity. Analytic formulas can be easily obtained setting the various equalities among all the state elements, because they are linear on  $\sin \varphi$  and  $\cos \varphi$ . They are not reported for the sake of conciseness.

#### 2.2.4.2 Numerical support to the optimal selection of $\varphi$

A numerical analysis of the presented necessary optimality condition is given in Figures 2.2, 2.3a and 2.3b. In Figure 2.2 1000 different initial KS conditions are simulated starting from evenly spaced values of  $\varphi$  between 0 and  $2\pi$ , for a 100 year simulation of the asteroids Apophis and 2010RF<sub>12</sub>. Despite a fixed and constant value of  $\varphi$  is not necessarily optimal for all the integration intervals arising from the frame switch, the comparison with the analytic optimal  $\varphi$  for the period averaged KS state obtained from the first initial condition can already prove the optimality of the choice. To better highlight the spikes, the plotted analytic function in Figure 2.2 is  $f(\varphi) = -\log(\min |\bar{p}_l(\varphi)|)$ , with  $l = 1, \dots, 8$ , normalized and shifted to be graphically superposed to the numerical results. As expected and well predicted by the analytical spikes, the number of time steps taken increases the smaller the smallest magnitude element of the averaged (i.e. manipulated initial) state vector become. Furthermore the absolute minima for the time steps are well predicted by the points of maximized minimum magnitude element (black envelope line) in Figures 2.3a and 2.3b, where the validity of the necessary optimality condition is also confirmed as such points feature two of the vector elements (grey) with equal magnitudes. The small oscillations that can be seen in the simulation results of Figure 2.2 are due to the chaotic full dynamics being integrated instead of the unperturbed two-body problem, whose magnitude is anyway much smaller than the time step reduction achieved by selecting the optimal  $\varphi$  instead of, for instance,  $\varphi = 0$ . Choosing the optimal  $\varphi$  is therefore proved to also be a robust choice for always step-wise nearly optimal simulations in the perturbed environment.

The simulations are performed in the same framework that will be better detailed in Section 2.5.1, also the evolution of the position error  $|\Delta \mathbf{r}| = |\mathbf{r}_{\text{simulation}} - \mathbf{r}_{\text{ephemerides}}|$  with respect to the ephemerides data for the two asteroids does not change with  $\varphi$  and is always the one presented in Figures 2.4a and 2.4b, thus  $\varphi$  can be exploited with the purpose of performing faster simulations.

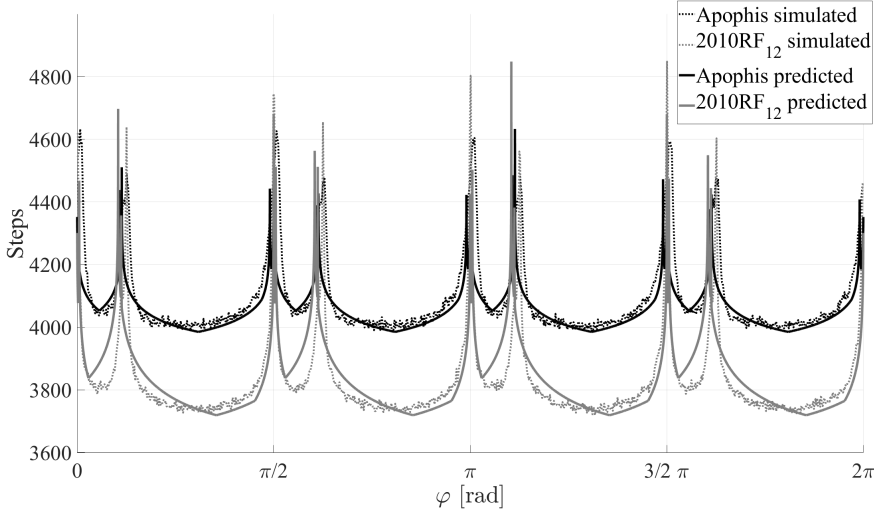
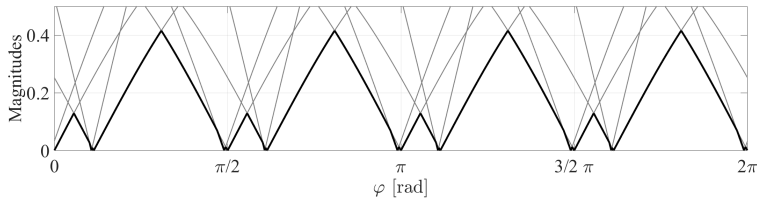
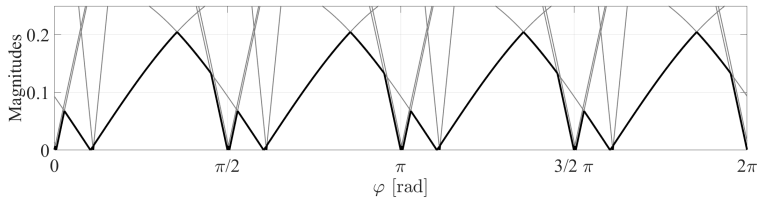


Figure 2.2: Time steps dependence on the fibration parameter  $\varphi$ .



(a) Apophis



(b) 2010RF<sub>12</sub>

Figure 2.3: Non-dimensional magnitude of the minimum magnitude element (black), compared against the magnitude variation of all the KS state elements (grey).

### 2.3 KS – CARTESIAN UNCERTAINTY MAPPING

Prior to the actual propagation of the uncertainty directly in KS coordinates, the uncertainty mapping strategy must be addressed. In fact, mapping a given probability distribution between phase spaces of different dimensions (6 for the Cartesian Case, 8 for KS coordinates) may not be straightforward. This aspect becomes fundamental for the adoption of statistical methods that aim to avoid MC-based techniques: an uncertainty description must be available in KS coordinates directly,

whose reconstruction by MC sampling and subsequent conversion may not be accurate enough.

Vallado [146] presents a wide selection of uncertainty mapping approaches for the orbital dynamics case. While useful and accurate in most scenarios, all the transformations proposed in [146] deal with phase spaces of the same dimension, which is not true for the KS-Cartesian case. Additionally, the proposed transformations all follow a Kalman Filter-based logic: after computing the Jacobian matrix with all the partial derivatives, its evaluation on the mean value is used to perform the covariance transformation. This approach relies on a linearization of the possibly non-linear transformation between coordinates, and, in general, more accurate solutions may exist. This section explores the theoretical foundations of alternative coordinate transformation approaches, with particular focus on the KS - Cartesian case, despite some of the proposed techniques could be applicable to other formulations as well. The different dimensionality of the Cartesian and KS phase spaces is also addressed.

### 2.3.1 Fibration of the KS space and uncoupled dimension raising

#### 2.3.1.1 Full and reduced fibration operators

Recalling Equation (2.13), the fibration property can be rewritten by means of vector notation, instead of the quaternion algebra concepts adopted so far. In particular, denoting  $\mathbf{v} = \mathbf{u}|_{\varphi=0} = \{v_1, v_2, v_3, 0\}^T \in \mathbb{R}^4$ ,  $\mathbf{u} = \{u_1, u_2, u_3, u_4\}^T \in \mathbb{R}^4$ , and the fibration parameter with  $\varphi$ , the following relation holds:

$$\begin{aligned} u_1 &= v_1 \cos \varphi \\ u_2 &= v_2 \cos \varphi + v_3 \sin \varphi \\ u_3 &= v_3 \cos \varphi - v_2 \sin \varphi \\ u_4 &= v_1 \sin \varphi \end{aligned} \tag{2.40}$$

and observing that the dependence of  $\mathbf{u}$  on  $\cos \varphi$  and  $\sin \varphi$  is linear, Equation (2.40) becomes

$$\mathbf{u} = \Phi(\varphi)\mathbf{v} \tag{2.41}$$

with the *fibration matrix*  $\Phi(\varphi) \in \mathbb{R}^{4 \times 4}$  defined as

$$\Phi(\varphi) = \begin{bmatrix} \cos \varphi & 0 & 0 & -\sin \varphi \\ 0 & \cos \varphi & \sin \varphi & 0 \\ 0 & -\sin \varphi & \cos \varphi & 0 \\ \sin \varphi & 0 & 0 & \cos \varphi \end{bmatrix} \tag{2.42}$$

At first glance, considering the definition of  $\mathbf{u}$  given in Equation (2.40), the last column of  $\Phi$  may seem arbitrary. Nonetheless, choosing it according to Equation (2.42) results in

$$\Phi(\varphi)^T \Phi(\varphi) = \Phi(\varphi)\Phi(\varphi)^T = \mathbf{I} \tag{2.43}$$

for any value of  $\varphi$ , with  $\mathbf{I}$  the identity matrix. This choice has several favorable implications, starting from the definition of the fibration as a standalone operator. In fact, the identity  $\Phi(\varphi)^T \Phi(\varphi) = \mathbf{I}$  allows the definition of the back-fibration that retrieves  $\mathbf{v}$  from  $\mathbf{u}$ :

$$\mathbf{v} = \Phi(\varphi)^T \mathbf{u} \quad (2.44)$$

Additionally, recalling the matrix-vector version of the mapping from KS to Cartesian coordinates presented in Equations (2.3) and (2.4), it also holds that:

$$\begin{aligned} \mathbf{L}(\mathbf{u}) &= \mathbf{L}(\mathbf{v})\Phi(\varphi)^T \\ \mathbf{L}(\mathbf{v}) &= \Phi(\varphi)\mathbf{L}(\mathbf{u}) \end{aligned} \quad (2.45)$$

Despite the evidently irrelevant role of the fibration parameter  $\varphi$  in the mapping between Cartesian coordinates and the "minimal" KS vector  $\mathbf{v}$ , the added dimension in the KS space is still required to write a complete set of KS equations of motion. In fact, other than its use for the performance optimization of numerical simulations, Equations (2.33) and (2.35) clearly show that perturbing forces result in a fourth acceleration component that is not null. Nonetheless, the uncertainty mapping is only linked to the actual mapping between KS and Cartesian coordinates, where the dynamical model used in the numerical simulation does not (yet) play any relevant part. For this reason, the introduction of a *reduced*, non-squared version of the fibration matrix may result useful. Introducing the *KS generator*  $\tilde{\mathbf{v}} = \{v_1, v_2, v_3\}^T \in \mathbb{R}^3$ , Equation (2.41) can be rewritten as

$$\mathbf{u} = \tilde{\Phi}(\varphi)\tilde{\mathbf{v}} \quad (2.46)$$

with the *reduced* fibration matrix  $\tilde{\Phi}(\varphi) \in \mathbb{R}^{4 \times 3}$  defined as:

$$\Phi(\varphi) = \begin{bmatrix} \cos \varphi & 0 & 0 \\ 0 & \cos \varphi & \sin \varphi \\ 0 & -\sin \varphi & \cos \varphi \\ \sin \varphi & 0 & 0 \end{bmatrix} \quad (2.47)$$

the one-way only identity  $\tilde{\Phi}(\varphi)^T \tilde{\Phi}(\varphi) = \mathbf{I}$  holds for the reduced fibration matrix, which however suffices to write the backward fibration as:

$$\tilde{\mathbf{v}} = \tilde{\Phi}(\varphi)^T \mathbf{u} \quad (2.48)$$

Additionally, the mapping between Cartesian position and the KS generator can be rewritten in terms of the matrix  $\tilde{\mathbf{L}}(\tilde{\mathbf{v}}) \in \mathbb{R}^{3 \times 3}$ :

$$\tilde{\mathbf{L}}(\tilde{\mathbf{v}}) = \begin{bmatrix} v_1 & -v_2 & -v_3 \\ v_2 & v_1 & 0 \\ v_3 & 0 & v_1 \end{bmatrix} \quad (2.49)$$

which preserves the retrieval of the Cartesian position components from the corresponding KS vectors, despite keeping the relation between three-dimensional vectors. In this way, the relation that retrieves the KS generator  $\mathbf{v}$  from the Cartesian position  $\mathbf{r}$  is uniquely defined by Equation (2.12).

The introduction of the reduced version of the fibration matrix and of the KS mapping highlights that the dimension raising part of the Cartesian-KS mapping can be confined to the fibration operation. In summary, the left-multiplication by  $\tilde{\Phi}(\varphi)$  results in the raising of the KS dimension, whereas the left-multiplication by its transpose reduces the KS space to its minimal representation.

*Since in general  $\tilde{\mathbf{v}}'_4 \neq 0$ , the use of the full matrix  $\Phi(\varphi)$  suffices to express the fibration of KS velocities.*

### 2.3.1.2 Fibration and statistics

Tensor notation provides a clear and concise framework to compactly write statistical moments of any order. For this reason, starting from this sub-section, all the relevant statistical quantities are presented using tensor notation, when needed. The reader can find an introduction to the tensor algebra concepts used in this work (and the related notation) in Appendix A.2.

The fibration operation is completely uncoupled by any statistics that is performed on KS variables. This can be concisely written with the relation between the  $k$ -th order raw statistical moments of the KS vectors  $\mathbf{v}$  and  $\mathbf{u}$ :

$$\mathbf{m}'_{\mathbf{u},k} = E[\mathbf{u}^{k\otimes}] = E[(\Phi(\varphi)\mathbf{v})^{k\otimes}] = \Phi(\varphi)^{k\otimes} E[\mathbf{v}^{k\otimes}] = \Phi(\varphi)^{k\otimes} \mathbf{m}'_{\mathbf{v},k} \quad (2.50)$$

which for the mean value immediately translates to

$$\mathbf{m}_{\mathbf{u}} = E[\mathbf{u}] = E[\Phi(\varphi)\mathbf{v}] = \Phi(\varphi)\mathbf{m}_{\mathbf{v}} \quad (2.51)$$

and results in the following relations for the  $k$ -th order central statistical moments:

$$\mathbf{m}_{\mathbf{u},k} = E[(\mathbf{u} - \mathbf{m}_{\mathbf{u}})^{k\otimes}] = E[\Phi(\varphi)^{k\otimes}(\mathbf{v} - \mathbf{m}_{\mathbf{v}})^{k\otimes}] = \Phi(\varphi)^{k\otimes} \mathbf{m}_{\mathbf{v},k} \quad (2.52)$$

Consequently, grouping the notation for either auto-correlation and covariance matrices with  $\mathbf{C}_{\mathbf{x}}$ , where the subscript  $\mathbf{x}$  refers to either the KS vector  $\mathbf{v}$  or  $\mathbf{u}$ , the following relations hold:

$$\begin{aligned} \mathbf{C}_{\mathbf{u}} &= \Phi(\varphi)\mathbf{C}_{\mathbf{v}}\Phi(\varphi)^{\top} \\ \mathbf{C}_{\mathbf{v}} &= \Phi(\varphi)^{\top}\mathbf{C}_{\mathbf{u}}\Phi(\varphi) \end{aligned} \quad (2.53)$$

Analogously, all the presented relations hold when using the reduced fibration matrix  $\tilde{\Phi}(\varphi)$ , making the fibration, and thus the dimension raising, irrelevant to study *any* statistics in the relationship between Cartesian coordinates and the KS generator-related moments. This aspect allows the focus on the uncertainty conversion to remain between three-dimensional spaces, delegating the dimension raising aspect to a subsequent, standalone step in the transformation process.

*The mean value notation is the only exception on the  $\mathbf{m}'$  superscript, that identify raw statistical moments, while the clean symbol  $\mathbf{m}$  represents central moments. More insight on statistics with tensor notation is given in Appendix A.2.*

## 2.3.2 Transformation methods

Taking the expectation of Equation (2.2) written in terms of the KS generator  $\tilde{\mathbf{v}}$ , that expresses the Cartesian position components as functions of the KS coordinates, gives

$$\mathbf{m}_r = E[\tilde{\mathbf{L}}(\tilde{\mathbf{v}})\tilde{\mathbf{v}}] \quad (2.54)$$

and writing it in terms of its components results in

$$\begin{aligned} m_{r_1} &= E[\tilde{v}_1^2] - E[\tilde{v}_2^2] - E[\tilde{v}_3^2] \\ m_{r_2} &= 2E[\tilde{v}_1\tilde{v}_2] \\ m_{r_3} &= 2E[\tilde{v}_1\tilde{v}_3] \end{aligned} \quad (2.55)$$

providing a direct conversion technique to obtain the Cartesian mean  $\mathbf{m}_r$ . Interestingly and compatibly with the squaring embedded in the KS to Cartesian transformation, the Cartesian mean depends on some of the KS autocorrelation matrix elements only. In particular, the transformation is similar to the original KS to Cartesian mapping, when applied to random variables.

Equation (2.2) can be rewritten for the KS generator  $\tilde{\mathbf{v}}$  case as the generic quadratic form for its  $i$ -th component:

$$r_i = \tilde{\mathbf{v}}^T \mathbf{H}_i \tilde{\mathbf{v}} \quad (2.56)$$

with  $i = 1, 2, 3$  and

$$\mathbf{H}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \quad \mathbf{H}_2 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{H}_3 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad (2.57)$$

The matrices  $\mathbf{H}_i$  can be seen as the Hessians of the transformations  $r_i = r_i(\mathbf{v})$ . Observing the properties of the fibration matrix formalism,  $\tilde{\Phi}(\varphi)$  can be used to map the fibrated and the generator-related Hessians as well:

$$\begin{aligned} \mathbf{H}_i(\mathbf{u}) &= \tilde{\Phi}(\varphi) \mathbf{H}_i(\tilde{\mathbf{v}}) \tilde{\Phi}(\varphi)^T \\ \mathbf{H}_i(\tilde{\mathbf{v}}) &= \tilde{\Phi}(\varphi)^T \mathbf{H}_i(\mathbf{u}) \tilde{\Phi}(\varphi) \end{aligned} \quad (2.58)$$

Introducing these quadratic forms, the components of covariance and autocorrelation matrices depend on the generic second order expectation  $E[r_i r_j]$ , with  $i, j = 1, 2, 3$ , that can be written as the bi-quadratic form

$$m'_{r_i r_j} = E[r_i r_j] = E[(\tilde{\mathbf{v}}^T \mathbf{H}_i \tilde{\mathbf{v}})(\tilde{\mathbf{v}}^T \mathbf{H}_j \tilde{\mathbf{v}})] \quad (2.59)$$

which can be re-arranged as

$$m'_{r_i r_j} = E[\tilde{\mathbf{v}}_2^T \mathbf{H}_{ij} \tilde{\mathbf{v}}_2] \quad (2.60)$$

with  $\tilde{\mathbf{v}}_2 = \{v_1^2, v_2^2, v_3^2, v_1v_2, v_1v_3, v_2v_3\}^T$  and the constant matrices  $\mathbf{H}_{ij}$  that can be derived by the product of the  $i$ -th and  $j$ -th quadratic forms:

$$\mathbf{H}_{11} = \begin{bmatrix} \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & -2\mathbf{I}_{3 \times 3} \end{bmatrix}, \mathbf{H}_{22} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ & 4 & 0 & 0 \\ \mathbf{0}_{3 \times 3} & 0 & 0 & 0 \\ & 0 & 0 & 0 \end{bmatrix}, \quad (2.61)$$

$$\mathbf{H}_{33} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ & 0 & 0 & 0 \\ \mathbf{0}_{3 \times 3} & 0 & 4 & 0 \\ & 0 & 0 & 0 \end{bmatrix}, \mathbf{H}_{12} = \begin{bmatrix} & & & 1 & 0 & 0 \\ & \mathbf{0}_{3 \times 3} & & -1 & 0 & 0 \\ & & & -1 & 0 & 0 \\ 1 & -1 & -1 & & & \\ 0 & 0 & 0 & & \mathbf{0}_{3 \times 3} & \\ 0 & 0 & 0 & & & \end{bmatrix},$$

$$\mathbf{H}_{13} = \begin{bmatrix} & & & 0 & 1 & 0 \\ & \mathbf{0}_{3 \times 3} & & 0 & 0 & 0 \\ & & & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \end{bmatrix}, \mathbf{H}_{23} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ & 0 & 2 & 0 \\ \mathbf{0}_{3 \times 3} & 2 & 0 & 0 \\ & 0 & 0 & 0 \end{bmatrix}$$

Equation (2.60) highlights the presence of 4<sup>th</sup> order moments of  $\tilde{\mathbf{v}}$  in the Cartesian autocorrelation matrix. In particular, since for any quadratic forms it holds that  $\mathbf{x}^T \mathbf{A} \mathbf{x} \equiv \text{Tr}(\mathbf{x}^T \mathbf{A} \mathbf{x})$  then, following [96], the cyclic property of the trace operator gives

$$\mathbb{E}[\text{Tr}(\mathbf{x}^T \mathbf{A} \mathbf{x})] = \mathbb{E}[\text{Tr}(\mathbf{A} \mathbf{x} \mathbf{x}^T)]$$

and exploiting the linearity of both expectation and trace operators results in

$$\mathbb{E}[\mathbf{x}^T \mathbf{A} \mathbf{x}] = \text{Tr}(\mathbf{A} \mathbb{E}[\mathbf{x} \mathbf{x}^T]) = \text{Tr}(\mathbf{A} \mathbf{R}_{\mathbf{x}}) \quad (2.62)$$

with  $\mathbf{R}_{\mathbf{x}}$  autocorrelation matrix associated to the random vector  $\mathbf{x}$ . Similarly, using  $\mathbf{P}_{\mathbf{x}} = \mathbf{R}_{\mathbf{x}} - \mathbf{m}_{\mathbf{x}} \mathbf{m}_{\mathbf{x}}^T$  leads to the following definition involving the covariance matrix  $\mathbf{P}_{\mathbf{x}}$ :

$$\begin{aligned} \mathbb{E}[\mathbf{x}^T \mathbf{A} \mathbf{x}] &= \text{Tr}(\mathbf{A} \mathbf{P}_{\mathbf{x}} + \mathbf{A} \mathbf{m}_{\mathbf{x}} \mathbf{m}_{\mathbf{x}}^T) \\ &= \text{Tr}(\mathbf{A} \mathbf{P}_{\mathbf{x}}) + \text{Tr}(\mathbf{m}_{\mathbf{x}}^T \mathbf{A} \mathbf{m}_{\mathbf{x}}) \\ &= \text{Tr}(\mathbf{A} \mathbf{P}_{\mathbf{x}}) + \mathbf{m}_{\mathbf{x}}^T \mathbf{A} \mathbf{m}_{\mathbf{x}} \end{aligned} \quad (2.63)$$

Thus, the elements of the Cartesian autocorrelation matrix can then be obtained as

$$m'_{r_i r_j} = \text{Tr}(\mathbf{H}_{ij} \mathbf{R}_{\tilde{\mathbf{v}}_2}) \quad (2.64)$$

$\mathbf{R}_{\tilde{\mathbf{v}}_2} = \mathbf{m}'_{\tilde{\mathbf{v}}_2, 2}$  contains, in general, the fourth order raw moments of  $\tilde{\mathbf{v}}$ . In summary, the terms of the Cartesian autocorrelation matrix  $\mathbf{R}_{\mathbf{r}}$  can be written as function of the statistical moments of the corresponding KS generator up to order 4.



## 2.3.2.1 Multivariate normal distribution case

Multivariate normal distributions feature a simpler structure, in terms of higher order central moments. In particular, for  $\mathbf{x} \in \mathbb{R}^N$  [68]:

- $m_{x_1^{\alpha_1}, \dots, x_N^{\alpha_N}} \equiv 0$  if  $\sum_{\eta=1}^N \alpha_\eta = k$  with  $k$  odd; more generally, the statement for the odd central moments is true for any symmetric distribution;
- consequently,  $m_{x_1^{\alpha_1}, \dots, x_N^{\alpha_N}} = \sum \prod m_{x_i x_j}$ , with  $\sum_{\eta=1}^N \alpha_\eta = k = 2\lambda$  ( $k$  even), i.e. summing the products of all the possible pairs of the covariance matrix elements.

In the case of normal distribution, the fourth order central moment (i.e. the kurtosis) depends only on the covariance matrix elements:

$$m_{x_i x_j x_k x_l} = m_{x_i x_j} m_{x_k x_l} + m_{x_i x_k} m_{x_j x_l} + m_{x_i x_l} m_{x_j x_k} \quad (2.65)$$

Equation (2.65) can be used to obtain all the possible fourth order moments simply by replacing  $(i, j, k, l)$  with the corresponding variable index (e.g.  $m_{x_i^2 x_j^2} = m_{x_i x_i x_j x_j}$ ).

Despite the null odd order central moments, the same does not apply to the raw moment case. For the third order this results in

$$m'_{x_i x_j x_k} = E[x_i x_j x_k] \neq 0 \quad (2.66)$$

However, the identity

$$m_{x_i x_j x_k} = E[(x_i - m_{x_i})(x_j - m_{x_j})(x_k - m_{x_k})] \equiv 0 \quad (2.67)$$

provides an expression for the third order raw moment in terms of lower order moments:

$$\begin{aligned} m'_{x_i x_j x_k} &= m_{x_i} E[x_j x_k] + m_{x_j} E[x_i x_k] + m_{x_k} E[x_i x_j] + m_{x_i} m_{x_j} m_{x_k} \\ &= m_{x_i} m'_{x_j x_k} + m_{x_j} m'_{x_i x_k} + m_{x_k} m'_{x_i x_j} + m_{x_i} m_{x_j} m_{x_k} \end{aligned} \quad (2.68)$$

and similarly, the fourth order raw moments can be retrieved expanding the following definition

$$m_{x_i x_j x_k x_l} = E[(x_i - m_{x_i})(x_j - m_{x_j})(x_k - m_{x_k})(x_l - m_{x_l})]$$

and combining it with Equation (2.65).

Finally, given a multivariate normal KS generator  $\tilde{\mathbf{v}}$  with mean  $\mathbf{m}_{\tilde{\mathbf{v}}}$  and covariance  $\mathbf{P}_{\tilde{\mathbf{v}}}$ , the elements of  $\mathbf{m}_{\mathbf{r}}$  can be retrieved applying Equation (2.55). The elements of the Cartesian autocorrelation matrix  $\mathbf{R}_{\mathbf{r}}$  can be seen as a function of also the KS mean instead:

$$\mathbf{R}_{\mathbf{r}} = \mathbf{R}_{\mathbf{r}}(\mathbf{m}_{\tilde{\mathbf{v}}}, \mathbf{P}_{\tilde{\mathbf{v}}}) \quad (2.69)$$

where the dependence on the elements of  $\mathbf{P}_{\bar{v}}$  is quadratic, contrarily to the explicit presence of fourth order moments in the general case (Equation (2.64)). The full expressions are given by:

$$\begin{aligned}
R_{r,11} &= 3P_{\bar{v},11}^2 - 2P_{\bar{v},11}P_{\bar{v},22} - 2P_{\bar{v},11}P_{\bar{v},33} - 4P_{\bar{v},12}^2 - 4P_{\bar{v},13}^2 + 3P_{\bar{v},22}^2 \\
&\quad + 2P_{\bar{v},22}P_{\bar{v},33} + 4P_{\bar{v},23}^2 + 3P_{\bar{v},33}^2 - 2m_{\bar{v},1}^4 + 4m_{\bar{v},1}^2m_{\bar{v},2}^2 \\
&\quad + 4m_{\bar{v},1}^2m_{\bar{v},3}^2 - 2m_{\bar{v},2}^4 - 4m_{\bar{v},2}^2m_{\bar{v},3}^2 - 2m_{\bar{v},3}^4 \\
R_{r,12} &= -4m_{\bar{v},1}^3m_{\bar{v},2} + 4m_{\bar{v},1}m_{\bar{v},2}^3 + 4m_{\bar{v},1}m_{\bar{v},2}m_{\bar{v},3}^2 + 6P_{\bar{v},11}P_{\bar{v},12} \\
&\quad - 6P_{\bar{v},12}P_{\bar{v},22} - 2P_{\bar{v},12}P_{\bar{v},33} - 4P_{\bar{v},13}P_{\bar{v},23} \\
R_{r,13} &= -4m_{\bar{v},1}^3m_{\bar{v},3} + 4m_{\bar{v},1}m_{\bar{v},2}^2m_{\bar{v},3} + 4m_{\bar{v},1}m_{\bar{v},3}^3 + 6P_{\bar{v},11}P_{\bar{v},13} \\
&\quad - 4P_{\bar{v},12}P_{\bar{v},23} - 2P_{\bar{v},13}P_{\bar{v},22} - 6P_{\bar{v},13}P_{\bar{v},33} \\
R_{r,22} &= 8P_{\bar{v},12}^2 - 8m_{\bar{v},1}^2m_{\bar{v},2}^2 + 4P_{\bar{v},11}P_{\bar{v},22} \\
R_{r,23} &= -8m_{\bar{v},2}m_{\bar{v},3}m_{\bar{v},1}^2 + 4P_{\bar{v},11}P_{\bar{v},23} + 8P_{\bar{v},12}P_{\bar{v},13} \\
R_{r,33} &= 8P_{\bar{v},13}^2 - 8m_{\bar{v},1}^2m_{\bar{v},3}^2 + 4P_{\bar{v},11}P_{\bar{v},33}
\end{aligned} \tag{2.70}$$

Equations (2.55) and (2.70) are a set of 3 linear and 6 quadratic equations in 9 unknowns, making it thus theoretically possible to retrieve  $(\mathbf{m}_{\bar{v}}, \mathbf{P}_{\bar{v}})$  from  $(\mathbf{m}_r, \mathbf{R}_r)$ . Despite two of the elements of the KS autocorrelation matrix could be retrieved directly, numerical solvers would need to work on a sum-of-square relaxation of the remaining 7 equations, resulting in a system of quadratic and fourth-power equations in 7 unknowns. The inspection of the Hessian matrices  $\mathbf{H}_1$  and  $\mathbf{H}_{ij}$ , with  $i, j = 1, 2, 3$ , shows that they all have *at least* one negative eigenvalue, making *each* of the 9 equations non-convex. In other words, the convergence to the actual solution of the non-linear system would happen only for very good initial guesses. This aspect makes the numerical solution of this system a NP-hard problem, despite its structural simplicity. Therefore, the use of the proposed solution process is recommended for the conversion of a distribution from KS to Cartesian, while alternative approaches should be explored for the inverse case.

Additionally, the proposed analytical process is valid for the KS position term only, although it would be required for the full state vector for a complete uncertainty mapping. Analogous solutions could be found for the position differentials, nonetheless the transformation of the cross and velocity terms would introduce a further non-linearity in the expectation operator. For this reason, exploring alternative conversion techniques may overcome the limitation of the proposed analytical solution, extending the conversion process to the full state vectors.

### 2.3.2.2 MC transformation

The conceptually simplest process to convert an uncertainty distribution relies on the MC approach and the definitions of sample mean

and sample covariance. The conversion process follows four main steps:

1. given a Cartesian/KS distribution with mean  $\mathbf{m}$  and covariance  $\mathbf{P}$  and knowing its distribution structure,  $N$  samples can be generated;
2. Equations (2.2) and (2.12) provide the direct and inverse conversion formulas for all the generated samples;
3. the sample mean can be estimated with  $\mathbf{m} \approx \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i$ ;
4. the sample covariance can be estimated with

$$\mathbf{P} \approx \frac{1}{N-1} \sum_{i=1}^N (\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^T$$

The MC approach is a robust method and can work on the full state vector, yet it may require a large number of samples, and thus a high computational burden, to reach high accuracy levels. Nonetheless, this technique can also be used to validate other possible conversion strategies.

### 2.3.2.3 Extended Kalman Filter-based transformation

The evolution of non-linear dynamical systems has been of primary interest in several space applications. For estimation problems, the Kalman Filter (KF) [77] still represent the reference technique for linear systems. Its primary goal is to generate an estimate of the future evolution of a propagated uncertainty, that can be used with observed data to enhance the control action robustness on the given dynamical system. For uncertainty transformations that are intrinsically non-linear, a local linearization may still provide a sufficiently accurate results. The core idea is to work on a Taylor expansion of the uncertainty transformation function truncated to the first order, originating the Extended Kalman Filter (EKF) concept. The earliest works on this technique employed it, for instance, for the on-board spacecraft position and velocity estimation, given a set of measured data [138], and applying it to the navigation software for interplanetary flybys [98].

Deepening the study on the many applications of KF and EKF is beyond the scopes of this dissertation, thus only a few key concepts are presented and used. Non-linear control problems typically use the EKF to get an estimate of the uncertainty at a future time, with respect to the present system observation. Nonetheless, the EKF formulation refers to the generic uncertainty transformation, without necessarily linking it to an evolution in time [77]. For the proposed application, assuming

zero control action, process, and observation noise is reasonable, and makes the generic prediction for mean  $\mathbf{x}$  and covariance  $\mathbf{P}$  become:

$$\begin{aligned}\mathbf{x}_{\text{predicted}} &= \mathbf{f}(\mathbf{x}) \\ \mathbf{P}_{\text{predicted}} &= \mathbf{F}\mathbf{P}\mathbf{F}^T\end{aligned}\quad (2.71)$$

with  $\mathbf{F}$  the Jacobian of the nonlinear function  $\mathbf{f}$ .

Writing the KS to Cartesian transformation,  $\mathbf{r}(\tilde{\mathbf{v}}) = \mathbf{L}(\tilde{\mathbf{v}})\tilde{\mathbf{v}}$  can be used to obtain an EKF-based estimation of the Cartesian mean:

$$\mathbf{m}_{\mathbf{r}} \approx \mathbf{L}(\mathbf{m}_{\tilde{\mathbf{v}}})\mathbf{m}_{\tilde{\mathbf{v}}}\quad (2.72)$$

In this context, the EKF-based estimation of the KS mean can be obtained simply applying Equation (2.12) on the Cartesian mean. The same equation can be used to write a Taylor expansion of the KS to Cartesian transformation about the mean  $\mathbf{m}_{\tilde{\mathbf{v}}}$ , truncated to the first order:

$$\begin{aligned}\mathbf{r} &\approx \mathbf{r}(\mathbf{m}_{\tilde{\mathbf{v}}}) + \nabla \mathbf{r}_{\tilde{\mathbf{v}}}|_{\mathbf{m}_{\tilde{\mathbf{v}}}}(\tilde{\mathbf{v}} - \mathbf{m}_{\tilde{\mathbf{v}}}) \\ &= \mathbf{L}(\mathbf{m}_{\tilde{\mathbf{v}}})\mathbf{m}_{\tilde{\mathbf{v}}} + 2\mathbf{L}(\mathbf{m}_{\tilde{\mathbf{v}}})(\tilde{\mathbf{v}} - \mathbf{m}_{\tilde{\mathbf{v}}}) \\ &= \mathbf{L}(\mathbf{m}_{\tilde{\mathbf{v}}})(2\tilde{\mathbf{v}} - \mathbf{m}_{\tilde{\mathbf{v}}})\end{aligned}\quad (2.73)$$

which once plugged in the definition of autocorrelation matrix  $\mathbf{R}_{\mathbf{r}} = \mathbb{E}[\mathbf{r}\mathbf{r}^T]$  gives

$$\begin{aligned}\mathbf{R}_{\mathbf{r}} &\approx \mathbb{E}[\mathbf{L}(\mathbf{m}_{\tilde{\mathbf{v}}})(2\tilde{\mathbf{v}} - \mathbf{m}_{\tilde{\mathbf{v}}})(2\tilde{\mathbf{v}} - \mathbf{m}_{\tilde{\mathbf{v}}})^T \mathbf{L}(\mathbf{m}_{\tilde{\mathbf{v}}})^T] \\ &= \mathbf{L}(\mathbf{m}_{\tilde{\mathbf{v}}})(4\mathbb{E}[\tilde{\mathbf{v}}\tilde{\mathbf{v}}^T] - 3\mathbf{m}_{\tilde{\mathbf{v}}}\mathbf{m}_{\tilde{\mathbf{v}}}^T)\mathbf{L}(\mathbf{m}_{\tilde{\mathbf{v}}})^T \\ &= \mathbf{L}(\mathbf{m}_{\tilde{\mathbf{v}}})(4\mathbf{R}_{\tilde{\mathbf{v}}} - 3\mathbf{m}_{\tilde{\mathbf{v}}}\mathbf{m}_{\tilde{\mathbf{v}}}^T)\mathbf{L}(\mathbf{m}_{\tilde{\mathbf{v}}})^T\end{aligned}\quad (2.74)$$

and recalling that  $\mathbf{P} = \mathbf{R} - \mathbf{m}\mathbf{m}^T$  leads to

$$\begin{aligned}\mathbf{P}_{\mathbf{r}} &\approx \mathbf{L}(\mathbf{m}_{\tilde{\mathbf{v}}})(4\mathbf{R}_{\tilde{\mathbf{v}}} - 3\mathbf{m}_{\tilde{\mathbf{v}}}\mathbf{m}_{\tilde{\mathbf{v}}}^T)\mathbf{L}(\mathbf{m}_{\tilde{\mathbf{v}}})^T - \mathbf{m}_{\mathbf{r}}\mathbf{m}_{\mathbf{r}}^T \\ &\approx \mathbf{L}(\mathbf{m}_{\tilde{\mathbf{v}}})(4\mathbf{R}_{\tilde{\mathbf{v}}} - 3\mathbf{m}_{\tilde{\mathbf{v}}}\mathbf{m}_{\tilde{\mathbf{v}}}^T)\mathbf{L}(\mathbf{m}_{\tilde{\mathbf{v}}})^T - \mathbf{L}(\mathbf{m}_{\tilde{\mathbf{v}}})\mathbf{m}_{\tilde{\mathbf{v}}}\mathbf{m}_{\tilde{\mathbf{v}}}^T\mathbf{L}(\mathbf{m}_{\tilde{\mathbf{v}}})^T \\ &= 4\mathbf{L}(\mathbf{m}_{\tilde{\mathbf{v}}})(\mathbf{R}_{\tilde{\mathbf{v}}} - \mathbf{m}_{\tilde{\mathbf{v}}}\mathbf{m}_{\tilde{\mathbf{v}}}^T)\mathbf{L}(\mathbf{m}_{\tilde{\mathbf{v}}})^T \\ &= 4\mathbf{L}(\mathbf{m}_{\tilde{\mathbf{v}}})\mathbf{P}_{\tilde{\mathbf{v}}}\mathbf{L}(\mathbf{m}_{\tilde{\mathbf{v}}})^T\end{aligned}\quad (2.75)$$

that in turn can be re-arranged to obtain the inverse formula:

$$\mathbf{P}_{\tilde{\mathbf{v}}} \approx \frac{1}{4}\mathbf{L}(\mathbf{m}_{\tilde{\mathbf{v}}})^{-1}\mathbf{P}_{\mathbf{r}}\mathbf{L}(\mathbf{m}_{\tilde{\mathbf{v}}})^{-T}\quad (2.76)$$

The linearization introduces the necessary simplification to deal with the full state conversion, including velocity and cross-correlation terms. In particular, the Cartesian velocity depends on both KS position and velocity. Rewriting Equation (2.15) in matrix form gives

$$\dot{\mathbf{r}} = 2\mathbf{r}\mathbf{L}(\tilde{\mathbf{v}}')\tilde{\mathbf{v}}\quad (2.77)$$

or equivalently, because of the conformality constraint of Equation (2.11)

$$\dot{\mathbf{r}} = 2\mathbf{r}\mathbf{L}(\tilde{\mathbf{v}})\tilde{\mathbf{v}}'\quad (2.78)$$

Despite the conformality constraint guarantees that  $\mathbf{L}(\tilde{\mathbf{v}})\tilde{\mathbf{v}}' \equiv \mathbf{L}(\tilde{\mathbf{v}}')\tilde{\mathbf{v}}$ , the two matrices differ in dimension, for the KS generator case. In the full-dimensional case, the elements of both matrices follow the definition presented in Equation (2.4), whereas a distinction must be made for  $\tilde{\mathbf{v}}$  and  $\tilde{\mathbf{v}}'$ , because in general  $\tilde{v}'_4 \neq 0$ .  $\mathbf{L}(\tilde{\mathbf{v}})$  remains the reduced matrix defined in Equation (2.49), while  $\mathbf{L}(\tilde{\mathbf{v}}')$  follows the original definition, where the conformality constraint allows to completely neglect its fourth row:

$$\mathbf{L}(\tilde{\mathbf{v}}') = \begin{bmatrix} \tilde{v}'_1 & -\tilde{v}'_2 & -\tilde{v}'_3 & \tilde{v}'_4 \\ \tilde{v}'_2 & \tilde{v}'_1 & -\tilde{v}'_4 & -\tilde{v}'_3 \\ \tilde{v}'_3 & \tilde{v}'_4 & \tilde{v}'_1 & \tilde{v}'_2 \end{bmatrix} \quad (2.79)$$

The EKF Taylor expansion for the Cartesian velocity as function of KS variables, truncated to the first order, is

$$\dot{\mathbf{r}} \approx \mathbf{m}_{\dot{\mathbf{r}}} + \nabla \dot{\mathbf{r}}_{\tilde{\mathbf{v}}|\mathbf{m}_{\tilde{\mathbf{v}}}, \mathbf{m}'_{\tilde{\mathbf{v}}}}(\tilde{\mathbf{v}} - \mathbf{m}_{\tilde{\mathbf{v}}}) + \nabla \dot{\mathbf{r}}_{\tilde{\mathbf{v}}'|\mathbf{m}_{\tilde{\mathbf{v}}}, \mathbf{m}'_{\tilde{\mathbf{v}}}}(\tilde{\mathbf{v}}' - \mathbf{m}'_{\tilde{\mathbf{v}}}) \quad (2.80)$$

where the two gradients are defined as

$$\nabla \dot{\mathbf{r}}_{\tilde{\mathbf{v}}|\mathbf{m}_{\tilde{\mathbf{v}}}, \mathbf{m}'_{\tilde{\mathbf{v}}}} = 2\mathbf{m}_r \mathbf{L}(\mathbf{m}_{\tilde{\mathbf{v}}'}) + 4\mathbf{L}(\mathbf{m}_{\tilde{\mathbf{v}}'}) \mathbf{m}_{\tilde{\mathbf{v}}} \mathbf{m}_{\tilde{\mathbf{v}}}^T \quad (2.81)$$

and, exploiting  $\mathbf{L}(\tilde{\mathbf{v}})\tilde{\mathbf{v}}' \equiv \mathbf{L}(\tilde{\mathbf{v}}')\tilde{\mathbf{v}}$ ,

$$\nabla \dot{\mathbf{r}}_{\tilde{\mathbf{v}}'|\mathbf{m}_{\tilde{\mathbf{v}}}, \mathbf{m}'_{\tilde{\mathbf{v}}}} = 2\mathbf{m}_r \mathbf{L}(\mathbf{m}_{\tilde{\mathbf{v}}}) \quad (2.82)$$

Replacing the expansions in the cross-covariance terms provides the following estimation for the position-velocity cross-correlation matrix  $\mathbf{R}_{\mathbf{r}, \dot{\mathbf{r}}}$ :

$$\mathbb{E}[\mathbf{r}\dot{\mathbf{r}}^T] \approx \mathbf{m}_r \mathbf{m}_{\dot{\mathbf{r}}}^T + \nabla \mathbf{r}_{\tilde{\mathbf{v}}|\mathbf{m}_{\tilde{\mathbf{v}}}} \mathbf{P}_{\tilde{\mathbf{v}}} \nabla \dot{\mathbf{r}}_{\tilde{\mathbf{v}}|\mathbf{m}_{\tilde{\mathbf{v}}}, \mathbf{m}'_{\tilde{\mathbf{v}}}}^T + \nabla \mathbf{r}_{\tilde{\mathbf{v}}|\mathbf{m}_{\tilde{\mathbf{v}}}} \mathbf{P}_{\tilde{\mathbf{v}}, \tilde{\mathbf{v}}'} \nabla \dot{\mathbf{r}}_{\tilde{\mathbf{v}}'|\mathbf{m}_{\tilde{\mathbf{v}}}, \mathbf{m}'_{\tilde{\mathbf{v}}}}^T \quad (2.83)$$

and thus the cross-covariance becomes

$$\mathbf{P}_{\mathbf{r}, \dot{\mathbf{r}}} = \nabla \mathbf{r}_{\tilde{\mathbf{v}}|\mathbf{m}_{\tilde{\mathbf{v}}}} \mathbf{P}_{\tilde{\mathbf{v}}} \nabla \dot{\mathbf{r}}_{\tilde{\mathbf{v}}|\mathbf{m}_{\tilde{\mathbf{v}}}, \mathbf{m}'_{\tilde{\mathbf{v}}}}^T + \nabla \mathbf{r}_{\tilde{\mathbf{v}}|\mathbf{m}_{\tilde{\mathbf{v}}}} \mathbf{P}_{\tilde{\mathbf{v}}, \tilde{\mathbf{v}}'} \nabla \dot{\mathbf{r}}_{\tilde{\mathbf{v}}'|\mathbf{m}_{\tilde{\mathbf{v}}}, \mathbf{m}'_{\tilde{\mathbf{v}}}}^T \quad (2.84)$$

with explicit dependence on both the KS cross-covariance and the KS position covariance. Similarly, the velocity autocorrelation matrix  $\mathbf{R}_{\dot{\mathbf{r}}}$  reads

$$\begin{aligned} \mathbb{E}[\dot{\mathbf{r}}\dot{\mathbf{r}}^T] &\approx \mathbf{m}_{\dot{\mathbf{r}}} \mathbf{m}_{\dot{\mathbf{r}}}^T + \nabla \dot{\mathbf{r}}_{\tilde{\mathbf{v}}|\mathbf{m}_{\tilde{\mathbf{v}}}, \mathbf{m}'_{\tilde{\mathbf{v}}}} \mathbf{P}_{\tilde{\mathbf{v}}} \nabla \dot{\mathbf{r}}_{\tilde{\mathbf{v}}|\mathbf{m}_{\tilde{\mathbf{v}}}, \mathbf{m}'_{\tilde{\mathbf{v}}}}^T + \nabla \dot{\mathbf{r}}_{\tilde{\mathbf{v}}|\mathbf{m}_{\tilde{\mathbf{v}}}, \mathbf{m}'_{\tilde{\mathbf{v}}}} \mathbf{P}_{\tilde{\mathbf{v}}, \tilde{\mathbf{v}}'} \nabla \dot{\mathbf{r}}_{\tilde{\mathbf{v}}'|\mathbf{m}_{\tilde{\mathbf{v}}}, \mathbf{m}'_{\tilde{\mathbf{v}}}}^T \\ &\quad + \nabla \dot{\mathbf{r}}_{\tilde{\mathbf{v}}'|\mathbf{m}_{\tilde{\mathbf{v}}}, \mathbf{m}'_{\tilde{\mathbf{v}}}} \mathbf{P}_{\tilde{\mathbf{v}}', \tilde{\mathbf{v}}} \nabla \dot{\mathbf{r}}_{\tilde{\mathbf{v}}|\mathbf{m}_{\tilde{\mathbf{v}}}, \mathbf{m}'_{\tilde{\mathbf{v}}}}^T + \nabla \dot{\mathbf{r}}_{\tilde{\mathbf{v}}'|\mathbf{m}_{\tilde{\mathbf{v}}}, \mathbf{m}'_{\tilde{\mathbf{v}}}} \mathbf{P}_{\tilde{\mathbf{v}}'} \nabla \dot{\mathbf{r}}_{\tilde{\mathbf{v}}'|\mathbf{m}_{\tilde{\mathbf{v}}}, \mathbf{m}'_{\tilde{\mathbf{v}}}}^T \end{aligned} \quad (2.85)$$

and thus the velocity covariance  $\mathbf{P}_{\dot{\mathbf{r}}}$  is

$$\begin{aligned} \mathbf{P}_{\dot{\mathbf{r}}} &\approx \nabla \dot{\mathbf{r}}_{\tilde{\mathbf{v}}|\mathbf{m}_{\tilde{\mathbf{v}}}, \mathbf{m}'_{\tilde{\mathbf{v}}}} \mathbf{P}_{\tilde{\mathbf{v}}} \nabla \dot{\mathbf{r}}_{\tilde{\mathbf{v}}|\mathbf{m}_{\tilde{\mathbf{v}}}, \mathbf{m}'_{\tilde{\mathbf{v}}}}^T + \nabla \dot{\mathbf{r}}_{\tilde{\mathbf{v}}|\mathbf{m}_{\tilde{\mathbf{v}}}, \mathbf{m}'_{\tilde{\mathbf{v}}}} \mathbf{P}_{\tilde{\mathbf{v}}, \tilde{\mathbf{v}}'} \nabla \dot{\mathbf{r}}_{\tilde{\mathbf{v}}'|\mathbf{m}_{\tilde{\mathbf{v}}}, \mathbf{m}'_{\tilde{\mathbf{v}}}}^T \\ &\quad + \nabla \dot{\mathbf{r}}_{\tilde{\mathbf{v}}'|\mathbf{m}_{\tilde{\mathbf{v}}}, \mathbf{m}'_{\tilde{\mathbf{v}}}} \mathbf{P}_{\tilde{\mathbf{v}}', \tilde{\mathbf{v}}} \nabla \dot{\mathbf{r}}_{\tilde{\mathbf{v}}|\mathbf{m}_{\tilde{\mathbf{v}}}, \mathbf{m}'_{\tilde{\mathbf{v}}}}^T + \nabla \dot{\mathbf{r}}_{\tilde{\mathbf{v}}'|\mathbf{m}_{\tilde{\mathbf{v}}}, \mathbf{m}'_{\tilde{\mathbf{v}}}} \mathbf{P}_{\tilde{\mathbf{v}}'} \nabla \dot{\mathbf{r}}_{\tilde{\mathbf{v}}'|\mathbf{m}_{\tilde{\mathbf{v}}}, \mathbf{m}'_{\tilde{\mathbf{v}}}}^T \end{aligned} \quad (2.86)$$

As expected by the explicit appearance of the KS position  $\tilde{\mathbf{v}}$ , the Cartesian velocity autocorrelation and covariance matrices depend on

*The quaternion product in Equation (2.16) results in a fourth component that is not null.*

*r is also a function of  $\tilde{\mathbf{v}}$ .*

all the position, cross, and velocity sub-matrices of the KS covariance. This fact makes the backward conversion, i.e. from Cartesian to KS covariance, not as direct as the simple inverse formula in the position case: the process must follow three sequential steps, the first being the retrieval of the KS position covariance with Equation (2.76), the second to compute the KS cross-covariance by reversing Equation (2.84), and the final to give the KS velocity covariance by re-arranging Equation (2.86).

#### 2.3.2.4 Unscented transform

The EKF operates with two separate approximations, i.e. the linearization of the uncertainty transformation function and the estimation of the transformed uncertainty. The Unscented Transform (UT) takes instead the original transformation function, and applies it to a set of suitably chosen points to estimate the transformed covariance [74, 75]. This has the advantage of retaining at least one more order of magnitude in the transformation accuracy, since the linear approximation introduced with the EKF is removed.

The first step of the UT-based process, is the sampling of the *sigma points*. For the KS generator  $\tilde{\mathbf{v}}$  with mean  $\mathbf{m}_{\tilde{\mathbf{v}}}$  and covariance  $\mathbf{P}_{\tilde{\mathbf{v}}}$ , a symmetric set of sigma points can be defined using the eigenvalues  $\lambda_i$  and the corresponding eigenvectors  $\mathbf{q}_i$  of  $\mathbf{P}_{\tilde{\mathbf{v}}}$ , with  $i = 1, 2, 3$  [74, 125]:

$$\begin{aligned}\sigma_{\tilde{\mathbf{v}},i} &= \mathbf{m}_{\tilde{\mathbf{v}}} + \sqrt{N\lambda_i}\mathbf{q}_i \\ \sigma_{\tilde{\mathbf{v}},i+N} &= \mathbf{m}_{\tilde{\mathbf{v}}} - \sqrt{N\lambda_i}\mathbf{q}_i\end{aligned}\quad (2.87)$$

each associated with a weight  $w_i = 1/2N$ . Then, the sigma points  $\sigma_{\tilde{\mathbf{v}},i}$ ,  $\sigma_{\tilde{\mathbf{v}},i+N}$  are all transformed according to the true transformation function (Equation (2.3)):

$$\begin{aligned}\sigma_{\mathbf{r},i} &= \mathbf{L}(\sigma_{\tilde{\mathbf{v}},i})\sigma_{\tilde{\mathbf{v}},i} \\ \sigma_{\mathbf{r},i+N} &= \mathbf{L}(\sigma_{\tilde{\mathbf{v}},i+N})\sigma_{\tilde{\mathbf{v}},i+N}\end{aligned}\quad (2.88)$$

Finally, the predicted mean and covariance are computed as

$$\begin{aligned}\mathbf{m}_{\mathbf{r}} &\approx \frac{1}{2N} \sum_{i=1}^{2N} \sigma_{\mathbf{r},i} \\ \mathbf{P}_{\mathbf{r}} &\approx \sum_{i=1}^{2N} w_i (\sigma_{\mathbf{r},i} - \mathbf{m}_{\mathbf{r}})(\sigma_{\mathbf{r},i} - \mathbf{m}_{\mathbf{r}})^{\top}\end{aligned}\quad (2.89)$$

Analogously, the exact same process could be repeated starting from the Cartesian mean  $\mathbf{m}_{\mathbf{r}}$  and covariance  $\mathbf{P}_{\mathbf{r}}$  to obtain the approximate KS mean  $\mathbf{m}_{\tilde{\mathbf{v}}}$  and covariance  $\mathbf{P}_{\tilde{\mathbf{v}}}$ , with the transformation function that follows Equation (2.12). Similarly to the EKF case presented in Section 2.3.2.3, the conversion process can be extended to the mapping between the full Cartesian and KS states. The common conversion

procedures can be followed, without the need to compute their gradients and directly replacing the products  $L(\sigma_{\bar{v},i})\sigma_{\bar{v},i}$  with the fully non-linear mapping routines.

#### 2.4 KS VARIABLES FOR UNCERTAINTY PROPAGATION

The potential advantages of employing the KS formulation and regularization approaches are not limited to improvements on the performance and precision of the single simulations. In fact, uncertainties can be intuitively seen as continuum entities, whose evolution is subject to a flow (the dynamics): the more non-linear the flow, the more complex the uncertainty evolution becomes, even in the case of initially small distributions.

For this reason, this section discusses the theoretical concepts on the role of the dynamics in the propagation of the uncertainty, aiming at assessing whether the adoption of KS variables mitigates the effects of non-linearities. This dissertation is not meant to propose an alternative and full uncertainty propagation technique (starting from uncertainty sampling to the assessment of its effect). Rather, MC simulated data accounting for gravitational and relativistic effects of all the major solar system bodies are used, to highlight the principal differences between the Cartesian and the KS evolution of the same uncertainty cloud, with results presented in Section 2.5.4.

The principal link of this work with the existing literature is, of course, the use of regularized formulations for uncertainty propagation applications. A work that takes advantage of these concepts is the study by Roa and Peláez [122], who applied the KS formulation to the linear, State Transition Matrix (STM)-based uncertainty propagation in orbit determination problem. Given the intrinsic, time-dependent nature of observations and the required time-dependent uncertainty evolution, they concluded that no significant advantages appear when KS-propagating the partial derivatives composing the STM. More recently, Hernando-Ayuso et al [64] adopted the DROMO formulation for the propagation of uncertainties associated to nominal trajectories of Near Earth Objects. The work also proposes a hybrid Monte-Carlo technique to mitigate the influence of Earth close approaches in the evolution of the uncertainty, still relying, however, on a STM-based propagation. While the improvements brought by the DROMO formulation are evident, their potential is still somehow hidden by the STM concept itself, which linearizes the dynamics about the nominal trajectory for the uncertainty cloud. Based on this considerations, this work aims at exploring the exact role of the dynamics in the uncertainty evolution, by studying in fine detail the effects of the (conceptually) simplest regularized formulation, i.e. KS variables.

### 2.4.1 Stability of the KS equations of motion

A direct look at the KS equation of motion

$$\mathbf{u}'' = \frac{\epsilon}{2} \mathbf{u}$$

reveals its intrinsic stability for closed orbits: if  $\epsilon < 0$ , then the equation replicates an harmonic oscillator, whose solution is stable by definition in case of small perturbations, either on the initial conditions or if given as small perturbing accelerations. Nonetheless, the introduction of KS variables implies the adoption of the fictitious time  $s$ , instead of the physical time  $t$ . While this concept is exactly what brings stability to the equation of motion, it is also what hinders its applicability in the physical time domain.

Additionally, the different initial conditions characterizing the uncertainties result into different values of the orbital energy  $\epsilon$  for each uncertainty location. In the KS context, this would result on harmonic oscillators with different frequencies, depending on the initial condition. Nonetheless, the adaptive non-dimensionalization strategy proposed in Section 2.2.2 resolves this problem in its entirety: since the reference length is the initial semi-major axis and the reference gravitational parameter is made equal to 1, any initial condition is adjusted so that its orbital energy becomes equal to  $1/2$ .

The physical time can be tracked separately, as adjointed state element (its evolution follows Equation (2.1)), and then fitted upon need, so that the stability and common frequency features of the KS formulation can be exploited in full.

### 2.4.2 Fictitious time synchronism on flyby propagation and post-encounter scattering mitigation

*The Sundman regularization of Equation (2.1) is a differential form.*

The time regularization only introduces a new independent integration variable that scales, in the KS case, like the eccentric anomaly, without prescribing its initial value which is, accordingly, completely arbitrary. For this reason the initial fictitious time  $s_0$  can be chosen freely, and does not necessarily need to be the same across all samples. This fact can be exploited to enforce a particular synchronization, that is not based on physical time. For instance,  $s_0$  could be chosen so that the uncertainty shrinks to its minimum size, or so that similar events (i.e. the exit from the SOI of a common planet, although happening at different times for different uncertainty samples) are characterized by a common synchronization point.



## 2.5 APPLICATIONS

### 2.5.1 *Single simulation accuracy and performance*

The performances of the different simulation strategies in the single long term simulation are shown for the two near Earth asteroids Apophis and 2010RF<sub>12</sub>, in terms of time steps taken, CPU runtime and error evolution with respect to SPICE data. The initial state was taken from SPICE ephemerides data on the 1<sup>st</sup> January 1989 at noon and the simulations are carried out 100 years forward in time in the J2000 reference frame. Both the asteroids feature a steep close encounter with Earth within this time span. Due to the prevalent interplanetary nature of the motion the integration neglects  $J_2$  and drag effects. Solar radiation pressure is not considered, because the product of the refraction coefficient for asteroids and the area-to-mass ratio is negligible, and also difficult to estimate given the irregular asteroid shapes and variable material composition. General relativity effects are included, based on a manipulation [94] of the post-Newtonian model proposed by Will [160] as written by Seidelmann [129]. The simulations have been performed using an Intel® Xeon® CPU E5-4620 V4 running at 2.10 GHz.

The following nomenclature is used to identify the tested cases: "FIXED" or "SWITCH" to consider whether switching the integration center in case of flybys, "COWELL" or "KS" for the adopted dynamics formulation (either the Cartesian or the KS formulation of the restricted N-body problem), "EN" or "AUY" for the energy-based and AU-Year non-dimensionalization strategies respectively, "SUN" or "SSB" for the center of the interplanetary legs (either the Sun's center of mass or the Solar System's barycenter (SSB)).

All the presented runs have been performed using the Runge Kutta 4/5 and 7/8 numerical schemes. Dimensional simulations are not presented, as the maximum number of time steps, set to  $10^5$  in this work, is reached before reaching 5% of the time span, also before the flyby events. Given the problem typical magnitudes ( $10^8$  km for positions,  $10^1$  km/s for velocities) dimensional simulations could likely be made faster if the values for the absolute tolerances were made high enough. The preferred approach is however to use suitable non-dimensionalization procedures, to preserve the robustness that high order schemes, such as the Runge Kutta 7/8 used in this work, have for propagations with stringent tolerances. All the presented analyses have been performed using Matlab® and interfacing with Jet Propulsion Laboratory (JPL)'s ephemerides data through the SPICE toolkit [2] for retrieving the coordinates of the N bodies, considered as all the Solar System's planets plus the Moon.

The benchmark time steps and runtimes are reported in Table 2.1.

Table 2.1: Simulation performance benchmark, AU-Y-COWELL-FIXED, average of 200 runs.

CASE	FEATURES		RK45		RK78	
			STEPS	RUNTIME	STEPS	RUNTIME
Apophis	Relativity	No	63218	140.63 s	7246	34.25 s
	Center	SSB				
	Relativity	No	63118	149.09 s	7357	34.88 s
	Center	SUN				
Apophis	Relativity	Yes	62863	180.61 s	7187	39.74 s
	Center	SSB				
	Relativity	Yes	62805	193.42 s	7316	44.24 s
	Center	SUN				
2010RF <sub>12</sub>	Relativity	No	59679	134.21 s	6820	32.29 s
	Center	SSB				
	Relativity	No	59596	142.12 s	6934	33.72 s
	Center	SUN				
2010RF <sub>12</sub>	Relativity	Yes	59664	167.13 s	6820	37.59 s
	Center	SSB				
	Relativity	Yes	59625	179.14 s	6929	42.09 s
	Center	SUN				

The same cases are re-run making use of the energy non-dimensionalization, shown in Table 2.2. One can already see that, despite the two objects are near-Earth asteroids, the better tuning of the reference quantities already reduces the number of time steps taken and the total runtime by more or less 10% for the correspondent center and force benchmark cases. No significant step and runtime difference was found running the SWITCH case for the same formulation and non-dimensionalization strategy.

Finally, Table 2.3 shows the time steps and the runtimes obtained using the KS formulation of the energy non-dimensional variables and dynamically switching the Center of the reference frame to the flyby body whenever the propagated object enters a SOI. The proposed KS barycentric formulation is used when the default center of the interplanetary phase is the SSB, whereas the standard perturbed KS formulation is adopted whenever a flyby happens and also for default Sun-centric integrations in the interplanetary phase. The presented steps and runtimes correspond to the fibration point selected according to the optimization presented in Section 2.2.4.1, in the interval  $[0, \pi/2]$ . To this extent, Figures 2.3a and 2.3b show the time steps that are obtained for other values of the fibration parameter  $\varphi$ . The runtime is not reported, although it is proportional to the increase in time

Table 2.2: Simulation performances, EN-COWELL-FIXED, average of 200 runs.

CASE	FEATURES		RK45		RK78	
			STEPS	RUNTIME	STEPS	RUNTIME
Apophis	Relativity	No	58949	133.11 s	6551	28.82 s
	Center	SSB				
	Relativity	No	58952	140.90 s	6646	32.51 s
	Center	SUN				
2010RF <sub>12</sub>	Relativity	Yes	58579	166.27 s	6549	37.40 s
	Center	SSB				
	Relativity	Yes	58519	174.61 s	6637	40.85 s
	Center	SUN				
2010RF <sub>12</sub>	Relativity	No	55770	128.99 s	6287	27.56 s
	Center	SSB				
	Relativity	No	55749	130.62 s	6404	31.96 s
	Center	SUN				
2010RF <sub>12</sub>	Relativity	Yes	55754	160.14 s	6282	36.04 s
	Center	SSB				
	Relativity	Yes	55727	165.68 s	6385	39.81 s
	Center	SUN				

steps since the dynamics formulation and implementation remains the same.

Simulations in KS coordinates without frame switch are not presented, because the numerical scheme led to the minimum step-size at the moment of close encounter, exceeding the maximum number of steps ( $10^6$ ) before the end of the integration span although matching the accuracy of the presented cases in the pre-flyby legs. In fact, the regularization concept introduced with the KS formulation makes the dynamics sensitive to flyby events, i.e. nearly-singular accelerations, since happening over a non-regularized center.

For all the presented cases the angle  $\varphi$  for the initial KS vector generation has been set according to the necessary optimality condition, searching for the best among the possible  $0 < \varphi < \pi/2$  candidates based on the averaged unperturbed KS state, every time the reference frame was switched. Note that the same condition is used for barycentric simulations, because the problem being weakly perturbed and heavily dominated by the Sun makes the difference between barycentric and Sun-centric coordinates small.

It can be clearly seen in Table 2.3 that the number of time steps taken drops of almost 40% with respect to the benchmark case, and the total runtime of about 30%, comparing the respective force and center cases. This 10% difference between runtime and time step improvements

can be explained by the switch events not computed at all in the previous cases. In any case, the improvement brought by the longer time steps that can be safely taken outbalances the advantages to the KS formulation even for cases where the event computation is not necessarily required, despite the extra function evaluations. Another 10% difference is added if the results obtained using the barycentric KS formulation are compared with the Sun-centric benchmark cases for the same force model.

Table 2.3: Simulation performances, EN-KS-SWITCH, average of 200 runs.

CASE	FEATURES	RK45		RK78		
		STEPS	RUNTIME	STEPS	RUNTIME	
Apophis	Relativity Center	No SSB	39760	102.68 s	4025	23.19 s
	Relativity Center	No SUN	39806	100.73 s	4423	25.22 s
	Relativity Center	Yes SSB	39502	123.03 s	4007	26.89 s
	Relativity Center	Yes SUN	39500	130.87 s	4407	31.60 s
2010RF <sub>12</sub>	Relativity Center	No SSB	37479	94.79 s	3765	21.88 s
	Relativity Center	No SUN	37449	95.94 s	4373	24.95 s
	Relativity Center	Yes SSB	37448	113.59 s	3782	25.34 s
	Relativity Center	Yes SUN	37445	123.49 s	4366	31.35 s

Figures 2.4a and 2.4b portrait the evolution of the position error  $|\Delta \mathbf{r}| = |\mathbf{r}_{\text{simulation}} - \mathbf{r}_{\text{ephemerides}}|$  with respect to JPL's ephemerides data [2] for the two different formulations, Cowell's method with the AU-year non-dimensionalization and KS formulation with energy non-dimensionalization and dynamic frame switch. The precision of the frame switch and the Energy non-dimensionalization was tested on the Cowell's method too, which is not shown because no visible difference with the results from the benchmark (AU-year, no switch) case was found. Similarly, the RK45 and RK78 numerical schemes are equivalent and non-distinguishable in terms of accuracy, although the latter always requires a lower computational effort. It can be clearly seen that the two respective force cases match, with or without accounting for general relativity effects. This already promotes the KS optimized formulation as the simulation method to be always preferred when compared to Cowell's, even for long term simulations

in the fully perturbed environment, as it runs significantly faster achieving the same precision for the correspondent force models. Furthermore, as it can also be seen considering the results presented in Table 2.3, the relativistic integration in KS variables is always faster than the Cowell's N-body integration, allowing for increased precision and reducing the required computational effort.

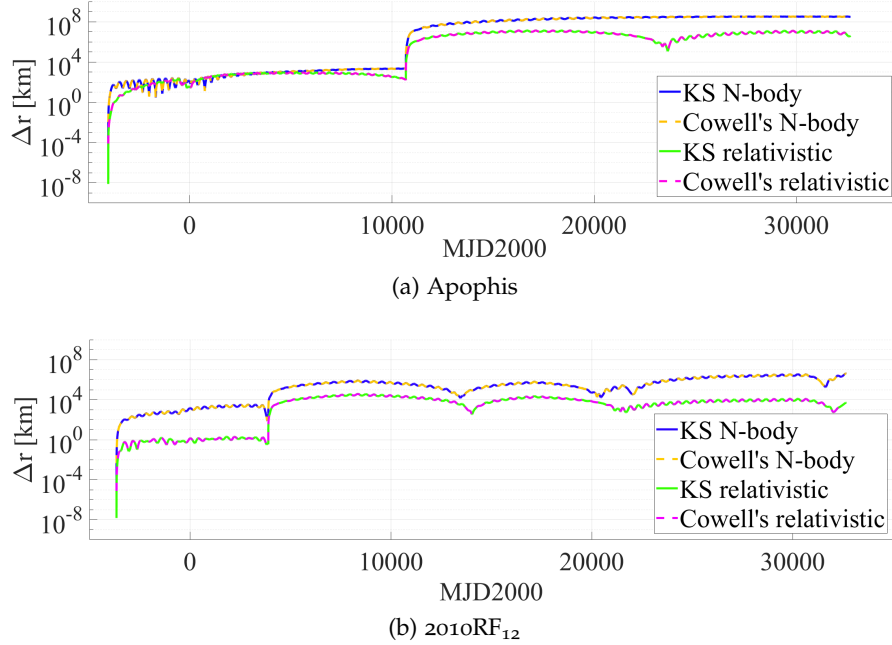
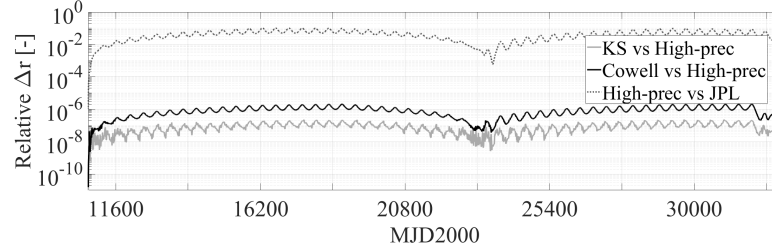


Figure 2.4: Position error evolution for the different force models and formulations, with respect to JPL's data.

Figures 2.5a and 2.5b still represent the evolution of the position error throughout the integration accounting for relativistic effects, despite showing its relative magnitude with respect to JPL's ephemerides data, and add a comparison with a higher accuracy integration performed on the EN-COWELL-SWITCH case but setting absolute and relative tolerances to  $10^{-14}$  with the RK78 scheme. The denser higher precision solution has been cubic spline-fitted to the already presented integration, particularly before the flyby events precision difference and fitting noise cannot be told apart. On the contrary, after the flyby it can be seen that the integration performed with the KS formulation remains nearly one order of magnitude closer to the higher precision solution. The relative error measure referred to JPL's ephemerides data for the two asteroids highlights again the flyby effect on long term simulations. For the particular case of Apophis, the steep flyby of Earth has the effect of amplifying by several orders of magnitude the error accumulated before the close encounter. Despite the physical model adopted in this work for the relativistic integration should match the one used by JPL (the user manual points to the model presented by Seidelmann [129] for the ephemerides generation), other

error sources are present, which could all explain the still low error accumulated before the flybys: JPL's ephemerides are generated with an Adams-Bashforth scheme and are then stored as coefficient of a polynomial interpolation, so that the user can request their value at specific epochs [2].



(a) Apophis

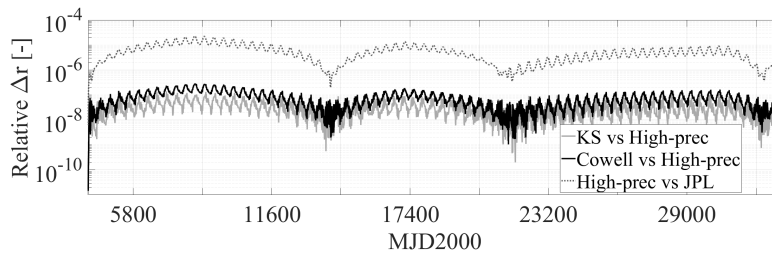
(b) 2010RF<sub>12</sub>

Figure 2.5: Relative position error evolution with higher precision relativistic simulation.

### 2.5.2 Solar Orbiter's upper stage of launcher MC PP analysis

The case of the upper stage of the launcher of Solar Orbiter (SolO) is presented, performing a MC simulation with samples generated from the uncertainty on the initial condition given as a covariance matrix, reported in Table 2.5 and with initial condition given in Table 2.4. Such data have been taken from [29], where this test case was studied first. Note that it refers to a mission profile later discarded, whose launch was originally scheduled for late 2018 and ultimately happened in February 2020. The presented results have been obtained with the same simulation routines used for the just discussed single simulation cases.

A total of 54114 samples has been generated for each case and simulated, based on the results of Wilson's expression [161] as done by Jehn [70] and Wallace [157], and following the implementation proposed in [29, 88, 89].

After the completion of the MC simulation, the impact probability of the disposal upper stage of launcher with Earth, Mars and Venus is computed by taking the ratio of the number of simulated impacts over all the generated samples. Tables 2.6 and 2.7 presents the results of

Table 2.4: Initial state of SolO's upper stage of launcher, J2000 reference frame [29].

$r_i$ [km]	$r_j$ [km]	$r_k$ [km]
132048839.02	63140185.88	27571915.38
$v_i$ [km/s]	$v_j$ [km/s]	$v_k$ [km/s]
-12.20	20.24	9.77
$t_0$ [MJD2000]		
6868.62		

Table 2.5: Elements of the covariance matrix of SolO's upper stage of launcher, J2000 reference frame [29].

POSITION COVARIANCE			
	$r_i$ [km]	$r_j$ [km]	$r_k$ [km]
$r_i$ [km]	$5.35139 \times 10^4$		
$r_j$ [km]	$5.40922 \times 10^4$	$1.35541 \times 10^5$	
$r_k$ [km]	$-2.56206 \times 10^4$	$4.50788 \times 10^3$	$1.72826 \times 10^5$
CROSS COVARIANCE			
	$r_i$ [km]	$r_j$ [km]	$r_k$ [km]
$v_i$ [km/s]	$2.48201 \times 10^{-1}$	$2.33655 \times 10^{-1}$	$-1.37013 \times 10^{-1}$
$v_j$ [km/s]	$2.74411 \times 10^{-1}$	$7.10015 \times 10^{-1}$	$5.01510 \times 10^{-2}$
$v_k$ [km/s]	$-1.20515 \times 10^{-1}$	$3.42692 \times 10^{-2}$	$8.33312 \times 10^{-1}$
VELOCITY COVARIANCE			
	$v_i$ [km/s]	$v_j$ [km/s]	$v_k$ [km/s]
$v_i$ [km/s]	$1.15577 \times 10^{-6}$		
$v_j$ [km/s]	$1.17908 \times 10^{-6}$	$3.72423 \times 10^{-6}$	
$v_k$ [km/s]	$-6.48488 \times 10^{-7}$	$3.07751 \times 10^{-7}$	$4.01929 \times 10^{-6}$

different MC simulations, parallelizing the simulations over 40 cores of the same kind of the one used for the single trajectory simulation. Particularly, the Cowell's case is propagated in one of the two fixed reference frames, SUN and SSB, and persistently checks whether an impact with Venus, Earth and Mars has happened or not at each time step. The KS case switches between either SSB or SUN and planetocentric frames, uses the energy non-dimensionalisation and checks for impacts only if entering any SOI. Also the same numerical scheme, Runge-Kutta 7/8, is used in both cases and for all the samples. In particular, Table 2.6 highlights the integration steps required by the different setups to detect the impact characterizing the barycenter of the sampled uncertainty cloud. The regularization benefits become particularly visible in this case: impacts can be detected almost twenty times faster, because of the removed mathematical singularity expe-

rienced by the Cartesian formulation when approaching any of the considered attractors, and more in general close approaches can be handled by the KS formulation with much larger time steps than what Cowell’s method does. The small differences between the barycentric and the Sun-centric results of the barycenter simulation may be due to the particular configuration of the selected case. Table 2.7 focuses

Table 2.6: Simulation outcome of the uncertainty barycenter of SolO’s upper stage of launcher.

CASE	BARYCENTER	
	RESULT	STEPS
AUY-COWELL-FIXED-SSB	Impact	1872
AUY-COWELL-FIXED-SUN	Impact	1889
EN-COWELL-SWITCH-SSB	Impact	1823
EN-COWELL-SWITCH-SUN	Impact	1798
EN-KS-SWITCH-SSB	Impact	88
EN-KS-SWITCH-SUN	Impact	84

instead on whole MC outcome. The total runtime is almost halved, achieving a reduction of more than 46%. As the number of time steps taken by the barycenter of the generated cloud (Table 2.6) tells, this, in turn, provides the observed performance enhancement with respect to the single simulation cases presented in Table 2.3: estimating impact probabilities requires analyzing what happens close to the encountered bodies, which is also the main advantage of the KS formulation, as it embeds an adaptive scaling of the physical time for different distances from the current primary attractor. Note finally that the

Table 2.7: PP analysis of SolO’s upper stage of launcher [29]

CASE	ESTIMATED	RUNTIME
	IMPACT PROBABILITY	
AUY-COWELL-FIXED-SSB	4.0211%	26.66 hours
AUY-COWELL-FIXED-SUN	4.0211%	26.45 hours
EN-COWELL-SWITCH-SSB	4.0248%	27.65 hours
EN-COWELL-SWITCH-SUN	4.0192%	28.75 hours
EN-KS-SWITCH-SSB	4.0192%	14.23 hours
EN-KS-SWITCH-SUN	4.0156%	13.22 hours

estimated impact probability remains basically unchanged, the slight difference might be due to both few borderline cases where the time step that would actually be within the impact region is skipped by the KS integration, and also because of the slightly different samples generated from the initial same covariance matrix (Table 2.5).



### 2.5.3 Covariance transformation

The proposed application studies the covariance transformation for the case of Solar Orbiter [43], using the same mean value and covariance matrix of Section 2.5.2 and assuming a multivariate normal distribution. To perform the analysis in relative terms, the mean value and the covariance have been normalized using the energy-based non-dimensionalization, computing the reference quantities on the mean value and using the Sun's gravitational parameter.

As a first, visual analysis, one million uncertainty samples are generated, and all converted from the Cartesian to the KS formulation using the optimal<sup>2</sup> fibration parameter of the mean value. Figure 2.6 shows the resulting distribution for the first two coordinates (i.e.  $x$  and  $y$  for the Cartesian samples), superposing both the Cartesian and the KS clouds. Visually, the KS distribution shrinks and rotates around the mean value although it seems to retain the multivariate normal distribution nature. Intuitively, this fact can be explained by the conformality property of the KS-Cartesian mapping [13]: relative angles are preserved, whereas distances, in general, are not. To confirm this observation, Figure 2.7 shows an estimation and superposition of the probability density function around the mean. In particular, for both the Cartesian and the KS cases, 1000 evenly spaced intervals have been created, from the minimum to the maximum values sampled, for each coordinate. Consequently, the number of samples falling in each box has been counted, and finally shown in Figure 2.7 as percentage of the maximum number of samples encountered in one box. Figure 2.7 seems to confirm the retained multivariate normal structure of the initial distribution.

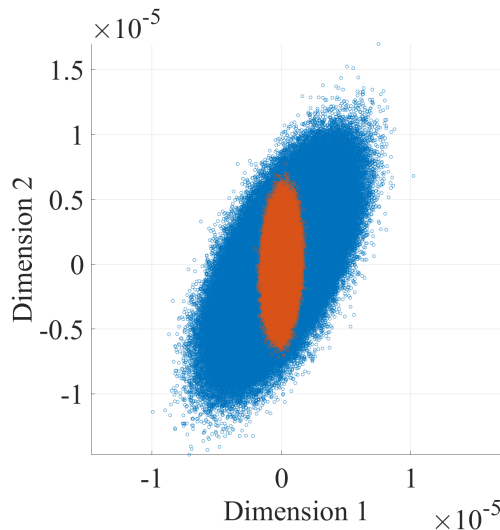


Figure 2.6: Superposition of Cartesian (blue) and KS (red) uncertainty distributions.

<sup>2</sup> The optimal fibration parameter selection method is presented in Section 2.2.4.

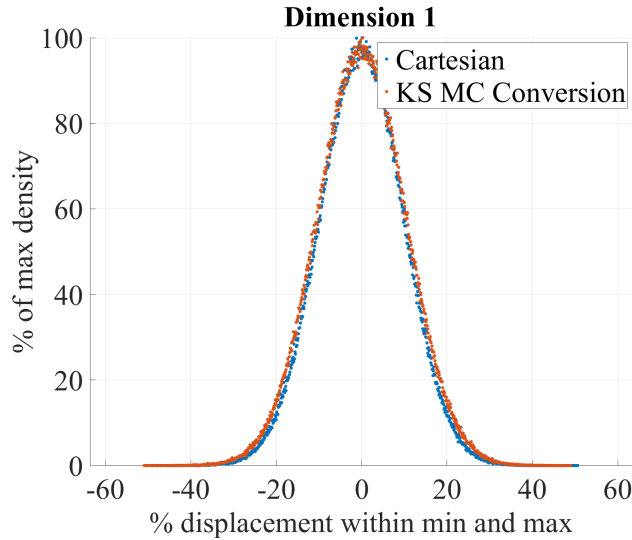


Figure 2.7: Superposition of Cartesian (blue) and KS (red) probability density estimations.

The following sections present the accuracy results of the proposed conversion technique, although not following the previous order. First, the MC approach is briefly introduced, followed by the UT-based approach, the EKF-based technique and finally a few considerations on the analytical method.

#### 2.5.3.1 MC transformation

In this section, the results are shown in terms of difference between the initial ( $\mathbf{m}_{x_0}$  and  $\mathbf{P}_{x_0}$ ) and the back-transformed Cartesian mean and covariance ( $\mathbf{m}_{x_b}$  and  $\mathbf{P}_{x_b}$ ), i.e. the results of a forward (Cartesian to KS) conversion followed by the backward transformation using the same technique. In particular, the process follows:

1. one million samples have been generated using the Matlab<sup>®</sup> `mvnrnd.m` function;
2. all the samples are transformed to obtain a set of one million KS samples;
3. the resulting KS mean and covariance have been estimated using the definition of sample mean and sample covariance, respectively;
4. based on Figures 2.6 and 2.7, a preserved multivariate normal distribution is assumed, and the newly computed KS mean and covariance are used again with the Matlab<sup>®</sup> `mvnrnd.m` function to generate a new set of one million KS samples;
5. the new KS samples are transformed back to the Cartesian formulation;

6. the newly back-transformed Cartesian samples are used again with the definitions of sample mean and sample covariance to obtain  $\mathbf{m}_{x_b}$  and  $\mathbf{P}_{x_b}$ .

The relative errors accumulated by the conversion amount to

$$\frac{|\mathbf{m}_{x_b} - \mathbf{m}_{x_0}|_2}{|\mathbf{m}_{x_0}|_2} = 7.9503 \times 10^{-8}$$

$$\frac{|\mathbf{P}_{x_b} - \mathbf{P}_{x_0}|_2}{|\mathbf{P}_{x_0}|_2} = 5.2877 \times 10^{-3}$$

where  $|\cdot|_2$  denotes the Euclidean matrix norm.

The MC technique shows reliable results and, by definition, its accuracy is expected to increase by increasing the number of samples. A more detailed discussion of how the number of samples affects the accuracy of the MC method is given in Section 2.5.3.2.

### 2.5.3.2 Unscented Transform

The unscented transform is tested as the difference between the UT-transformed and the MC-transformed KS mean and covariance matrices, exploiting the validation of the MC-based technique presented in Section 2.5.3.1. In particular, Figure 2.8 shows the error, measured again as Euclidean matrix norm, made between the MC-based conversion and the UT-based technique, for all the covariance sub-parts (position, velocity, cross). Interestingly, all the errors decrease with increasing number of samples: this means that the UT-based technique is far more precise than the MC-based one, and can be used as *the* transformation technique for the KS-Cartesian mapping. A possible explanation relies on the core idea which the UT method is built upon: the UT applies the exact non-linear transformation to an approximating probability distribution. In other words, Figure 2.8 proves that the preserved multivariate normal structure is not an assumption at all, but corresponds to the true transformed probability distribution shape.

Following the just presented result, a back-to-back conversion is proposed, similar to the one proposed in Section 2.5.3.1 but performed with the UT-based technique instead of MC, resulting in

$$\frac{|\mathbf{m}_{x_b} - \mathbf{m}_{x_0}|_2}{|\mathbf{m}_{x_0}|_2} = 1.2166 \times 10^{-10}$$

$$\frac{|\mathbf{P}_{x_b} - \mathbf{P}_{x_0}|_2}{|\mathbf{P}_{x_0}|_2} = 5.2477 \times 10^{-11}$$

These results highlight once again the accuracy of both the UT-based approach and the claim of preserved multivariate normal structure, which is not far from machine precision. In fact, most of the accumulated error may be due to the numerical computation of eigenvalues and eigenvectors<sup>3</sup>.

<sup>3</sup> Computed with the Matlab<sup>®</sup> `eig.m`.

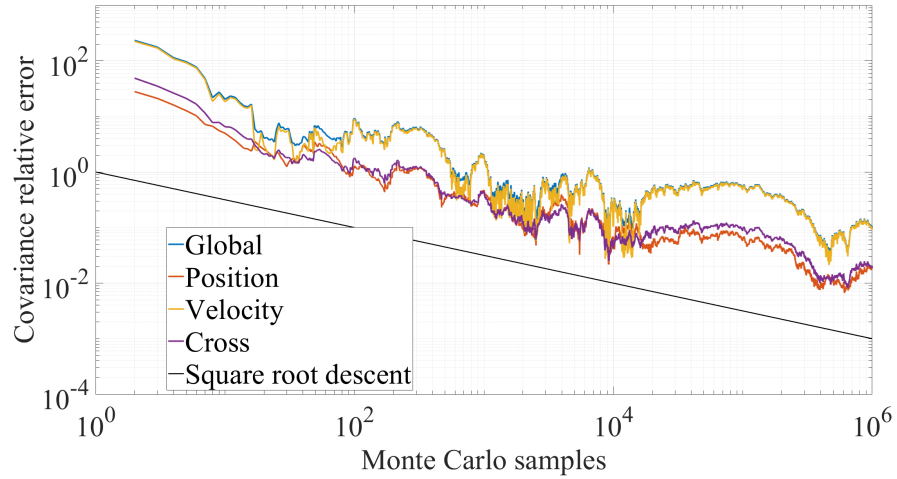


Figure 2.8: Accuracy of the UT-based technique against increasing number of samples for the MC-based conversion.

### 2.5.3.3 *Extended Kalman filter*

Given the results obtained in Section 2.5.3.2, the accuracy of the EKF-based approach is assessed against the UT-based technique, directly on the Cartesian to KS transformation. Since the method relies on the linearization of the transformation about the mean, the conversion is tested for varying magnitude of the covariance matrix, expecting better results in the case of smaller uncertainty distribution. In particular, the magnitude of the original covariance matrix (the same used for all test cases) has been first normalized, and then scaled by a factor  $f$ , evenly sampled on one million log-spaced points between  $10^{-9}$  and 1. Figure 2.9 shows that the position sub-part of the EKF-estimated covariance is accurately estimated, with decreasing precision for increasing covariance magnitude. On the contrary, the estimation worsens at any magnitude for cross and velocity terms, thus affecting the global covariance. This may be due to the stronger non-linearity introduced in the computation of velocity dependent terms, whose accuracy is inevitably reduced compared to the linearization of the simpler position transformation.

### 2.5.3.4 *Analytical solution*

Given the considerations made in Section 2.3.2.1, the accuracy of the analytical technique is assessed against the analytical conversion of the UT-based results for the position mean and covariance only, from KS to Cartesian, directly using the expressions presented in Equation (2.70).

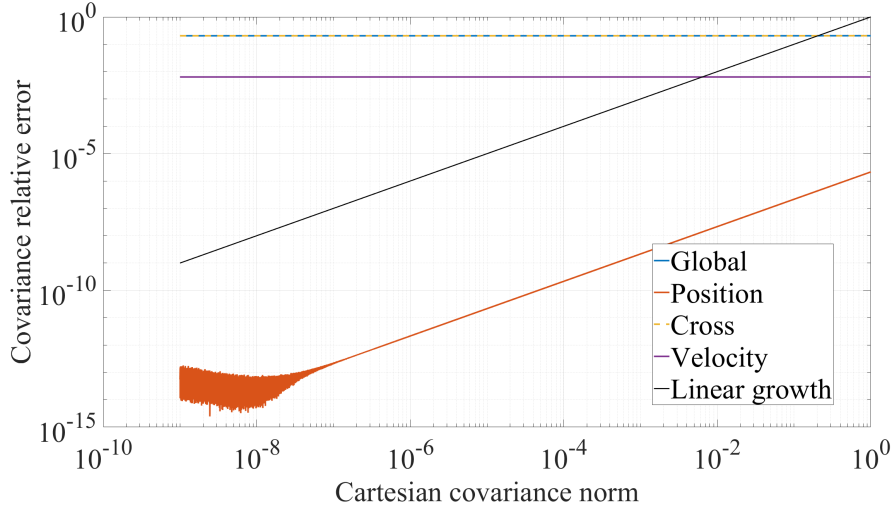


Figure 2.9: Accuracy of the EKF-based technique against UT-based conversion.

Referring to the initial mean values and autocorrelation ( $\mathbf{m}_0, \mathbf{R}_0$ ) and to the analytically transformed ( $\mathbf{m}_a, \mathbf{R}_a$ ), the relative errors result in

$$\frac{|\mathbf{m}_a - \mathbf{m}_0|_2}{|\mathbf{m}_0|_2} = 2.9122 \times 10^{-12}$$

$$\frac{|\mathbf{R}_a - \mathbf{R}_0|_2}{|\mathbf{R}_0|_2} = 4.5066 \times 10^{-12}$$

for lower values than the UT-based approach. However, the error between the covariance matrices  $\mathbf{P}_0$  and  $\mathbf{P}_a$  becomes

$$\frac{|\mathbf{P}_a - \mathbf{P}_0|_2}{|\mathbf{P}_0|_2} = 6.8414 \times 10^{-5}$$

The possible explanation for the worse performance on the covariance may be the high magnitude difference between the mean value and the dispersion. In fact, after normalization the reference euclidean norms for the position mean and covariance are

$$|\mathbf{m}_0|_2 = 1.2662$$

$$|\mathbf{R}_0|_2 = 1.3006 \times 10^{-11}$$

Scaling the covariance by a factor  $10^8$  and repeating the process leads to

$$\frac{|\mathbf{m}_a - \mathbf{m}_0|_2}{|\mathbf{m}_0|_2} = 2.9123 \times 10^{-4}$$

$$\frac{|\mathbf{R}_a - \mathbf{R}_0|_2}{|\mathbf{R}_0|_2} = 4.5081 \times 10^{-4}$$

$$\frac{|\mathbf{P}_a - \mathbf{P}_0|_2}{|\mathbf{P}_0|_2} = 1.0694 \times 10^{-3}$$

and, repeating the back-to-back conversion with the UT-based approach and considering only the position terms

$$\frac{|\mathbf{m}_{\text{UT}} - \mathbf{m}_0|_2}{|\mathbf{m}_0|_2} = 2.9123 \times 10^{-4}$$

$$\frac{|\mathbf{P}_{\text{UT}} - \mathbf{P}_0|_2}{|\mathbf{P}_0|_2} = 1.1255 \times 10^{-3}$$

These results show that most of the error made in the back-to-back conversion happens in the conversion from Cartesian to KS. While accurate, the UT-based approach still relies on the numerical computation of the covariance eigenvalues and eigenvectors, as well as the square rooting involved by the Cartesian to KS conversion, which remain possible error sources. Nonetheless, this application proves that the UT-based approach remains the only viable option. In fact, trying to solve the inverse problem using the analytical solution with a semi-definite programming [60] approach gives

$$\frac{|\mathbf{P}_{\alpha, \text{KS}} - \mathbf{P}_{\text{UT}, \text{KS}}|_2}{|\mathbf{P}_{\text{UT}, \text{KS}}|_2} = 3.0030 \times 10^2$$

proving the NP-hard nature of the Cartesian to KS analytical covariance conversion.

#### 2.5.4 *KS variables for uncertainty propagation*

The proposed application studies a MC simulation based on the asteroid Apophis, with initial condition taken from JPL ephemeris [2] on 01/01/1989 at midnight. 2000 normally distributed samples are generated using the following covariance matrix presented in Section 2.5.2 for the Solar Orbiter test case, with all elements reduced by a factor 100 to avoid an excessive spreading of the different samples. The simulations are run in non-dimensional KS coordinates, while the correspondent Cartesian and dimensional counterparts are obtained post-processing the simulation results. The study is split on two different phases: flyby and post-flyby<sup>4</sup>, according to the propagated trajectory of the nominal initial condition. In particular, the initial fictitious time  $s_0$  is set equal to 0 for all the samples at the beginning of each phase, i.e. entrance/exit to/from the SOI, following an event-synchronous approach, similarly to what introduced in Section 2.4.2. In all the upcoming plots, the color scale represents a relative comparison between the initial, per-phase, semi-major axis of all the samples, from the smallest (blue) to the largest (yellow).

<sup>4</sup> The initial uncertainty is small compared to the size of the initial conditions. Consequently, until the flyby event, no relevant phenomenon can be observed, the separation among the different samples remains nearly one million times smaller than the dynamics' scale.

2.5.4.1 *Flyby*

The first differences among the various simulated initial conditions start appearing in the flyby phase, whose dynamics magnifies any of the previously small variations by several orders of magnitude. Figure 2.10 shows the evolution of the semi-major axis when synchronism with respect to the physical time is in place. The sudden divergence of the different samples happen because of the strong Moon perturbation acting along the trajectory, whose distance is presented in Figure 2.11 together with Earth's, for one randomly sampled trajectory. Nonetheless, the slightly different times suggests that not all flybys happen exactly at the same physical time, in terms of entrance in Earth's SOI. Depending on the initial conditions also the difference in the initial hyperbolic semi-major axis is magnified.

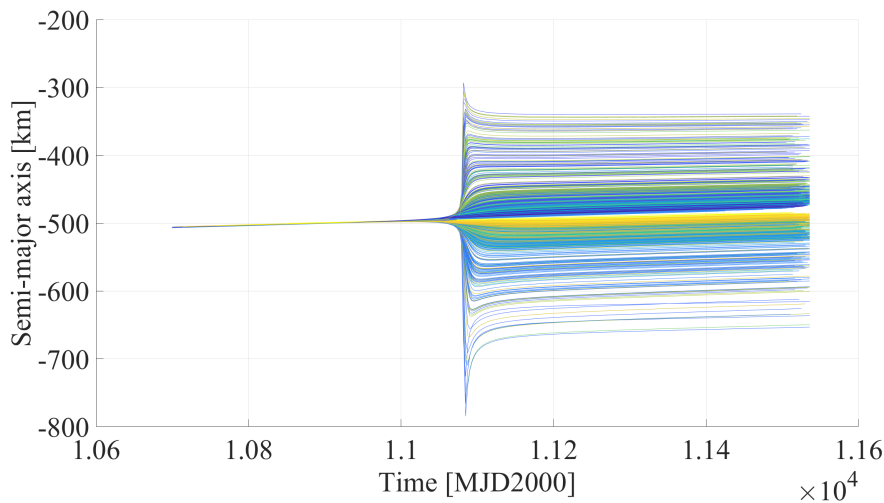


Figure 2.10: Physical time-synchronous evolution of the semi-major axis, for the flyby leg.

Figure 2.12 shows the evolution of the semi-major axis when synchronism with respect to the fictitious time is in place. The combined effect of the time regularization and of the per-sample non-dimensionalization has stretched the previously sudden variation in semi-major axis, providing a smoother change compared to Figure 2.10. Even if the regularization effects are most beneficial with respect to Earth, its effect has a significant impact in the more regular tracking of other perturbing effects (closely approaching the Moon in this case).

Figure 2.13 depicts the evolution of the Cartesian  $x$  coordinate with time. While this representation may be meaningless per se, comparing the coordinate evolution between the Cartesian and the KS representation contributes to highlighting the improvements brought by the regularization. Figure 2.14 shows instead the variation of the first hyperbolic KS coordinates, synchronized on the fictitious time. This evolution is significantly more regular than all the previously pre-

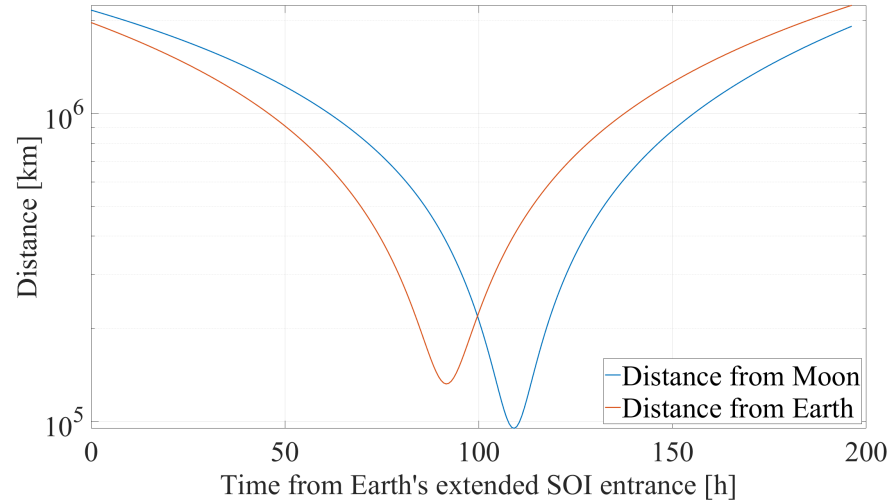


Figure 2.11: Random trajectory sample distance from the Moon (blue) and Earth (red).

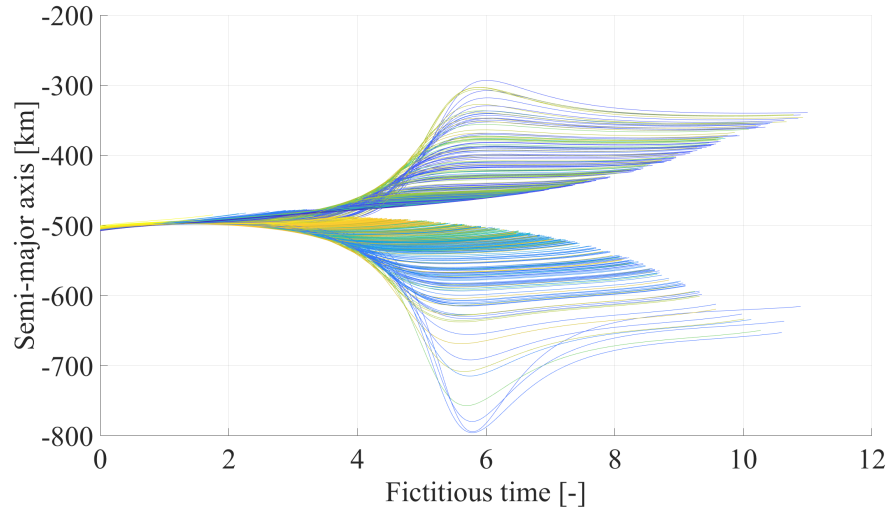


Figure 2.12: Fictitious time-synchronous evolution of the semi-major axis, for the flyby leg.

sented cases. All the samples follow a nearly exponential evolution, due to the quasi-linearity of the equations of motion. Despite exponential and non-periodic, the resulting uncertainty evolves smoothly on the fictitious time, removing the chaotic phenomena typically introduced by the close approach on the different uncertainty regions.

#### 2.5.4.2 *Post-flyby*

The challenge posed by post-flyby legs is all connected to the separation that the different uncertainty areas feature, despite all being the initial conditions of the post-flyby leg. The separation is then intro-



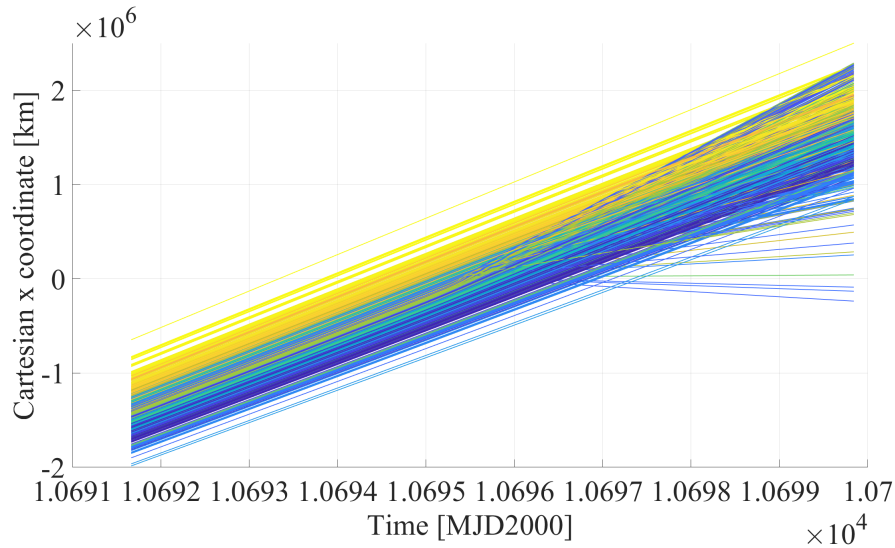


Figure 2.13: Physical time-synchronous evolution of the Cartesian  $x$  coordinate, for the flyby leg, in the Earth-Centered ECLIPJ2000[2] reference frame.

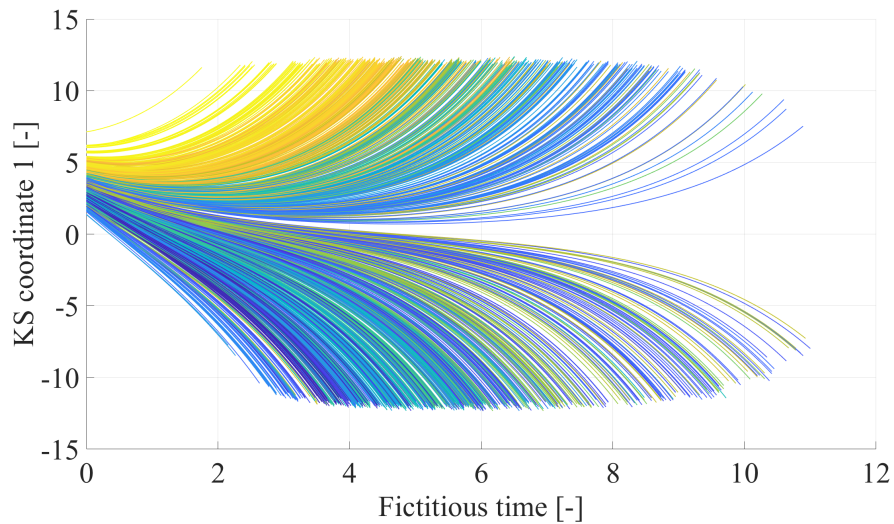


Figure 2.14: Fictitious time-synchronous evolution of the first KS coordinates, for the flyby leg.

duced not only in terms of different positions and velocities at the end of the flybys, but also as the different physical times that characterize the end of the close approach and the beginning of the post-encounter leg for the different uncertainty regions. Figure 2.15 shows the fictitious time span, on the  $x$  axis, with the initial semi-major axis, on the  $y$  axis. Since the propagation is stopped when the final physical time is reached, all the different samples end at different fictitious times. Nonetheless, the value of fictitious time where each propagation ends is highly predictable and strongly related to the semi-major axis: recalling the role of the orbital energy as the square of the natural

frequency in the KS dynamics, larger orbits (normally characterized by longer time scales) have a lower dimensional frequency, which results in a shorter fictitious time span once made non-dimensional. In other words, the energy-based non-dimensionalization intrinsically accelerates the simulation of larger orbits, while slowing down the smaller ones. Additionally, recalling that the fictitious time evolves according to the eccentric anomaly, the loci of common physical time can be estimated simply by solving Kepler's equation from the initial sample condition, shifting the value of the current fictitious time by the initial eccentric anomaly. The black lines of Figure 2.15 have been computed in the just discussed way, and prove that the fictitious time span can be reliably estimated using only the initial condition of the different samples/uncertainty regions. Building on this, Figure 2.16

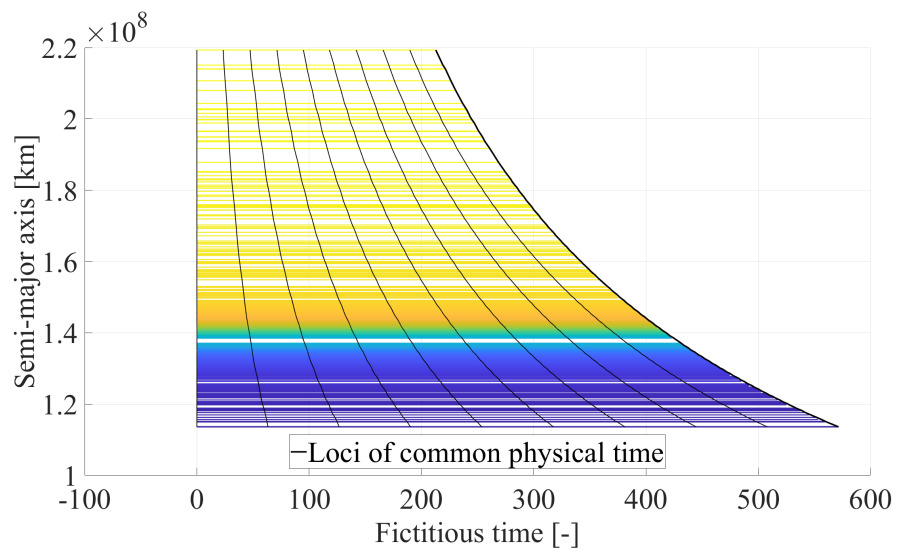


Figure 2.15: Different fictitious time spans, for the post-flyby leg.

shows the evolution of the physical time with respect to the fictitious time, for all the uncertainty samples. Samples with higher semi-major axis reach the final physical time (limit in the y direction) in a shorter fictitious time interval, as a consequence of the energy-based non-dimensionalization. The small oscillations that make time flow faster or slower are the actual effect of the regularization, with implicit slow-downs happening nearby the orbit's pericenter. Despite the apparent complexity introduced by the non-uniform non-dimensionalization, its effects remain highly predictable, as it is based only on the initial condition of the given uncertainty sample.

Figure 2.17 shows the evolution of the Cartesian  $x$  coordinate across the whole post-flyby phase. The initial different oscillation frequency and amplitude, clearly due to the different sizes of the orbits, result in an evolution that becomes more and more chaotic over time. Similarly to what presented for the flyby case, propagating upon the physical time results in a behavior of the whole uncertainty that seems discon-

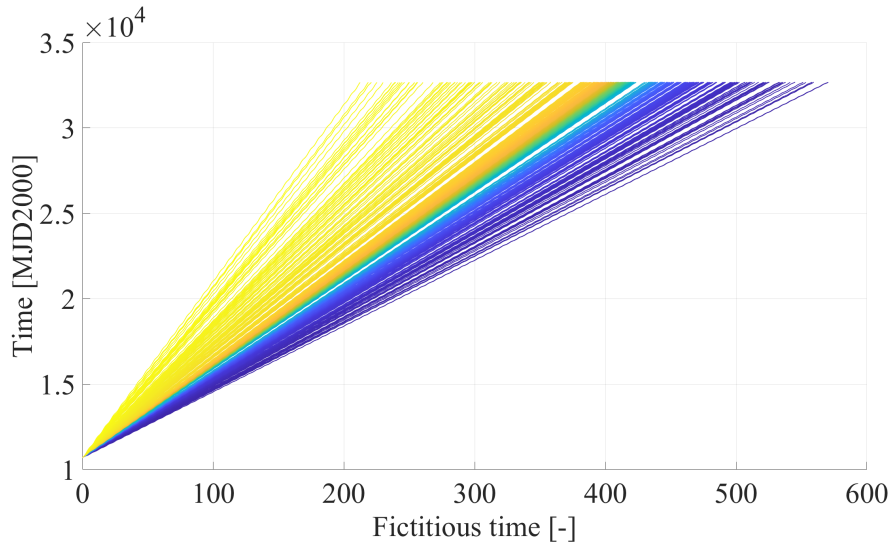


Figure 2.16: Fictitious time-synchronous evolution of the physical time, for the post-flyby leg.

tinuous. No sudden perturbation is experienced in this case, however the longer propagation span is all it takes to lose track of the initial original close trajectories.

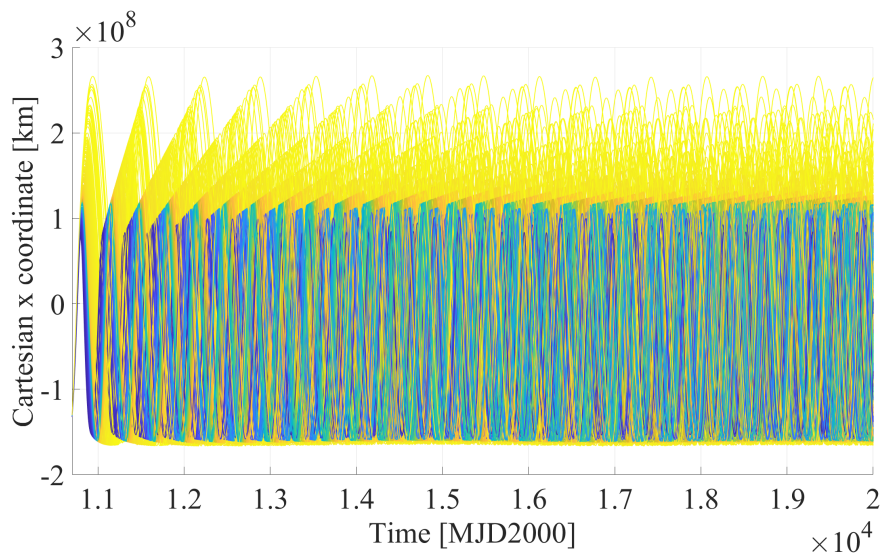


Figure 2.17: Physical time-synchronous evolution of the Cartesian  $x$  coordinate, for the post-flyby leg, in the SSB-Centered ECLIPJ2000[2] reference frame.

To overcome this continuity issue, the evolution of KS coordinates can be considered directly. Figure 2.18 and Figure 2.19 show, for different fictitious time spans, how the first of the four KS coordinates evolve in time. The KS dynamics highlights a nearly linear behavior for weakly perturbed two-body orbits, which also explains the observed regular and sinusoidal trend. This regularity appears even without

having introduced KS integrals of motion, and already demonstrates a smooth and continuum-like behavior of the uncertainty, when propagated and studied in non-dimensional KS coordinates. As it can be observed in Figure 2.18, the different fictitious time span result simply in the early or late truncation of the propagation, that follows the already discussed semi-major axis-related pattern. Figure 2.19 zooms instead on the initial propagation window and highlights the common oscillation frequency introduced by the KS transformation and the energy-based non-dimensionalization. Despite the normalization, the color scale highlights the retained dependence of the non-dimensionalization units on the initial condition, even in KS coordinates, confirmed by analyzing the evolution of the second KS coordinate in Figure 2.20.

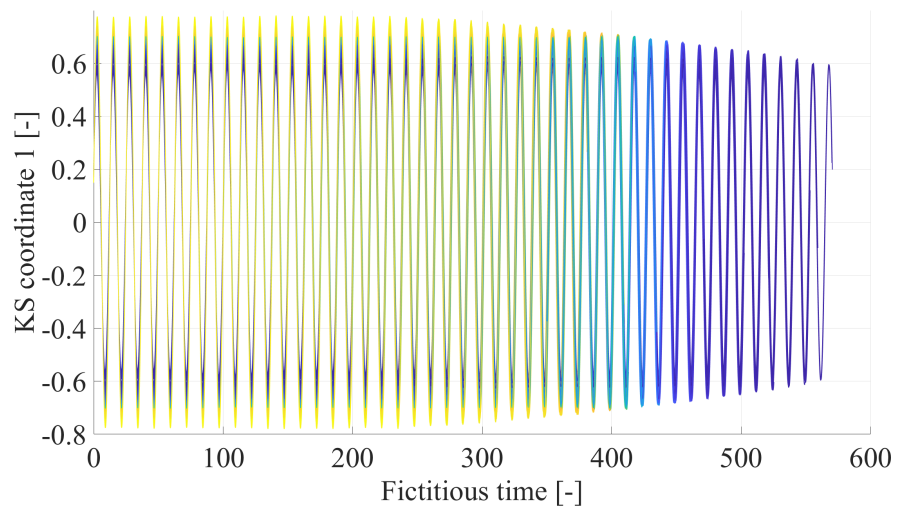


Figure 2.18: Fictitious time-synchronous evolution of one of the KS coordinates, for the post-flyby leg.

## 2.6 SUMMARY

### 2.6.1 KS variables for interplanetary simulations and PP/SDM

Figure 2.21 shows the step and runtime improvements brought by the different formulations and implementations, for the single simulations of the asteroids Apophis and 2010RF<sub>12</sub> in the relativistic case, with respect to the AUY-COWELL-FIXED case. The results for the Newtonian simulations that could be plotted from the values available in Tables 2.1, 2.2, and 2.3 are analogous, despite a lower improvement margin introduced by the barycentric simulations.

As dimensional propagations are not efficient in general, the adaptive energy-based non-dimensionalization improves the usual simulation techniques using Cartesian coordinates. For the transition to KS

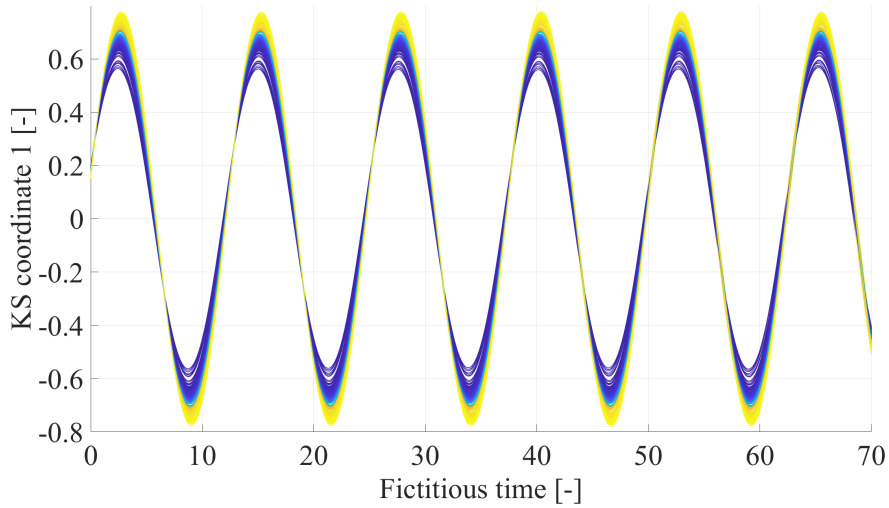


Figure 2.19: Initial window of the fictitious time-synchronous evolution of the first KS coordinates, for the post flyby leg.

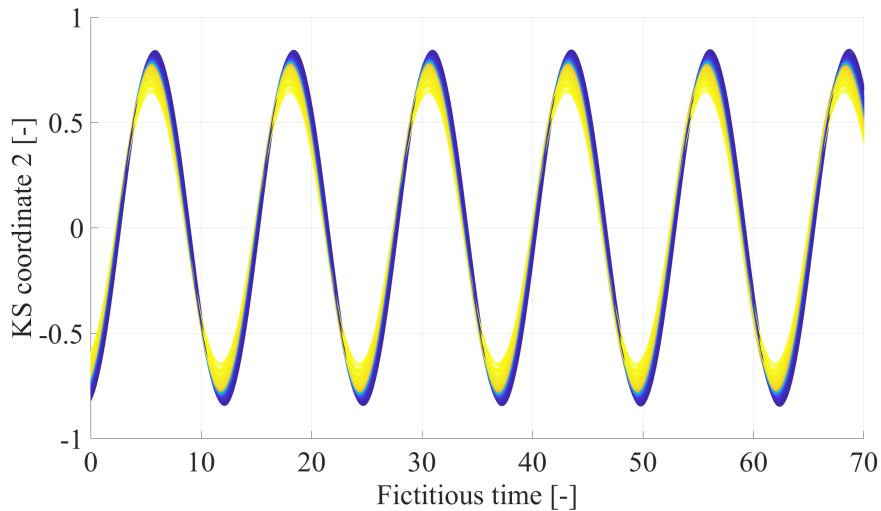


Figure 2.20: Initial window of the fictitious time-synchronous evolution of the second KS coordinates, for the post flyby leg.

coordinates, this choice of reference quantities becomes necessary, as well as switching the center of reference frame becomes mandatory for the simulation convergence.

In general, the KS approach improves the efficiency of numerical simulations as larger time steps can be taken without any precision loss. Despite the on-paper lost linearity property, the barycentric KS formulation exhibits the best performances overall, both in terms of time steps taken and total runtime required, especially for the relativistic case. In this context, the runtime reduces more than the number of steps, with respect to the KS Sun-centric case. The reduced time steps may be explained by an overall more regular dynamics being

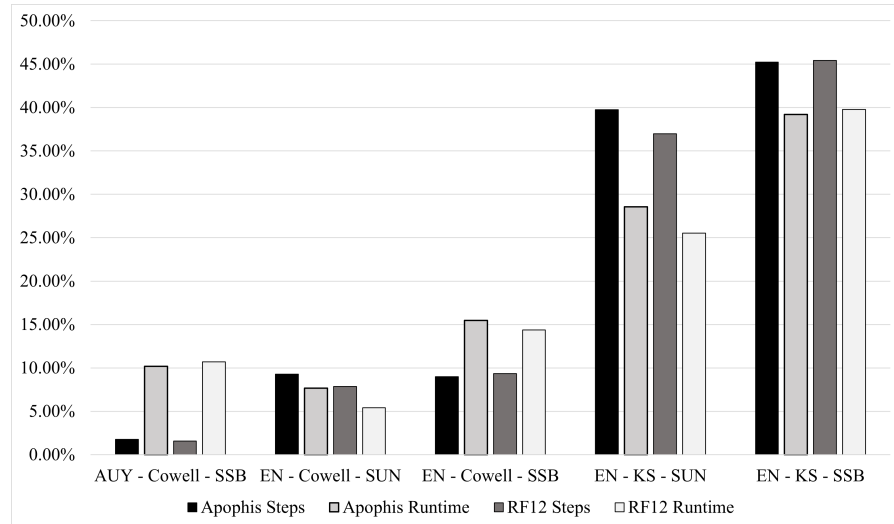


Figure 2.21: Speedup provided by the different formulations and implementations.

propagated, not affected by tidal terms. The further runtime reduction especially evident in relativistic simulations is experienced because evaluating the dynamics function itself is much more efficient in the barycentric case, also being the original frame where the dynamical model is given [129]. Despite not as much as in the relativistic case, an improvement is also observed in the Newtonian dynamics simulations, again because of the absence of tidal terms in the force model.

The energy-based non-dimensionalization has been exploited to obtain a closed form expression for the optimal pre-processing of the KS initial condition, for problems whose primary dynamics remains two-body. The degree of freedom left in the mapping to the four-dimensional KS space has been fixed maximizing the magnitude of the averaged (or initial, equivalently) minimum magnitude element in the unperturbed problem. Because of the way adaptive numerical integration schemes take the initial time steps, the ranges of possible initial conditions that lead to minimized integration steps could be accurately predicted. Furthermore, the improvement is not limited to the step and runtime reduction: as Figures 2.5a and 2.5b show, the simulations performed with the KS formulation remain nearly one order of magnitude more accurate than the correspondent Cartesian cases, for the same numerical scheme and absolute and relative integration tolerances.

The proposed KS formulation has been finally adopted to perform the PP analysis of Solar Orbiter's upper stage of launcher. An even larger relative reduction of the total runtime is obtained, with respect to the presented single simulation cases: impacts, therefore close approaches, are the objective of analysis and the KS formulation is exactly built to better handle small distances from the primary. This aspect was highlighted particularly by the much lower time steps taken by



the impacting barycenter, it can be seen in Table 2.6, more than twenty times lower than the usual Cowell's approach.

### 2.6.2 *KS - Cartesian uncertainty mapping*

Accurately mapping the uncertainty between the Cartesian and the KS realms is the first key step to enable orbital uncertainty propagation in KS coordinates. The raised dimension of the phase space may have been the main conceptual difficulty, nevertheless, expressing the fibration property of the KS space in matrix form resulted in the retrieval of a mapping between spaces with an equal number of degrees of freedom.

The accuracy of four different mapping techniques has been assessed through the uncertainty generated by Solar Orbiter's upper stage of launcher. Considering the transformation of the position part of the covariance, an analytical solution is proposed for the transformation KS to Cartesian of multivariate normal distributions. While solving the inverse of the same equation would, in principle, provide the backward transformation, the problem becomes NP-hard, as it involves quartic powers of the KS coordinates. As a result, convexity cannot be ensured, for numerical solvers getting blocked in local minima without converging to the actual solution. Anyway, the analytical solution has been proved to be the most accurate approach for the KS to Cartesian autocorrelation position mapping, although the uncertainty magnitude introduces numerical instability for the retrieval of the covariance, in the case of small uncertainties. Other techniques should still be used for both the inverse and the conversion of cross and velocity terms.

While a MC approach ensures a robust conversion, its accuracy is heavily dependent on the number of samples used in the transformation. Moreover, it does not exploit the properties of the KS transformation, making an overall robust, yet inefficient and inaccurate technique.

The UT-based approach provides the best overall results, for a mostly completely preserved accuracy in the case of small uncertainties (typical PP/SDM size). Additionally, the conversion of cross and velocity covariance terms happens with the same degree of accuracy of the position case. For the position conversion of multivariate normal distribution, it matches the accuracy of the KS to Cartesian transformation. The position accuracy slightly degrades in the inverse conversion (of  $10^{-4}$  in relative terms), yet it remains reliable enough to be considered the reference transformation technique in all cases.

Finally, an EKF-based technique is proposed, that relies on the linearization of the KS transformation about the mean value. While it provides satisfactory accuracy results in the Position case, the strong

non-linearity present in the KS velocity expression makes its global performance much worse compared to the UT case.

### 2.6.3 *KS variables and continuity of the propagated uncertainty*

The choice of independent variable to rely upon plays a key role in the statistical behavior of the propagated uncertainty. From the regularity viewpoint, introducing the KS coordinates results in a remarkable improvement, so significant to make evident a continuum-like behavior even in the case of post-flyby scattered trajectories.

While it is true that the physical meaning of the fictitious time synchronization may not be simple to understand and interpret, this framework does provide mathematical stability and quasi-linearity to the propagated uncertainty. The general recommendation is to treat the physical time as an additional state variable, similarly to what the theory of relativity does when dealing with time as the fourth dimension of space, and separating the UP phase from its analysis. If considering off-sampled points, a simple fitting should already accurately capture its evolution, based on how tightly and regularly the evolution of the physical time is linked to the initial semi-major axis. Therefore, the physical time behavior of the uncertainty can always be reconstructed *a posteriori* on a two-step process, building first the fictitious-time synchronous Cartesian/Keplerian uncertainty, and then finding the loci of points of common physical time. The KS description would make a more convenient and efficient framework for both the propagation of uncertainty samples and the use of continuum-based descriptions. The proposed sample-based analysis shows that the uncertainty becomes more regular overall in KS coordinates for both flybys and interplanetary legs, which can be significantly ease the implementation of computationally efficient uncertainty quantification techniques, such as Machine Learning models and statistical continua.

Despite the method may already be suitable for dominantly two-body trajectories, it should also be tested on low-energy and three-body-like trajectories. While a regularization approach may still be used and should anyway ensure an improved stability of the propagated uncertainty, the new equations of motion may not be quasi-linear. Additionally, the proposed approach requires to split trajectories that undergo conceptual discontinuities into separate branches and only acts on the common legs. For instance, if any part of the uncertainty performs an additional flyby, this model remains suitable only up to the occurrence of that event. The newly created post-flyby branch should be studied separately.



# 3

---

## ENCOUNTER CHARACTERIZATION AND KEYHOLE MAPS

---

Ever since the first interplanetary exploration missions, the use of gravity assist maneuvers has been adopted as enabling feature to reach distant locations in the Solar System. Nonetheless, close encounters need to be studied not only as a mean to achieve otherwise unfeasible trajectories, but also to gain insight and awareness on possible side effects or natural phenomena. Be that the need to implement center-of-integration-switch in numerical simulators, or to compute impact probabilities with given celestial bodies, or to simply assess how a trajectory changes through a flyby, encounters still represent a bottleneck for the complete understanding of certain orbital phenomena, including the actual threshold to determine what a flyby is, and what it is not.

When taking the focus of the analysis on the encounter itself, PP/SDM and planetary defense tasks become basically the same study. An uncontrolled object is moving in space, subject only to the pure ballistic orbital dynamics, and its impact probability has to be assessed. Were it an asteroid or a human-crafted launcher upper stage, the effects it undergoes remain the same, all due to the major Solar System bodies.

Keeping the PP/SDM application of this dissertation at the center of the discussion, mission analysts need to study close encounters to assess whether a given trajectory profile complies with COSPAR [31] policies or not. Similarly, Boutonnet and Rocchi [17] proposed an analysis of Earth-impacting trajectories stemming from SDM requirements [87]. In particular, many of the collisions were found to be out of what is nominally considered a colliding trajectory, i.e. out of a nominal orbital resonance with the target planet. Boutonnet and Rocchi [17] went deeper in the analysis, and concluded that a first, distant interaction triggered a new trajectory profile that could not be initially predicted, which, in turn, led to subsequent collisions. This aspect motivates the research presented in this chapter, attempting to study shallow encounters of the type that Boutonnet and Rocchi describe. Even after decades of research on close encounters, a clear distinction between what can be considered a flyby and what can instead be neglected remains unclear. This fact has indeed led to a

further sub-classification for flybys, labeling with the term "shallow" all those interactions that happen at great distances and/or with low relative velocity magnitudes. The literature covering the extension of flyby theories to shallow encounters is very limited, with a conceptual "jump" from analytical, patched conics-based methods to the fully numerical simulation of the Circular Restricted Three Body Problem (CR3BP). On the contrary, the astrodynamics community has so far profused significant efforts in gaining a better understanding of deep encounters with the existing theoretical tools, such as the b-plane [25, 148] and the Tisserand-Poincaré graph [23], key aspect to enable the realization of complex missions such as Solar Orbiter [43] and JUICE [44].

This chapter attempts to improve the existing theoretical framework for the characterization of close encounters, with the objective to propose techniques and tools that encompass both distant and deep interactions. First, a brief overview of the most commonly used theoretical frameworks and SOI concepts is given in Section 3.1, together with some works that propose non-traditional SOI definitions. Section 3.2 presents an approach that attempts to use perturbation methods to include third-body effects at the boundary of the SOI. Even though the proposed results actually dispute the use of perturbation techniques for this specific problem, this study eventually led to two of the publications listed in Section 1.5. Hence, the related research is included in this dissertation. A novel concept of SOI is proposed in Section 3.3, based on the maximum eigenvalue of the three-body dynamics' Jacobian. Finally, Section 3.4 combines the theoretical results of Section 3.3 with a simple surrogate model of the CR3BP dynamics, proposing the "Keyhole Map", a visualization tool for the detection of high impact probability regions in the phase space, even for off-resonance nominal trajectories.

### 3.1 ENCOUNTERS AND SPHERE OF INFLUENCE DEFINITIONS

The line to cross to assess whether a close encounter is happening or not is not clear per se, which inevitably becomes blurred in the case of shallow interactions. The commonly adopted frameworks do not fit perfectly well these borderline cases, for different reasons.

The patched conics approximation simplifies the flyby problem into two distinct two-body phases, and assumes a zero-radius sphere of influence for an instantaneous flyby effect, in the outer scale. In the inner part, instead, the trajectory is approximated by a hyperbolic solution of the two-body problem, and the trajectory deflection caused by the flyby is simply given by the rotation of the asymptotic velocity vectors [97]  $\mathbf{v}_{\infty}^{\pm}$ :

$$\mathbf{v}_{\infty}^{\pm} = \mathbf{v}^{\pm} - \mathbf{v}_{p1} \quad (3.1)$$

While in the secondary-centric frame the asymptotic velocities have equal magnitude, the rotation of  $\mathbf{v}_\infty^-$  into  $\mathbf{v}_\infty^+$  results, from the interplanetary viewpoint, in a net instantaneous  $\Delta\mathbf{v}$  supplied to the outer trajectory:

$$\Delta\mathbf{v} = \mathbf{v}^+ - \mathbf{v}^- \quad (3.2)$$

This approximation may seem reasonable from the interplanetary viewpoint, nonetheless it becomes inaccurate in some cases. Distant and continuous along-trajectory interactions are completely neglected, as well as the the finite time it takes to perform a complete flyby.

On the contrary, the CR<sub>3</sub>BP introduces a synodic reference frame whose  $x$  axis is attached to the line that connects the two primary bodies. The dynamical model considers both bodies at the same time and in full, without neglecting either body in specific trajectory phases. The non-dimensional CR<sub>3</sub>BP equations of motion, written in the rotating frame, are [97]:

$$\begin{aligned} \ddot{x} &= 2\dot{y} + x - \frac{1-\pi_2}{\sigma^3}(x+\pi_2) - \frac{\pi_2}{\psi^3}(x-1+\pi_2) \\ \ddot{y} &= -2\dot{x} + y - \frac{1-\pi_2}{\sigma^3}y - \frac{\pi_2}{\psi^3}y \\ \ddot{z} &= -\frac{1-\pi_2}{\sigma^3}z - \frac{\pi_2}{\psi^3}z \end{aligned} \quad (3.3)$$

with

$$\begin{aligned} \pi_2 &= \frac{\mu_2}{\mu_1 + \mu_2} \\ \psi &= \sqrt{(x-1+\pi_2)^2 + y^2 + z^2} \\ \sigma &= \sqrt{(x-1+\pi_2)^2 + y^2 + z^2} \end{aligned} \quad (3.4)$$

where  $\mu_1$  and  $\mu_2$  are the gravitational parameters of primary and secondary, respectively. The non-dimensionalization length is given by the distance between the primary and secondary body, and the reference gravitational parameter is equal to  $\mu_1 + \mu_2$ . The CR<sub>3</sub>BP has one single integral of motion, the Jacobi constant  $C_J$ :

$$C_J = n^2(x^2 + y^2) + 2\left(\frac{\mu_1}{r_1} + \frac{\mu_2}{r_2}\right) - (\dot{x}^2 + \dot{y}^2 + \dot{z}^2) \quad (3.5)$$

with  $r_1$  and  $r_2$  distances from body 1 and 2, respectively, and  $n$  the angular rate of the synodic frame. Far from the secondary body and with  $\mu_1 \gg \mu_2$ , the Jacobi Constant can be approximated by the Tisserand parameter  $\mathcal{T}$ :

$$\mathcal{T} = \frac{1}{a} + 2\sqrt{a(1-e^2)} \cos i \approx C_J \quad (3.6)$$

where  $a$  is the non-dimensional semi-major axis, expressed with reference length as the semi-major axis of body 2 with respect to body 1.

Despite more complete, in some phases of the trajectory the effects of one of the two bodies could be actually neglected. In addition, the more complex dynamical model prevents the use of analytical solutions, thereby restricting possible analysis to simulation and map-based approaches.

### 3.1.1 *Sphere of Influence and Hill's sphere*

The two most common definitions for the boundary that splits heliocentric and planetary domains are the SOI and the Hill's sphere. Denoting the semi-major axis of body 2 around body 1 with  $a_2$ , the former is identified by

$$r_{\text{SOI}} \approx a_2 \left( \frac{\mu_2}{\mu_1} \right)^{2/5} \quad (3.7)$$

and stems from equaling the perturbations of the two gravitational fields. For instance, in the Sun-Earth case, the SOI approximates the distance where the gravitational perturbation of the Sun on the Earth's gravitational field equals the perturbation of the Earth on the Sun's gravitational field [97]. On a different approach, the Hill's sphere radius is defined from the CR<sub>3</sub>BP, as the average distance from the planet's of the Lagrangian points L<sub>1</sub> and L<sub>2</sub>

$$r_{\text{SOI}} \approx a_2 \sqrt[3]{\frac{\mu_2}{3\mu_1}} \quad (3.8)$$

Also called libration points, Lagrangian points are equilibrium points in the CR<sub>3</sub>BP, i.e. local maxima of the CR<sub>3</sub>BP potential. Five of this points exist, and can be found as the points where the centrifugal acceleration of the synodic frame balances the combined gravitational field of both bodies [97].

### 3.1.2 *Numerical investigations*

The dynamics' Jacobian is traditionally linked to the step size control, for numerical simulations. In particular, the maximum eigenvalue influences the stability of the numerical scheme [118]. Contrarily to predictor-corrector integrators (e.g. the Runge-Kutta family), other adaptive schemes use the knowledge on the dynamics Jacobian to minimize the truncation error at each step. In the orbital dynamics case, the work of Debatin et al. [33] used an analytical approximation of the maximum Jacobian eigenvalue of the N-body dynamics, building a fast integration algorithm with step size control. Debatin et al. approximated the square of the maximum Jacobian eigenvalue

$\lambda_{\max}^2$  as the sum of the squares of all the separate two-body Jacobian eigenvalues:

$$\lambda_{\max}^2 \approx \sum_{i=1}^N \lambda_i^2 \quad (3.9)$$

with the subscript  $i$  denoting the  $i$ -th body. This approximation becomes particularly reliable far from the boundaries of any SOI/Hill's sphere, since in these regions either the Sun or the planet flown by heavily dominates the dynamics.

The later work of Romano [125] used a similar approximation approach to implement a flyby detection criterion. In particular, if the ratio between the eigenvalues of a given planet and the Sun grows above a user-specified tolerance, then a flyby event is detected:

$$\frac{\lambda_i}{\lambda_{\text{Sun}}} \geq \text{tol} \quad (3.10)$$

Romano also showed that this criterion encompasses the usually defined SOI/Hill's sphere: in the case of threshold set equal to 1, the critical ratio is triggered at greater distances from the secondary body, compared to SOI and Hill's sphere. Figure 3.1 provides an example with the evolution of the eigenvalue ratio for the 2010RF<sub>12</sub> in the Sun-Earth case. The eigenvalue ratio equals 1 at greater distances from Earth (red lines) compared to the SOI case (black dashed lines), resulting in a wider definition of sphere of influence and, therefore, of encounter.

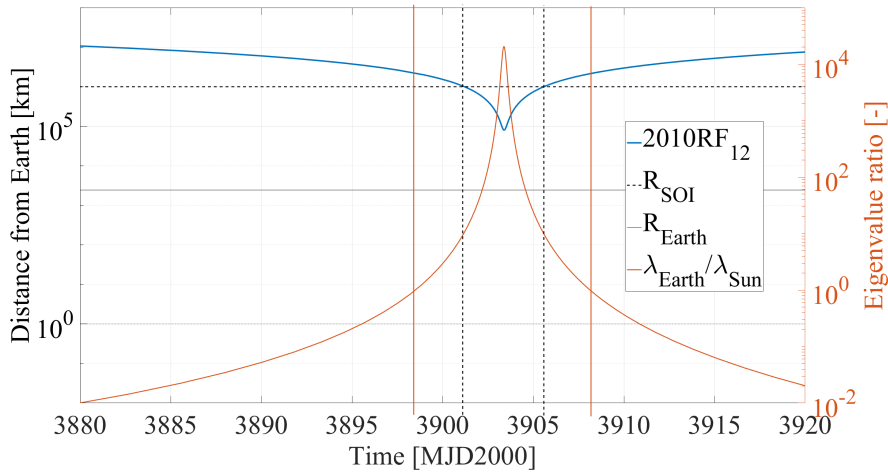


Figure 3.1: Distance from Earth and eigenvalue ratio evolution for the asteroid 2010RF<sub>12</sub> the Sun-Earth case.

Most recently, Cavallari et al [26] studied the case of Earth's SOI searching for a suitable size also accounting for velocity-related three-body effects on the dynamics, proposing an optimization-based approach that results in different SOI radii depending on the initial

conditions, according to the local quality of the patched conics approximation when compared to the CR<sub>3</sub>BP. The definition of a radii database is also included, which can be used with interpolation techniques to generalize the results for other, non-tested initial states. Cavallari et al [26] find that the identified radii all fall within the optimal switch range proposed by Amato et al. [3]. Further details on [3] are given in Section 3.3.

### 3.2 ANALYTICAL METHODS FOR FLYBY PERTURBATIONS

Perturbation approaches rely on the orbital dynamics being a Hamiltonian system. In short, a given system is Hamiltonian if its dynamics is governed by Hamilton's equation:

$$\begin{aligned}\dot{\xi} &= \frac{\partial \mathcal{H}}{\partial \Xi} \\ \dot{\Xi} &= -\frac{\partial \mathcal{H}}{\partial \xi}\end{aligned}\tag{3.11}$$

where  $(\xi, \Xi)$  are a pair of coordinate and its conjugate momentum, respectively, and  $\mathcal{H}$  is a scalar function, called the *Hamiltonian*.

In the orbital dynamics case, the Hamiltonian is simply the total orbital energy per unit mass. The simplest example of Hamiltonian formulation of the orbital dynamics are Cartesian coordinates, where the components of the velocity vector are the conjugate momenta of each corresponding position coordinate. For the restricted two body problem:

$$\mathcal{H} = -\frac{\mu}{r} + \frac{1}{2}v^2$$

Despite its flexibility and its Hamiltonian structure, Cartesian coordinates do not use the integrals of motion that characterize the orbital dynamics. Similarly, whilst Keplerian elements do introduce integrals in the two-body problem, they are not a set of coordinates and conjugate momenta. Hence, a new set of orbital elements, named after Charles-Eugène Delaunay [14], has been developed. Delaunay elements are commonly identified by  $(l, g, h, L, G, H)$  and represent a set of action-angle canonical variables, including the integrals of motion of the orbital dynamics. For closed orbits:

- $(l, g, h)$  identify the mean anomaly  $M$ , the argument of the pericenter  $\omega$  and the right ascension of the ascending node  $\Omega$ , respectively;
- $L = \sqrt{\mu a}$  is an analog for the orbital energy, with  $a$  the orbit's semi-major axis;
- $G, H$  both stem from  $L$  and the eccentricity function  $\eta = \sqrt{1 - e^2}$ , with  $G$  being an analog for the angular momentum of the orbit, or its semilatus-rectum  $p$ , and  $H$  being its out-of-plane component that identifies the inclination of orbital plane  $i$ .

In perfect analogy with the elliptic case, Delaunay elements can also be defined for the hyperbolic case [45]:

- $L$  takes a different definition, since  $a < 0$  for hyperbolas, i.e.  $L = -\sqrt{\mu|a|} < 0$ . Consequently, also the definition of the mean motions  $n$  differs in sign, with  $n_{\text{elliptic}} = \mu^2/L^3 > 0$  and  $n_{\text{hyperbolic}} = -\mu^2/L^3 > 0$ ;
- $G, H$  are also affected by the different eccentricity and the eccentricity function  $\eta = \sqrt{e^2 - 1}$ .

Using Delaunay variables, the two-body Hamiltonian can be written as

$$\mathcal{H} = \pm \frac{\mu u^2}{2L^2} \quad (3.12)$$

and using Hamilton's equation

$$\frac{dl}{dt} = \frac{d\mathcal{H}}{dL} = \pm n \quad (3.13)$$

remains the only non-zero right hand side, as five integrals of motion appear, with  $\pm n$  the elliptic/hyperbolic mean motion.

The Hamiltonian formalism allows the description of perturbation problems by Lie transforms [14]. Provided that the perturbation force can be described as a series expansion, Lie transforms introduce a systematic analytical procedure, that replaces the actual integration of the dynamics' differential equation(s) with a set of canonical coordinate transformations. A brief reference to this technique and the related notation is given in Appendix B.

*With a little notation abuse, "+" and "-" represent elliptic and hyperbolic case, respectively.*

### 3.2.1 Third body perturbation

Considering the gravitational effects of a primary body 0 and a perturbing body  $k$  on a test particle, in a reference frame centered on the primary the Hamiltonian  $\mathcal{H} = \mathcal{H}_0 + \mathcal{H}_k$  takes the following expression:

$$\mathcal{H} = -\frac{\mu}{2a} - \mu_k \left( \frac{1}{|\mathbf{r} - \mathbf{r}_k|} - \frac{\mathbf{r} \cdot \mathbf{r}_k}{r_k^3} \right) \quad (3.14)$$

with  $\mathbf{r}$  and  $\mathbf{r}_k$  denoting the position vectors of test particle and of third body in the reference frame centered on the primary. The distances  $r$  and  $r_k$  are simply defined as the magnitudes of the respective position vectors.  $\mathcal{H}_k$  is the sum of two terms, the direct pull of the body  $k$  over the test particle, and a tidal term, arising from the action of the body  $k$  on the primary. The pull term can be re-written using the cosine law:

$$\mathcal{H} = -\frac{\mu}{2a} - \frac{\mu_k}{r_k} \left( \left( 1 + \frac{r^2}{r_k^2} - 2 \frac{r}{r_k} \cos \psi \right)^{-1/2} - \frac{r}{r_k} \cos \psi \right) \quad (3.15)$$

with the relative angle  $\psi$  between the position vectors  $\mathbf{r}$  and  $\mathbf{r}_k$  defined by

$$\cos \psi = \frac{\mathbf{r}_k \cdot \mathbf{r}}{r_k r} \quad (3.16)$$

Following Lidov and Kozai [82, 90], the first term of  $\mathcal{H}_k$  can be expanded in terms of the Legendre polynomials  $P_j(\cos \psi)$ :

$$\left(1 + \frac{r^2}{r_k^2} - 2\frac{r}{r_k} \cos \psi\right)^{-1/2} = \sum_{j=0}^{+\infty} \left(\frac{r}{r_k}\right)^j P_j(\cos \psi) \quad (3.17)$$

where, for instance,  $P_0(x) = 1$ ,  $P_1(x) = x$ ,  $P_2(x) = 1/2(3x^2 - 1)$ ,  $P_3(x) = 1/2(5x^3 - 3x)$ , and so on. The series converges if  $r < r_k$ , and few orders suffice to reach a good accuracy level if  $r \ll r_k$ . Many examples retaining terms up to the fourth order are available in the literature for the luni-solar perturbations on Earth orbits.

$P_0(\cos \psi) = 1$ ,  $P_1(\cos \psi)$  cancels with the tidal term and the faster flyby dynamics allows the approximation of  $r_k$  as constant. Thereby  $\mathcal{H}_k$  can be re-written as:

$$\mathcal{H}_k = -\frac{\mu_k}{r_k} \sum_{j=2}^{+\infty} \left(\frac{r}{r_k}\right)^j P_j(\cos \psi) \quad (3.18)$$

which, once re-arranged in a Taylor-like series expansion over  $\varepsilon = 1/r_k$ , becomes

$$\mathcal{H}_k = -\frac{\mu_k}{r_k} \sum_{j=2}^{+\infty} \frac{\varepsilon^j}{j!} r^j P_j(\cos \psi) \quad (3.19)$$

The just described expressions hold regardless the bound or unbound geometry of the unperturbed trajectory. The assumption  $\varepsilon = 1/r_k$  small, with respect to the primary dynamics, is a good approximation both in case of closed trajectories and at the boundaries of the SOI in the inner Solar System: for near-Earth orbits  $r_{\text{SOI}} \approx 0.006 a_{\text{Earth}}$ , whereas for near-Venus orbits  $r_{\text{SOI}} \approx 0.0055 a_{\text{Venus}}$ . Moreover, these values are the maximum possible in the respective flyby cases, within the defined SOI.

### 3.2.2 Arbitrary order analytical procedure

Having identified the small parameter  $\varepsilon = 1/r_k$  and written  $\mathcal{H}_k$  as a series expansion, an analytical solution by Lie transforms can be proposed. Modifying the notation, the complete Hamiltonian  $\mathcal{H}$  can be written as

$$\mathcal{H} = \sum_{j=0}^{+\infty} \frac{\varepsilon^j}{j!} \mathcal{H}_{j,0} \quad (3.20)$$

with

$$\begin{aligned} \mathcal{H}_{0,0} &= \mathcal{H}_0 = -\mu/(2a) \\ \mathcal{H}_{1,0} &= \mathcal{H}_k|_{j=1} \equiv 0 \\ \mathcal{H}_{j,0} &= \mathcal{H}_k|_{j>1} = -\frac{\mu_k}{r_k} j! r^j P_j(\cos \psi) \end{aligned} \quad (3.21)$$

*The typical time scale for flybys in the inner Solar System is in the order of hours or two-three days at most, against months or years for the planets' motion.*



The Lie derivative operator for the Kepler perturbed problem in Delaunay variable can be written as:

$$\mathcal{L}_0(-) = n \frac{\partial}{\partial l}(-) \quad (3.22)$$

and thus the homological equation (Equation (B.18)) becomes:

$$\mathcal{L}_0(\mathcal{W}_n) = n \frac{\partial \mathcal{W}_n}{\partial l} = \tilde{\mathcal{H}}_n - \mathcal{H}_{0,n} \quad (3.23)$$

### 3.2.3 Hori's approach to the main problem

The peculiar feature of the considered problem, which makes it different from the typical orbital applications, is that hyperbolic trajectories are non-periodic, with short-term, long-term and secular effects that cannot be distinguished. Hori [66] proposed an analytical solution for the  $J_2$  perturbation on hyperbolic trajectories, written in hyperbolic Delaunay variables. In particular, denoting with  $R$  and  $J_2$  Earth's radius and equatorial oblateness respectively, starting from the perturbed Hamiltonian written in the Earth-centered equatorial frame

$$\mathcal{H} = \mathcal{H}_{0,0} + \mathcal{H}_{1,0} = -\frac{\mu^2}{2L^2} - \frac{\mu R^2}{r} \frac{1}{r^2} \frac{1}{2} J_2 (3 \sin^2 i \sin^2(g+f) - 1) \quad (3.24)$$

the integration of the homological equation (Equation (3.23)) for the first order gives:

$$\mathcal{W}_1 = \frac{1}{n} \int (\mathcal{H}_{1,0} - \mathcal{H}_{0,1}) dl + Z(-, g, -, L, G, H) \quad (3.25)$$

with  $Z$  pure function of the kernel of the Lie derivative operator. A closed form expression of the integral can be found using the differential relation

$$dl = \frac{\eta^2}{(1 + e \cos f)^2} df \quad (3.26)$$

to change the integration variable from  $l$  to  $f$ . Since the system is non-periodic, the choice of  $Z(-, g, -, L, G, H)$  is not unique, and is necessarily associated with the integration constants of the simplified problem.

To this extent, Hori sets  $\mathcal{H}_{0,1} = 0$ , integrating the hyperbolic Kepler problem in prime variables, finding the kernel terms such that

$$(l, g, h, L, G, H)|_{t=t_0} = (l', g', h', L', G', H')|_{t=t_0} \quad (3.27)$$

$t_0$  must be found to have  $(l_{0,1}, g_{0,1}, h_{0,1}, L_{0,1}, G_{0,1}, H_{0,1})|_{t=t_0} = 0$ , and should be used to find  $Z(-, g, -, L, G, H)$ .

The dependence on  $h$  does not appear in  $Z(-, g, -, L, G, H)$  because also the original perturbed Hamiltonian is not function of  $h$ , whereas the dependence on  $l$  is confined to the term of  $\mathcal{W}_1$  not belonging to

the kernel of the Lie derivative operator. This term is therefore used to find  $t_0$  as

$$L_{0,1} = \{L; \mathcal{W}_1\} = -\frac{\partial \mathcal{W}_1}{\partial l} = -\frac{1}{n}(\mathcal{H}_{1,0} - \mathcal{H}_{0,1}) \quad (3.28)$$

which in Hori's case means  $\mathcal{H}_{1,0} = 0$ , because of the introduced simplification  $\mathcal{H}_{0,1} = 0$ , thus

$$t_0 = \pm\infty, \quad f(t_0) = f_{\infty}^{\pm} = \arccos\left(-\frac{1}{e}\right) \quad (3.29)$$

with  $t_0 = -\infty$  chosen in Hori's case.

Re-writing  $\mathcal{W}_1$  of Equation (3.25) as

$$\mathcal{W}_1 = \mathcal{W}_1^{(l)} + Z(-, g, -, L, G, H) \quad (3.30)$$

the next steps of the procedure take a similar form for the other first order correction terms, e.g.

$$G_{0,1} = \{G; \mathcal{W}_1\} = -\frac{\partial \mathcal{W}_1^{(l)}}{\partial g} - \frac{\partial Z}{\partial g} \quad (3.31)$$

At  $t = t_0$ ,  $G_{0,1} = 0$  is required, hence the  $g$ -dependent terms of  $Z$  are determined by integrating

$$\frac{\partial Z}{\partial g} = -\frac{\partial \mathcal{W}_1^{(l)}}{\partial g} \Big|_{t=t_0} \quad (3.32)$$

$t$  appears in  $\mathcal{H}_{1,0}$  and the derived expressions through trigonometry functions of  $g + f$ , which, once expanded, can be replaced by the eccentricity-dependent infinity expression for  $\cos(f_{\infty}^{\pm})$ . Integrating:

$$Z(-, g, -, L, G, H) = -\int \frac{\partial \mathcal{W}_1^{(l)}}{\partial g} \Big|_{t=t_0} dg + Z_1(-, -, -, L, G, H) \quad (3.33)$$

and updating  $\mathcal{W}_1$ :

$$\begin{aligned} \mathcal{W}_1 &= \mathcal{W}_1^{(l)} - \int \frac{\partial \mathcal{W}_1^{(l)}}{\partial g} \Big|_{t=t_0} dg + Z_1(-, -, -, L, G, H) \\ &= \mathcal{W}_1^{(l,g)} + Z_1(-, -, -, L, G, H) \end{aligned} \quad (3.34)$$

An analogous process is taken for  $(l_{0,1}, g_{0,1}, h_{0,1})|_{t=t_0} = 0$ , leading to a unique definition of all the terms of  $Z(-, g, -, L, G, H)$ , e.g.:

$$l_{0,1} = \{l; \mathcal{W}_1\} = \frac{\partial \mathcal{W}_1^{(l,g)}}{\partial L} + \frac{\partial Z_1}{\partial L} \quad (3.35)$$

integrating this time for  $Z_1$  setting always  $t = t_0$ , and so on until all the terms of  $Z$  are found. Regardless the order the remaining Delaunay variables are processed with, the last step returns only a pure constant

term of  $Z$ , which Hori sets equal to 0. The explicit complete expression of  $\mathcal{W}_1$  can be found in [66].

A numerical propagation was made to test Hori's solution, for the following Keplerian elements at the entrance of Earth's SOI:

$$\begin{aligned} a &= -2459.37877 \text{ km}, & e &= 4, & i &= \frac{\pi}{10} \text{ rad}, \\ \Omega &= \frac{\pi}{3} \text{ rad}, & \omega &= \frac{2}{5}\pi \text{ rad}, & f &= -1.8203 \text{ rad}. \end{aligned} \quad (3.36)$$

The relative position errors with respect to the numerical integration of the perturbed motion are presented in Figure 3.2, in blue for the sole Keplerian propagation, in red for a direct implementation of the equations provided in [66], in dashed yellow for Hori's solution re-computed with the Lie-transform based analytical procedure proposed in Section 3.2.4.2, truncated to the first-order terms. The accuracy improvements obtained correspond to the prediction of the perturbation theory, since the new analytical solution difference from the numerical reference has become of order  $O(J_2^2)$ .

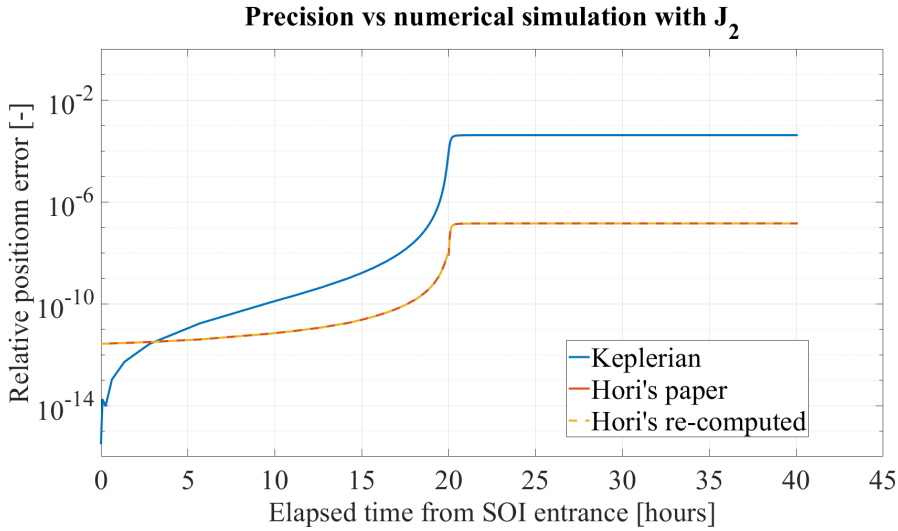


Figure 3.2: Errors of Hori's solution [66] for the  $J_2$  perturbation against the numerical simulations of the dynamics including Keplerian hyperbolic and  $J_2$  effects.

### 3.2.4 Hori's approach for the third body perturbation: boundary conditions

Hori's approach can be followed in the case of the third body perturbation as well, provided the integrability of all the involved expressions.

#### 3.2.4.1 Analytical setup

Since  $\mathcal{H}_{1,0} \equiv 0$ ,  $\mathcal{W}_1 \equiv 0$  as well, allowing the Lie transform method to start directly at the second order, implying also that  $y_{0,2} = \{y, \mathcal{W}_2\}$  with, in general,  $y = y(l, g, h, L, G, H)$ .

The second order term is defined as

$$\mathcal{H}_{2,0} = -\frac{\mu_k}{r_k} 2 r^2 \frac{1}{2} (3 \cos^2 \psi - 1) \quad (3.37)$$

and recalling  $\cos \psi = (\mathbf{r} \cdot \mathbf{r}_k) / (r r_k)$

$$\mathcal{H}_{2,0} = -\frac{\mu_k}{r_k} \left( 3 \frac{\mathbf{r} \cdot \mathbf{r}_k}{r_k^2} - r^2 \right) \quad (3.38)$$

Since  $\mathbf{r}_k$  is assumed constant, having  $\mathbf{r}$  only made of polynomial, trigonometric or exponential terms should suffice to ensure the integrability. This was not required for the integration of the first order term of the  $J_2$  perturbation, because of the cancellation of the denominator with the term  $R^2/r^2$ . For the third body case the presented description of  $\mathbf{r}$  on the true anomaly  $f$  must however be abandoned, adopting the hyperbolic anomaly  $u$  instead, particularly  $r = -a(e \cosh u - 1)$ . Writing  $\mathbf{r} = \{r_1, r_2, r_3\}^T$  as function of the Keplerian elements and  $u$  gives

$$\begin{aligned} r_1 &= a\eta \sinh u (\cos \Omega \sin \omega + \sin \Omega \cos i \cos \omega) \\ &\quad - a (\cos \Omega \cos \omega - \sin \Omega \cos i \sin \omega) (e - \cosh u) \\ r_2 &= a\eta \sinh u (\sin \Omega \sin \omega - \cos \Omega \cos i \cos \omega) \\ &\quad - a (\sin \Omega \cos \omega + \cos \Omega \cos i \sin \omega) (e - \cosh u) \\ r_3 &= -a \sin i \sin \omega (e - \cosh u) - a\eta \cos \omega \sin i \sinh u \end{aligned} \quad (3.39)$$

which ensures the integrability of the homological equation (Equation (3.23) since the differential relation

$$dl = (e \cosh u - 1) df \quad (3.40)$$

does not modify the structure of the expressions. Moreover, the relations between Delaunay and Keplerian elements are also known and can be used to change the integration variables in case of need. Given the already lengthy expressions of the position vector elements, they will be kept implicit in the following lines and have all been computed using Matlab<sup>®</sup> Symbolic Toolbox<sup>™</sup>.

#### 3.2.4.2 Hori-like procedure

The null first order terms lead directly to the expression of  $\mathcal{W}_2$  from the first homological equation solution

$$\begin{aligned} \mathcal{W}_2 &= \frac{1}{n} \int (\mathcal{H}_{2,0} - \mathcal{H}_{0,2}) dl + Z(-, g, h, L, G, H) \\ &= \mathcal{W}_2^{(1)} + Z(-, g, h, L, G, H) \end{aligned} \quad (3.41)$$

Differently from Hori's case, also the dependence on  $h$  appears in  $Z$  and the choice of  $\mathcal{H}_{0,2}$  is initially left implicit.

The motion is still hyperbolic, thereby requiring to find the terms of  $Z$  according to the initial condition presented in Equation (3.27). For  $L_{0,2}$  then

$$L_{0,2} = \{L; \mathcal{W}_2\} = -\frac{\partial \mathcal{W}_2}{\partial l} = -\frac{1}{n}(\mathcal{H}_{2,0} - \mathcal{H}_{0,2}) \quad (3.42)$$

In the third body case  $r \rightarrow +\infty$  for  $t \rightarrow \pm\infty$ , the only self-vanishing case of  $\mathcal{H}_{2,0}$  happens for  $\psi_{1,2,3,4}^*$  such that  $\cos^2 \psi^* = 1/3$ .  $\psi_{1,2,3,4}^*$  correspond to peculiar relative positions between the perturbing body and the test particle, not necessarily associated with a specific hyperbolic condition e.g. pericenter or infinity and varying case by case.

Following Hori's simplification criterion  $\mathcal{H}_{0,2} = 0$  or at most constant can be chosen, leading to the Keplerian problem in prime variables. In this case, from Equation (3.42) the initial condition  $t_0$  to have  $L_{0,2}|_{t=t_0} = 0$  should be chosen so that  $\mathbf{r}(t_0)$  satisfies  $\cos^2 \psi(t_0) = 1/3$ . In general, configurations where no points on the trajectory have such a feature may exist.

Alternatively, Equation (3.42) can be satisfied choosing  $\mathcal{H}_{0,2}$  as a non-null constant to impose a desired  $t_0$ , i.e.  $\mathcal{H}_{0,2} = \mathcal{H}_{2,0}|_{t=t_0}$ , since it would always lead to  $L_{0,2}|_{t=t_0} \equiv 0$  regardless the specific  $t_0$  chosen.

The terms of  $Z$  can then be found for any made choice of  $t_0$ , following Hori's approach for the second order terms. For  $G_{0,2}$ , similarly to Equations (3.31) to (3.34):

$$G_{0,2} = \{G; \mathcal{W}_2\} = -\frac{\partial \mathcal{W}_2^{(1)}}{\partial g} - \frac{\partial Z}{\partial g} \quad (3.43)$$

Requiring again  $G_{0,2} = 0$  brings

$$\frac{\partial Z}{\partial g} = -\frac{\partial \mathcal{W}_2^{(1)}}{\partial g} \Big|_{t=t_0} \quad (3.44)$$

and integrating:

$$Z(-, g, h, L, G, H) = -\int \frac{\partial \mathcal{W}_2^{(1)}}{\partial g} \Big|_{t=t_0} dg + Z_1(-, -, h, L, G, H) \quad (3.45)$$

Finally the updated  $\mathcal{W}_2$  becomes:

$$\begin{aligned} \mathcal{W}_2 &= \mathcal{W}_2^{(1)} - \int \frac{\partial \mathcal{W}_2^{(1)}}{\partial g} \Big|_{t=t_0} dg + Z_1(-, -, h, L, G, H) \\ &= \mathcal{W}_2^{(1,g)} + Z_1(-, -, h, L, G, H) \end{aligned} \quad (3.46)$$

with the next step, e.g. to have  $H_{0,2} = 0$ , initiated by

$$H_{0,2} = \{H; \mathcal{W}_2\} = -\frac{\partial \mathcal{W}_2^{(1,g)}}{\partial h} - \frac{\partial Z_1}{\partial h} \quad (3.47)$$

Proceeding once again according to the just presented steps, all the terms of the kernel function  $Z$  can be found. The final expression of  $\mathcal{W}_2$  has been obtained with the symbolic processor, not reported here for the sake of conciseness. To check the correctness of the derivation, exactly as what found by Hori the last step returned a pure constant term for  $Z$  was obtained, regardless the chosen  $t_0$ .

Writing the corrections in term of the prime variables the following expressions for the original Delaunay elements are obtained:

$$\xi = \xi' + \frac{\varepsilon^2}{2!} \xi_{0,2}(l', g', h', L', G', H'), \quad \xi \in \{l, g, h, L, G, H\} \quad (3.48)$$

and the evolution of the prime variables simply corresponds to the Keplerian unperturbed trajectory.

### 3.2.4.3 Proceeding to higher orders

The null first order terms lead to the following third order homological equation:

$$\begin{aligned} \mathcal{W}_3 &= \frac{1}{n} \int (\mathcal{H}_{3,0} - \mathcal{H}_{0,3}) dl + Z(-, g, h, L, G, H) \\ &= \mathcal{W}_3^{(1)} + Z(-, g, h, L, G, H) \end{aligned} \quad (3.49)$$

and, given the null first order terms, Deprit's triangle for the third order correction yields

$$y_{0,3} = \{y; \mathcal{W}_3\}, \quad y = y(l, g, h, L, G, H) \quad (3.50)$$

If  $\mathcal{H}_{3,0} = 0$ , then also  $\mathcal{H}_{0,3} = 0$  similarly to the first order case, thus  $\mathcal{W}_3 = Z(-, g, h, L, G, H)$  is only function of the Lie derivative kernel. Since  $\mathcal{W}_3^{(1)} \equiv 0$ , applying the first step of Hori's procedure gives

$$G_{0,3} = \{G; \mathcal{W}_3\} = -\frac{\partial Z}{\partial g} \quad (3.51)$$

which implies  $Z \neq Z(g)$  to have  $G_{0,3} = 0$  at the desired  $t_0$ , but also at any other  $t$  by consequence. Updating the terms of  $\mathcal{W}_3$  it can be seen that  $\mathcal{W}_3^{(1,g)} \equiv 0$  as well, thus by induction and for all the remaining steps it then follows that  $\mathcal{W}_3 \equiv 0$  to have  $y_{0,3}|_{t=t_0} = 0$ . Consequently,  $y_{0,3} \equiv 0 \forall t$ .

Otherwise, if  $\mathcal{H}_{3,0} \neq 0$  then the same already presented procedure would follow for all the terms of  $\mathcal{W}_3$ . The initial condition  $t_0$  determined for the second order term must be kept in the higher orders, thus writing the expansion up to the order 3

$$\begin{aligned} y &= y_{0,0}(l', g', h', L', G', H') + \frac{\varepsilon^2}{2!} y_{0,2}(l', g', h', L', G', H') \\ &\quad + \frac{\varepsilon^3}{3!} y_{0,3}(l', g', h', L', G', H') \end{aligned} \quad (3.52)$$

highlights that choosing a different  $t_0$  to find the kernel term of  $\mathcal{W}_3$  would imply  $y_{0,3} \neq 0$  at  $t_0$ , making the initial condition of Equation (3.27) invalid. The proposed procedure guarantees that choosing  $\mathcal{H}_{0,3} = \mathcal{H}_{3,0}|_{t=t_0}$  makes Equation (3.27) hold.

The fourth order terms do not vanish if  $\mathcal{H}_{4,0} = 0$ , because also terms like  $\{y_{0,2}; \mathcal{W}_2\}$  appear in  $\tilde{\mathcal{H}}_4$ . The full definition of the homological equation applies:

$$\begin{aligned} \mathcal{W}_4 &= \frac{1}{n} \int (\tilde{\mathcal{H}}_4 - \mathcal{H}_{0,4}) dl + Z(-, g, h, L, G, H) \\ &= \mathcal{W}_4^{(1)} + Z(-, g, h, L, G, H) \end{aligned} \quad (3.53)$$

and once again having  $L_{0,4}|_{t=t_0}$  requires  $\mathcal{H}_{0,4} = \tilde{\mathcal{H}}_4|_{t=t_0}$ . The terms of the kernel function  $Z$  are always found according to the already presented steps.

As a final remark, despite increasingly complex, the initial choice of using  $u$  instead of  $f$  for the definition of  $\mathbf{r}$  should ensure the integrability of all the expressions at any order, since fractional functions of the Delaunay elements appear at most for the definition of  $\cos i$  and  $\sin i$ .

#### 3.2.4.4 Numerical results

The presented analytical solution has been tested for a planetary flyby of Venus, modeling the effects of the Sun, within the planet's SOI. The entrance condition, in the Ecliptic J2000 reference frame centered on Venus is given in Table 3.1.

Table 3.1: Initial state of the test particle, ECLIPJ2000 reference frame centered on Venus [2].

$r_i$ [km]	$r_j$ [km]	$r_k$ [km]
-104551.25	-597237.74	-110314.51
$v_i$ [km/s]	$v_j$ [km/s]	$v_k$ [km/s]
3.25	17.76	3.67
$t_0$ [MJD2000]		
8119.84		

The precision of the proposed method is assessed comparing the evolution of the position errors of four different models against a relativistic simulation including all the Solar System's planets:

1. The two-body unperturbed Keplerian hyperbolic trajectory.
2. The numerical simulation including the complete Sun's perturbing potential.
3. The numerical simulation of the Legendre expansion of the Sun's perturbing potential truncated to the second order.

4. The presented analytical solution truncated to the second order corrections for retrieving the original variables.

Moreover, the presented test case features two points in the benchmark trajectory that satisfy  $\cos^2 \psi = 1/3$ , one before and one after the pericenter, as it can be visualized by the function  $\cos \psi(t)$  plotted in Figure 3.3, also highlighting the time scale of the considered dynamics.

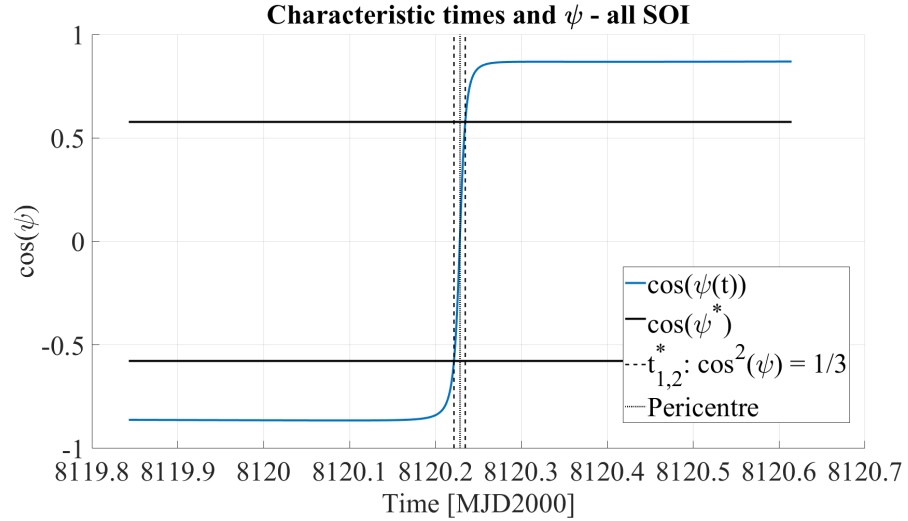


Figure 3.3: Evolution of  $\cos \psi(t)$  within Venus' SOI.

The coordinates of all the planets have been taken from JPL's ephemerides data, with the Matlab<sup>®</sup> version of the SPICE toolkit [2], and all the numerical simulations have been performed setting a relative tolerance of  $10^{-14}$ .

Figures 3.4 and 3.5 show the quality of the numerical integration for different orders of the Legendre polynomial expansion of the perturbing force (yellow, purple, green, cyan, crimson), as the position error against the complete relativistic model, and comparing it against the two-body model (blue), and the complete three body problem (red), including only the Sun's effects. As expected, the longer term integration accuracy increases with increasing order. A local maximum of the simulated position error with respect to the complete relativistic model is experienced nearby the pericenter for all orders, including the conditions  $\cos^2 \psi = 1/3$ . Such times correspond to the null second order term, the increased integration error may be due to the analytical cancellation of the lower order terms. The difference among orders four, five and six cannot be told apart on the plotted scale and considered integration time.

Despite the outlined analytical process for the higher orders, the presented numerical results only involve the terms up to the second order. An attempt was made to reach the fourth order, however the symbolic



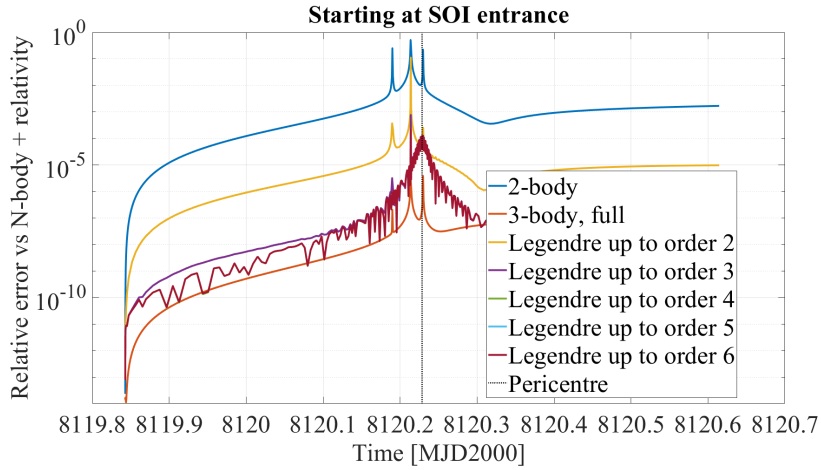


Figure 3.4: Numerical accuracy assessment of the Legendre polynomial expansion, integration starting from the entrance of the SOI.

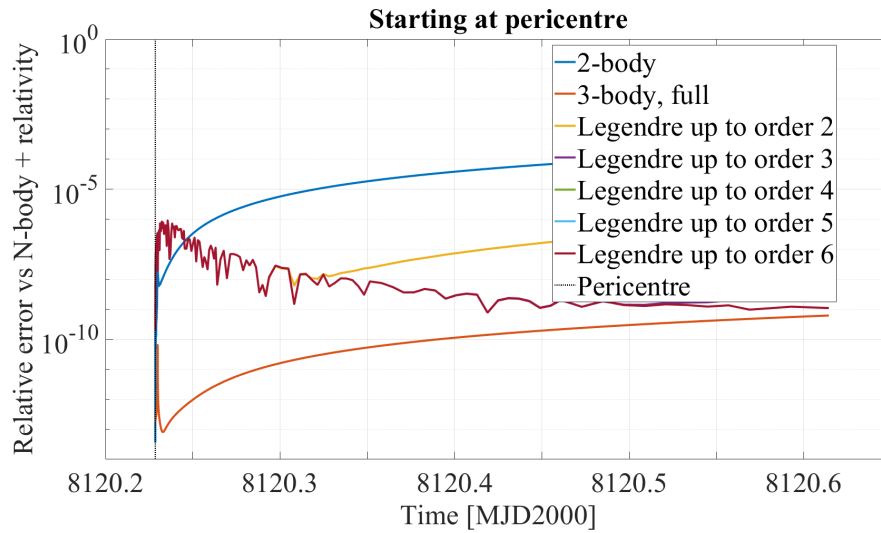


Figure 3.5: Numerical accuracy assessment of the Legendre polynomial expansion, integration starting from the hyperbola pericenter and going forward in time.

processor could not deliver the complete solution perhaps because of the too long reached expressions, albeit the ensured integrability.

#### 3.2.4.5 Accuracy analysis

The presented figures all feature the same color code: the position error of the two-body unperturbed solution is given in blue, the full three-body numerical integration in red, the numerical integration of the three-body potential approximated with the Legendre expansion truncated to the second order in yellow, and the analytical solution by Lie transforms with second order corrections in purple. Especially in the neighborhood of  $t_0$ , the proposed analytical solution (purple)

should converge to the numerical simulation of the Legendre approximated case (yellow).

Figure 3.6 shows the evolution of the position errors for the four presented cases starting the integration at the entrance of the SOI, and stopping it at the exit. As expected, the numerical simulation

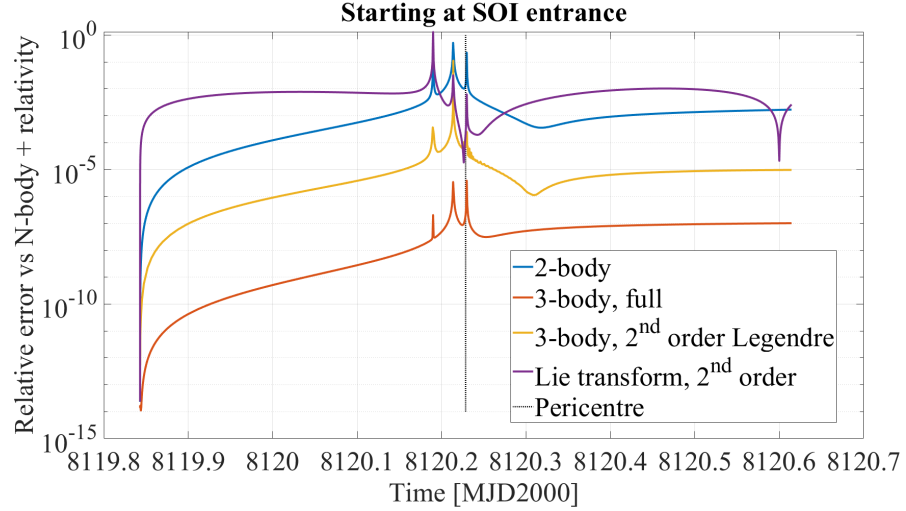


Figure 3.6: Evolution of the position errors within Venus' SOI, setting  $t_0$  as the entrance of the SOI.

of the Legendre expansion truncated to the second order (yellow) reaches an intermediate accuracy level with respect to the two-body (blue) and the complete three body model (red). As it can be observed by the analytical solution by Lie transforms (purple), the procedure guarantees that the initial condition imposed is fulfilled, i.e. the prime variables (corresponding to the two-body case) correspond to the original ones at the starting time. The solution seems although to diverge quickly in time, leading to an higher position error if compared to all the other cases.

Figure 3.7 shows the evolution of the position errors for the four presented cases starting the integration at the hyperbola's pericenter, dividing the domain into two branches to integrate backward to the entrance and forward to the exit of the SOI. Most of the already made observations for the integration started at the entrance of the SOI (Figure 3.6) still hold. An initial error growth can be observed in the numerically simulated Legendre case (yellow), reaching two nearly symmetric relative maxima that, if superposed to Figure 3.3, will match the conditions where  $\cos^2 \psi = 1/3$ . Also, the analytical solution (purple) divergence has become less steep, although apparently still not following the "targeted" model (yellow) nearby the initial time, despite once again seeing the correspondence between prime variables and original ones at the very starting time.

Figure 3.8 shows the evolution of the position errors for the four presented cases starting the integration at the first encountered point

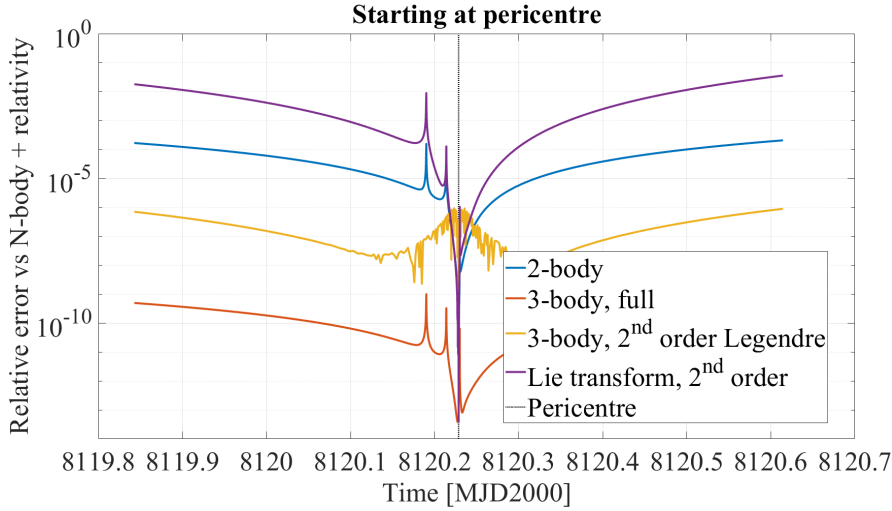


Figure 3.7: Evolution of the position errors within Venus’ SOI, setting  $t_0$  as the hyperbola’s pericenter.

in the trajectory where  $\cos^2 \psi = 1/3$ , dividing the domain into two branches to integrate backward to the entrance and forward to the exit of the SOI. The analytical solution still diverges far from the initial

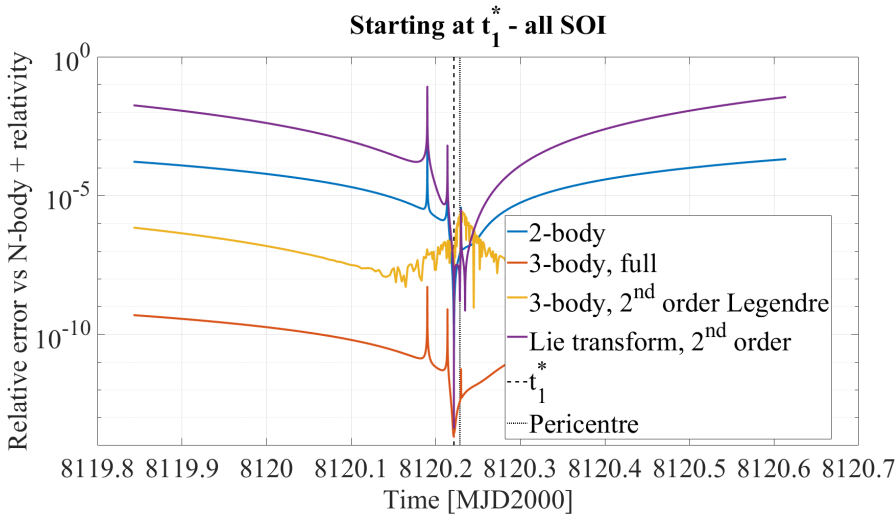


Figure 3.8: Evolution of the position errors within Venus’ SOI, setting  $t_0 = t_1^*$  as the time when  $\cos \psi(t_1^*) = -1/\sqrt{3}$ .

condition, as well as the correspondence between prime variables and original ones is observed again. Differently from the previous cases, the initial error growth seems to be even less steep. Figure 3.9 presents the same information already given in Figure 3.8 but zoomed nearby the initial condition. In this case it can be seen that, initially, the "targeted" (yellow) model is on average followed by the analytical solution, providing a slightly wider time horizon where the proposed approximation seems to hold. The inclusion of higher order terms might contribute to increase the size of this region. The relative order

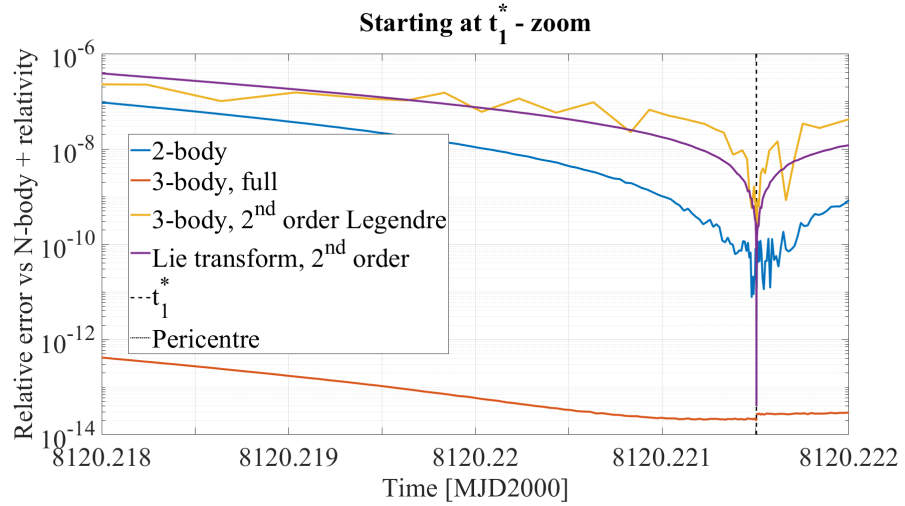


Figure 3.9: Zoom of Figure 3.8 nearby  $t_0 = t_1^*$ .

of the small parameter  $\epsilon$  is  $10^{-3}$ , thus adding the third order terms would provide corrections in the order of  $10^{-9}$ . At the initial time, such a difference is not visible, because the position differences grow to about  $10^{-7}$ . On the contrary, if such a difference was higher than the expected  $10^{-6}$  order, it would clearly appear in the presented graph.

Figure 3.10 shows the evolution of the position errors for the four presented cases starting the integration at the second encountered point in the trajectory where  $\cos^2 \psi = 1/3$ , dividing the domain into two branches to integrate backward to the entrance and forward to the exit of the SOI. The same observations made for the first encountered

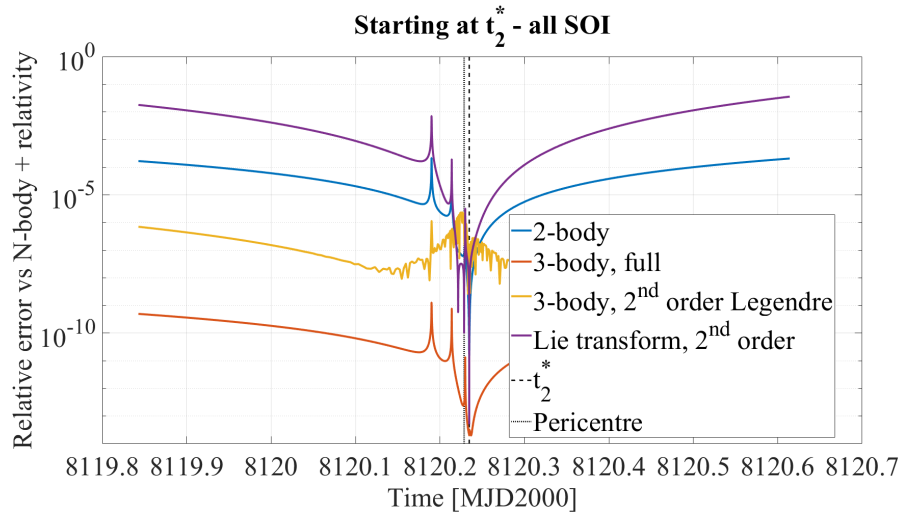


Figure 3.10: Evolution of the position errors within Venus' SOI, setting  $t_0 = t_2^*$  as the time when  $\cos \psi(t_2^*) = 1/\sqrt{3}$ .

point where  $\cos^2 \psi = 1/3$  hold, highlighting the equivalent dynamical nature of the two points. Again, Figure 3.11 presents the same information already given in Figure 3.10 but zoomed nearby the initial

condition. The differences that can be observed between Figures 3.9

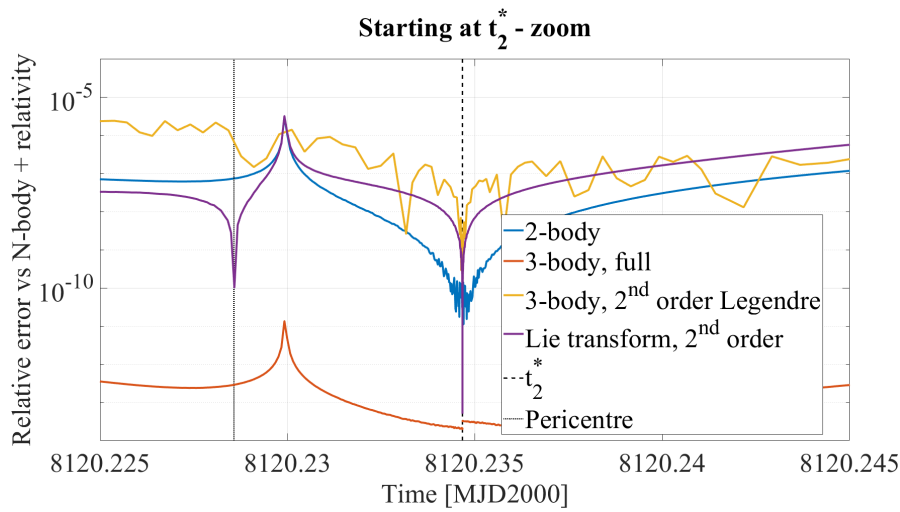


Figure 3.11: Zoom of Figure 3.10 nearby  $t_0 = t_2^*$ .

and 3.11 simply correspond to a slightly wider time region plotted for the latter, nonetheless, the already made observations for Figure 3.9 still hold.

### 3.3 DYNAMIC FLYBY CHARACTERISATION: JACOBIAN EIGENVALUES

This section proposes a method to bridge the gap between the patched conics and the CR<sub>3</sub>BP realms for the study of close encounters, aiming at identifying a new and robust criterion for the definition of the SOI of a planet. Such a criterion should make a more general framework compared to the traditional SOI and Hill's sphere definitions, possibly allowing the study of shallow and distant interactions.

The numerical results presented in [33], [125], and [26] all superpose with the detailed analysis proposed by Amato et al. [3]. In fact, Amato et al. identify the best distance to switch from heliocentric to planetocentric orbital propagation, in terms of numerical stability of the propagation, in the area encompassed by one to three Hill's radii. This suggests that this distance range may have a particular dynamical meaning, hence this section searches for possible analytical solution and the related physical meaning connected to these regions of space. In particular, the criterion proposed by Romano [125], for the ratio between the Jacobian eigenvalues with tolerance equal to 1, represents a fairly simple equation where to build an analytical solution upon, and is seemingly falling perfectly within the one to three Hill's range.

### 3.3.1 Analytical spheroidal locus of points

Considering a generic Barycentric Restricted Three-Body Problem (BR<sub>3</sub>BP) (body 1 - the major, body 2 - the minor, and test particle), the Jacobian  $\mathbf{J}$  of the dynamics is defined as:

$$\mathbf{J} = \begin{bmatrix} \mathbf{o} & \mathbf{I} \\ \mathbf{G} & \mathbf{o} \end{bmatrix} \quad (3.54)$$

with  $\mathbf{o}$  and  $\mathbf{I}$  the  $3 \times 3$  null and identity matrices, respectively, and  $\mathbf{G}$  defined as:

$$\mathbf{G} = \sum_{i=1}^2 \mathbf{G}_i = \sum_{i=1}^2 \frac{\mu_i}{|\mathbf{d}_i|^5} (|\mathbf{d}_i|^2 \mathbf{I} - 3\mathbf{d}_i \mathbf{d}_i^T) \quad (3.55)$$

with  $\mathbf{d}_i = \mathbf{r} - \mathbf{r}_i$ . The eigenvalues can be computed as

$$\det(\lambda \mathbf{I} - \mathbf{J}) = 0$$

which becomes, for the properties of block-square matrices [135]

$$\det(\lambda^2 \mathbf{I} - \mathbf{G}) = 0$$

In the two body case  $\mathbf{G} \equiv \mathbf{G}_i$ , thereby making  $\lambda_{j,J} \equiv \lambda_{j,G}^2$ , with  $j = 1, 2, 3$ . Analytical expressions exist for the two-body Jacobian eigenvalues:

$$\lambda_{i,1}^2 = \frac{2\mu_i}{d_i^3}, \quad \lambda_{i,2}^2 = \lambda_{i,3}^2 = -\frac{\mu_i}{d_i^3} \quad (3.56)$$

Following [125], the eigenvalue ratio with tolerance equal to  $\gamma$  can be re-written expressing the position in a reference frame centered on the body 2 (the secondary), leading to the following equality between the eigenvalues  $\lambda_{1,1}^2$  and  $\lambda_{2,1}^2$ :

$$\frac{\mu_1}{|\mathbf{r}_2 - \mathbf{r}_{21}|^3} = \gamma \frac{\mu_2}{r_2^3} \quad (3.57)$$

where  $\mathbf{r}_{21}$  is the position vector of body 1 with respect to body 2. Expanding  $|\mathbf{r}_2 - \mathbf{r}_{21}|$  with the cosine law gives

$$|\mathbf{r}_2 - \mathbf{r}_{21}| = \sqrt{r_2^2 + r_{21}^2 - 2r_2 r_{21} \cos \theta} \quad (3.58)$$

where  $\theta$  is the angle between the vectors  $\mathbf{r}_2$  and  $\mathbf{r}_{21}$ . Consequently, Equation (3.57) can be re-arranged as

$$\frac{r_2^2}{(r_2^2 + r_{21}^2 - 2r_2 r_{21} \cos \theta)^2} = \left( \gamma \frac{\mu_2}{\mu_1} \right)^{2/3} = \alpha(\gamma) \quad (3.59)$$

Equation (3.59) is quadratic on  $r_2$ :

$$(1 - \alpha(\gamma))r_2^2 + 2\alpha(\gamma)r_{21} \cos \theta r_2 - \alpha(\gamma)r_{21}^2 = 0$$

and can be solved analytically, provided that

$$r_2^2 + r_{21}^2 - 2r_2 r_{21} \cos \theta \neq 0$$

obtaining

$$\begin{aligned} r_2(\theta) &= \frac{-\alpha(\gamma)r_{21} \cos \theta + \sqrt{\alpha(\gamma)^2 r_{21}^2 \cos^2 \theta - (1 - \alpha(\gamma))\alpha(\gamma)r_{21}^2}}{1 - \alpha(\gamma)} \\ &= r_{21} \frac{-\alpha(\gamma) \cos \theta + \sqrt{\alpha(\gamma)(1 - \alpha(\gamma) \sin^2 \theta)}}{1 - \alpha(\gamma)} \end{aligned} \quad (3.60)$$

with minimum and maximum radii, respectively for  $\theta = 0$  and  $\theta = \pi$

$$\begin{aligned} r_2(0) &= \frac{\sqrt{\alpha(\gamma)}}{\sqrt{\alpha(\gamma)} + 1} \\ r_2(\pi) &= \frac{\sqrt{\alpha(\gamma)}}{1 - \sqrt{\alpha(\gamma)}} \end{aligned}$$

for an average radius of

$$\bar{r}_2 = \frac{\sqrt{\alpha(\gamma)}}{1 + \alpha(\gamma)}$$

Equation (3.60) describes a spheroidal locus of points, the *Jacobian spheroid*, axially symmetrical about the line connecting the bodies 1 and 2, as function of the distance  $r_{21}$  between the two bodies and the parameter  $\gamma$ . The radius of the spheroid is given as function of  $\theta$ , the angle between the test particle direction and the line connecting body 1 and body 2. Figure 3.12 shows the spheroidal loci of points on the ecliptic plane with  $\gamma$  in the range  $[0.01, 100]$  in the color scale, for the Sun-Earth case. The spheroid identified by  $\gamma = 1$  is highlighted in red. These spheroids are flattened in the direction that points toward the Sun, whereas enlarged on the opposite side, resulting in a slight eccentricity. Figure 3.12 also compares these loci of points against the traditional SOI and Hill's sphere definitions, as well as against the distance of three times the Hill's radius identified by Amato et al. [3]. In particular, the range one-to-three Hill's radii corresponds rather precisely to the range of the parameter  $\gamma$  that goes from 0.1 to 10. Figure 3.13 presents the same analysis of Figure 3.12, although made on the Sun-Jupiter case. In particular, the higher ratio between the gravitational parameters of the two bodies results in a more eccentric shape of the spheroids. Nonetheless, the correspondence between the one-to-three range identified in [3] still corresponds, on average, to the interval  $\gamma \in [0.01, 10]$ .

### 3.3.2 Dynamical analysis of the Jacobian eigenvalues

The results shown at the end of Section 3.3.1 make a relevant question arise: why and how can the range of optimal switch distance be

Of the two  $\pm$  solutions of a quadratic equations,  $r_2 = r_2^+$  only: since  $\mu_2 < \mu_1$ , then  $0 < \alpha < 1$ , making  $r_2^- < 0$  (not physical).



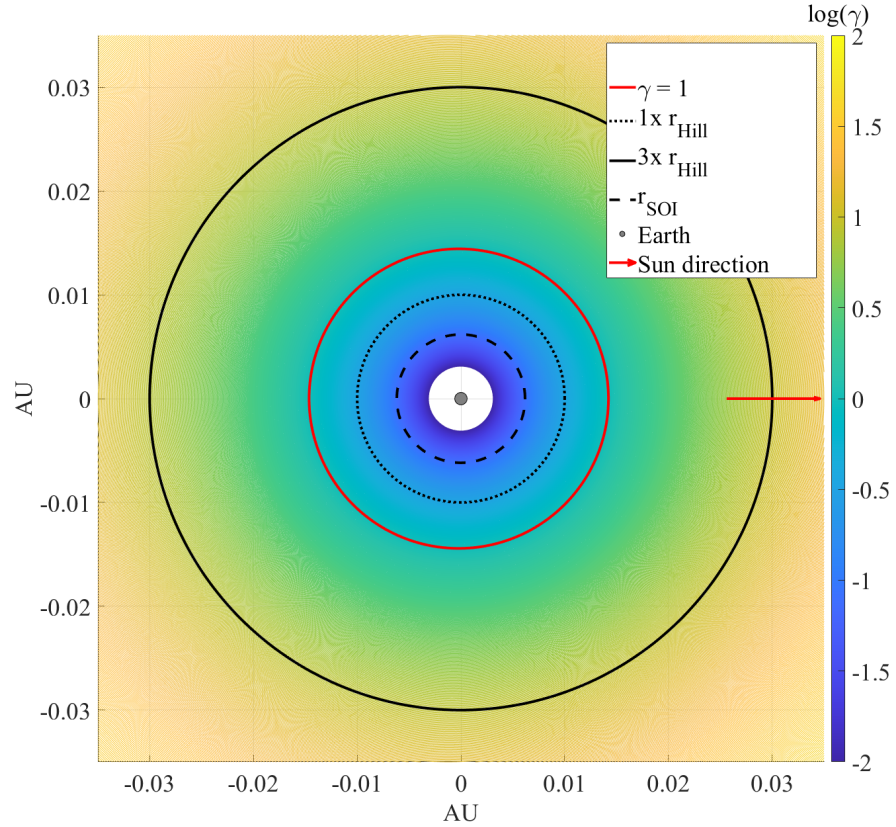


Figure 3.12: Spheroidal loci of points and values of the parameter  $\gamma$  for the Sun-Earth case.

closely represented by values of  $\gamma \in [0.01, 10]$ ? Is there any physical meaning behind the Jacobian eigenvalues, or their square? It may be worth re-analyzing where the solution stems from and, consequently, review whether a physical meaning behind the eigenvalues of the Jacobian of the orbital dynamics exists. Possibly, the interpretation of flybys themselves, in terms of their effect on the dynamics, should be considered.

Observing the BR<sub>3</sub>BP Jacobian expression in Equations (3.54) and (3.55), the approximation as if only the currently dominant body, say 1, were present reflects on the approximated Jacobian as  $\mathbf{G} \approx \mathbf{G}_1$ , thereby making the matrix Jacobian error  $\Delta\mathbf{G} = \mathbf{G} - \mathbf{G}_1$  equal to  $\mathbf{G}_2$ . Since  $\mathbf{G}_i$  is symmetric, its euclidean matrix norm equals its spectral radius:

$$|\Delta\mathbf{G}|_2 \equiv |\mathbf{G}_i|_2 \equiv \rho(\mathbf{G}_i) \equiv \lambda_{\max, \mathbf{G}_i}$$

In other words, the maximum eigenvalue of  $\mathbf{G}_2$  directly measures the error of  $\mathbf{G} \approx \mathbf{G}_1$ . Conversely, the maximum eigenvalue of  $\mathbf{G}_1$  directly measures the error of  $\mathbf{G} \approx \mathbf{G}_2$ .

Denoting the gravitational potential with

$$V = -\mu_1/|\mathbf{d}_1| - \mu_2/|\mathbf{d}_2|$$



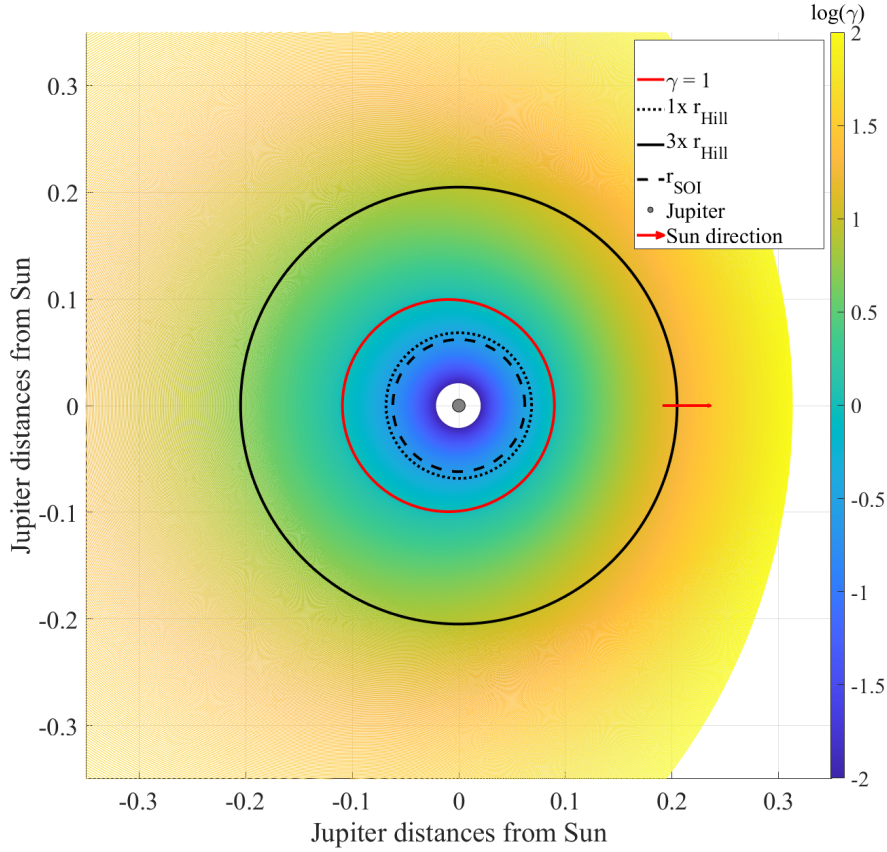


Figure 3.13: Spheroidal loci of points and values of the parameter  $\gamma$  for the Sun-Jupiter case.

it holds that  $\mathbf{G} \equiv \text{Hess}(V)$ . Additionally, denoting with  $\mathbf{K}$  the symplectic matrix defined in Appendix B, it also holds that

$$\mathbf{J} = \mathbf{K} \text{Hess}(\mathcal{H})$$

where  $\mathcal{H}$  represents the Hamiltonian (total energy) of the BR<sub>3</sub>BP. This suggests that, being the Jacobian eigenvalues closely linked with the Hessian of the gravitational potential, they also locally represent the curvature of thereof. On a different viewpoint, the Jacobian of the orbital dynamics represents its local variation across the phase space. Further insight can be obtained studying the eigenvalues of the Hessian of the Hamiltonian  $\text{Hess}(\mathcal{H})$ . Since

$$\text{Hess}(\mathcal{H}) = \begin{bmatrix} \mathbf{G} & \mathbf{o} \\ \mathbf{o} & \mathbf{I} \end{bmatrix}$$

and hence the characteristic polynomial is given by

$$\det \begin{bmatrix} \mathbf{G} - \lambda \mathbf{I} & \mathbf{o} \\ \mathbf{o} & (1 - \lambda) \mathbf{I} \end{bmatrix} = 0$$

which highlights that the eigenvalues related to the velocity terms are all equal to 1, characterizing a phase space of constant curvature along

all the velocity components. Therefore, the remaining eigenvalues are simply given by

$$\det(\mathbf{G} - \lambda \mathbf{I}) = 0$$

Consequently, the Jacobian eigenvalues and the local curvature of the gravitational potential are simply related by a square root operation.

In conclusion, elaborating a criterion to characterize flybys, and thus the boundary that defines whether a close approach is happening or not, choosing a Jacobian-related metric means considering how and where the dynamics itself is changing. Rather than using static criteria, e.g. the traditional SOI that is based on perturbing forces or the Hill's sphere that considers saddle points in the gravitational potential, the Jacobian holds all the information for the local variation of the dynamics, including the body which is affecting it the most. Interestingly, the velocity components of the phase space do not play any role in the proposed characterization (Equation (3.56)), despite the given definition has been obtained in Section 3.3.1 without neglecting the velocity-related effects, but simply by the properties of eigenvalues and block-square matrices. As close approaches are, by nature, phenomena whose dynamics is notably faster than the interplanetary scales, criteria that intrinsically account for variations in the dynamics represent a robust alternative to the traditional "static" SOI and Hill's definitions.

Based on the just made observation and aiming at achieving a domain characterization that follows the proposed metric, the Jacobian error function  $J_E$  can be defined:

$$J_E = \max \left[ \frac{\rho(\Delta \mathbf{G}_1)}{\rho(\mathbf{G})}, \frac{\rho(\Delta \mathbf{G}_2)}{\rho(\mathbf{G})} \right] \quad (3.61)$$

Other than which of the two bodies is locally contributing to the value of  $J_E$ , its magnitude may be a useful metric to highlight regions in the domain where either approximation, i.e. considering only one of the two bodies, is poor, i.e. where a complete three-body model should be used instead. Figure 3.14 shows  $J_E$  in the logarithmic color scale, for the Sun-Jupiter case, comparing the values taken by  $J_E$  against SOI and Hill's sphere. As it could be expected by the continuity of the domain, the transition between the realm dominated by the Sun to the one dominated by Jupiter is smooth. Two higher error regions, called "Thickened regions" in Figure 3.14, appear nearly perpendicularly to the Sun-Jupiter direction, whose possible physical meaning is discussed later in this section.

Figure 3.15 extends Figure 3.14, comparing the analytical loci of points obtained with Equation (3.60) against the computed values of the Jacobian error (on the color scale), for  $\gamma = 0.1, 1, 10$ . The cases  $\gamma = 0.1, 10$  (dashed red lines) well bound the regions where the Jacobian approximation error is higher than 10%. Thickened regions aside, the case  $\gamma = 1$  (solid red line) perfectly predicts the "critical" distances where the error is maximized.

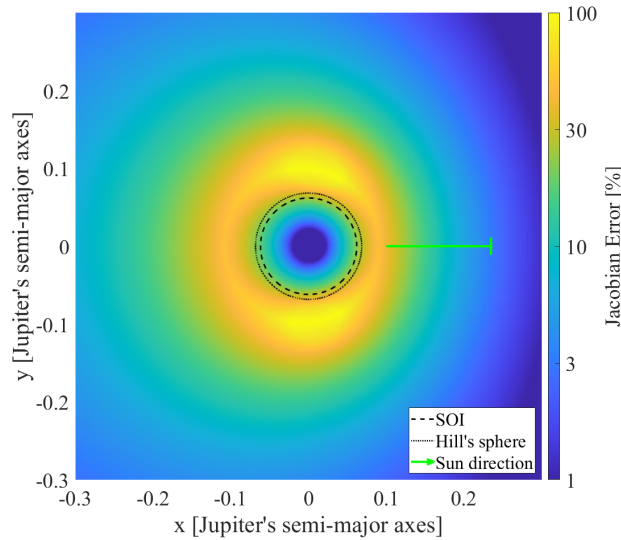


Figure 3.14: Jupiter's Jacobian percent error (color scale), compared against Hill's surfaces (dotted) and SOI (dashed).

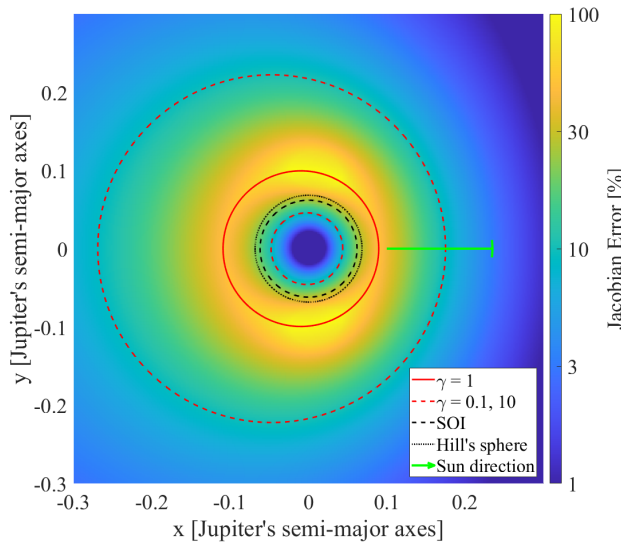


Figure 3.15: Jupiter's Jacobian percent error (color scale), compared against Hill's surfaces (dotted), SOI (dashed), and analytical loci of points (red).

While Figures 3.14 and 3.15 help understanding the physical meaning of the proposed Jacobian eigenvalue-based criterion for flyby detection and characterization, Figures 3.12 and 3.13 provide a better explanation to the ideal range for frame center switch identified by Amato et al. [3]. Nonetheless, the renewed understanding of the Jacobian criterion may give physical support to the range of  $\gamma$  values previously identified. In conclusion, the frame center switch should happen in a region where it is fairly irrelevant for how the dynamics is changing locally, that roughly corresponds to the region where either body dominates the change in dynamics by less than 90%. In any case,

the proposed boundaries remain arbitrary, as even in [3] the ideal range resembled a continuously growing trend outside the identified thresholds. Aiming instead at proposing a new definition of Sphere of Influence, the "critical" spheroid identified by  $\gamma = 1$  should be used, as it represents the region of space where none of the two bodies can dominate over the other, for the local variation on the dynamics. Equivalently, it can be seen as the spheroid where the Jacobian approximation considering either body is locally the worse, when compared against the full BR<sub>3</sub>BP case.

The thickened regions observed in Figures 3.14 and 3.15 may be analyzed with the help of some CR<sub>3</sub>BP concepts. Figure 3.16 shows the perfect alignment of the zero-velocity surfaces, plotted as grey/shadowed areas, with the Jacobian error, on the plane containing Jupiter's orbit about the Sun. The red lines represent different values of  $\gamma$ , equal to 1 (solid), 0.1 or 10 (dotted) and 0.01 (dashed). The selected Jacobi constant to plot the zero-velocity surfaces has been chosen only to highlight their alignment with the thickened regions, without particular meaning. On the zero-velocity curves the kinetic energy content of the test particle, the centrifugal reaction due to the non-inertial rotating frame, and the gravitational attraction of both bodies all balance out. Simplifying the dynamical model along this curves may become inaccurate, particularly if close to the Hill sphere boundaries, as highlighted by the thickened error regions. In other words, the mutual effect of the two bodies on how the dynamics changes is more prominent along the zero-velocity surface direction, on a wider region compared to the pure "critical spheroid line" identified by  $\gamma = 1$ .

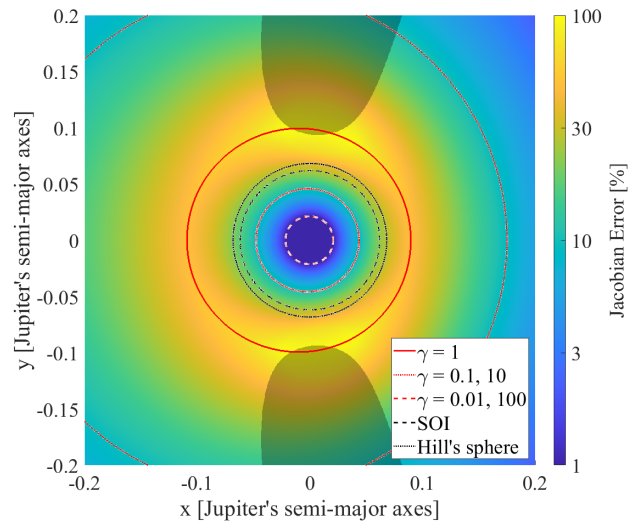


Figure 3.16: Jupiter's Jacobian percent error (color scale), compared against Hill's surfaces (dotted), SOI (dashed), analytical loci of points (red), and zero-velocity surfaces (grey/shadowed).

## 3.4 APPLICATION: KEYHOLE MAPS

Following the results presented in Section 3.3, this section proposes to apply the concept of Jacobian spheroid to the study of shallow encounters, aiming at developing tools and method to aid PP/SDM compliance analyses. In particular, the proposed approach tries to answer the question about why many impacting trajectories in [17] happen out of nominal resonances. While Boutonnet and Rocchi [17] identify a few families for this off-nominal impacting trajectories, this section takes a step back and analyses only the dynamics of one encounter at a time. In this first framework definition, the goal is to find and describe which parameters of the dynamics are leading to off-nominal impacting trajectories.

## 3.4.1 Detection of shallow encounters

The first step to be taken addresses a value of the spheroid parameter  $\gamma$ , so that the detection of shallow encounter can be arbitrarily robust. Since, as of the time of this dissertation, a link between the value of  $\gamma$  and possible maximum and minimum boundaries in the variation of the orbital elements has not been found, an overly conservative threshold is desired, avoiding possible losses of generality due to some small, but still significant, encounter deflections being missed. Figure 3.17 shows the Jacobian spheroids in the case of Earth, compared to the traditional SOI, for  $\gamma = 1, 10, 100$ .

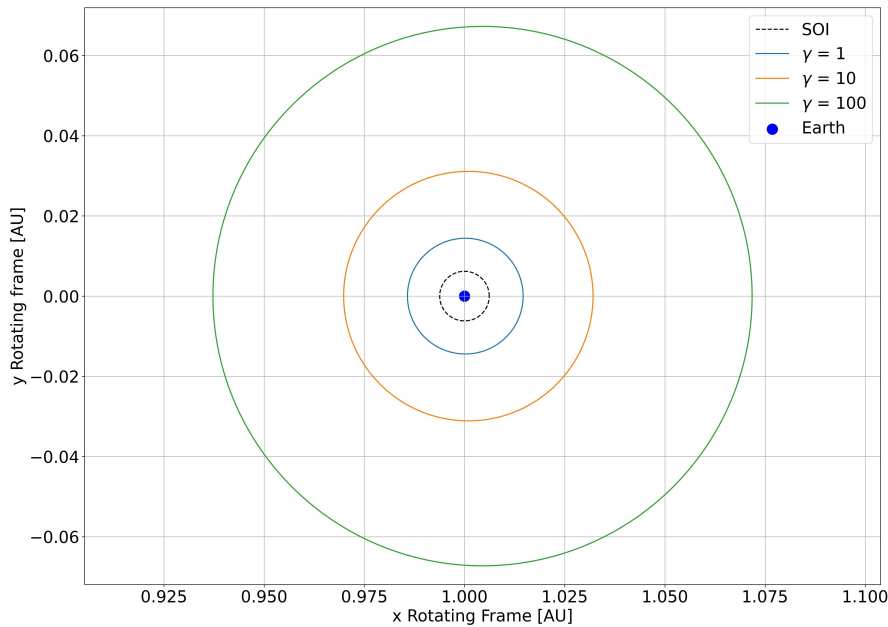


Figure 3.17: Sample spheroids for  $\gamma = 1, 10, 100$  and SOI, centered on Earth in the Sun-Earth synodic frame.

The eigenvalue ratio value is reciprocal to the  $\gamma$  parameter used to identify the spheroid, e.g.  $\lambda = 0.01$  corresponds to  $\gamma = 100$ .

Figure 3.18 compares the evolution of the orbital period for one of the samples propagated in Section 2.5.2 with its distance from Venus, tracking the values taken by the Jacobian eigenvalues of Venus, Earth, and Jupiter<sup>1</sup>. The orbital period is chosen for its prominent role in the identification of orbital resonances and, therefore, future impacts. Using the traditional definition of sphere of influence would allow the detection of the two steepest variations only, without classifying some of the intermediate encounters, i.e. that indeed lead to the next deep flyby, as significant. The common x axis highlights the correspondence of the variations in the orbital period with the spikes in the eigenvalue evolution.

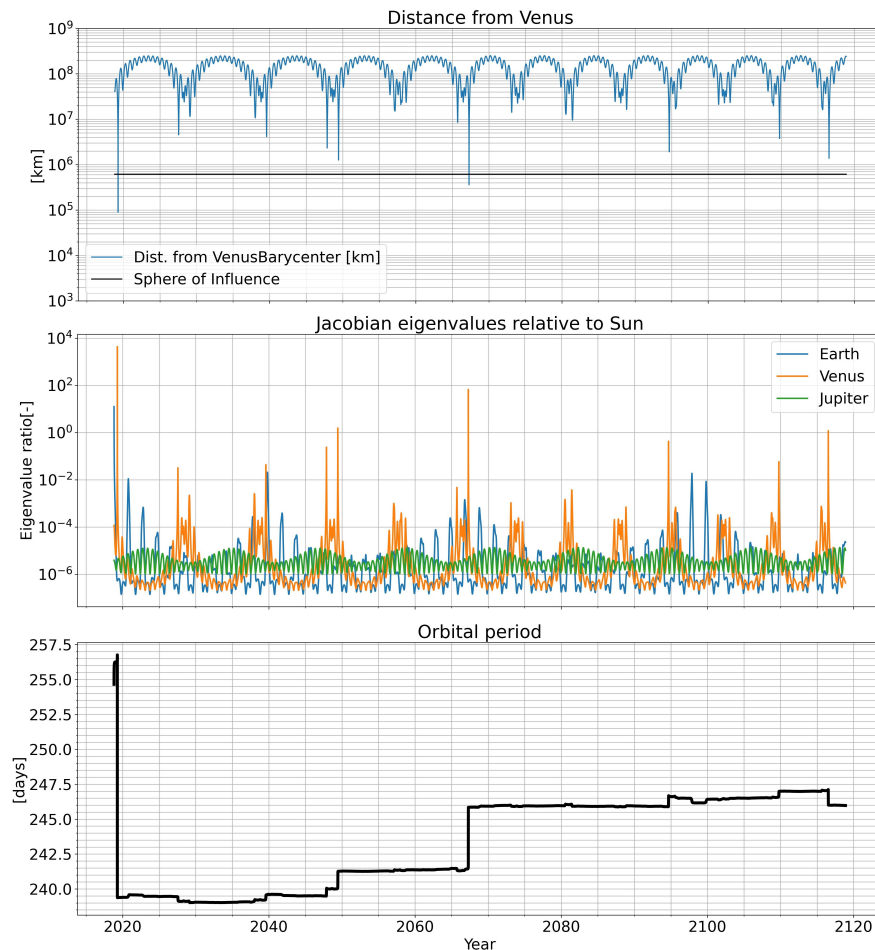


Figure 3.18: Orbital period, distance from Venus, and Jacobian eigenvalues evolution. Example from the PP analysis of Solar Orbiter.

Figure 3.19 extends Figure 3.18, highlighting the time steps that fall inside the  $\gamma = 100$  spheroid in red. With this approach, notably smaller variations of the orbital period can be detected.

Based on this considerations, Figure 3.20 has been built aggregating simulation data of the Asteroids Apophis and 2010RF<sub>12</sub>, and of the ESA

<sup>1</sup> The eigenvalue related to Jupiter is shown only as magnitude reference.



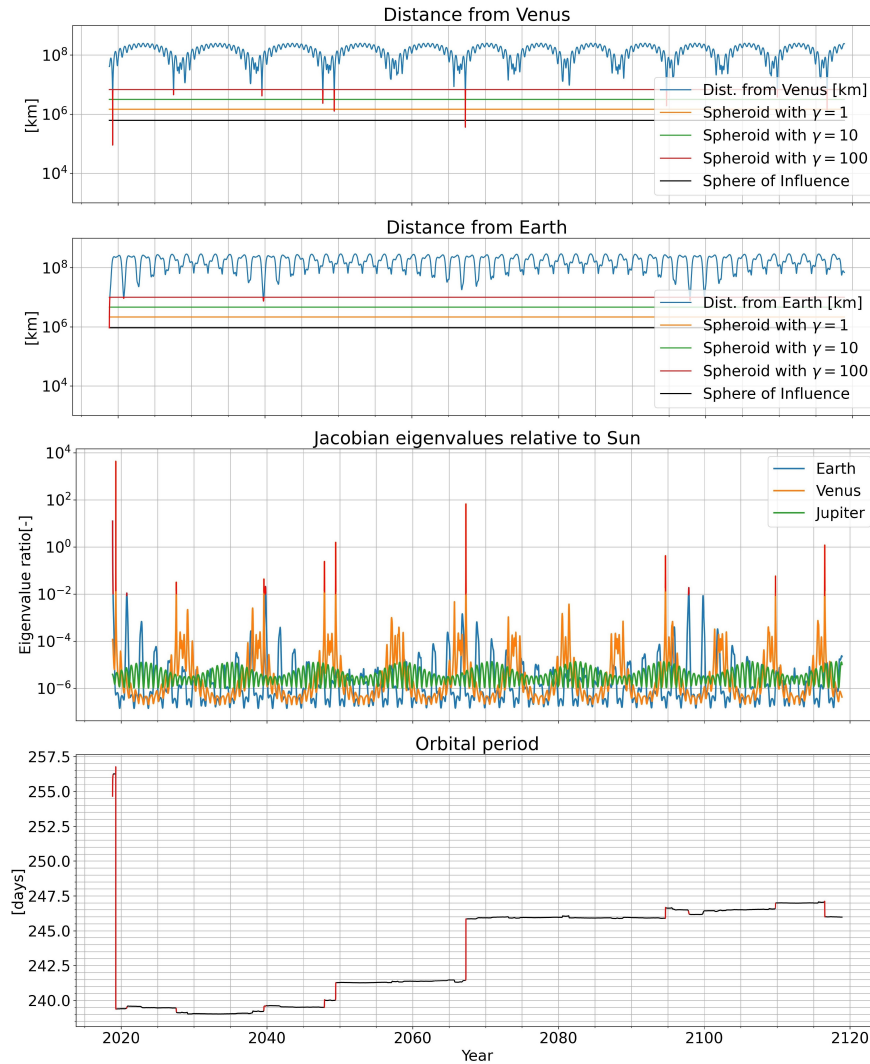


Figure 3.19: Orbital period and Distance from Venus and Earth. Example from the PP analysis of Solar Orbiter.

missions Solar Orbiter [43] and JUICE [44]. Any encounter entering the  $\gamma = 1000$  spheroid for any sample in the MC simulations, and for any of these four cases, is represented as a dot. The encounters are only split based on the encountered body (i.e. Venus or Earth). The  $x$  axis shows the maximum value taken by the eigenvalue ratio during the encounter, whereas the  $y$  axis depicts the relative orbital period variation with respect to the pre-encounter state. The color scale identifies the flyby alignment, in the range  $[-1, 1]$ , representing flybys from completely in front of the planet (alignment close to  $-1$ ) to completely behind (alignment close to  $1$ ). Even considering an eigenvalue ratio of  $0.01$ , the maximum detected variations of the orbital period remain well below  $0.25\%$ . The few outliers observed, all happening for low eigenvalue ratios, may be explained by multiple-body interactions or by rather low relative velocities. For instance,

in the Earth case, the Moon may add its effects on the observed small orbital period changes. In any case, given the still extremely small variation experienced by the vast majority of the samples, the value  $\gamma = 100$  is chosen as conservative threshold for the study of off-nominal resonances. In particular, the spheroid associated with  $\gamma = 100$  is chosen as boundary for two different models. Trajectories are assumed to follow a two-body heliocentric dynamics only outside this spheroid, whereas the full CR<sub>3</sub>BP is considered in the inside.

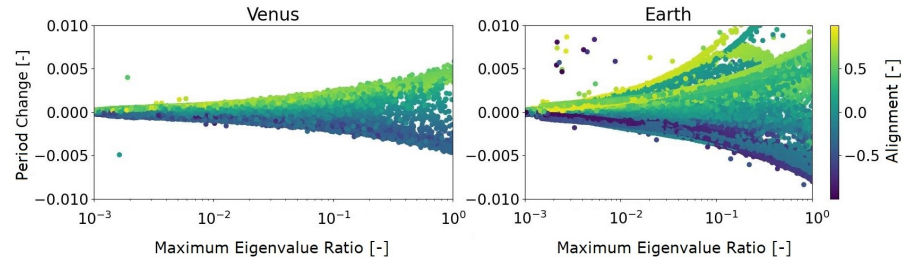


Figure 3.20: Aggregate orbital period variations for MC simulations of, Apophis, 2010RF<sub>12</sub>, Solar Orbiter, and JUICE.

#### 3.4.2 Encounter phasing in the Restricted Three Body Problem

Having defined the criterion to split the domain between unperturbed two-body and CR<sub>3</sub>BP realms, a few more implications arise. Since any point of the  $\gamma = 100$  spheroid boundary corresponds to prescribed position vector, say  $\mathbf{r}_i$ , only three degrees of freedom (the velocity components) remain, in principle, to the full determination of the orbital state at the boundary of the spheroid. Additionally, assuming the interactions to be fully ballistic and limiting, for now, the analysis to one single secondary body, the Jacobi constant  $C_J$  is conserved throughout the full integration span, adding thus a constraint between the orbit's total energy and angular momentum. Therefore, the remaining two degrees of freedom are broken down to the following parameters:

- The orbit's semi-major axis  $a$ , tightly linked to the orbital period around body 1;
- a phasing parameter, to be better identified in the following lines.

The phasing parameter should be defined based on what actually makes the trajectory become subject to an encounter. In particular, any flybys in the proposed framework are either tangent or secant to the  $\gamma = 100$  spheroid boundary. Following the derivation made by Campiti [24], prescribed values of semi-major axis and Jacobi constant (in the Tisserand approximation of Equation (3.6)) results, in general, in a set



of four orbits passing on the same position vector  $\mathbf{r}_i$ : the direction of the angular momentum  $\hat{\mathbf{h}}$  must fulfil the following conditions

$$\begin{cases} \hat{\mathbf{h}} \cdot \hat{\mathbf{k}} = \cos i \\ \hat{\mathbf{h}} \cdot \mathbf{r}_i = 0 \\ |\hat{\mathbf{h}}| = 1 \end{cases} \quad (3.62)$$

that, once expanded and re-arranged, lead to

$$\begin{cases} \hat{h}_3 = \cos(i) \\ \hat{h}_2 = -\frac{1}{r_{i,2}} [r_{i,3}\hat{h}_3 + r_{i,1}\hat{h}_1] \\ \hat{h}_1 = \pm \sqrt{1 - \hat{h}_2^2 - \hat{h}_3^2} \end{cases} \quad (3.63)$$

The  $\pm$  in the expression for  $\hat{h}_1$  highlights the existence of two possible directions of the angular momentum,  $\hat{\mathbf{h}}_1$  and  $\hat{\mathbf{h}}_2$ . In turn, this leads to two possible directions for the line of nodes,  $\hat{\mathbf{n}}_1$  and  $\hat{\mathbf{n}}_2$ , thereby giving two distinct values of the right ascension of the ascending node  $\Omega$ . Finally, writing explicitly the true anomaly  $f$  with the conics equation gives two opposite true anomalies that correspond to the distance  $r_i$  from the primary [24]:

$$f_{1,2} = \pm \arccos \left( \frac{1}{e} \left( \frac{a(1-e^2)}{r_i} - 1 \right) \right) \quad (3.64)$$

Restricting the analysis to the planar case, with inclination  $i = 0$ , removes the two cases related to the different right ascensions of the ascending node<sup>2</sup>, with only two remaining orbits completely defined by the true anomalies  $f_1$  and  $f_2$ . This implies that, given a semi-major axis  $a$  and a Jacobi constant  $C_J$ , any trajectory crossing or touching (the limit case with  $f_1 = f_2$ ) the spheroid at a given point can be described by these two true anomaly values only.

In the remaining of the section, the Sun-Earth case is considered, although the concepts could be extended to any significant pair of primary and secondary body. Figure 3.21 shows the different phasing of a set of heliocentric trajectories with common Jacobi constant  $C_J = 2.95$  and semi-major axis (leading to the 3 : 2 resonance with Earth), in the Sun-Earth rotating frame. The phasing is described with the *spheroid angle*  $\theta$ , the angle measured counterclockwise from the Sun-to-Earth line on the spheroid surface, shown in the color scale. Because of the prescribed Jacobi constant, not all the spheroid points result in a physical orbit, as constraints on pericenter and apocenter distances appear for a given semi-major axis. Nonetheless, Figure 3.21 encompasses the full useful phasing range, with outside trajectories that certainly do not experience any encounter with Earth.

Evidently, a strong link between the true anomaly  $f$  at the spheroid boundary and the spheroid angle  $\theta$  exists, highlighted in Figures

*Formally, it differs only by a flip of sign in the spheroid angle  $\theta$  defined in Section 3.3.1*

<sup>2</sup> Its value becomes arbitrary, and usually set equal to 0.

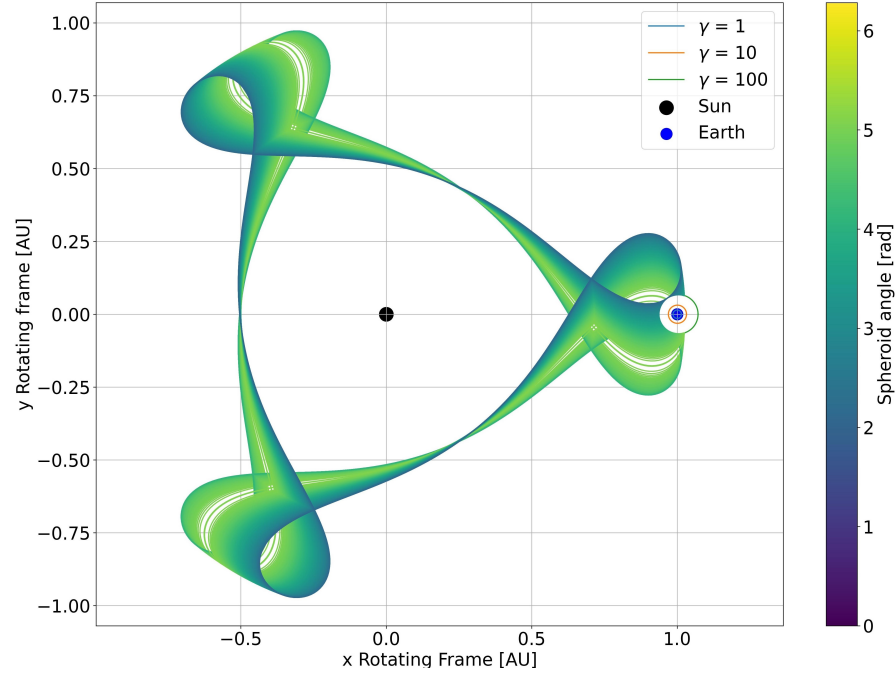


Figure 3.21: Visualization of phased trajectories with the  $\gamma = 100$  spheroid, for  $C_J = 2.95$  and the 3 : 2 resonance trajectory with Earth.

3.22a and 3.22b. In particular, the previously defined spheroid angle  $\theta$  coincides with the *inner* definition of Figure 3.22a and is also depicted in the color scale in all the presented Figures (3.21, 3.22a, 3.22b).

The outer definition of the angle can also be seen as

$$\cos \theta_{\text{outer}} = \hat{\mathbf{r}}_{\text{Heliocentric}} \cdot \hat{\mathbf{r}}_{\text{Sun-Earth}} \quad (3.65)$$

whereas for the inner one

$$\cos \theta_{\text{inner}} = \hat{\mathbf{r}}_{\text{Earth-centric}} \cdot \hat{\mathbf{r}}_{\text{Sun-Earth}} \quad (3.66)$$

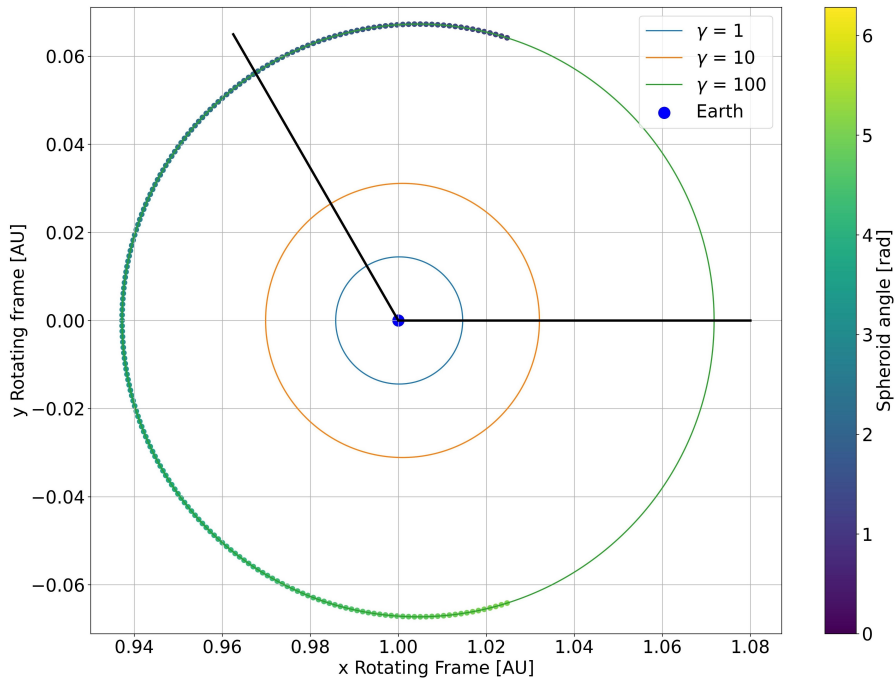
with the "hat" symbol denoting the direction unit vectors, providing a purely geometric link between inner and outer angles for *any* point on the spheroid surface, dependent only on the Sun-earth distance and the size of the spheroid. Recalling the arbitrary choice of  $\Omega$  associated with the planar problem, if the node line is set along the Sun-Earth line, then it holds that

$$\theta_{\text{outer}} = \omega + f \quad (3.67)$$

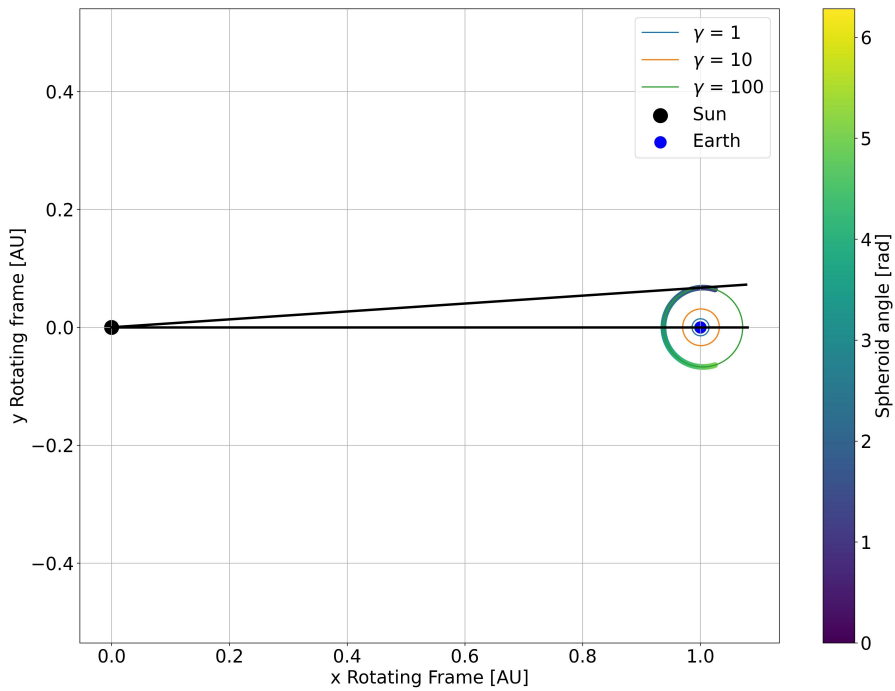
with  $\omega$  the pericenter anomaly. In particular, defining the *rotating* right ascension of the ascending node  $\Omega_{\text{rot}}$  as

$$\Omega_{\text{rot}} = \Omega_0 - nt \quad (3.68)$$

where  $n$  is the angular rate of the rotating frame. In other words, for the purposes of this analysis the rotation of the Sun-Earth synodic frame can be seen as a constant-rate backward precession of the orbit's



(a) Inner spheroid angle



(b) Outer spheroid angle

Figure 3.22: Visualization of phased trajectories (colored dots) with the  $\gamma = 100$  spheroid, for  $C_J = 2.95$  and the 3 : 2 resonance trajectory with Earth. Inner and outer spheroid angles marked with the black lines.

right ascending node. If  $\Omega_0$  is chosen as the Sun-Earth direction at the time  $t = 0$ , then letting  $\Omega = \Omega_{\text{rot}}$  pairs  $\theta_{\text{outer}}$  with the heliocentric

orbital elements. In summary,  $\theta_{\text{outer}}$  can be seen as equal to the *rotating longitude*  $l$ , making the phasing of the orbit, i.e. the particular point of entrance/exit to/from the spheroid, function of the orbital parameters and the initial condition only. Figure 3.23 shows the link between the rotating longitude  $l$  and the spheroid angle  $\theta$ . Since  $\theta$  does not have ambiguous values (opposite to  $l$ , near the spheroid upper and lower edges, as Figure 3.22b also shows), the spheroid angle  $\theta$  is chosen as the reference phasing parameters in the following analyses.

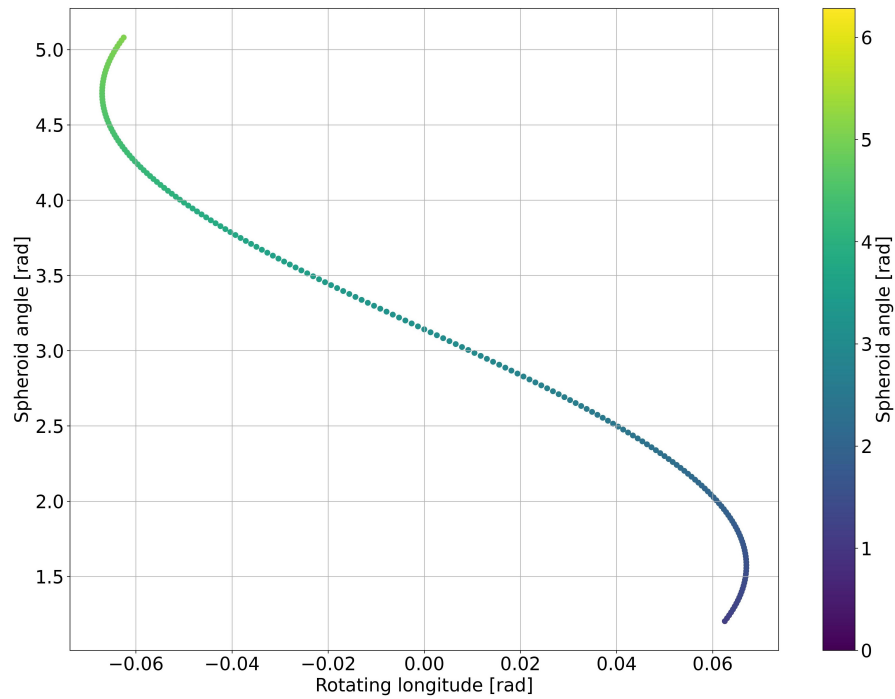


Figure 3.23: Link between the spheroid angle  $\theta$  and the rotating longitude  $l = \theta_{\text{outer}}$  in the case of Earth.

Since the presented phasing definition acts only on  $\Omega$  and no assumption has been introduced on the sum between  $\Omega$  and  $l$ , it can be generalized to the case of trajectories with  $i \neq 0$  without any change, provided that generic spherical geometry rules or general Cartesian/Keplerian conversion routines are used. The only modification appears in the spheroid-phased generated trajectories, that would become tightly linked to the chosen inclination and, in general, would include all the four solutions arising from the two possible values of  $\Omega$ .

For the sake of clarity, Figure 3.21 does not show the trajectories going inside the spheroid, letting Earth and the spheroids to appear in the picture. This observation opens a new characterization possibility for the trajectories generated starting from a given  $(a, C_J)$  pair. Figure 3.24 shows a complete classification of the spheroid-phased trajectories.

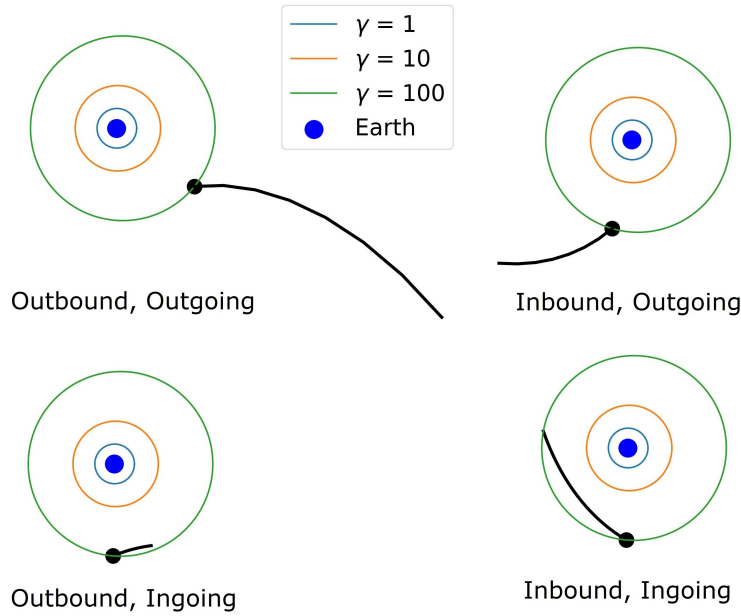


Figure 3.24: Topology of spheroid-phased trajectories.

In particular, based on whether the trajectory is entering or exiting the spheroid, it can be considered as either *ingoing* or *outgoing*. This classification follows the simple dot product between the Earth-centric position and velocity vectors  $\mathbf{r}_{EC}$  and  $\mathbf{v}_{EC}$ :

- ingoing trajectories are characterized by  $\mathbf{r}_{EC} \cdot \mathbf{v}_{EC} < 0$ ;
- outgoing trajectories feature  $\mathbf{r}_{EC} \cdot \mathbf{v}_{EC} > 0$ .

Tangent trajectories are found with  $\mathbf{r}_{EC} \cdot \mathbf{v}_{EC} = 0$ , and are neither ingoing nor outgoing. This criterion is a good approximation for spheroids with small eccentricity, a general and more refined version would replace the direction  $\mathbf{r}_{EC}$  with the local perpendicular direction to the spheroid surface. Additionally, based on the true anomaly value, a trajectory can be:

- *inbound*, with  $-\pi < f < 0$  or  $\pi < f < 2\pi$ ;
- *outbound*, with  $0 < f < \pi$  or  $-2\pi < f < -\pi$ .

The values  $f = 0$  and  $f = \pi$  correspond to the limit pericenter and apocenter cases, which are neither inbound nor outbound.

### 3.4.3 Keyhole maps and generalized orbital resonances

Having extended the encounter definition and characterization, this section tries to study the effects of flybys in the extended sense, including shallow interactions and using the CR<sub>3</sub>BP. In particular, new

insight on impacts that nominally lie outside the traditional orbital resonances is sought for.

The definition of trajectory phasing presented in Section 3.4.2 fixes the remaining degree of freedom to completely map the set of encounters, for a given Jacobi constant and semi-major axis pair  $(C_J, a)$ . Similarly to what presented in Figure 3.21, for 100 evenly spaced semi-major axis values within the interval between the 3 : 2 to the 1 : 2 resonances, a grid of 200 evenly spaced points on the spheroid angle  $\theta$  is generated, all with the same Jacobi constant  $C_J = 2.95$ . Then, the trajectory set is first filtered for non-physical cases (e.g. unreachable points on the spheroid for the given  $(C_J, a)$  pair) and split according to the trajectory topology of Figure 3.24. The first distinction between ingoing and outgoing trajectories determines the dynamical model used in the study: ingoing trajectories are propagated up to the spheroid exit (unless an impact is detected), whereas outgoing orbits are propagated forward in time for 11 years<sup>3</sup>, or until a new entrance in the  $\gamma = 100$  Earth's spheroid is detected.

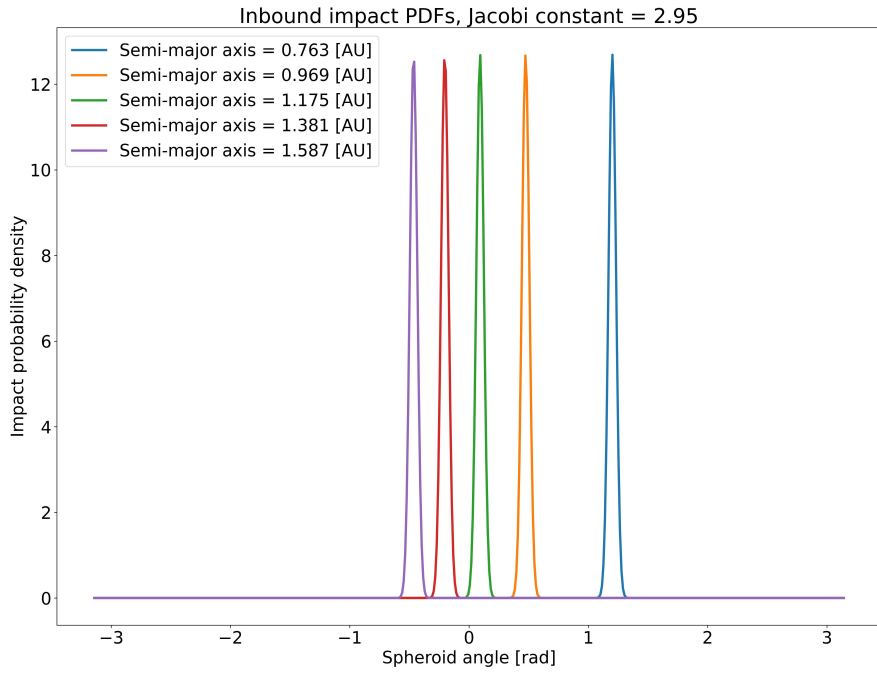
The ingoing family is split into inbound and outbound, and the propagated trajectories are used to train two simple surrogate models that replace the full CR<sub>3</sub>BP propagations, yielding a model of impact probability density as function of Jacobi constant, semi-major axis and spheroid angle  $\theta$ , according to the following steps:

1. the spheroid angle  $\theta^*$  associated with the minimum altitude encounter is found, for each sub-family and each value of semi-major axis;
2. the  $\theta^*$  values are used to train a cubic polynomial regression, that predicts the spheroid angle of maximum flyby depth as function of the semi-major axis, for each sub-family;
3. finally, a standard deviation value equal to the spheroid angle discretization step ( $0.01\pi$ , for the 200 angle samples of this case) is introduced on the modeled  $\theta^*$ .

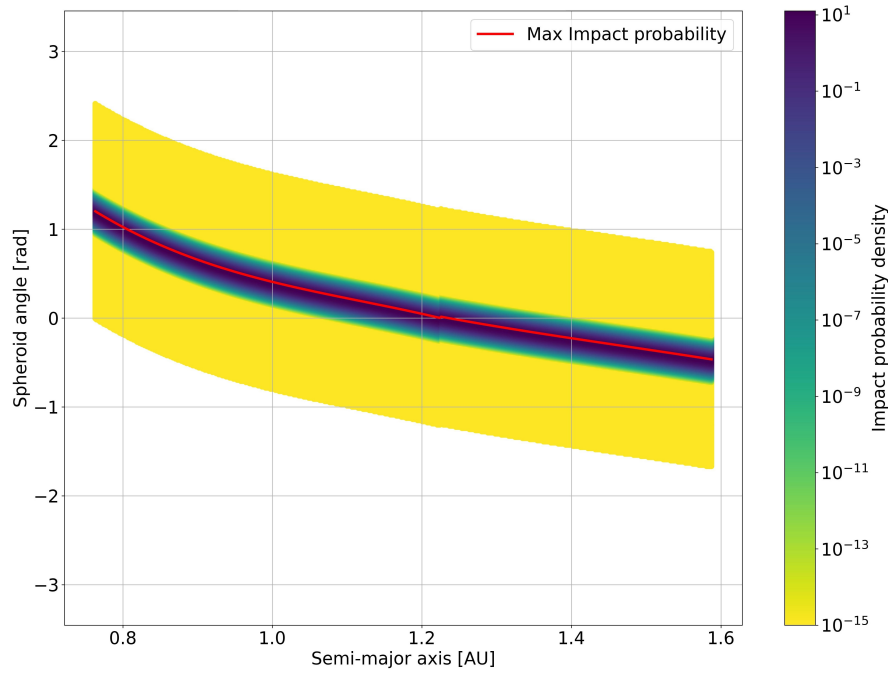
The model represents the mean spheroid angle about which an impact is most likely to be found. Figure 3.25a samples a few semi-major axis values and shows the associated probability density functions, whereas Figure 3.25a shows the full surrogate model.

The outgoing family results in propagated trajectories that either end at the end of the integration span, or that experience a new spheroid entrance. For the latter case, the entrance spheroid angle  $\tilde{\theta}$  is computed, together with the true anomaly  $\tilde{f}$  at the end of the simulation, to detect whether the entrance is happening with inbound or outbound nature. Trajectories with inbound exit resulting in an inbound entrance (or alternatively, outbound exit to outbound entrance) would resemble full resonances. Instead, trajectories with a switch in the sub-family

<sup>3</sup> The time frame of this analysis of this case



(a) Sampled probability density functions



(b) Overall probability density function

Figure 3.25: Surrogate model for the impact probability density.

from exit to entrance would look like pseudo resonances, as the next encounter would happen in a position on the heliocentric orbit much different from the one at exit.

Figure 3.26 represents the trajectories of the outgoing family with dots, plotting the pair semi-major axis  $a$  (on the  $x$  axis) and spheroid

angle  $\theta$  (on the  $y$  axis and splitting the inbound (bottom) and outbound (top) sub-families, for the  $C_J = 2.95$  case. The predicted impact probability density associated with each outgoing initial condition is given in the color scale, from low (yellow) to high (blue). Outgoing trajectories that do not result in a new spheroid entrance, within the analyzed time span, are marked with a lighter yellow color, and are assimilated to low impact probability areas in the Keyhole map. The yellow tone is kept slightly different to remark the distinction between trajectories re-entering the spheroid and not. Each sub-plot in Figure 3.26 is a *Keyhole Map*: keyholes become clearly visible as the regions of the Keyhole Map that tend toward the dark blue color. The light grey lines represent the nominally resonant semi-major axis values in the 11 year time span, and provide a useful reference for the keyhole identification: since many of the predicted keyholes appear outside a line of nominal resonance, the proposed impact probability mapping strategy has successfully extended the analysis to nominally off-resonant trajectories.

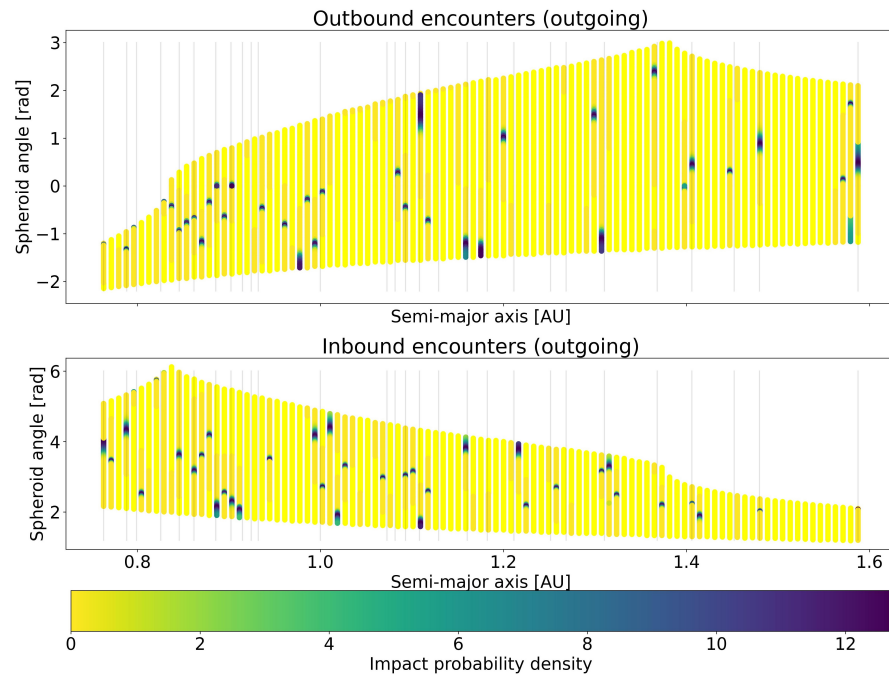


Figure 3.26: Sun-Earth Keyhole Map up to one encounter and 11 year time span.

The Keyhole Map in Figure 3.26 is constructed by evaluating the surrogate model for the impact probability density at the immediate next spheroid entrance, after propagating the trajectories in the two-body problem: in other words, while the Keyhole Map has the *initial* outgoing spheroid angle  $\theta$  on the  $y$  axis, the spheroid angle at the end of the propagation (i.e. at the next spheroid entrance) is used for the evaluation with the ingoing surrogate model. Additionally, the surrogate model to be evaluated (inbound vs outbound) is



chosen according to the true anomaly at the spheroid entrance, for each propagated trajectory. Essentially, the surrogate model is what provides the color to each dot in the Keyhole Map, allowing regions of high impact probability to be highlighted. The peculiar shape of the Keyhole Map is due to the chosen value of Jacobi constant: specific semi-major axes are physically prevented to feature some inbound or outbound conditions, because of the constraint in the eccentricity value, that results in a threshold of the spheroid angle that certain trajectories can reach.

Figure 3.27 proposes a validation of the Keyhole map approach. The samples at the center of each of the three Keyholes, among the ones already available from the generation of the Keyhole map, are selected and propagated following a two-body heliocentric model up to the spheroid threshold. Here, the propagation switches to the CR3BP. The propagated trajectories are shown in corresponding colors to the highlighted keyholes, i.e. in blue, red, and magenta. While the blue trajectories is supposed to be in nominal resonance with Earth, the propagation result is not an impact, rather a very deep flyby. This may be explained by the non-perfect phasing, which leads to a still dangerous encounter although not colliding. Similarly, the magenta trajectory also experiences a low-altitude encounter, although not being in strict resonance with Earth. Finally, the red trajectory ends its propagation with an Earth impact, despite again being nominally out of the resonance condition. In this case, the encounter phasing perfectly compensates the non-resonant orbital period, leading to a subsequent impact that could not be predicted using two-body approaches.

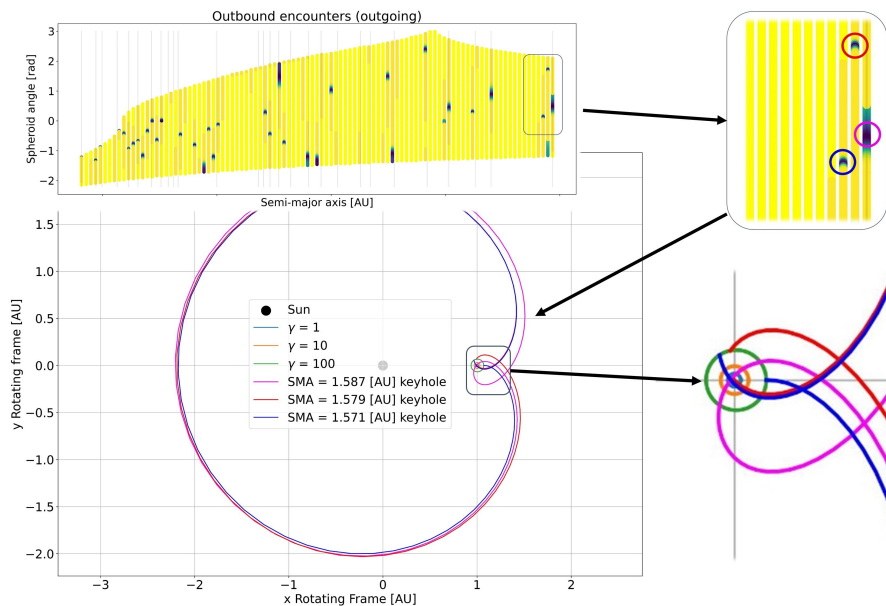


Figure 3.27: Keyhole impacting trajectory prediction examples.

The results shown for the Keyhole Map validation in Figure 3.27, in particular two out of the three trajectories not impacting Earth, do not represent an incomplete validation. Rather, the Keyhole Map is a statistical graphical tool that has the sole purpose of highlighting regions in the semi-major axis and phasing space that are intrinsically associated to high impact probability levels. In fact, all the trajectories propagated in Figure 3.27 experience a very low-altitude encounter, in the order of 1000 km: this proves the success of the Keyhole Map in this predictions. The map suggests that impacts are indeed located around those regions, even in nominally off-resonant cases. The Keyhole Map can support the analysis of impacting trajectories, the recommended use superposes an outgoing probability distribution, given in terms of semi-major axis and exit spheroid angle, for the computation of impact probabilities corresponding to regions in the Keyhole Map that are "wet" by real uncertainty distributions.

### 3.5 SUMMARY

#### 3.5.1 *Analytical methods for flyby perturbations*

As in Hori's approach [66], the choice made for the integration constant is not arbitrary. In the flyby case, that integration constant makes the difference in obtaining (or not) a physically correct analytical solution. The main problem straightforwardly suggests to impose null perturbing effects at infinity, as at great distances from the primary the non-uniform gravity field of the planet flown by is negligible.

Yet, in practical examples for the Solar or a planetary system, the great distances from the flyby body are characterized by stronger effects of the main attractor of the whole system. The analysis shown in Section 3.2.4.5 shows that a Hori-like boundary condition, using the Keplerian solution at infinity, is not suitable for the third body case: this unique setting corresponds to the second order term only being null, rather than the full original potential. In addition, trajectories such that  $\cos^2 \psi = 1/3$  is never verified may exist. Considering the higher order terms, a value of  $\cos \psi$  that makes all the expansion terms to vanish cannot be found. This aspect remarks that the method presented in Section 3.2 remains valid for the main problem only, or more in general for vanishing perturbations at infinity.

In clear contrast with the improvements obtained for the main problem case, the Keplerian boundary condition where the perturbing potential vanishes provides even worse predictions than the unperturbed case, without converging toward the numerically simulated trajectory for the same physical model. Other types of boundary conditions should be explored in dedicated works, for instance the magnitude difference between the perturbation and the Keplerian term in specific regions of the hyperbolic trajectory.

### 3.5.2 *Dynamic flyby characterisation: Jacobian Eigenvalues*

A flyby characterization approach that accounts for the dynamical nature of the encounter is proposed, focusing on the changes on the dynamics caused by the body flown by. The actual smooth transition of the motion from interplanetary to planetary is also modeled, highlighting regions of space where none of the two body is clearly dominating, and approximating the dominance that each body has in each point of the planet neighborhood.

Apart from the regions nearby the critical distance and along the zero-velocity curves, the proposed parametric analytical model accurately predicts the loci of points of common Jacobian error. Setting the parameter  $\gamma = 1$  allows the use of the "critical" spheroid as robust flyby detection criterion. Given the parametric nature of the criterion, this approach is suitable to improve the characterization of shallow encounters. A specific value of the parameter  $\gamma$  can successfully bound the corresponding region of the Jacobian error surrounding the critical distance, meaning that, within those analytical boundaries, variations on the nominal trajectory can be expected, against the pure two-body dynamics. Future works should address this point in deeper detail, aiming at finding a direct link between the value of  $\gamma$  and the expected maximum or minimum variations, perhaps on relevant orbital element-based indicators.

### 3.5.3 *Keyhole Map*

The Keyhole Map implements the Jacobian spheroids as a robust threshold to distinguish regions of space where a CR<sub>3</sub>BP approach should be preferred to the pure two-body motion around the primary. The overall logic relies on this domain split, considering a sequence of two-body and CR<sub>3</sub>BPs instead of an overall N-body model: since the influence of any celestial body outside its own spheroid should be negligible, then such effects should be statistically irrelevant, provided that all the "major" interactions are instead included. This approach should ensure a high computational efficiency, given that it keeps the dynamics as simple as possible, without the need of ephemeris models or data. Future works should develop the Keyhole Map keeping a focus on this aspect, aiming at statistically validating this statement: only then can the Keyhole Map be proved a suitable candidate to replace the existing MC-based technique for the computation of interplanetary impact probabilities.

The application of the Keyhole Map to the detection of Earth keyholes proves the potential in the characterization of nominally off-resonant impacting trajectories. Despite shown for a limited time interval and for a single encounter, the concept can be extended to longer time spans and the multiple encounter case. Since, in the CR<sub>3</sub>BP,

for multiple encounters with the same celestial body the Jacobi constant/Tisserand parameter is conserved the same surrogate model used for the definition of the high impact probability phasing parameters can be used throughout all encounters. The major extension is represented by the required chaining of multiple flybys, that results into the ingoing conditions being mapped to new outgoing trajectories. Figure 3.28 represents the newly mapped post-encounter trajectories, resulting from the same simulations used to train the surrogate model in Section 3.4.3. The color scale shows the flyby depth, i.e. the distance from Earth's surface. As expected, the map's shape resembles the Keyhole Map of Figure 3.26, with the encounter effect that becomes evident as a "sliding" along the minimum altitude line.

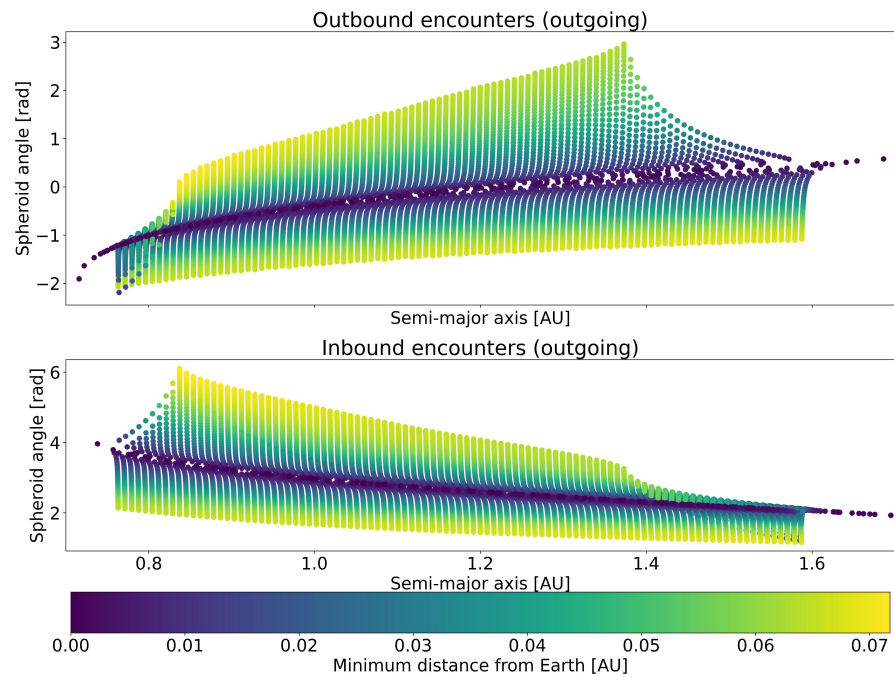


Figure 3.28: New outgoing trajectories after the encounter.

The complete modeling of the encounters, in terms of including the effects of Figure 3.28 in the map chaining process required to build a multi-encounter analysis, is the next key step for the extension of the Keyhole Map theory. Given the importance of distance interactions, these effects should be accurately mapped, to ensure that the robustness associated with the  $\gamma = 100$  spheroid definition is preserved. Eventually, the Keyhole Map should also include out-of-plane trajectories and interactions with other planets. The former task may in some cases be neglected, as for instance most interplanetary missions preserve nearly planar initial nominal orbits, which includes their disposal objects. The latter may instead be studied, since encounters with a third body may trigger new impacting pathways with, in general, any body in the considered system. The main difficulty of this case is the modification that would inevitably affect the Jacobi constant. The

map chaining would then need to follow a tree-based logic, with only maps in the same branch that can follow the simplest multi-encounter stacking principle.

In the end, the multi-body and/or multi-encounter Keyhole Map should not look much different from Figure 3.26: denser regions and new Keyholes may appear, although its shape would be preserved. Similarly, it should be used combined with a real-case probability distribution given in terms of semi-major axis and spheroid angle, so that only the relevant regions of the domain may contribute to the final impact probability computation. The Keyhole Map remains a statistical tool useful to gain insight on potentially dangerous trajectories, described as their flyby exit: it is not meant to be a complete replacement of trajectory simulators. Yet, the promising preliminary results proposed in this chapter highlight a significant potential, for the reduction of the computational burden of impact probability-related tasks.



# 4

---

## PICARD-CHEBYSHEV INTEGRATION AND AUGMENTATION

---

As any other engineering application that involves simulated data, the choice of the numerical solver that handles the propagation of orbits plays a fundamental role in the final accuracy and efficiency of the simulation software. Each numerical integration scheme may have its own advantages and drawbacks: for instance, the Runge-Kutta-Fehlberg RKF78 scheme is significantly more efficient than the Runge Kutta RK45 scheme due to its higher order. However, it does not feature an exact interpolation routine, while the RK45 scheme does [61].

Propagation algorithms differ in many ways, and a general and complete characterization of the currently adopted techniques in the astrodynamics field is beyond the scopes of this dissertation. Among the main features worth mentioning for the proposed application, the choice of single-step or multi-step routines is the first aspect that can be considered. For instance, JPL uses an Adams-Bashforth scheme for the generation of planetary ephemeris data [2], whereas state-of-the-art software such as SNAPPshot [29] and CUDAjectory [57, 128] implement the aforementioned RKF78 method. Romano [125] explored the use of a wide class of schemes for interplanetary N-body propagation and planetary protection applications, comparing both implicit and explicit routines, fixed-step and variable-step methods, and also exploring the use of Symplectic integrators: these schemes take advantage of the Hamiltonian structure of the Cartesian formulation of the orbital dynamics, and enforce the conservation of the total orbital energy preventing its secular drift due to truncation errors. Romano found that these schemes become particularly effective in the case of interplanetary, distant propagation, losing however their advantage in case of flybys [125]. In conclusion, the broad analysis proposed by Romano demonstrated that the choice of the numerical simulation scheme does not affect the overall accuracy of the MC planetary protection analysis, as the impact of the initial uncertainty widely outclasses the different error accumulated by the different algorithms.

This chapter explores the use of a different numerical simulation concept for interplanetary simulations, that does not rely on a forward

step taken during the propagation, rather starting from an initial full trajectory guess that is refined through an iteration process. Provided a sufficiently high simulation accuracy, the choice of numerical scheme does not affect the results of a PP/SDM analysis, nevertheless significant improvements in terms of efficiency may still arise by a more optimal use of computational resources. In this sense, this chapter studies the performance of the PC (PC) method [119] for interplanetary simulations, and proposes a few applications and extensions where this technique may excel. An introduction to the mathematical formulation of this method is given in Section 4.1, followed by the theoretical foundations of an interplanetary resonant trajectory optimization approach, that relies on the fixed-point nature of the PC iterations to minimize the cost associated to reading ephemeris data, in Section 4.2. Section 4.3 then extends the numerical scheme to the integration of augmented systems, aiming at maximizing its computational performance and making the scheme more suitable to massively parallel computations. Finally, application test cases covering the concepts of the chapter are presented in Section 4.4.

#### 4.1 PC NUMERICAL SCHEME

Picard iterations [61] are an analytical technique that can be used to obtain an approximation of the solution of initial/boundary value problems. Denoting the state of dimension  $n$  with  $\mathbf{x}$ , the independent variable with  $t$ , the initial/boundary condition with  $\mathbf{x}_0$  and the dynamics function with  $\mathbf{f}(\mathbf{x}, t)$ , the problem is defined as:

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}, t), \quad \mathbf{x}_0 = \mathbf{x}(t_0) \quad (4.1)$$

Starting from an initial approximation  $\mathbf{x}^{(0)}(t)$  of the actual solution  $\mathbf{x}(t)$  in the interval  $[t_0, t]$  of the initial/boundary value problem presented in Equation (4.1), the  $i$ -th Picard iteration improves the previous approximation  $\mathbf{x}^{(i-1)}(t)$  of  $\mathbf{x}(t)$  with  $\mathbf{x}^{(i)}(t)$  as in [61]:

$$\mathbf{x}^{(i)}(t) = \mathbf{x}^{(0)}(t) + \int_{t_0}^t \mathbf{f}(\mathbf{x}^{(i-1)}(s), s) ds \quad (4.2)$$

The method converges for a good enough initial approximation  $\mathbf{x}^{(0)}(t)$  and for  $i \rightarrow +\infty$  [61].

In the analytical Picard iteration context, performing more than one iteration is in general hard. The increasingly complex expressions for  $\mathbf{x}^{(i)}(t)$  make it difficult to retrieve closed form solutions after the first 2-3 steps [8]. At the same time, numerically computing the integral functions by quadrature might not suffice in accuracy, as only the first few iterations in general improve the function approximation. In the attempt to develop parallelizable routines for the integration of the dynamical motion, the PC method was built combining the Picard



iterations with the Chebyshev polynomial approximation [119]. A possible derivation of the method that follows the work of Fukushima [48] can be summarized in three steps:

1. Select a good enough initial guess  $\mathbf{x}^{(0)}(t)$ .
2. Approximate  $\mathbf{f}(\mathbf{x}, t)$  and  $\mathbf{x}^{(0)}(t)$  with their Chebyshev polynomial expansion.
3. Perform a Picard iteration to update the coefficients of the interpolating Chebyshev polynomials.

The Picard iterations halt when the stopping conditions are met, based on the maximum difference between two consecutive iterations dropping below some user-specified tolerance.

The so defined method allows to easily perform several more Picard iterations than the analytical case. The involved expressions remain always of the same type, i.e. Chebyshev polynomials. The function approximation becomes an interpolation through nodes that should be close to the true trajectory, instead of a global function whose value after the iterations still depends on the initial guess choice. Furthermore, few iterations suffice to drop below a low tolerance if the real solution  $\mathbf{x}(t)$  differs from the initial guess  $\mathbf{x}^{(0)}(t)$  only because of small perturbations [61]. Starting from the unperturbed Keplerian solution for the generic weakly perturbed two body problem, a relatively fast convergence of the method is ensured [48]. In the context of orbital simulations, Macomber [93] referred to this type of initial guess as warm-starting the PC iteration method, because the analytical solution of the dominant dynamics part is used to reduce the number of iterations required. Differently, "cold start" has been defined by simply setting all the trajectory samples as equal to the initial condition. In general, the closer the initial guess to the true trajectory, the lower the number of iterations will be. Semi-analytical initial guesses or results of propagations from simpler models are also an option, and in the case of three-body-like perturbed trajectories would be a better choice compared to the Keplerian approximation. Macomber also introduced the concept of hot start in the case of time spans covering multiple Earth planetary orbits [93], where the first orbit was used to compute the difference between the Keplerian guess and the converged trajectory. The near-periodicity of the spherical harmonics perturbation was then exploited, including this difference in the starting trajectory, achieving a further reduction of the iterations required for convergence.

#### 4.1.1 *Matrix form for vectorized and parallel computation*

The suitability of the method for parallel and vector implementation has been studied by Shaver [133] and Fukushima [49], in particular

for the evaluation of the dynamics function and the execution of the matrix multiplications. More recent works over this technique by Bai and Junkins developed the modified PC method [9] and a CUDA<sup>®</sup> implementation for NVIDIA GPUs [8]. For compactness and to better highlight the parallelization possibilities, the method is presented following the matrix notation introduced by Koblick et al [79].

For  $N$  Chebyshev nodes and the integration interval  $[t_0, t_{N-1}]$ , the independent variable  $t$  is sampled for  $j = 0, 1, \dots, N - 1$  up-front as

$$t_j = \omega_2 \tau_j + \omega_1 \quad (4.3)$$

with

$$\tau_j = -\cos\left(\frac{j\pi}{N-1}\right), \quad \omega_1 = \frac{t_{N-1} + t_0}{2}, \quad \omega_2 = \frac{t_{N-1} - t_0}{2} \quad (4.4)$$

Given the  $n$ -dimensional sampled states  $\mathbf{y}^{(i-1)}(t_j) = \mathbf{y}_j^{(i-1)}$ ,  $j = 0, \dots, N$  as a matrix  $\mathbf{y}^{(i-1)}$  of dimension  $N \times n$  computed at the Picard iteration  $i - 1$ , the whole process can be summarized in three sequential steps to obtain the states at the iteration  $i$ . The first one collects the evaluations of the dynamics function  $\mathbf{f}$  in the  $N \times n$  force matrix  $\mathbf{F}$  [79]:

$$\mathbf{F}_{j+1}^{(i)} = \omega_2 \mathbf{f}(\mathbf{y}_j^{(i-1)}, t_j), \quad j = 0, \dots, N - 1 \quad (4.5)$$

Secondly, identifying with  $\mathbf{A}$ ,  $\mathbf{C}$ ,  $\mathbf{S}$  the method's constant matrices whose definition can be found in [79], the  $N \times n$  matrix  $\mathbf{B}$  is obtained by rows as

$$\mathbf{B}_1 = \mathbf{S}\mathbf{A}\mathbf{F} + 2\mathbf{y}_0, \quad \mathbf{B}_j = \mathbf{A}\mathbf{F}, \quad j = 2, \dots, N \quad (4.6)$$

Third and last, the  $N \times n$  matrix of the state guesses  $\mathbf{y}^{(i)}$  for the  $i$ -th Picard iteration is

$$\mathbf{y}^{(i)} = \mathbf{C}\mathbf{B} \quad (4.7)$$

The iteration process stops when the maximum state difference between two consecutive Picard iterations  $\mathbf{y}^{(i)}$  and  $\mathbf{y}^{(i-1)}$  drops below a specified relative or absolute tolerance, upon user's choice.

Despite the proved theoretical convergence, large integration spans may lead to numerical instabilities, due to the cumulation of round-off errors even with large  $N$ , as multiple orbital revolutions take place [8, 9, 48]. Fukushima [48] suggests a piece-wise approach as a workaround, which has been implemented in the application of Section 4.4.1.2 and uses the modified PC method to integrate orbit by orbit in sequence, until the end of the span.

The core steps of the proposed algorithm follow the presented scheme [8, 9], together with the automatic generation of the Keplerian initial guess spanning one nominal orbital period.

*The proposed implementation automatically handles either forward or backward integration*

#### 4.1.2 *Latest developments of the method*

The modified PC method has been continuously developed in the past few years, both in its formulation and implementation sides, and outlining possible applications for Earth orbits where it contributed to increase the efficiency of the numerical analyses. Junkins et al [76] analyzed the performances of the method comparing the efficiency against the Runge-Kutta-Nystrom 12(10) integrator, proposing also a second order PC version. Later, Koblick and Shankar [80] extended the analysis to the propagation of accurate orbits testing difference force models with NASA's Java Astrodynamics toolkit. Woollands et al. [165–167] applied the method as numerical integrator for the solution of the Lambert two-point boundary value problem, also assessing the benefits of adopting the KS formulation of the dynamics, and proposing a solution for the multi-revolution trajectory design. Swenson et al. [141] applied the modified PC method on the circular restricted three-body problem, using the differential correction approach. Singh et al. [137] used the method as the numerical integration scheme for their feasibility study on quasi-frozen, near polar and low altitude lunar orbits, including the N-bodies and the spherical harmonics perturbations. The fixed point nature of the method was exploited by Koblick et al. [81] to design low-thrust trajectories as an optimal control problem, discretizing the control impulses and also included the Earth's oblateness  $J_2$  perturbation. Macomber et al. [92] introduced the concepts of cold, warm, hot starts of the method, addressing possible efficiency improvements by means of better initial conditions, and variable-precision force models taking advantage of the fixed-point nature of the algorithm. Woollands et al. [164] extended the optimal low-thrust design to a high-fidelity model for the non-spherical Earth, considering an arbitrary number of spherical harmonics in the perturbing acceleration. Woollands and Junkins [163] developed the Adaptive PC method, including an integral error feedback that accelerates the convergence of the Picard iterations and an empirical law to determine segment length and polynomial degree of the method, based on previous stability analyses. Atallah et al. [7] compared the method with other sequential integration techniques on different Earth-based orbital cases.

## 4.2 FIXED-POINT HIGH-PRECISION RESONANT FLYBY OPTIMISATION

Orbital resonances have been exploited in several ways for mission design purposes and in many different contexts, such as the Earth-Moon case (for example in the works of Topputo et al. [143], to reach the moon with low fuel consumption, of Ceriotti et al. [27] to increase the coverage of polar orbits, and Short et al. [134], as the scientific orbit

of the Transiting Exoplanet Survey Satellite mission) or the exploration of Jupiter's and Saturn's moon systems (for example the works of Lantoine et al. [86], Campagnola et al. [21, 22] and Vaquero et al. [150]). The planned introduction of the Lunar Gateway in 2024 has drawn the attention of more recent works on the cis-lunar space. The 9:2 resonant Near Rectilinear Halo Orbits are extensively analyzed by Zimovan et al. [171], both as possible candidates for the hosting the Gateway, and in terms of the transfer possibilities toward other cis-lunar orbits by McGuire et al., with and without the aid of low-thrust propulsion [99]. Singh et al. [136] investigate eclipse-aware low-thrust transfer strategies to such orbits, proposing a method whose concept resembles the one of this section, leveraging the perturbation effects through the use of high-fidelity analogues of the invariant manifolds of the Circular Restricted Three Body Problem. Other applications also regard pure interplanetary orbits, for instance the ESA/NASA mission Solar Orbiter [41] as the latest example: resonant trajectories with Venus are exploited to raise the orbital inclination up to almost 30 degrees [43] over the ecliptic, to better observe the near-polar regions of the Sun.

In this last case, the use of resonant close encounters allows to save a considerable amount of fuel because of the repeated sequential flyby maneuvers. Nonetheless, such a phenomenon remains difficult to accurately model and understand, especially at the boundaries of the planet's SOI where none of the two dynamics, planetary or interplanetary, has a dominant role. This effect is amplified for shallow encounters, where either the small relative velocity with respect to the flyby planet or the high miss distance worsen the patched conics approximation. However, accurate predictions are required for steep close approaches too: a small deviation from the nominal condition may be amplified by several orders of magnitude during the flyby, requiring trajectory correction maneuvers.

In this section, the basic formulation of the modified PC integration method [9] is combined with the b-plane flyby prediction capabilities and applied to the design of multi-flyby trajectories in reverse cascade. The exit requirements of the current flyby are computed to meet the entrance condition of the next one. Consequently, the back-integration of the optimized exit state yields a new entrance condition to be targeted, within a dynamic programming-like backward recursion logic. The proposed method extends an unperturbed version of the design algorithm [95], that exploits the b-plane formalism to design a series of two body resonant orbits in the patched conics case. The here proposed strategy uses the unperturbed b-plane solution to prune the trajectory design in the perturbed environment. Starting from the Keplerian initial guesses for the patched conics interplanetary arcs, a continuity link between the planetary and interplanetary legs is introduced at the boundaries of the planet's SOI. The core of the

presented approach numerically integrates the full dynamics using the PC method, embedded in a multi-layer optimization problem that minimizes an artificial correction at a user-specified point in the interplanetary cruise. This application also tests the PC integration techniques to interplanetary orbits, where the fixed point nature of the algorithm introduces further benefits compared to the sole Earth case. In particular, the numerical propagation scheme is used to remove the patched conics approximation, and to surf the complex perturbing accelerations from the N-bodies and general relativity. In summary, the b-plane formalism is used for both the preliminary design of the patched-conics initial trajectory guess and for the description of the optimization variables. The PC integration scheme is then used at the core of the optimization, exploiting the fixed point nature for increased computational performance when including the effects of N-bodies and general relativity perturbations.

#### 4.2.1 *B-plane search of optimal resonant flyby exit state*

The b-plane formalism presents an analytic theory for the characterization of flybys, based on a manipulation of Öpik's variables [69] originally proposed by Carusi et al. [25], and further developed by Valsecchi et al. [147–149]. Fixed values of the post-encounter semi-major axis are represented as circles in the b-plane, which can therefore be targeted a priori as the link with the orbital period is well known [148].

##### 4.2.1.1 *Close encounters in the b-plane*

Assuming the planet in a circular orbit around the Sun, an intermediate frame needs to be defined for the b-plane flyby representation. Such a frame was first introduced in the framework of Öpik's theory [69] by Greenberg et al. [59] and later used by Carusi et al. [25] for the characterization of close encounters, aiming at finding analytical expressions for post-flyby orbital parameters. Considering a frame centered on the planet's center of mass, the  $(x, y, z)$  axes are directed as the heliocentric position, velocity  $\mathbf{v}_p$  and angular momentum of the planet, respectively, as shown in Figure 4.1.  $\mathbf{U}$  and  $\mathbf{U}'$  denote the pre-encounter and post-encounter planetocentric velocities, respectively.

All the involved quantities are non-dimensional, such that the planet's distance from the Sun and the Sun's gravitational parameter are both equal to 1. The non-dimensionalization gives in turn  $|\mathbf{v}_p| = 1$  and makes the orbital period of the planet equal to  $2\pi$ . The angles  $\varphi$ ,  $\varphi'$  and  $\chi$  appear in the works of Carusi and Valsecchi [25, 148] for other analyses, whereas are not necessary for the purposes of the presented design algorithm.  $\gamma$  identifies the flyby turn angle,  $\theta$  and  $\theta'$  the pre and post encounter angles between the corresponding

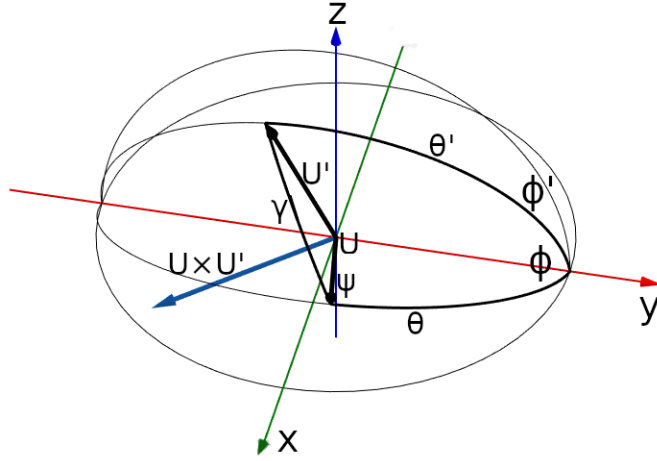


Figure 4.1: Graphical representation of the reference frame of analysis. Picture re-drawn based on original from [25].

planetocentric velocity  $\mathbf{U}$  or  $\mathbf{U}'$  and the planet's velocity  $\mathbf{v}_p$ , and  $\psi$  identifies the direction for the rotation of  $\mathbf{U}$  into  $\mathbf{U}'$  caused by the flyby, measured counter-clockwise from the major circles identified by  $\mathbf{U}$  and  $\mathbf{v}_p$ .

The flyby effect, interplanetary-wise in any patched conics approximation, is modeled as an instantaneous rotation of the planetocentric velocity vector  $\mathbf{U}$  without magnitude change. With the above defined quantities the  $b$ -plane reference frame can be introduced, whose axes  $(\hat{\xi}, \hat{\eta}, \hat{\zeta})$  are defined as by Öpik [69]:

$$\hat{\eta} = \frac{\mathbf{U}}{|\mathbf{U}|}; \quad \hat{\xi} = \frac{\mathbf{v}_p \times \mathbf{U}}{|\mathbf{U}| |\mathbf{v}_p|}; \quad \hat{\zeta} = \hat{\xi} \times \hat{\eta}. \quad (4.8)$$

In the following lines, the definition  $b$ -plane will be used to identify the plane perpendicular to the  $\hat{\eta}$  axis, because

$$\xi^2 + \zeta^2 = b^2 \quad (4.9)$$

with  $b$  the impact parameter as in Milani et al. [108].

Recalling [25], from an interplanetary point of view the flyby can be modeled as an instantaneous rotation of  $\mathbf{U}$  into  $\mathbf{U}'$ . The superscript  $'$  is used to denote the post-encounter quantities.

#### 4.2.1.2 $B$ -plane circles

A certain post-encounter semi-major axis  $a'$  is fully determined by  $\theta'$  [148]:

$$\cos \theta' = \frac{1 - 1/a' - U^2}{2U} \quad (4.10)$$

From the  $b$ -plane properties and some spherical geometry analysis, the  $b$ -plane locus of points of a given post-encounter semi-major axis  $a'$  is a circle centered on the  $\hat{\zeta}$  axis [148]:

$$\xi^2 + \zeta^2 - \frac{2c \sin \theta}{\cos \theta' - \cos \theta} \zeta + \frac{c^2 (\cos \theta' + \cos \theta)}{\cos \theta' - \cos \theta} = 0 \quad (4.11)$$

which is equivalent to

$$\xi^2 + \zeta^2 - 2D\zeta + D^2 = R^2 \quad (4.12)$$

with the center's  $\zeta$  coordinate  $D$  and the radius  $R$  explicitly defined as

$$D = \frac{c \sin \theta}{\cos \theta' - \cos \theta} \quad R = \left| \frac{c \sin \theta'}{\cos \theta' - \cos \theta} \right| \quad (4.13)$$

where, analogously to  $\theta'$ ,  $\theta$  is the angle between  $\mathbf{U}$  and  $\mathbf{v}_p$ , and  $c = \mu_p/|\mathbf{U}|^2$ . As already mentioned, any reachable post-encounter semi-major axis can be drawn as a circle in the b-plane, and need not be resonant. The sole exception are flybys that do not modify the value of  $\alpha$ , and thus feature  $\theta \equiv \theta'$ , which are defined as the straight horizontal line [148]:

$$\zeta = \cot \theta \quad (4.14)$$

#### 4.2.1.3 Perturbations in the b-plane

Previous results led to the semi-analytical definition of the b-plane circles arising from the effects of a generic perturbation source [95]. All the perturbing effects can be condensed in three angular variations:

- of the turn angle  $\gamma$ ,  $\Delta\gamma$ ;
- of the angle  $\psi$  that identifies the direction of the rotation of  $\mathbf{U}$  into  $\mathbf{U}'$ ,  $\Delta\psi$ ;
- of the post-encounter angle  $\theta'$ ,  $\Delta\theta'$ .

Figures 4.2a and 4.2b compare the resonant circles drawn with the unperturbed theory (Figure 4.2a on the left) and the new perturbed model (Figure 4.2b on the right) with the simulated resonant samples, highlighted in yellow, coming from the PP analysis of the upper stage of the launcher of Solar Orbiter [43, 126]<sup>1</sup>. The b-plane circles, on purpose nearly visible and drawn in light grey, have become the black bounded belt shaped loci of points, because also almost perfectly phased resonant returns have been considered extending each circle over its own neighbourhood.

In the case of Figures 4.2a and 4.2b the angles  $\Delta\gamma$ ,  $\Delta\psi$  and  $\Delta\theta'$  remain small in magnitude, nevertheless the difference they make in the characterization of the b-plane circles is significant. This gives a further proof to the need of precise models for the flyby phase, which is required if the desired post-encounter prediction must be accurate.

<sup>1</sup> More detailed information about this analysis and the related validation can be found in the work of Colombo et al. [29], Colombo et al. [30] and Masat [95].



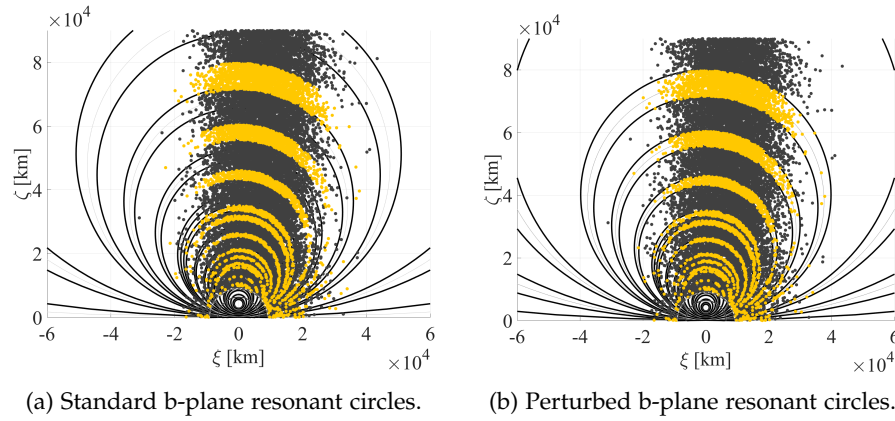


Figure 4.2: Visual accuracy improvement of the b-plane circle model. The analytical belts are bounded by the black circles, the yellow dots highlight the numerically detected resonances on the whole simulated cloud of initial conditions.

#### 4.2.2 *B-plane for backward-recursive flyby design*

Significant trajectory deflections can be achieved using flybys, however such an amplifying effect requires a high precision measure of the entrance state to the planet's SOI. In fact, it is well known that even small errors on the entrance conditions can lead to completely undesired exit states, which might be disastrous for the forthcoming mission phases. This issue can be mitigated increasing the precision of the models used to simulate the trajectories, nonetheless the high computational complexity of some perturbation effects hinders their practical use for the mission analysis. Among those, other than their computational burden, complex gravitational fields generated by the N-body environment build an overall chaotic dynamical system. This makes it extremely difficult to search for solutions similar to each other, since such systems are characterized by diverging trajectories, even for small differences on the initial conditions. This work introduces an efficient computational framework to account for such perturbing effects, taking also advantage of the chaotic force environment to minimize the artificial trajectory correction maneuvers. Being the goal the development of the design technique itself to exploit chaotic perturbations, without focusing on the particular test-case trajectory, solar radiation pressure effects are neglected. On top of the Newtonian gravitational effects, general relativity contributions are included as well, to highlight that even perturbations with the most complex physics can be exploited by the proposed setup. General relativity effects have been implemented as in [30, 94], based on the post-Newtonian model of the Einstein-Infeld-Hoffmann equations as presented by Seidelmann [129]. The same set of equations is used by JPL for the generation of



ephemerides data [2], which are also used in this work to fetch the state of the N-bodies at each sampling time  $t_j$ .

The b-plane theory is used to prune the optimization of a given multi-flyby trajectory. Knowledge of desired macro-properties are assumed to be known, such as semi-major axis, eccentricity, inclination and flyby planets and times, the overall algorithm can be summarized in two steps:

1. Obtaining the unperturbed patched-conics solution using the b-plane theory, as explained in Section 4.2.2.1, for the interplanetary orbits and the planetocentric details of all of the possibly multiple flybys.
2. Making the solution continuous in time and space, accounting for perturbing effects and exploiting them to minimize the corrections required to enter subsequent flybys.

The presented steps are explained in more detail in the following sections.

#### 4.2.2.1 Patched conics b-plane solution for resonant orbits

Valsecchi et al. [147, 149] found an analytical solution for the computation of the post-encounter orbital parameters for a given b-plane point at the entrance of the SOI. They successfully identify fixed values of eccentricity and inclination that conserve the Tisserand parameter, for each point belonging to a fixed semi-major axis circle. Although analytical, the relationship is unfortunately given as a full algorithmic procedure made of highly non-linear equations: this makes it difficult to build the inverse relation, i.e. to retrieve the b-plane entrance to the SOI given the full set of post-encounter orbital parameters, even in a numerical or optimization context as convexity cannot be in general ensured.

An alternative approach was developed in [95], defining an efficient optimization problem that uses the spherical geometry relations that generate the b-plane circles.

Specifically mentioning to the case of resonances, another optimization layer was developed [95]: find a set of intermediate resonant trajectories to gradually move from an initial interplanetary orbit to a final one, which is not reachable with a single flyby, for a fixed number of intermediate flybys. A set of intermediate tentative  $\Delta\mathbf{v}$  targets is defined, which the algorithm tries to match while preserving the resonance condition. A block-scheme diagram of the unperturbed design algorithm is given in Figure 4.3.

In the unperturbed and patched conics context, 1-2 seconds only [95] are required by a MATLAB<sup>®</sup> implementation of this approach to design a set of resonant orbits with Venus, which are already very close to the actual optimized mission profile from Solar Orbiter's mission redbook [43].

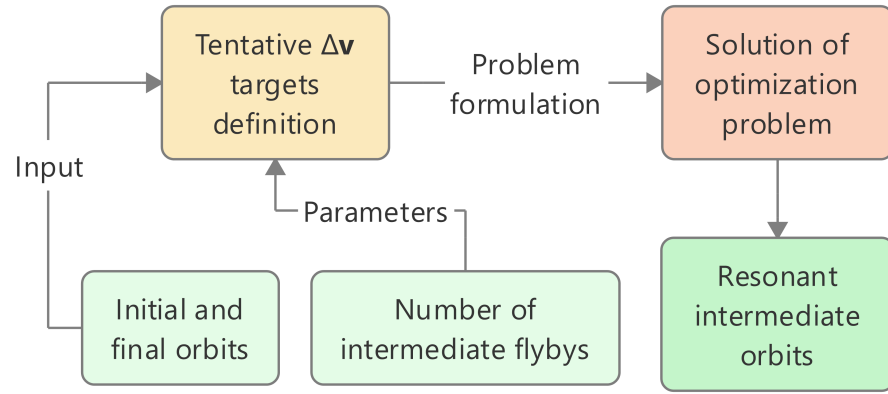


Figure 4.3: Block-scheme diagram of the unperturbed design algorithm developed in [95].

As shown in Section 4.2.1.3, when accounting for perturbing effects in the b-plane, a modification of the circles is inevitably introduced, as already shown in Figures 4.2a and 4.2b. Nevertheless, the  $\Delta\mathbf{v}$  variation due to perturbing effects is much smaller than the difference between the two set of circles, in relative terms [95].

#### 4.2.2.2 PC optimisation with perturbations

A recursive strategy for multi-flyby design can be built connecting the presented concepts presented. The whole multi-flyby problem is broken down to a discrete set of orbital arcs, each being covered between two gravity assist maneuvers. When placed in a backward design, the proposed algorithm tries to give an optimal solution to the following question: how should flyby  $j$  occur, so that flyby  $j + 1$  happens according to some already specified features and accounting for any perturbing effect?

The b-plane design strategy [95] provides a unique entrance (and thus exit) to the SOI in the patched conics approximation. All that remains to do, conceptually, is to properly provide the interface conditions between the two legs, accounting for all the possible perturbations sources and replacing the zero/infinity link with a continuity relationship. In the following lines the subscripts in and out shall denote the specific points of entrance and exit to/from the SOI for the current flyby.

Consider the entrance conditions to flyby  $j + 1$ , happening at the time  $t_{in}^{(j+1)}$ , as the Sun-centric position  $\mathbf{r}_{in}^{(j+1)}$  and velocity  $\mathbf{v}_{in}^{(j+1)}$ , already fulfilling the mission requirements for  $t > t_{in}^{(j+1)}$  together with possible future manoeuvres already defined. Consider also a deep space correction maneuver happening at the time  $\tilde{t} > t_{out}^{(j)}$ . The whole entrance condition  $(t_{in}^{(j+1)}, \mathbf{r}_{in}^{(j+1)}, \mathbf{v}_{in}^{(j+1)})$  is back-integrated in the perturbed environment with the modified PC method to the time  $\tilde{t} < t_{in}^{(j+1)}$ , obtaining the connection state  $(\tilde{\mathbf{r}}, \tilde{\mathbf{v}})$ .

Assume that the unperturbed solution for flyby  $j$  is expressed in the b-plane formalism, which can also mean a manipulation of the solution of the Lambert problem [97] with the related planetocentric phase and not necessarily from the already mentioned b-plane algorithm [95], particularly as:

- the outgoing time  $t^{(j)}$ ;
- the outgoing b-plane coordinates  $(\xi, \zeta)$ ;
- the outgoing planetocentric asymptotic velocity  $\mathbf{U}'$ .

Based on this, the time spent in the flyby phase  $\delta t^{(j)}$  can be estimated with the time law for the hyperbolic motion<sup>2</sup>, forcing the remaining b-plane coordinate  $\eta$  such that the distance from the flyby planet equals the radius of the SOI. In turn,  $\delta t^{(j)}$  can be used to get another estimate, that is the actual exit from the SOI  $t_{\text{out}}^{(j)} = t^{(j)} + \delta t^{(j)}/2$ . The time  $t_{\text{out}}^{(j)}$  is actually the outer optimization variable of the proposed algorithm. Intuitively, the time estimate arising from  $t^{(j)}$  and  $\delta t^{(j)}$  might not be the best possible time when to abandon the SOI starting the phase toward flyby  $j + 1$  and performing the minimum cost correction maneuver at  $\tilde{t}$ , especially because of perturbing effects acting on the way. The claim that is made treats  $t_{\text{out}}^{(j)}$  as a very good starting guess for an outer optimization layer, using a "perturbation"  $\Delta t^{(j)}$  of the exit time as optimization variable and bounding the search to a relatively small domain. A similar reasoning is made for the b-plane coordinates  $(\xi, \zeta)$  and the outgoing planetocentric velocity  $\mathbf{U}'$ , considering the unperturbed solution as initial optimization guess and searching over small variations thereof. Theoretical support comes in this case from the results of the perturbed b-plane circles: the relatively small difference between the selected points in the perturbed and unperturbed cases suggests to use the variations of the b-plane coordinates  $(\Delta\xi, \Delta\zeta)$  as two optimization variables and to bound them again in a relatively small search space. The set of optimization variables is completed with  $\Delta\mathbf{U}'$ , a variation of  $\mathbf{U}'$  bounded in a small domain as well. The use of the b-plane interface between flyby and interplanetary leg combined with the small bounded variation approach also has a more practical reason: despite working in a backward time recursion, a perturbed trajectory that minimizes the maneuver cost at  $\tilde{t}$  may in general excessively differ from the mission requirements. The b-plane intrinsically constrains the interface to be an actual flyby, furthermore the small and bounded search space should ensure a perturbed trajectory not too different from the desired profile for  $t < t_{\text{in}}^{(j)}$ . Given the initial values  $(t_{\text{out}}^{(j)}, \xi, \zeta, \mathbf{U}')$  and the generic variations  $(\Delta t^{(j)}, \Delta\xi, \Delta\zeta, \Delta\mathbf{U}')$ , the initial conditions  $(\mathbf{r}_{\text{out}}^{(j)}, \mathbf{v}_{\text{out}}^{(j)})$  at the time  $t_{\Delta}^{(j)} = t_{\text{out}}^{(j)} + \Delta t^{(j)}$  for the forward PC integration from  $t_{\Delta}^{(j)}$  to  $\tilde{t}$  are uniquely defined through the following steps:

<sup>2</sup> Not reported here. See for instance Vallado [97] for more details.

1. the flyby planet's state  $(\mathbf{r}_p^{(j)}, \mathbf{v}_p^{(j)})$  can be retrieved by reading the ephemeris database for the time  $t_\Delta^{(j)}$ ;
2. from  $(\xi + \Delta\xi, \zeta + \Delta\zeta)$  the third b-plane coordinate  $\eta$  is fixed by requiring the distance from the planet to equal the radius of the SOI;
3. the b-plane coordinates  $(\xi + \Delta\xi, \eta, \zeta + \Delta\zeta)$  can be converted to the planetocentric Cartesian coordinates  $\mathbf{r}_{pl}$ , because the axes of the b-plane reference frame are uniquely defined as in Equation (4.8) and the planetocentric velocity vector is  $\mathbf{U}' + \Delta\mathbf{U}'$ ;
4. The Sun-centric coordinates  $(\mathbf{r}_{out}^{(j)}, \mathbf{v}_{out}^{(j)})$  are retrieved by the simple summations  $\mathbf{r}_{out}^{(j)} = \mathbf{r}_{pl} + \mathbf{r}_p^{(j)}$  and  $\mathbf{v}_{out}^{(j)} = (\mathbf{U}' + \Delta\mathbf{U}') + \mathbf{v}_p^{(j)}$ .

The initial value problem identified by  $(\mathbf{r}_{out}^{(j)}, \mathbf{v}_{out}^{(j)})$  at the time  $t_\Delta^{(j)} = t_{out}^{(j)} + \Delta t^{(j)}$  is solved numerically forward in time with the modified PC method, to the connection maneuver at an arbitrary time  $\tilde{t}$ . Using a concise notation, the initial value problem to be numerically integrated will be identified by the dynamics functions  $\mathbf{r}_f(t)$  and  $\mathbf{v}_f(t)$ , with  $t_{out}^{(j)} \leq t \leq \tilde{t}$ , for position and velocity respectively, and setting the initial conditions:

$$t_0 = t_{out}^{(j)}, \quad \mathbf{r}(t_0) = \mathbf{r}_{out}^{(j)}, \quad \mathbf{v}(t_0) = \mathbf{v}_{out}^{(j)} \quad (4.15)$$

In general, the forward-integrated state  $(\mathbf{r}_f(\tilde{t}), \mathbf{v}_f(\tilde{t}))$  at the correction maneuver time  $\tilde{t}$  will differ from the back-integrated state that leads to flyby  $j + 1$  by

$$\widetilde{\Delta\mathbf{r}} = \mathbf{r}_f(\tilde{t}) - \tilde{\mathbf{r}} \neq \mathbf{0} \quad \text{and} \quad \widetilde{\Delta\mathbf{v}} = \mathbf{v}_f(\tilde{t}) - \tilde{\mathbf{v}} \neq \mathbf{0} \quad (4.16)$$

The physics of the correction maneuver performed at the maneuver time  $\tilde{t}$  embeds the mandatory constraint of the position where it is to happen, theoretically defined as  $\widetilde{\Delta\mathbf{r}} = \mathbf{0}$ . The motion is numerically integrated, hence leaving the maneuver position as a pure equality constraint might severely affect the computational performance of the optimization: a full PC integration would be required to evaluate the constraint function, since the optimization variables are nothing but the b-plane form of the initial state  $(\mathbf{r}_{out}^{(j)}, \mathbf{v}_{out}^{(j)})$  and the simulation to the connection time  $\tilde{t}$  would always be needed. At the opposite side, the actual maneuver to be designed may not have any physical sense if omitted, as the continuity requirement may be lost. Nonetheless, in a numerical context an absolutely negligible value of  $\widetilde{\Delta\mathbf{r}}$  suffices to satisfy the physical meaning of the correction maneuver. These observations have led to the choice of explicitly implementing the position constraint with a penalty method [60], that is penalizing the objective function (the correction  $|\widetilde{\Delta\mathbf{v}}|$  in this case) adding a large term direct function of the position difference  $\widetilde{\Delta\mathbf{r}}$ .

Therefore, defining  $J_v = |\widetilde{\Delta \mathbf{v}}|$ , omitting the explicit dependencies on the optimization variables for conciseness and denoting the components of  $\Delta \mathbf{U}'$  with  $\Delta \mathbf{U}'^{(1,2,3)}$ , the general maneuver design can be written as the following optimization problem:

$$\begin{aligned} & \underset{\Delta \xi, \Delta \zeta, \Delta \mathbf{U}'}{\text{minimize}} && J_v(\tilde{t}, t_{\Delta}^{(j)}) + \alpha J_r(\tilde{t}, t_{\Delta}^{(j)}) \\ & \text{subject to} && |\Delta \xi| \leq \Delta \xi_{\max}, \\ & && |\Delta \zeta| \leq \Delta \zeta_{\max}, \\ & && |\Delta \mathbf{U}'^{(1,2,3)}| \leq \Delta \mathbf{U}'_{\max} \end{aligned} \quad (4.17)$$

with  $J_r = |\widetilde{\Delta \mathbf{r}}|$  and the weighting factor  $\alpha$  of the penalty method sufficiently large. The choice of  $\alpha$  is in general arbitrary, it will be discussed in Section 4.4.2.2 for the presented test case.

A block-scheme diagram summarizing all the presented features and steps of the algorithmic optimization problem in Equation (4.17) is given in Figure 4.4. The initial unperturbed solution expressed in the b-plane formalism is converted to a Cartesian state and used to prune the optimization process. Subsequently, the fixed-point nature of the PC nature requires to sample an initial trajectory guess on fixed time nodes: as already mentioned, this feature is also exploited to perform the sampling of the N-bodies ephemerides data only once, not only for the PC integration but also for the whole optimization run. Finally, the "closed loop" that can be seen in Figure 4.4 is entered, where each objective function evaluation involves the PC forward integration of some coordinates, generated from the optimization variables expressed as b-plane variations. The exit conditions strongly depend on the chosen implementation, although any already existing scheme could be followed (for instance, relative state and objective function variations smaller than some user-defined tolerance in this work, as it is discussed in Section 4.4.2.2).

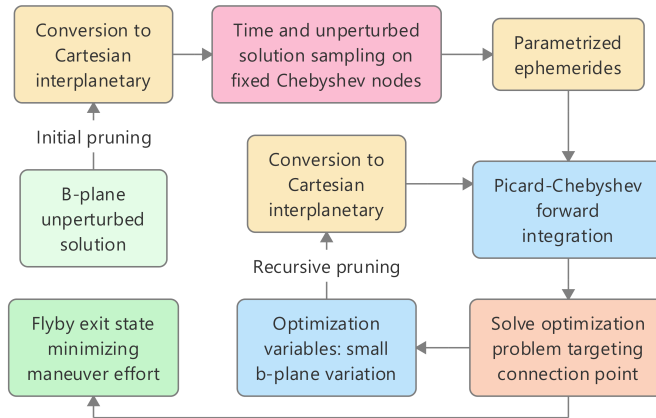


Figure 4.4: Block-scheme diagram of features and steps embedded in the solution of the optimization problem of Equation (4.17).

Finally, assuming the position constraint to be fulfilled in the optimization problem of Equation (4.17) whose result gives  $J^*(\tilde{t}, t_{\Delta}^{(j)}) = J_v^*(\tilde{t}, t_{\Delta}^{(j)}) + \alpha J_r^*(\tilde{t}, t_{\Delta}^{(j)})$ , for optimizing the flyby time it is enough to use its definition  $t_{\Delta}^{(j)} = t_{\text{out}}^{(j)} + \Delta t^{(j)}$ , with  $\Delta t^{(j)}$  the new optimization variable:

$$\begin{aligned} & \underset{\Delta t^{(j)}}{\text{minimize}} && J^*(\tilde{t}, t_{\text{out}}^{(j)} + \Delta t^{(j)}) \\ & \text{subject to} && |\Delta t^{(j)}| \leq \Delta t_{\text{max}} \end{aligned} \quad (4.18)$$

No choice has been made yet about the optimization algorithms, which might be sensitive to the search space size and the function relative steepness within the different regions. Moreover, it should also be tailored on the available computational resources, i.e. preferring parallelizable routines over dominantly sequential algorithms for high performance computing facilities.

The optimization problem of Equation (4.17) is a sub-problem of the optimization problem of Equation (4.18). This somehow enhances the flexibility of the approach, i.e. the former might be used for search space exploration purposes without the need of a finely refined solution in terms of starting time  $t_{\Delta}^{(j)}$ . The optimization problem of Equation (4.17) is explicitly dependent on the maneuvering point  $\tilde{t}$ , which in fact can and for practical applications should be optimized as well. In this application it remains a problem parameter, as more focus is put toward exploring the effect of small variations of the departure time  $t_{\Delta}^{(j)}$ . Completing the description, another optimization level can be easily defined to find the best  $\tilde{t}$  similarly to what done for  $t_{\Delta}^{(j)}$  in the optimization problem of Equation (4.18), and in the presented formalism it straightforwardly includes the innermost level defined by the optimization problem of Equation (4.17). A summary of the relations among the different optimization levels is given in Table 4.1.

Table 4.1: Optimization levels for the full design of the arc  $j$  to  $j + 1$ .

$J_v + \alpha J_r$ (Equation (4.17))	$\xrightarrow[\Delta \varepsilon, \Delta \zeta, \Delta U]{\text{Optimize}}$	$J^*$ (Equation (4.18))
$J^*$ (Equation (4.18))	$\xrightarrow[\Delta t^{(j)}]{\text{Optimize}}$	$j$ to $j + 1$ , maneuver at $\tilde{t}$
$j$ to $j + 1$ , maneuver at $\tilde{t}$	$\xrightarrow[\tilde{t}]{\text{Optimize}}$	$j$ to $j + 1$ , optimal

Some observations regarding the expected computational performances of the optimization can be made based on the analysis of the PC integration, already presented in Figures 4.10a, 4.10b, 4.10c, 4.10d for the accuracy behavior and Figure 4.11 for the computational time variation with increasing number of Chebyshev nodes. For sequential executions, the higher the value of  $\tilde{t}$  the higher the runtime will be, if the number of nodes per period is kept constant. The optimiza-

tion problem of Equation (4.17) is going to benefit from the minimal ephemerides overhead as a whole: the boundary times are fixed, hence the ephemerides data set can be scanned only once and the related values can be considered as parameters not only within the PC integration, but also for all the iterations of the optimization algorithm. Finally, the penalty approach [60] used to define the optimization problem of Equation (4.17) allows for massively parallel strategies to be implemented as well, because all the remaining constraints are of boundary type.

#### 4.3 PC AUGMENTATION FOR LARGE SETS OF INITIAL CONDITIONS

##### 4.3.1 One-level augmentation

Instead of the evolution of the sole trajectory determined by the initial condition  $\mathbf{y}_0$ , the system undergoing the PC integration can be rewritten so that  $M$  different trajectories sampled on the same  $N$  time nodes are processed sharing the iteration calls. At the iteration  $i$ , the matrix  $\mathbf{Y}^{(i)}$  can be defined, collecting all the samples of all the trajectories. The  $j$ -th row is related to the  $j$ -th time sample of the  $m$ -th trajectory by:

$$\mathbf{Y}_j^{(i)} = \begin{bmatrix} \mathbf{y}_{j,1}^{(i)} & \cdots & \mathbf{y}_{j,m}^{(i)} & \cdots & \mathbf{y}_{j,M}^{(i)} \end{bmatrix}, \quad j = 1, \dots, N \quad (4.19)$$

and similarly the dynamics function evaluations can be collected in the matrix  $\mathbf{F}^{(i)}$ , whose  $j$ -th row is:

$$\mathbf{F}_j^{(i)} = \begin{bmatrix} \mathbf{F}_{j,1}^{(i)} & \cdots & \mathbf{F}_{j,m}^{(i)} & \cdots & \mathbf{F}_{j,M}^{(i)} \end{bmatrix}, \quad j = 1, \dots, N \quad (4.20)$$

whose elements are still computed per sample:

$$\mathbf{F}_{j,m}^{(i)} = \omega_2 \mathbf{f}(\mathbf{y}_{j,m}^{(i-1)}, t_{j-1}), \quad j = 1, \dots, N \quad (4.21)$$

In principle, building the augmented system only requires to define  $\mathbf{Y}^{(i)}$  and  $\mathbf{F}^{(i)}$  by stacking the different  $M$  trajectory and dynamics matrices along the columns. The structure of the PC iterations remains unchanged, and features the usual steps. First, evaluate the dynamics function for all the  $N$  states of all the  $M$  trajectories with  $\mathbf{F}_{j,m}^{(i)} = \omega_2 \mathbf{f}(\mathbf{y}_{j,m}^{(i-1)}, t_{j-1})$ . Second, perform the matrix operations  $\mathbf{B}_1 = \mathbf{S}\mathbf{A}\mathbf{F} + 2\mathbf{Y}_0$  and  $\mathbf{B}_j = \mathbf{A}\mathbf{F}$ , for  $j = 2, \dots, N$ . Third and last, update the guesses for all the  $M$  trajectories with  $\mathbf{Y}^{(i)} = \mathbf{C}\mathbf{B}$ .



### 4.3.2 Two-level and multi-level augmentation

The stack-along-column rule can be applied again, this time collecting in one single matrix  $P$  groups of different  $M_p$  trajectories each. The augmented matrix  $\mathbf{Y}^{(i)}$  is now built as

$$\mathbf{Y}_j^{(i)} = \begin{bmatrix} \mathbf{Y}_{j,1}^{(i)} & \cdots & \mathbf{Y}_{j,p}^{(i)} & \cdots & \mathbf{Y}_{j,P}^{(i)} \end{bmatrix}, \quad j = 1, \dots, N \quad (4.22)$$

with

$$\mathbf{Y}_{j,p}^{(i)} = \begin{bmatrix} \mathbf{y}_{j,p,1}^{(i)} & \cdots & \mathbf{y}_{j,p,m}^{(i)} & \cdots & \mathbf{y}_{j,p,M_p}^{(i)} \end{bmatrix}, \quad j = 1, \dots, N \quad (4.23)$$

In principle, infinite augmentation levels could be built relying on the same logic, none of them would require modifications in the core PC algorithm structure. Nevertheless, a re-definition of the iteration error may be helpful for practical purposes, since the augmentation rationale is purely computational.

Two strategies can be addressed. The first uses a traditional error definition, that treats the trajectory samples as if they were part of a unique system, whose maximum will be compared against the iteration stopping condition. The second introduces a more flexible per-block error definition, that treats the different trajectory blocks as independent, for which the augmentation has then a sole computational purpose. Both the approaches have advantages and drawbacks. The former would allow a simpler implementation and is inevitably computationally more efficient than the latter, because of the reduced overhead compared to maintaining the group split. However, dissimilar trajectories requiring a significantly different number of iterations would keep the computational resources busy for already converged blocks, while the per-block definition allows far more flexibility on this regard.

### 4.3.3 CPU and GPU implementation paradigms

Among the implementations outlined in this section, the case of independent PC runs for all the propagated trajectories is considered as benchmark. This allows the direct assessment of the performance of the augmented PC algorithm, against the original integrator, for the same test case. All the algorithms have been implemented using the C language with OpenMP [28] parallelization, with the exception of the GPU program that was coded in CUDA<sup>®</sup> [112]. The reader can refer to Appendix C for an overview of parallel and GPU computing fundamentals, whose concepts are extensively used in this section.

The basic workflow of the independent PC runs is given in the block-scheme of Figure 4.5. The only parallelization possibilities, for high numbers of trajectories, apply at the highest level, inevitably



introducing a considerable overhead for both the inner sequential execution and the still parallelizable inner functions. In fact, all the per-trajectory steps of the PC process would still be parallelizable algorithms per se.

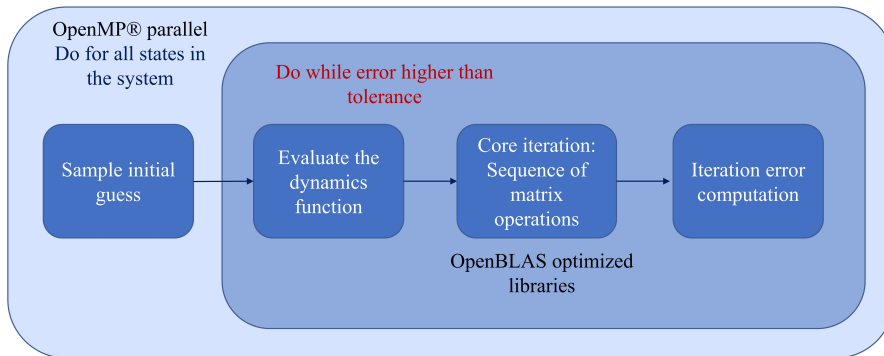


Figure 4.5: Standard PC workflow.

#### 4.3.4 Sequential Augmented PC workflow

The implementation of the augmented PC integration follows a one-level augmentation only, to highlight the pipeline benefits in terms of overhead that this implementation introduces. A block-scheme representation of the augmented PC integration workflow is given in Figure 4.6. The conceptual change, from the PC iteration viewpoint, is only the initial sampling, happening in a single array that contains all the state vectors of all the trajectories of the augmented system.

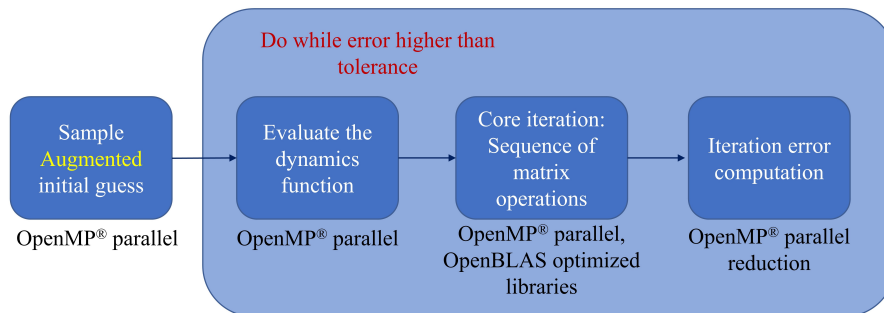


Figure 4.6: Augmented PC workflow.

#### 4.3.5 OpenMP parallelized Augmented PC workflow

The parallelization of the augmented system integration, whose block-scheme representation is given in Figure 4.7, becomes fine-grained, acting directly on the single state vectors for the dynamics function evaluation. In addition, reduction operations can be made through

OpenMP<sup>®</sup> for a cooperated and parallel search of the maximum error.

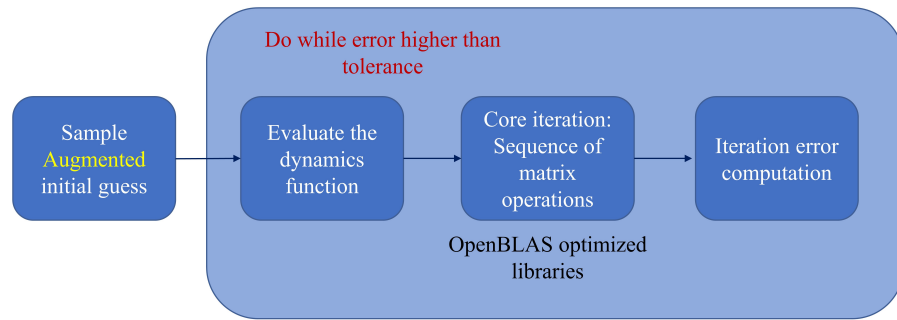


Figure 4.7: Augmented and OpenMP<sup>®</sup> parallelized PC workflow.

The parallel dynamics function is implemented with the OpenMP<sup>®</sup> parallelization of the for loop, collapsing all the states of the augmented system into a single loop. The multiple workers access the shared state array, then computing the acceleration values and temporarily storing them into thread-private variables, eventually copying them back to a shared and global acceleration array. OpenMP<sup>®</sup> exploits the flexibility of the CPU architecture, thus no significant modification are required to the innermost parts of the dynamics function to make an efficiently parallelized program. OpenMP<sup>®</sup> is also used "indirectly" for the matrix multiplications of the PC method. The optimized OpenBLAS [158, 168, 169] libraries are used to implement this part of the program, they already include the OpenMP parallelization.

#### 4.3.6 CUDA Augmented PC workflow

The block-scheme representation of the CUDA<sup>®</sup> algorithm is given in Figure 4.8. The two-level augmentation concept is exploited, assigning one higher level augmented system to each CUDA<sup>®</sup> stream and using the thread-based parallelism on the lower level augmented systems.

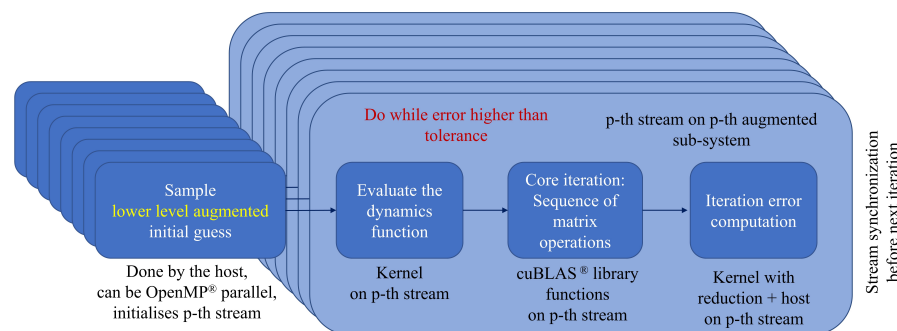


Figure 4.8: Augmented CUDA<sup>®</sup> PC workflow.

The principal benefit is the cooperation between CPU and GPU for the overall execution, with as many operations as possible executed

concurrently. Each lower level augmented system is initially sampled by the host and then moved to the GPU. The CUDA<sup>®</sup> stream management API allows to overlap the CPU sampling of the next higher level augmented systems with memory transfers and kernel executions of the already launched ones. Similarly, the last step of the PC iteration requires to retrieve the computed iteration error for each stream from the GPU to the CPU for loop control purposes, which is also subject to the stream concurrency benefits. A stream synchronization at the end of each while loop iteration is necessary to achieve the overlapping behaviour of all the streams. Running independent loops for each higher level augmented system would result in completely sequential and non-overlapped executions. If a single CUDA<sup>®</sup> stream is generated, the standard one-level augmented system case is reproduced, albeit with the GPU computing acceleration instead of the previously described OpenMP<sup>®</sup> implementation.

The warp-centric programming model of CUDA<sup>®</sup> kernels requires a small modification on the lower level augmented system definition. Contiguous array elements should be of the same component type (i.e. contiguous  $x$  coordinates, then contiguous  $y$  coordinates, and so on), instead of storing state vector by state vector. This aspect might seem an implementation detail, however it is fundamental to ensure coalesced global memory access. A too high latency would happen otherwise, which cannot be hidden even by intensive parallelized GPU computations. The just discussed modification has no effect on the overall algorithm structure, all it requires is the dynamics and error kernels to be implemented following this array element logic. This aspect is discussed in more detail in the following section, together with the implications it has especially on the evaluation of the dynamics function.

*This is true as long as the CPU memory is allocated as paged with specific CUDA<sup>®</sup> functions [112].*

#### 4.3.6.1 Dynamics model, array sorting, and CUDA kernel

At the core of the PC integration scheme lies the evaluation of the dynamics function at each Picard iteration. This task can be performed in parallel for all the states of the system being integrated, however, although conceptually simple, its implementation may not be straightforward in the GPU computing case. The more complex the dynamical model becomes, the more intertwined its implementation inevitably gets, possibly requiring to access data distributed in multiple arrays, possibly of notably different sizes. The accuracy requirements of the proposed application demand to work under the restricted relativistic N-body problem, following the Einstein-Infeld-Hoffmann equations [129], which has a dynamics function of the form:

$$\ddot{\mathbf{r}} = \mathbf{f}(\mathbf{r}, \dot{\mathbf{r}}, \mathbf{r}_i, \dot{\mathbf{r}}_i, \ddot{\mathbf{r}}_i) \quad (4.24)$$

with  $i$  condensing the dependence on the states of all the major bodies in the ephemeris model, e.g. the solar system planets, and  $\mathbf{r}$ ,  $\dot{\mathbf{r}}$ ,  $\ddot{\mathbf{r}}$

denoting position, velocity, and acceleration, respectively. The relations are unfortunately non-linear: as a consequence, each CUDA<sup>®</sup> thread cannot perform simple operations on one single array element, since both position and velocity of each state are required to compute any acceleration component. Moreover, ephemerides data for  $\mathbf{r}_i$ ,  $\dot{\mathbf{r}}_i$ ,  $\ddot{\mathbf{r}}_i$  also enter the dynamics function. These aspects suggest to implement the dynamics kernel having each CUDA<sup>®</sup> thread to process one full state vector, rather than one element. At the same time, coalescing the global memory access remains crucial to obtain a well-performing kernel. The PC method introduces however a partial constraint on the array shapes: the matrix multiplications that build the method need the sampled trajectory states to be stored as rows of an overall matrix, fixing the different times to identify each row. For a column-major sorted augmented state matrix, contiguous state elements are interrupted by the ending time nodes, likely leading to non-coalesced memory access for numerous warps. Row-major sorted state arrays feature instead non-coalesced access for all the state elements.

To cope with these issues, the lower-level augmented state is reformulated by stacking along the columns the same components of all the state vectors in the augmented system:

$$\mathbf{Y}_j^{(i)} = \begin{bmatrix} x_{j,1}^{(i)} & \cdots & x_{j,M}^{(i)} & \cdots & p_{j,1}^{(i)} & \cdots & p_{j,M}^{(i)} & \cdots & \dot{z}_{j,1}^{(i)} & \cdots & \dot{z}_{j,M}^{(i)} \end{bmatrix},$$

$$j = 1, \dots, N, \quad p = y, z, \dot{x}, \dot{y} \quad (4.25)$$

where  $(x, y, z)$  are the Cartesian components of  $\mathbf{r}$ . The advantage is obvious in the column-major sorting case, since all the common components are found in adjacent memory addresses. Since in the presented application the number of states in the augmented system is much larger than the number of sampled trajectory nodes, many contiguous state components also appear in the row-major sorted array case. In addition, the higher-level augmented system definition may remain unaltered, since different CUDA<sup>®</sup> streams would be processing each lower-level sub-systems. The augmented force matrix  $\mathbf{F}_j^{(i)}$  can be adapted accordingly, without introducing any modification to the matrix multiplication characterizing the PC iteration. Lastly, bank conflicts (explained in Section C.5) are automatically avoided [112] with this array sorting approach.

The kernel design is tied with the array sorting strategy: a key role is played by the fixed time nodes. The augmented system logic is a consequence of the shared time nodes among the different state vectors, this feature should also be exploited to design the thread blocks to make the most of the available shared memory. In particular, the amount of memory required to store the planetary ephemerides is minimized if all the threads in a block process state vectors corresponding to the same time node. For this reason, the proposed program implements a row-major sorting strategy of state and force matrices,

as shown in Listing 4.1. The augmented state matrix is accessed as a two-dimensional block array: one dimension (the rows) follows the different time nodes, whereas the other is used to split the states of a common time node into smaller chunks, each containing 32 states. In this way, all the states in the same block of the two-dimensional block array require exactly the same ephemerides data, because they are all associated to the same time node. For fixed-time thread blocks bank conflict is automatically avoided also when reading ephemerides data from the shared memory, since all the threads are forced to access the same ephemerides item or vector component [112]. As shown in the CUDA code of Listing 4.1, the proposed implementation highlights how low level the programmer should work, managing explicitly the memory location and accesses to achieve good performance results.

Listing 4.1: Summary of dynamics CUDA kernel.

```
// Dynamics kernel sample - Cartesian dynamics
__global__ void dynamics(double* dstate, double*
    state, void* other_params, int NN, int NO) {

    /*
        Retrieve the global index of this
        sample from the ids of this block
        and this thread
    */
    int idx = blockIdx.x * 32 + threadIdx.x;
    // Retrieve the time sample id
    int timeID = idx / NO;
    // Retrieve the state id in the augmented
    system
    int stateID = idx - NO * timeID;

    // Allocate the shared memory amount
    __shared__ double sbuf[288];

    // Define pointers to manage shared memory
    buffers
    double* accs = &sbuf[192];
    double* sstate = sbuf;

    // set initial acceleration values (in shared
    memory) to 0
    setZero(accs);

    // loop counter
    int i;

    // avoid accessing out-of-bounds memory
    accesses
    if (idx < NN * NO) {
```

*32 is the warp size for most NVIDIA<sup>®</sup> graphics cards [112]. For augmented systems with a number of states that is not a multiple of the block size, the last block of threads processes the remainder of the integer division between the number of states and the block size.*

```

// copy states from global to shared
// memory
// index2D and index3D are macros
// that access multi-dimensional
// array elements
#pragma unroll
for (i = 0; i < 6; i++){
    sstate[index2D(i, threadIdx.x
        , 32)] = state[index3D(
        timeID, i, stateID, 6, N0)
        ];
}
// copy velocity elements in output (
// dstate
#pragma unroll
for (i = 0; i < 3; i++){
    dstate[index3D(timeID, i,
        stateID, 6, N0)] = sstate[
        index2D(i + 3, threadIdx.x
        , 32)];
}

// call acceleration function
compute_acceleration(accs, sstate,
    other_params);
// copy acceleration from shared to
// global memory
for (i = 0; i < 3; i++){
    dstate[index3D(timeID, i + 3,
        stateID, 6, N0)] = accs[
        index2D(i, threadIdx.x,
        32)];
}
}
}
}

```

The computation of the dynamics function is a compute-bound task: most of the effort lies on performing computations on a limited amount of data, rather than on the movement of a large amount of information between two memory locations. Furthermore, the values of the state elements need to be repeatedly accessed. For these reasons and because of the limited capacities of thread-private registers, shared memory is used to also temporarily store the state values, because of its lower latency compared to the global GPU memory [112]. The resulting dynamics kernel is summarized in Figure 4.9.

In addition, a close look at the Einstein-Infeld-Hoffmann equations reveals that the acceleration of each propagated body depends not only on its state and the gravitational parameter and states of major bodies in the ephemeris model, but also explicitly on the gravitational potential and the acceleration of such bodies [129]. These contributions

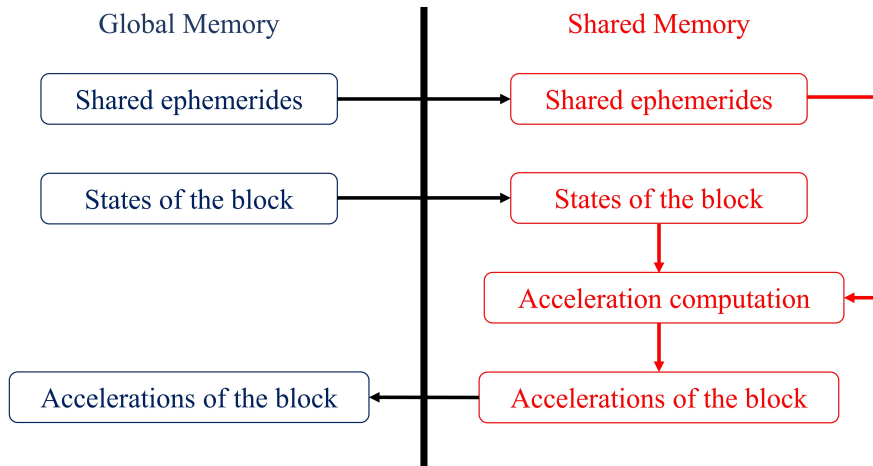


Figure 4.9: Dynamics kernel memory management.

can be computed before the evaluation of the dynamics function itself in the restricted problem of the proposed application, saving the computational burden of a task that would be repeated at each Picard iteration. The values of gravitational potential and the acceleration of the major bodies of the ephemeris model are computed by the CPU before starting the Picard iterations, then moved to the global GPU memory, and eventually loaded in the shared memory for the time node associated with each block, along with the ephemeris states and gravitational parameters.

#### 4.3.6.2 CPU-GPU cooperated iteration error computation

All the involved CUDA<sup>®</sup> kernels are run at each Picard iteration. Their execution must be called by the CPU, which also stops the Picard iteration while loop when the error between two consecutive steps falls below the desired tolerance. Since the updated states already reside on the GPU, the GPU massive parallelism can be exploited to accelerate the computation of the iteration error, transferring only a limited amount of data to the CPU to be used for the loop control. Despite dealing with the augmented state system, the error is still defined on a per-state basis, with the maximum of the errors of all the states that is used to control the loop.

The error computation process involves two separate kernels and a CPU function. The first kernel computes both the position and the velocity errors and stores the maximum between these two, for all the states in the augmented system. Then, as shown in Listing 4.2, the second kernel computes the maximum error of groups of 4096 states: 1024 thread-sized blocks are created, discerning the first maxima while reading four consecutive chunks of state errors into the shared memory. Consequently, 1024 threads cooperate to find the actual maximum error among the remaining 1024 state errors, with reduction-driven

parallelism<sup>3</sup>. Eventually, the maximum error is copied back to the global memory, in a new array consisting of reduced errors only. Finally, this whole array is copied back to the CPU, which finds the actual maximum with a traditional sequential for loop-based approach. Even with augmented states made of millions of state vectors, this approach makes the CPU search sequentially only over hundreds to thousands candidates at most, with a negligible computational cost compared to the other steps. Since reduction operations require the cooperation of all the threads in a block, synchronization barriers become necessary, and are set manually with the `__syncthreads()` function.

Listing 4.2: Summary of error reduction CUDA kernels and device functions.

```
// kernel for maximum error computation
__global__ void maxErr (double* newerrblocks, double*
    olderrblocks, int oldblocks) {

    int i = blockIdx.x * 4096 + threadIdx.x;

    // copy to shared memory - dynamic allocation
    double dum = 0.0;
    __shared__ double err_s[1024];

    // do first reduction while loading from
    // global to shared memory
    err_s[threadIdx.x] = 0.0;
    if (i < oldblocks)
    {err_s[threadIdx.x] = olderrblocks[i];}
    i += 1024;
    if (i < oldblocks){
        dum = olderrblocks[i];
        if (dum > err_s[threadIdx.x])
        {err_s[threadIdx.x] = dum;}
        i += 1024;
    }
    if (i < oldblocks){
        dum = olderrblocks[i];
        if (dum > err_s[threadIdx.x])
        {err_s[threadIdx.x] = dum;}
        i += 1024;
    }
    if (i < oldblocks){
        dum = olderrblocks[i];
        if (dum > err_s[threadIdx.x])
        {err_s[threadIdx.x] = dum;}
    }
}
}
```

<sup>3</sup> Like the dynamics kernel, the error kernels are implemented so that the last thread block processes the remainder between the integer division between the number of states and the block size (4096 in the case of the reduction kernel), with the `maxreduce` device function in Listing 4.2. The reduction steps are controlled accordingly: only those corresponding to a number of threads less than or equal to the number of states in the block are activated.



```

    }
    // synchronize threads before proceeding
    __syncthreads();

    // do reduction in shared memory
    maxreduce(err_s, threadIdx.x, blockDim.x);

    // the reduced element is in thread 0 -- copy
    // it back to global memory
    if (threadIdx.x == 0)
        {newerrblocks[blockIdx.x] = err_s[0];}
}

// device function for reduction computation in
// shared memory
__device__ void maxreduce(volatile double* sdata, int
    threadID, int blockSize) {

    if (threadID < 512) {
        if (sdata[threadID] < sdata[threadID + 512])
            {sdata[threadID] = sdata[threadID + 512];}
        __syncthreads();
        if (threadID < 256) {
            if (sdata[threadID] < sdata[threadID + 256])
                {sdata[threadID] = sdata[threadID + 256];}
            __syncthreads();
            if (threadID < 128) {
                if (sdata[threadID] < sdata[threadID +
                    128])
                    {sdata[threadID] = sdata[threadID + 128];}
                __syncthreads();
                if (threadID < 64) {
                    if (sdata[threadID] < sdata[threadID +
                        64])
                        {sdata[threadID] = sdata[threadID + 64];}
                    __syncthreads();
                    // finally do reduction for the single
                    // warp
                    if (threadID < 32)
                        {warpMaxReduce(sdata, threadID);}
                }
            }
        }
    }

}

// single warp max reduction
__device__ void warpMaxReduce(volatile double* sdata
    , int tid){
    if (sdata[tid] < sdata[tid + 32])
        {sdata[tid] = sdata[tid + 32];}
    if (sdata[tid] < sdata[tid + 16])

```

```

        {sdata[tid] = sdata[tid + 16];}
        if (sdata[tid] < sdata[tid + 8])
        {sdata[tid] = sdata[tid + 8];}
        if (sdata[tid] < sdata[tid + 4])
        {sdata[tid] = sdata[tid + 4];}
        if (sdata[tid] < sdata[tid + 2])
        {sdata[tid] = sdata[tid + 2];}
        if (sdata[tid] < sdata[tid + 1])
        {sdata[tid] = sdata[tid + 1];}
    }

```

#### 4.4 APPLICATIONS

##### 4.4.1 *Single trajectory propagation*

The SPICE toolkit [2] is used together with JPL's ephemerides data to retrieve the states of the N bodies at any integration step, required for the computation of both the Newtonian and the relativistic perturbations due to the Solar System bodies. This aspect is the most computationally expensive task in the general integration accounting for N-body effects, for instance making around 60% of the total account in the work by Colombo et al. [29]. In fact, a binary source must be scanned seeking for the closest saved samples, which must then be interpolated to fit the actual supplied time, for each step and for each of the bodies in the integration. Time steps cannot be foreseen with the standard integration methods, that continuously adapt the step size and sequentially move forward or backward from a given state, thus requiring repeated toolkit calls.

The fixed point nature of the modified PC method brings a significant advantage to this regard: the restricted N-body problem equation for a test particle written in barycentric Cartesian coordinates is

$$\ddot{\mathbf{r}}(t) = - \sum_{i=1}^N \frac{\mu_i (\mathbf{r}(t) - \mathbf{r}_i(t))}{|\mathbf{r}(t) - \mathbf{r}_i(t)|^3} \quad (4.26)$$

with  $\mathbf{r}(t)$ ,  $\dot{\mathbf{r}}(t)$  and  $\ddot{\mathbf{r}}(t)$  position, velocity and acceleration vectors respectively. If the time  $t$  is used as the independent variable to integrate the motion of the test particle with the modified PC method, it must be sampled a-priori on the Chebyshev nodes, by the definition of the method itself. Using a dataset for the ephemerides instead of requiring a custom integration of the full N-body problem makes  $\mathbf{r}_i$  sole function of the time  $t$ . In turn, the states of the N bodies can also be sampled a-priori, as the sampling times are never going to change through the whole integration process. Then, such samples can be given as input not only to the dynamics function evaluation, but become a parameter for all the required Picard iterations. This aspect can dramatically speed up numerical simulations in the interplanetary environment, provided that the precision achieved is satisfactory.

#### 4.4.1.1 *Flyby event detection*

The single simulation test case proposed in Section 4.4.1.2 requires the detection of flybys. This aspect is fundamental to overcome the instability issues connected to the PC scheme. In fact, if two trajectory samples are too far from each other and the trajectory in between experiences a flyby, the effect of the close approach may be missed. The proposed approach solves this issue implementing a flyby detection routine at the trajectory sampling stage, i.e. before its update through Picard iterations.

In particular, the initial one-orbit sampling is performed, and crossings of the planet's SOI are checked through all the samples. If a crossing is detected, the sample immediately within the SOI is considered as the terminal sample for the integrated interval, and the trajectory is re-sampled and PC-integrated up to this specific time with newly distributed Chebyshev nodes. Then, the final point is sampled and PC-integrated up to the exit of the SOI. Finally, the last sample at the SOI exit is used as new starting condition to run the heliocentric PC integration over a new whole orbital period, initiating a new propagation loop that runs until the end of the integration span.

#### 4.4.1.2 *Integration accuracy and performance*

The integration is performed piece-wise orbit-by-orbit, as suggested by Fukushima [48]: new time nodes are generated, thus new ephemerides data are sampled, one orbital period by one orbital period until the end of the time span is reached, or only once for the time spent within the SOI in case of flyby phases.

Figures 4.10a, 4.10b, 4.10c, 4.10d show the evolution of the relative position error with respect to JPL's data for the near-Earth asteroid 2010RF<sub>12</sub> from 1<sup>st</sup> January 1989, 100 years forward in time, integrating in the Sun-centered J2000 reference frame and varying the number of Chebyshev nodes per orbit from 15 to 200. Such asteroid was chosen because it performs a flyby of Earth, so that the hyperbolic phase could be tested too. The three different legs (pre-flyby in Figure 4.10a, flyby in Figure 4.10b, post-flyby in 4.10c) are shown on their own, as well as the overall global view is given in Figure 4.10d. The color scale portrays the different number of nodes and is the same for all the sub-figures, reported in Figure 4.10d. For each leg, the initial guess is the Keplerian solution, elliptical or hyperbolic depending on the current status, generated from the initial state (entrance to the SOI in case of planetary flybys). The integration accuracy increases with the number of nodes per orbit, and converges to the precision of a Runge-Kutta simulation strategy plotted with the black solid line. The

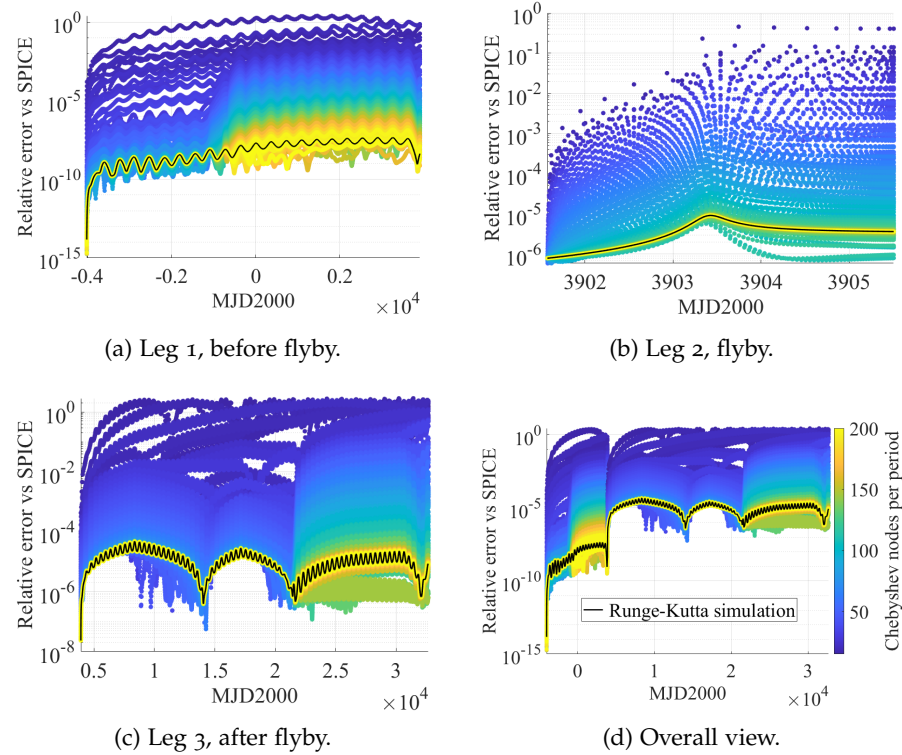


Figure 4.10: PC and Runge-Kutta RK78 integration errors, as relative position difference with respect to JPL’s data for the asteroid 2010RF<sub>12</sub>, for the pre-flyby (4.10a), flyby (4.10b) and post-flyby legs (4.10c), as well as globally for the whole integration span (4.10d). The color scale is the same for all the sub-figures, and reported in (4.10d).

latter has been performed using the RK78 method, adopting the same dynamical model in both cases<sup>4</sup>.

Figure 4.11 presents instead the relative relationship between the execution time of the sequential modified PC method and the number of Chebyshev nodes. A MATLAB<sup>®</sup> non-parallel implementation with a MEX<sup>®</sup> function for the dynamics<sup>5</sup> requires about 15 seconds to complete the full integration presented in Figure 4.10d, on a single core of a local workstation equipped with an Intel<sup>®</sup> Core<sup>™</sup> i7-7700 CPU (3.60 GHz).

#### 4.4.2 PC for Solar Orbiter-like flyby optimization

The proposed resonant trajectory optimization procedure has been tested on a phase of the ongoing mission Solar Orbiter [41], taking the initial data from the trajectory profile with launch in January

<sup>4</sup> The test case has been extensively discussed in [94] for the validation of the implementation of relativistic effects.

<sup>5</sup> The MEX<sup>®</sup> function has been generated with MATLAB<sup>®</sup>’s Code Generation Toolbox.

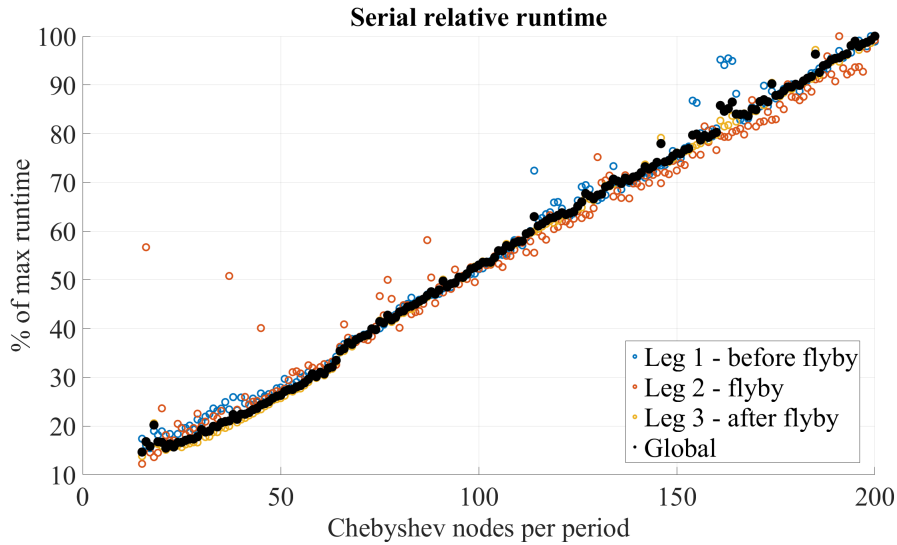


Figure 4.11: PC serial execution runtimes for the asteroid 2010RF<sub>12</sub>, reported as relative runtime with respect to the maximum runtime obtained for the different number of Chebyshev nodes.

2017, available in the mission redbook [43]. The algorithm has been entirely implemented in MATLAB<sup>®</sup>. A small computational acceleration is introduced compiling the PC iterations into a MEX<sup>®</sup> function with MATLAB<sup>®</sup> Coder<sup>™</sup>. The optimization problem of Equation (4.17) is solved with the `fmincon.m` function of MATLAB<sup>®</sup>'s Optimization Toolbox, using both the Interior-point and Sequential Quadratic Programming methods [60], based on the dimension of the search space. The selected local optimization algorithms and tolerances sufficed for the test case of this work to converge, proving the methodology concept of efficiently designing trajectories that take advantage of chaotic perturbations. The use of global search approaches and/or different tolerance setups could for sure increase the robustness of the approach, at the cost of possibly increasing the total computational load. In the case of practical use by mission analysts, the optimization algorithm selection should also be tailored on the hardware availability.

Solar Orbiter's first resonant phase with Venus is reproduced accounting for perturbing effects from the N bodies and general relativity. Following the notation from the mission redbook [43], the two gravity assist maneuvers are identified with V<sub>2</sub> and V<sub>3</sub>, with V standing for the flyby planet (Venus) and the numbers 2 and 3 representing the second and third close approach with Venus from the mission launch, respectively. The interplanetary leg between the two flybys is identified with V<sub>2</sub>-V<sub>3</sub>. The goal is to design flyby V<sub>2</sub> so that V<sub>3</sub> can lead to a desired post-encounter trajectory almost ballistically, i.e. minimizing the correction maneuver required in the phase between V<sub>2</sub> and V<sub>3</sub>. The maneuver is designed with maneuvering time  $\bar{t}$  at the apocenter

*Later discarded, the actual mission left Earth on February 2020.*

Table 4.2: Flyby V<sub>3</sub> entrance state.

$r_{\text{out},x}^{(j+1)}$ [km]	$r_{\text{out},y}^{(j+1)}$ [km]	$r_{\text{out},z}^{(j+1)}$ [km]
-67030683.03	-85738232.37	2563856.42
$v_{\text{out},x}^{(j+1)}$ [km/s]	$v_{\text{out},y}^{(j+1)}$ [km/s]	$v_{\text{out},z}^{(j+1)}$ [km/s]
30.54	-4.05	1.79
$t_{\text{in}}^{(j+1)}$ [MJD2000]		
8119.84		

Table 4.3: Retrieved optimal b-plane coordinates  $(\xi^*, \zeta^*)$ , exit time  $t_{\text{out}}^{(j)}$ , and planetocentric velocity  $\mathbf{U}^*$ .

$\xi^*$ [km]	$\zeta^*$ [km]	$t_{\text{out}}^{(j)}$ [MJD2000]
-8057.07	-5497.19	7446.52
$U_x^*$ [km/s]	$U_y^*$ [km/s]	$U_z^*$ [km/s]
3.08	17.78	3.66

of the first nominal orbit after V<sub>2</sub>, albeit, as already mentioned in Section 4.2, even this aspect can and should be optimized.

#### 4.4.2.1 Boundary conditions, b-plane pruning and method parameters

Generally, the required boundary condition is the state vector that allows a specified entrance to flyby  $j + 1$ . It may come from a previous step of the presented flyby design algorithm, as the output of the back-integration of  $(\mathbf{r}_{\text{out}}^{(j+1)}, \mathbf{v}_{\text{out}}^{(j+1)})$ , or simply being given, if no close approach is to happen after flyby  $j + 1$ . Considering the Solar Orbiter-like mission, flyby V<sub>3</sub> may be entered as the interplanetary state written in the ecliptic J2000 reference frame reported in Table 4.2.

Solar Orbiter's first resonant phase with Venus is in a 3 : 4 resonance. The output of the b-plane preliminary unperturbed design [95] enforcing the 3 : 4 resonance has produced the pruning quantities reported in Table 4.3, together with the exit time from flyby  $j$  set as  $t_{\text{out}}^{(j)} = 7446.52$  MJD2000.

The maneuvering time is set as a parameter, particularly at the nominal apocenter of the first interplanetary resonant orbit, with the correspondent state  $(\tilde{\mathbf{r}}, \tilde{\mathbf{v}})$  reported in Table 4.4.

The maximum values where to bound  $(\Delta\xi_{\text{max}}, \Delta\zeta_{\text{max}}, \Delta\mathbf{U}'_{\text{max}})^{(1,2,3)}$  have been set as 1% of the impact parameter [108]  $b = \sqrt{\xi^2 + \zeta^2}$  and of  $|\mathbf{U}'|$  for the b-plane coordinates and the velocity components respectively. The boundary value for the exit time variation  $\Delta t_{\text{max}}$  is set to 1% of Venus' orbital period. Trivially, the optimization starts with all the variables  $(\Delta\xi, \Delta\zeta, \Delta\mathbf{U}', \Delta t^{(j)})$  set equal to zero. The cost functions  $J_r$  and  $J_v$  are in all the cases computed as the relative values  $|\Delta\mathbf{v}|/|\tilde{\mathbf{v}}|$  and  $|\Delta\mathbf{r}|/|\tilde{\mathbf{r}}|$  with respect to the known maneuvering point, to remove the possible dimension sensitivity.

*For orbital parameters equal to the desired post-maneuver trajectory.*

Table 4.4: Maneuvering point, apocentre of unperturbed initial pruning solution.

$\tilde{r}_x$ [km]	$\tilde{r}_y$ [km]	$\tilde{r}_z$ [km]
-133524954.60	-32036518.08	-4418791.75
$\tilde{v}_x$ [km/s]	$\tilde{v}_y$ [km/s]	$\tilde{v}_z$ [km/s]
5.09	-20.43	1.65
$\tilde{t}$ [MJD2000]		
7570.92		

Specifically for the modified PC method, 160 nodes per period are used and the iterations are stopped when the maximum of the relative difference between two consecutive state updates drops below  $10^{-14}$ . The first arc to be designed, i.e. the one defining the optimal exit state and the maneuver, spans less than one orbital period, thus proportional nodes to the defined 160 per period based on its total time length are set, according to the fixed nodes per period logic. The optimal time found is then used for a single run of the optimization problem of Equation (4.17) with 200 Chebyshev nodes, assessing the influence of the number of nodes in the design precision, comparing both the node cases against a relativistic simulation.

#### 4.4.2.2 Optimization implementation

Despite the narrow region where the optimization variables are set to vary, even the smallest variations have a relevant impact in the convergence of the algorithm, especially if the marching position constraint is made strict. For this reason and to preserve the robustness of the approach, the optimization problem of Equation (4.17) is solved several times in a continuation procedure, using the result of the previous step as the new starting guess. Particularly:

- the search space dimension is reduced by 10 times for each optimization problem, up to an absolute minimum of  $10^{-8}$  starting from the already introduced  $\pm 1\%$  for each variable;
- within the optimization solver, the initial minimum relative step size between two iterations is of  $10^{-6}$ , reduced by a factor 10 each time up to  $10^{-15}$ ;
- the penalty factor  $\alpha$  is initially set to  $10^5$  to improve the convergence also for the  $J_v$  contribution, although the position constraint is then made stricter by raising the value of  $\alpha$  by a factor 10 each time, up to  $10^9$ ;
- the Interior-point algorithm in `fmincon.m` is selected for the first half optimization problems, whereas Sequential Quadratic



Table 4.5: Optimization results, in terms of position difference residual  $\Delta\mathbf{r}^*$  and correction effort  $\Delta\mathbf{v}^*$  at the maneuvering time.

$\Delta r_x^*$ [m]	$\Delta r_y^*$ [m]	$\Delta r_z^*$ [m]
-0.52	-0.52	-1.19
$\Delta v_x^*$ [m/s]	$\Delta v_y^*$ [m/s]	$\Delta v_z^*$ [m/s]
-1.28	1.57	0.22

Table 4.6: Optimization results, in terms of initial position residual and  $\Delta\mathbf{v}$  magnitude.

$ \Delta\mathbf{v}^* $ [m/s]	$ \Delta\mathbf{v}^* / \tilde{\mathbf{v}} $ [-]
2.04	$9.66 \times 10^{-5}$
$ \Delta\mathbf{r}^* $ [m/s]	$ \Delta\mathbf{r}^* / \tilde{\mathbf{r}} $ [-]
1.39	$1.02 \times 10^{-11}$

Programming is used in the last ones because of the smaller search space;

- MATLAB<sup>®</sup>'s `globalsearch` algorithm solves the current optimization problem if the previous step has returned the starting guess without improvements, searching for a global minimizer.

*For performance reasons, a maximum of half of the iterations can run the global search. In the presented test case at most two have been experienced, out of all the ten steps.*

The optimization problem of Equation (4.18) is solved with a grid search approach. The time span is always sampled with the initial supplied value plus 40 evenly spaced values of  $\Delta t^{(i)}$ , reducing  $\Delta t_{\max}$  by a factor 10 for 5 times, from the initial grid size equal to  $\pm 1\%$  of Venus' orbital period. The best value from the previous search is used as starting point for the new one. This approach resembles the algorithm used in MATLAB<sup>®</sup>'s `patternsearch.m` function, implemented manually in this work to keep a low number of trial  $\Delta t^{(i)}$ .

#### 4.4.2.3 Results

Solving the optimization problem of Equation (4.17) with the above described implementation has taken roughly 2-3 minutes, on a single core of a local workstation equipped with an Intel<sup>®</sup> Core<sup>™</sup> i7-7700 CPU (3.60 GHz). The 200 nodes algorithm converged to the residual  $\Delta\mathbf{r}^*$  and impulsive action  $\Delta\mathbf{v}^*$  for the required maneuver presented in Table 4.5.

Despite the execution point  $\tilde{\mathbf{r}}$  is yet to be optimized, and the fulfillment of the position constraint, the correction effort is small. The presented maneuver is modeled as a single impulse, nevertheless given its magnitude it can be achieved by the current low thrust propulsion technologies, as shown in Table 4.6.

*The difference with the  $\Delta\mathbf{v}$  resulting from the 160 nodes run is negligible, the position constraint is slightly worse fulfilled but in the same order of magnitude.*

Most of the computational time is due to the fulfillment of the position constraints. If the presented algorithm were used with a wider



Table 4.7: Optimization results, in terms of initial optimal state  $(\mathbf{r}_{\text{out}}^{(j)*}, \mathbf{v}_{\text{out}}^{(j)*})$ .

$r_{\text{out},x}^{(j)*}$ [km]	$r_{\text{out},y}^{(j)*}$ [km]	$r_{\text{out},z}^{(j)*}$ [km]
-64960957.28	-85998225.22	2682290.24
$v_{\text{out},x}^{(j)*}$ [km/s]	$v_{\text{out},y}^{(j)*}$ [km/s]	$v_{\text{out},z}^{(j)*}$ [km/s]
31.00	-3.45	1.7

Table 4.8: Retrieved optimal b-plane coordinates  $(\xi^*, \zeta^*)$  and planetocentric velocity  $\mathbf{U}^*$ .

$U_x^*$ [km/s]	$U_y^*$ [km/s]	$U_z^*$ [km/s]
3.25	17.76	3.67
$\xi^*$ [km]	$\zeta^*$ [km]	
-8057.22	-5700.49	

but still good position tolerance, it would likely run significantly faster, even before its parallel implementation.

The best starting time obtained in the optimization is  $t_{\text{out}}^{(j)*} = 7446.52$  MJD2000, slightly higher than the initial guess  $t_{\text{out}}^{(j)}$ . The interplanetary optimal starting state is given in Table 4.7.

The b-plane coordinates  $(\xi^*, \zeta^*)$  and the planetocentric velocity  $\mathbf{U}^*$  retrieved from  $(\mathbf{r}_{\text{out}}^{(j)*}, \mathbf{v}_{\text{out}}^{(j)*})$  and Venus' position at  $t_{\text{out}}^{(j)*}$  are presented in Table 4.8, proving the optimal pruning brought by the b-plane prediction.

The value of  $\zeta^*$  looks slightly (2%) out of the initial bounds despite the constraint, which may have two different explanations. First, the optimization variables are updated concurrently: variations on  $\mathbf{U}'$  also change the orientation of the b-plane axes, which in turn result on different b-plane coordinates for a given fixed position in space. Secondly, the domain reduction sequential procedure may find a minimum close to the initial boundaries, centering there the next narrower search. The difference is in any case rather small in magnitude, as it can be also seen in the real-scale difference shown in Figure 4.12: with the pruning point identified by the blue dot, whereas the optimized one is plotted in dark orange.

Figure 4.13 shows the difference between the designed trajectory with respect to the a relativistic simulation of the same case, both featuring the optimized maneuver at  $\tilde{t}$ . The two trajectories basically coincide even if using the lower number of Chebyshev nodes, with a relative difference that remains in the order of  $10^{-8}$ , as expected from what already seen in Figures 4.10a, 4.10b, 4.10c, 4.10d. Again, as expected the higher number of nodes yields a more accurate solution, with the difference from the relativistic simulation reduced by more than 10 times. Even if small, the error inevitably accumulates and is amplified if multiple gravity assists are present, hence, a higher

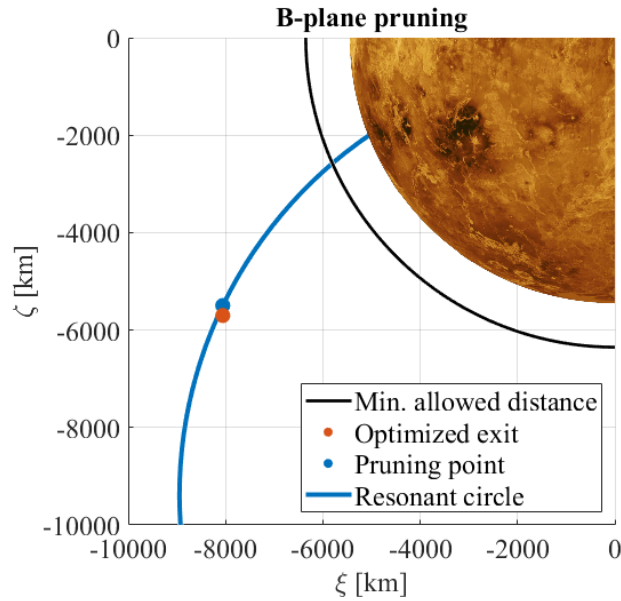


Figure 4.12: Visual representation of the b-plane pruning strategy. The pruning point corresponds to the blue dot, whereas the optimized point is depicted in dark orange.

*Because of the definition of Chebyshev nodes in Equations (4.3) and (4.4).*

number of nodes should be kept for more precise design requirements. The periodic "hills" visible in Figure 4.13 happen far from the domain boundaries, and are located where the Chebyshev nodes become sparser, also corresponding to the neighbourhood of the pericenter of the pre-maneuver arc in this case. The periodic error increase is likely due to the faster orbital dynamics nearby the pericenter, not followed by the Chebyshev node density, as the domain boundaries are located at the correction maneuver (at the apocenter). This effect could be mitigated by adapting the PC integration intervals so that the node distribution becomes denser nearby the pericenter, for instance splitting the optimization horizon into two sub-intervals, the first from the flyby exit to the pericenter, the second from the pericenter to the connection maneuver point, and finally following the same concept for the fixed post-maneuver arc. If the the full trajectory were required with as high precision as possible this should be considered, anyway, even with the tested setup, the long-term error evolution remains low, and already allows a precise design at the event points (flybys and correction maneuver). The "noise" over the hills may be explained by interpolation of the PC solution over the standard simulation time steps, necessary to visualize the presented difference measurement. The impact of the correction maneuver, despite small, can also be assessed: at the time of the close approach  $V_3$  the position would differ of thousands of kilometers from the desired condition, preventing the correct occurrence of the flyby.

Finally, Figures 4.14a and 4.14b show the continuous trajectory that embeds the planetocentric phases for both the flybys  $V_2$  and  $V_3$ ,

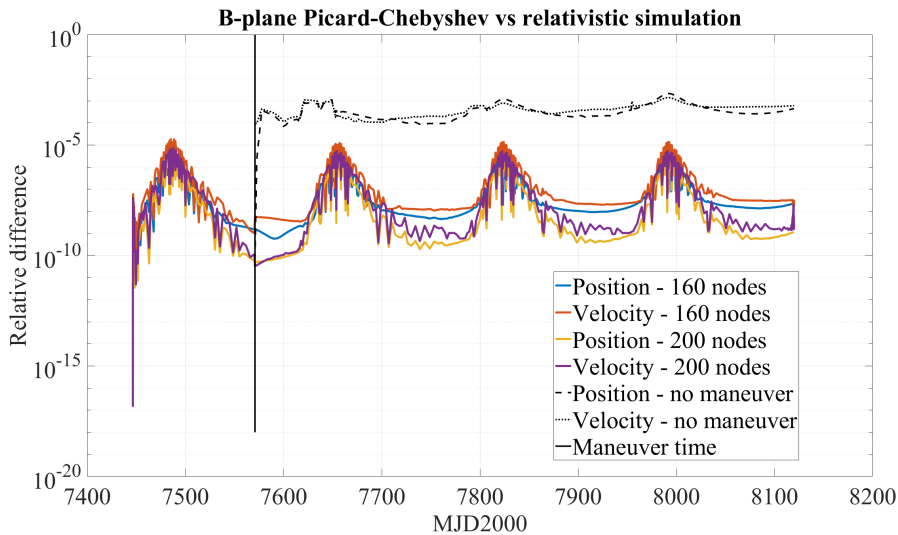
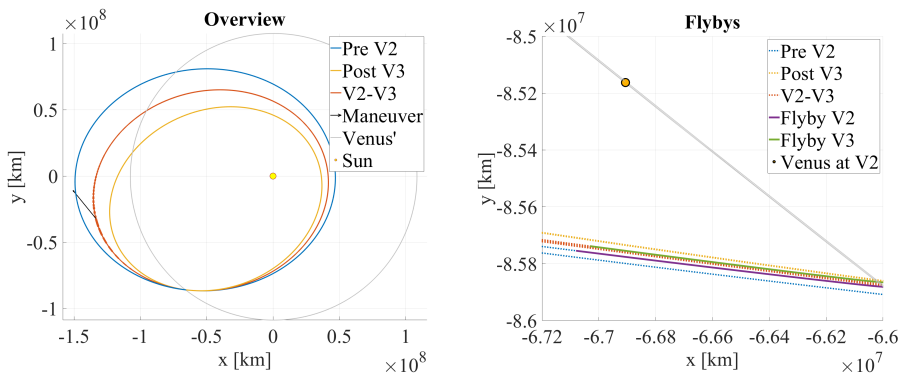


Figure 4.13: Design difference with respect to relativistic simulation between  $V_2$  and  $V_3$  for the two node cases, and without manoeuvre.

together with the pre- $V_2$  and post- $V_3$  solutions, all generated with the PC approach. As expected, all the orbits are all very similar to the original mission profile shown in Figure 4.14a. Zooming over the flyby regions the new continuity feature can be recognized (Figure 4.14b).



(a) Deep space correction maneuver and overview. (b) Zoom over Venus' flybys  $V_2$  and  $V_3$ .

Figure 4.14: Solar Orbiter's continuous first resonant phase with Venus.

#### 4.4.3 CPU and GPU performance of the augmented PC method

This application uses the two augmentation levels presented in Section 4.3.2 to build a hybrid approach, that treats the outer blocks as independent, and the inner ones as a single system in a more strict sense. In this way, groups of similar trajectories can be considered as unique but separate augmented system, allowing to maximize the integration performances. The two level augmentation also provides a

framework to deal with single trajectories within the same high performance computing context, considering them as a group made of only one member. As a practical example, the outer augmentation level could be used to "isolate" a sub-group of trajectories experiencing a planetary flyby, whose dynamics would become significantly different from the rest of the samples. The test case follows what computed in Section 4.4.2.

#### 4.4.3.1 Computational setup

To build a common framework for the pure algorithm performance evaluations, the 13509 initial conditions generated in the optimization process to eventually obtain the results of Table 4.5 are re-run, using the C and the CUDA<sup>®</sup> implementations of the PC integration. The execution of the 13509 independent runs with the C implementation of the algorithm is considered as benchmark case, both completely sequential and parallelized with OpenMP [28]. The matrix operations featured in the PC iterations are performed using the OpenBLAS library [158, 168, 169]. All the presented runs of the C algorithm have been executed on a machine running Ubuntu Linux 20.04, equipped with 40 physical / 80 logical cores of the type Intel<sup>®</sup> Xeon<sup>™</sup> CPU E5-4620 V4 running at 2.1 GHz, with varying number of OpenMP threads and the "o3" gcc compiler optimization enabled. Because of the physical machine where the GPU was available, the CUDA<sup>™</sup> code has been run on the same workstation of the Matlab<sup>®</sup> optimal solution computation. In this case a four core OpenMP<sup>®</sup> parallelization on a Intel<sup>®</sup> Core<sup>™</sup> i7-7700 CPU (3.60 GHz), is combined for concurrent executions with a NVIDIA<sup>®</sup> GTX 1050 (1.3GHz) graphics card.

*The NVIDIA<sup>®</sup> GTX 1050 is a 2016 low-end gaming card model, whose design purpose is far from the double precision computing of this work. Modern gaming/professional cards could run up to 60 times faster, data center cards up to 400-500 times faster than this model for the presented application.*

#### 4.4.3.2 Accuracy comparison

Figure 4.15 shows the evolution of the propagation error, measured as relative position and velocity error with respect to the independent runs case, for the C augmented and the CUDA<sup>®</sup> programs. In both cases the error remains lower than the specified relative tolerance used to halt the PC iterations, set to  $10^{-12}$ .

Figure 4.16 shows instead the evolution of the error for the dynamics function only, distinguishing two different compilations of the CUDA<sup>®</sup> programs, with (red dashed) and without (orange) the `-use-fast-math` flag enabled. Despite the error experienced in Figure 4.15, the C augmented program shows no error for the computation of the dynamics function. That means, the error is accumulated throughout the PC iterations only because of the different structure of the matrix multiplications. The implementation in the OpenBLAS library [158, 168, 169] is optimized on the matrix size. This is a known issue in high-performance computing problems, where the floating point representation of numbers interferes with a change in the order of

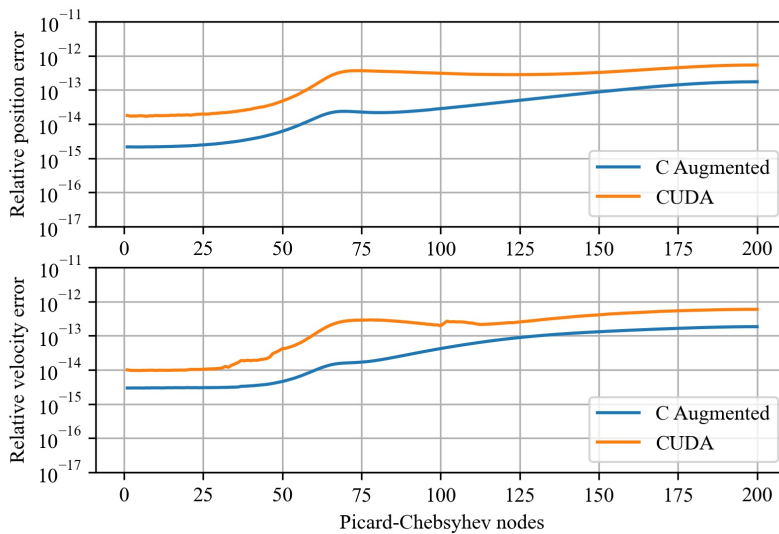


Figure 4.15: Errors of C augmented and CUDA<sup>®</sup> programs, as the average of the error of all the states in the augmented systems.

the basic math operations on the matrix elements, producing small deviations with respect to a reference non-parallelized solutions. No differences have been observed between parallelizing and not parallelizing the augmented C programs. On top of this consideration, the CUDA<sup>®</sup> program is also subject to the errors introduced by the different compilers. Other than being two different environments (Ubuntu Linux with GNU<sup>®</sup> compilers for the C augmented program, Windows 10 Pro with Microsoft<sup>®</sup> Visual Studio<sup>®</sup> and Nvidia<sup>®</sup> CUDA<sup>®</sup> compilers for the CUDA<sup>®</sup> program), small differences can also be observed by setting different optimization flags in the compilation. The comparison between enabling and disabling the fast math optimization options are run to exclude the possibility of implementation problems for the dynamics kernel. As it can be observed comparing the two compilations, the error spikes reaching  $10^{-12}$  happen in a seemingly unpredictable manner and can be completely attributed to the compiler, because happening at different PC nodes for the two cases. Otherwise, the error level remains more than two orders of magnitude lower. In any case, the accumulated effect of this error source is not taking the overall error above the  $10^{-12}$  iteration tolerance, as Figure 4.15 already highlighted.

#### 4.4.3.3 Performance comparison

Table 4.9 shows the runtime difference between the sequential C programs. The improved efficiency of the augmented integration can be immediately seen even in this sequential case, where the augmented system approach runs 23.9% faster, because of the minimized overhead

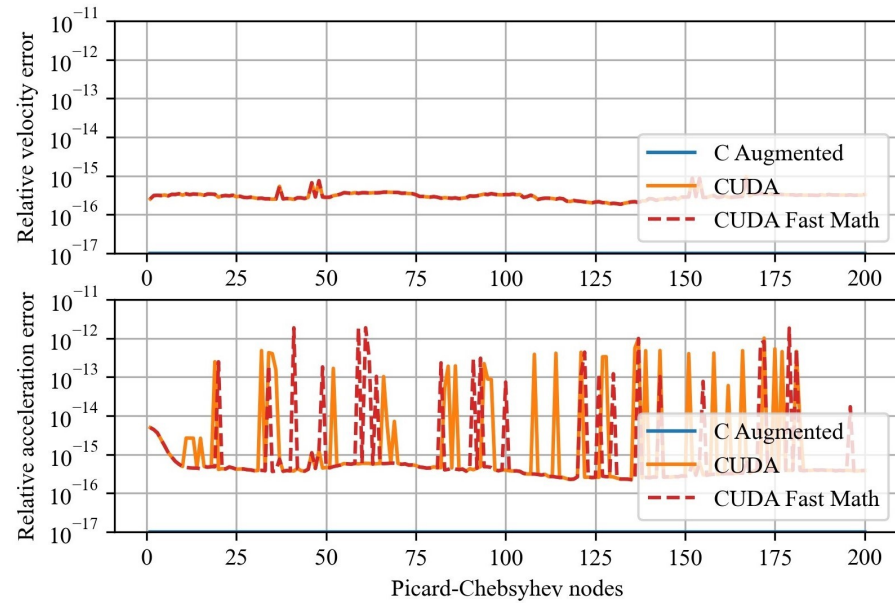


Figure 4.16: Errors of C augmented and CUDA<sup>®</sup> programs, as the average of the error of all the states in the augmented systems.

experienced by sharing the outer while loop. In the considered application the various trajectories all require 41 or 42 PC iterations, making it negligible to keep running trajectories even if their PC process has already converged.

Table 4.9: Sequential runtimes for the independent runs and the augmented system executions.

CASE	RUNTIME [s]
Independent runs	245.02
Augmented system	186.54

The scalability properties of the integration of independent trajectories and the augmented system are studied on the C implementations, i.e. assessing how well the execution of the two programs accelerates with increasing number of OpenMP<sup>®</sup> threads. Figure 4.17 shows that the augmented system features excellent scalability properties, for a runtime that keeps decreasing for increasing number of OpenMP<sup>®</sup> threads. On the contrary, the integration of independent trajectories experiences even higher runtimes after a certain number of threads. This happens because the parallelization itself introduces some overhead to the overall program execution, which cannot be compensated for by newly created parallel threads.

Figure 4.18 shows the achieved speedup, defined as the ratio between the sequential runtime and the parallel runtime, varying the number of OpenMP<sup>®</sup> threads. It provides a measure of how par-

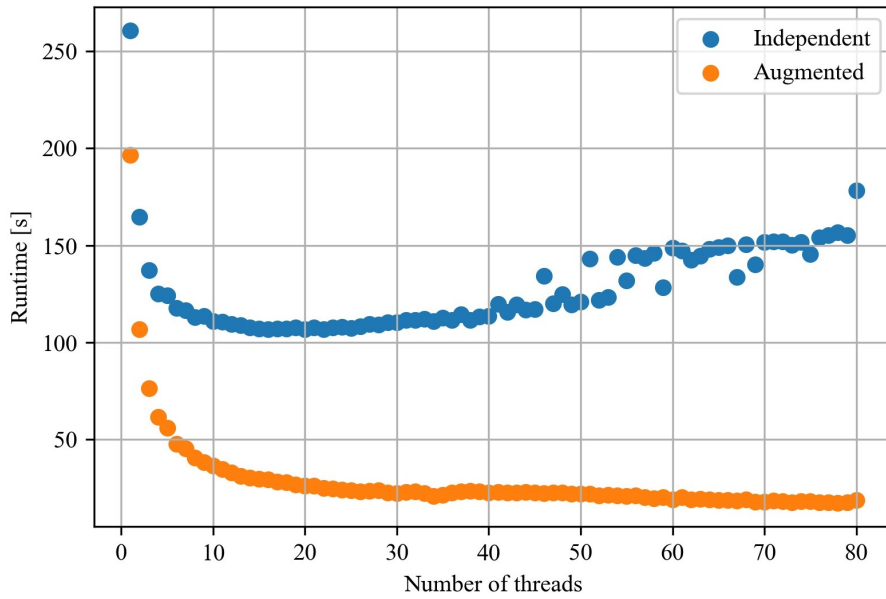


Figure 4.17: Augmented system and independent integrations C code runtime comparison with OpenMP® parallelization and varying number of threads.

allelizable the algorithm is, with higher values underlining higher accelerations. The augmented system integration shows once again excellent scalability properties even at high number of threads, suggesting the efficiency of the GPU computing transition even without having assessed the performances of the CUDA® implementation yet. If the number of threads is set equal to one, the algorithm conceptually

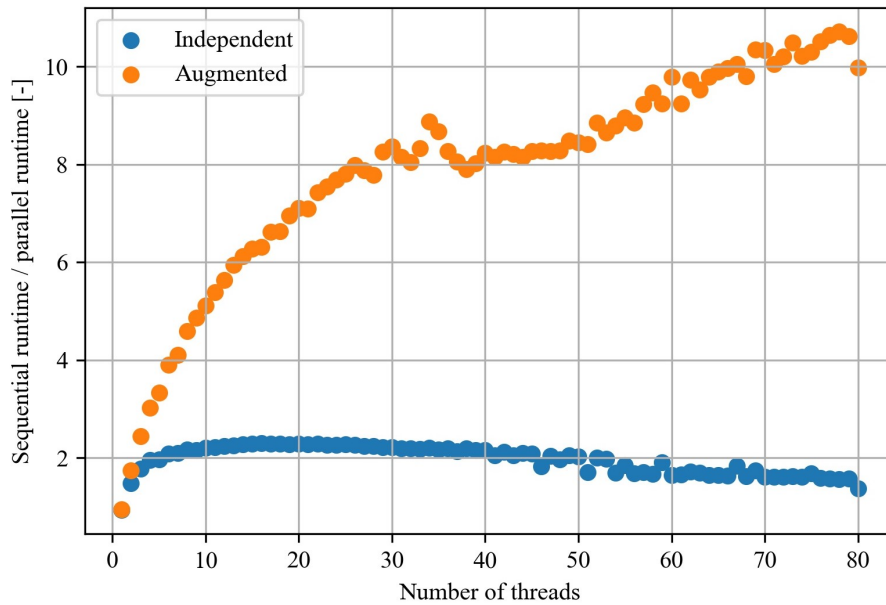


Figure 4.18: Augmented system and independent integrations C code speedup comparison with OpenMP® parallelization and varying number of threads.



reduces to a sequential case. However, enabling compiler optimization and the OpenMP<sup>®</sup> flag in the compilation, a parallel program introduces data distribution and retrieval tasks across the possibly multiple workers. With OpenMP<sup>®</sup>, the number of threads is specified by an environment variable right before running the program. Therefore, activating one thread only exposes all the parallelization-induced overhead without having any computational benefit at all. For this reason the one-thread runtimes for the independent and the augmented systems both result higher than the benchmark, sequential runtimes, which have been compiled with optimization enabled but without active OpenMP<sup>®</sup> flag.

Finally, the execution of the CUDA<sup>®</sup> implementation<sup>6</sup> results the faster overall, taking 15.84 seconds. The whole trajectory set has been split into 10 streams. The stream definition guideline should in practice fit the application the propagator would run on, being the sole flexibility degree left by the implementation. However, from the GPU viewpoint, larger kernels always imply a better GPU exploitation, thus creating too many outer augmentation levels, i.e. activating a large number of independent CUDA<sup>®</sup> streams, with too few trajectories each would result in performance degradation, eventually obtaining what already observed with the independent integration cases for the single trajectories.

Table 4.10 summarizes the runtime results discussed in the previous lines for the different cases, for selected number of cores in the C implementation cases.

*By the result of the integer division of the 13509 states by 9, with the tenth stream containing a number of states equal to the remainder of that division.*

Table 4.10: Runtimes for the independent runs and the augmented system executions. Average of 10 different runs each.

CASE	THREADS	GPU	RUNTIME [s]
C Independent	1	-	245.02
C Augmented	1	-	186.54
C Independent	8	-	113.09
C Augmented	8	-	40.60
C Independent	40	-	113.54
C Augmented	40	-	22.67
C Independent	80	-	178.18
C Augmented	80	-	18.70
CUDA <sup>®</sup> Augmented	4	GTX 1050	15.84

Figure 4.19 shows the increasing speedup achieved by the augmented system when compared to the sequential and independent simulations of all the samples. The CUDA<sup>®</sup> implementation runs more

<sup>6</sup> Only the results obtained for the compilation without the `-use-fast-math` options are shown. No significant runtime difference (less than 0.1 seconds) has been observed.



than 15 times faster compared to the baseline case, suggesting the suitability of the augmented PC algorithm to high performance and GPU computing facilities.

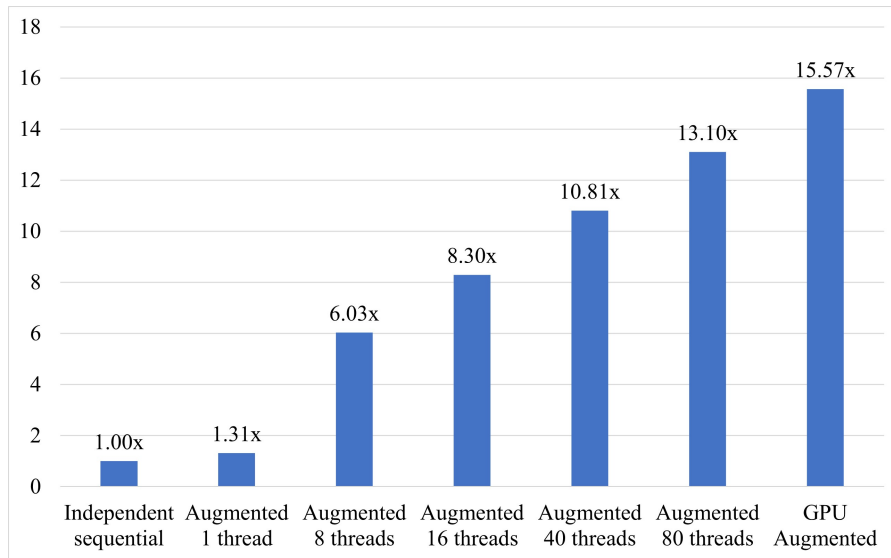


Figure 4.19: Speedup comparison among C and CUDA<sup>®</sup> implementations.

#### 4.4.3.4 CUDA<sup>®</sup> Kernel profiling and optimization - GTX 1050

The final kernel implementations are the result of a detailed profiling and optimization process, performed with the Nsight<sup>®</sup> Compute tool and a NVIDIA<sup>®</sup> GeForce<sup>®</sup> GTX 1050 graphics card. The choice of the 32 units block size of the dynamics kernel has been driven mostly by the need of using the shared memory also for the temporary storage of the double precision state and the acceleration vector elements, just too large to be kept on the thread-private registers. The error computation kernels do not feature this bottleneck, thus the full amount of 1024 threads per block can be activated, minimizing the number of memory transactions and maximizing the reduction effects. The further four times of reduction while reading the error from ephemerides data (better detailed in Section 4.3.6.2) are the result of multiple trials: adding more reduction layers results in a kernel slowdown not compensated by the consequent speedup in the CPU function final call. Analogously, removing some of them resulted in a CPU function slowdown not compensated by the kernel speedup. The chosen number of while-reading reductions may however be affected by the problem size of the selected test case. Larger (or smaller) sets of initial conditions to be propagated may feature a different optimal implementation of the maximum error computation. On the contrary, 32 is already the minimum block size for the dynamics kernel and is not related to the problem size. Other non-relativistic dynamics

function may drive the use of the shared memory in a different way, possibly allowing the use of larger block sizes.

The kernel performance is condensed into three numbers that result from the profiling process, i.e. the average multiprocessor occupancy for compute operations (COMPUTE in Tables 4.11 and 4.12), the shared memory utilization (SHARED MEMORY in Tables 4.11 and 4.12), and the memory throughput (THROUGHPUT in Tables 4.11 and 4.12). Despite profiling tools provide more detailed information, the presented indicators already allow the description of the kernel performances in sufficient detail. The profiling indicators of the cuBLAS<sup>®</sup> kernels for matrix-matrix product (dgemm), matrix-vector product (dgemv), and element-wise summation (daxpy)<sup>7</sup> used for the matrix multiplications embedded in the PC method are also shown, providing a performance comparison against well-known and heavily optimized library functions. In addition, also the kernel runtime as measured in the profiling activity is added to the comparison, to highlight the difference for the two tested augmented system sizes.

Table 4.11 shows the kernel profiling results for an augmented system made of all the 13509 states to be propagated. The compute-bound kernels can be easily recognized as the ones exploiting the most the GPU's compute capability (Dynamics (dynamics), dgemm, error computation (errCompute), and maximum error reduction (maxReduce)). The matrix-matrix multiplication is more memory bound than the dynamics function, because larger arrays must be loaded at the same time on the shared memory to perform the computation. On the contrary, the dynamics kernel features a more relevant computational bottleneck, because of the much more complex algorithm compared to the simple products and summations of the dgemm case. The remaining kernels, as well as the lower-intensity but still highly parallelizable error computation, all feature a high memory throughput. The GTX 1050 card used as a maximum memory band width of 112.1 Gb/s: the closer the throughput to that value the better the memory transactions are managed, essential feature for memory-bound problems. If some computations are added in the kernel, some throughput is inevitably lost, as latency sources are introduced between transaction to/from the shared/global memory.

Table 4.12 shows the kernel profiling results for an augmented system made of 1501 states to be propagated. The more compute-bound kernels do not show a significant loss of performance compared to the 13509-sized single augmented state case of Table 4.11, whereas the other kernels do. This highlights the suitability of GPU computing to extremely intense and parallelizable tasks, where the kernel call overhead is heavily compensated for by the massive task paral-

*Non-compute intensive but still highly parallelizable tasks may have the memory transfer as final bottleneck, the throughput measures how much of the CPU-GPU communication band width is used.*

*1501 results from splitting the full augmented system of 13509 states into 10 CUDA<sup>®</sup> streams, each but the last with a number of states equal to the integer division between 13509 and 9. The tenth and last streams contains a number of states equal to the remainder of the previous integer division.*

<sup>7</sup> For the sake of conciseness the kernels are identified with the names of the cuBLAS<sup>®</sup> API functions, although optimized kernels are called for each specific GPU architecture and problem size.

Table 4.11: 13509-sized augmented state kernel profiling results, for the GTX<sub>1050</sub> GPU.

KERNEL	COMPUTE [%]	SHARED MEMORY [%]	THROUGHPUT [Gb/s]	RUNTIME [ $\mu$ s]
dynamics	97.11	24.96	1.91	137470
dgemm	94.15	26.67	4.06	164210
dgemv	47.14	84.54	93.24	1400
daxpy	18.18	74.78	60.26	32.48
errCompute	99.41	53.26	59.36	4740
maxReduce	36.80	71.36	79.25	279.36

lization. Nevertheless, the capability of successfully processing also smaller-sized problems remains crucial for the program flexibility, particularly for what concerns the adoption of the proposed two-level augmentation scheme comprising smaller sub-systems significantly different from each other.

Table 4.12: 1501-sized augmented state kernel profiling results, for the GTX<sub>1050</sub> GPU.

KERNEL	COMPUTE [%]	SHARED MEMORY [%]	THROUGHPUT [Gb/s]	RUNTIME [ $\mu$ s]
dynamics	96.86	24.92	1.91	15300
dgemm	93.67	26.65	4.08	18390
dgemv	42.59	78.74	87.53	170.85
daxpy	9.19	33.95	36.31	6.62
errCompute	98.25	53.44	59.78	529.34
maxReduce	26.29	63.35	68.20	42.88

Table 4.13 compares instead the runtimes of the CUDA<sup>®</sup> program, for the cases of 1 and 10 active streams. Theoretically, the former has the advantage of a better GPU resource exploitation, whereas the latter makes a more aggressive use of the CPU-GPU concurrency. Despite the lower GPU efficiency, the achieved runtimes are almost identical. Therefore, the two-level augmentation scheme can efficiently tackle the case of differently-sized lower-level augmented subsystems, showing the flexibility of the proposed computational scheme, despite the fine grain code optimization necessary for its successful implementation.

Table 4.13: 1 and 10 streams CUDA<sup>®</sup> program executions. Average of 10 different runs each, for the GTX 1050 GPU.

CASE	RUNTIME [s]
1 stream	15.78
10 streams	15.84

#### 4.4.3.5 Profiling on the RTX A6000

The profiling of the CUDA<sup>®</sup> kernels presented in the previous section is repeated, using the Nvidia<sup>®</sup> RTX A6000 graphics card and running the computations on a Ubuntu Linux 22.04 machine, equipped with 2 Intel<sup>®</sup> Xeon<sup>®</sup> Platinum 8352V at 2.1 GHz. The RTX A6000 mounts 48 Gb of memory and is equipped with roughly 15 times more CUDA kernels compared to the GTX 1050, hence featuring a much higher memory throughput (768 Gb/s against 112 Gb/s). For this reason, the block size for the dynamics kernel has been increased from 32 to 128.

Table 4.14 shows the profiling results for the same test case presented in Table 4.11, that is the propagation of the full augmented system on a single CUDA<sup>®</sup> stream. For the dynamics kernel, the main difference with respect to the GTX 1050 case is the lower utilization of the GPU, visible both in terms of computational occupancy and shared memory utilization. Being a more powerful graphics card, the kernel implementation could be optimized for higher computational resources, better exploiting the available shared memory and possibly increasing the overall performance.

Table 4.14: 13509-sized augmented state kernel profiling results, for the RTX A6000 GPU.

KERNEL	COMPUTE [%]	SHARED MEMORY [%]	THROUGHPUT [Gb/s]	RUNTIME [μs]
dynamics	89.10	3.27	23.83	19080
dgemm	97.09	3.86	13.77	20340
dgemv	35.97	91.66	653.49	216.03
daxpy	16.04	41.52	250.64	5.63
errCompute	89.86	59.32	431.99	705.66
maxReduce	29.11	73.37	514.91	45.70

Table 4.15 shows instead the profiling of the kernels when the trajectories are split in 10 different streams, analogously to Table 4.12. As expected and as already observed, the efficiency slightly decreases with augmented systems of smaller size.

Table 4.15: 13509-sized augmented state kernel profiling results, for the RTX A6000 GPU.

KERNEL	COMPUTE [%]	SHARED MEMORY [%]	THROUGHPUT [Gb/s]	RUNTIME [μs]
dynamics	87.24	3.02	21.98	2170
dgemm	95.72	3.78	12.50	2310
dgemv	31.97	74.01	469.83	33.22
daxpy	2.87	7.39	44.14	3.55
errCompute	84.84	54.54	394.22	83.65
maxReduce	18.93	47.54	318.97	8.19

Finally, Table 4.16 compares the total runtimes of the single and the 10-stream cases. While each kernel runs roughly 8 times faster on the RTX A6000, the global runtime reduces of "only" a factor 4. This result may be explained by the CPU cluster occupancy at the time of testing, as well as for the problem becoming more and more CPU-bound: the massive parallelization of all the GPU parts of the program has reduced the device-related runtime, possibly enough to make the sequence of iteration control instructions issued by the CPU the new bottleneck.

Table 4.16: 1 and 10 streams CUDA<sup>®</sup> program executions. Average of 10 different runs each, for the RTX A6000 GPU.

CASE	RUNTIME [s]
1 stream	3.87
10 streams	3.84

The profiling activity performed with a different device highlights once again the importance of low-level programming in GPU computing: the configurations that led to the optimal performance with the GTX 1050 are not exploiting the RTX A6000 at its full potential, which should be adapted for greater shared memory use to achieve full occupancy and, thereby, optimizing the overall execution. At the same time, the CPU side of the program should also be reviewed, aiming at reaching total runtimes aligned with the reduction factors observed for the CUDA<sup>®</sup> kernels, if possible.

## 4.5 SUMMARY

### 4.5.1 *Fixed-point high-precision resonant flyby optimisation*

The complexity of the multi-flyby design problem in the continuous environment has been successfully broken down into a backward recursive approach, that designs each of the flybys in cascade, considering the next encounter as the target condition for how to perform the current one. Given the results of an unperturbed patched conics analysis, the b-plane has been proven to be a powerful formalism to enforce a continuity condition with, and particularly well suited for pruning purposes, making the dimension of the optimization search space minimal.

A first possible development direction is the inclusion of tighter mission constraints, such as a minimum pericenter distance as Solar Orbiter needs [41]. This aspect might be tackled with the proposed strategy before the design of any maneuver, seeking for quasi-ballistic solutions that surf the effects of orbital perturbations, even if chaotic, aiming to minimize the required artificial corrections.

Despite it might be already satisfactory, the computational performance of the method is for sure what can be improved the most by future works. First of all, the optimized versions of the modified PC method can be adopted and the sequential execution can be accelerated by a complete implementation in a compiled programming language, instead of the MATLAB<sup>®</sup> platform proposed in this work. Furthermore, although the multi-step solution of the optimization problem of Equation (4.17) proposed in the presented application is robust, it could be better tailored by prior analysis of the search space or made more lightweight already scanning with finer tolerances.

In conclusion, a systematic framework to surf a complex perturbation environment such as the relativistic N-body problem is proposed. Provided the model to be sufficiently accurate, the available technology might become the new bottleneck for practical purposes: some uncertainty is inevitably introduced by the execution of the control maneuvers, as well as the connected orbital determination measurements. Future works might also deepen this aspect using models of real life equipment, studying in turn what consequences non-precise measurements or thruster firings might have on the high-fidelity designed trajectory, together with the possible required mission planning actions.

Looking to possible applications in other environments, the presented approach may be used in the design of moon tour missions towards the giant planets, which feature the available fuel as a major constraint. Quasi-ballistic solutions are always sought for, to swing by the numerous bodies multiple times maximizing the exploration outcome.

#### 4.5.2 *PC augmentation for large sets of initial conditions*

The proposed development of the PC scheme explores the benefits that high-performance CPU clusters and GPU computing architectures bring to the short leg orbital propagation of large sets of initial conditions. The tested case studies the runs required to design a Solar Orbiter-like resonant phase with Venus, optimized to surf the relativistic N-body environment, proposing a two-level augmentation strategy implemented in the C and the CUDA<sup>®</sup> programming languages.

Propagating the augmented system always outperforms propagating the trajectories independently, both in the sequential and the parallelized case. The augmentation benefits appear in two different aspects, the first being the reduced overhead compared to the repeated independent runs, the second represented by a finer grain parallelization also exploiting optimized linear algebra libraries.

The approach scalability allows the implementation on GPU architectures, with low end graphics card already capable of matching the performance of a common-size cluster node. Data center card models can make the algorithm run around 400-500 times faster, while the newest gaming GPUs should already allow a 60 times program acceleration. In addition, implementing the second order version of the modified PC method with error feedback should introduce a further two/three-fold acceleration.

Despite being shown on a test case derived by a trajectory optimization application, the proposed scheme represents a completely general orbital propagator. Any application requiring the propagation of large sets of initial conditions could benefit by the high computational efficiency provided by this algorithm, not necessarily requiring the use of super-computing facilities, in favor of much less expensive graphics cards.

Future works may keep developing the software tool, managing longer integration spans in sequence, and, as a general propagator, implementing event detection and management procedures. At the same time, the optimization scheme that led to the presented trajectory design application can be modified to accommodate massively parallel search approaches. On the implementation side, the benefits of using CUDA<sup>®</sup> graphs instead of the stream-based management of the outer level augmented systems and the use of Tensor Core<sup>®</sup> capabilities of modern devices may be explored.





# 5

---

## KS VARIABLES IN CUDAJECTORY

---

Following the theoretical and experimental results presented in this dissertation, some of the proposed concepts and state-of-the-art advancements are already suitable for implementation and application. In particular, the performance improvement obtained in Chapter 2 justifies the goal of this chapter, that covers the implementation and the validation of KS variables in CUDAjectory [32, 57, 128]. A summary of parallel and GPU computing concepts and fundamentals can be found in Appendix C, as this chapter makes an extensive use of GPU and CUDA<sup>®</sup> nomenclature and terminology. After an introduction on the general structure of CUDAjectory in Section 5.1, Section 5.2 outlines the implemented software extensions. Finally, Section 5.3 validates the KS propagation in CUDAjectory, running a compliance analysis of the ESA mission JUICE with SDM policies.

### 5.1 CUDAJECTORY GENERAL ARCHITECTURE

CUDAjectory [32] is a GPU object-oriented software for massively parallel trajectory propagation, using the Runge-Kutta-Fehlberg RKF78 integration method. As the name suggests, CUDAjectory is written in CUDA<sup>®</sup> C++, hence making it suitable to be run with NVIDIA<sup>®</sup> graphics cards.

CUDAjectory has been developed and is maintained by the Mission Analysis section at the European Space Operations Centre (ESOC), at ESA, and is available to the general public by an ESA Community License. The first implementation follows the Master's thesis work of Geda [57], including most of the model features, i.e. gravitational effects from the Solar System bodies, solar radiation pressure, spherical harmonics perturbation for Earth orbits, and high-order perturbations based on point mascons [57]. CUDAjectory manages ephemeris data in a unique way, with data written in the CUBE (acronym for CUdjectory Binary Ephemeris) format and accessed from texture memory, improving the performance of the software. In this way, CUDA<sup>®</sup> threads can access ephemeris data more efficiently than in lookup-based approaches (commonly adopted with SPICE's spk files [2]). More recent works have widened the force pool with an atmospheric model and aerodynamic drag acceleration [6], as well as the

capability of interrupting simulations based on the total radiation absorption in the Jovian environment [144]. Additionally, Inno [67] improved the efficiency of memory management and event detection routines.

Figure 5.1 shows the overall software structure. Other than the pure command line program, CUDAjectory includes a PyBind<sup>®</sup> [117] interface, that exposes CUDAjectory’s simulators to Python programs and Python Numpy<sup>®</sup> [111] input/output array types, providing Python users a simple interface to GPU parallelized propagations. For the command line execution of CUDAjectory, input/output samples and trajectories are given in csv format, while the only configuration file required can be written either in json [73] or yaml [170] format, for the Python library implementation as well.

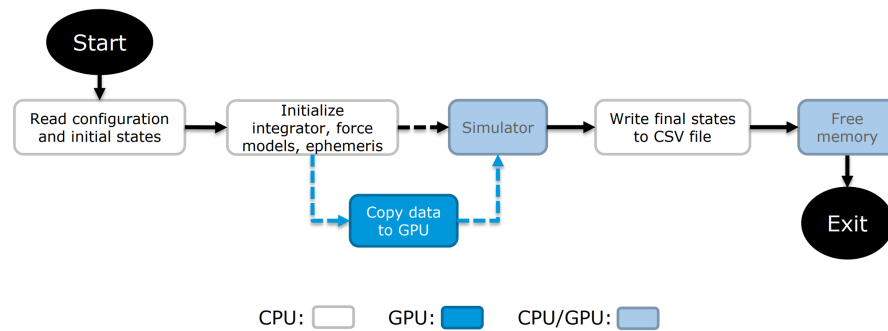


Figure 5.1: CUDAjectory overall structure. Picture from [57].

Figure 5.2 shows instead the simulation approach implemented in ClusterSimulator. A limited set of steps is taken on the GPU, then the simulation data are passed to the CPU for storage and processing. The process is repeated in a while loop until the end of the simulation is reached. This approach has the advantage of reducing the occupancy in the device memory, since, if needed, the full trajectories are gradually stored on the CPU, keeping the limited GPU memory free. Additionally, the simulation samples can be sorted to maximize the software performance, for instance gradually removing early-ending samples (e.g. in case of detection of impacts).

## 5.2 IMPLEMENTATION OF THE KS DYNAMICS

Figures 5.1 and 5.2 highlight the blocks of CUDAjectory where the modifications required to run simulations in KS coordinates should take place. At first glance, the following routines are required:

- non-dimensionalization of Cartesian state vectors;
- conversion between KS and Cartesian formulations;
- interface with the existing force model for the computation of the dynamics function;

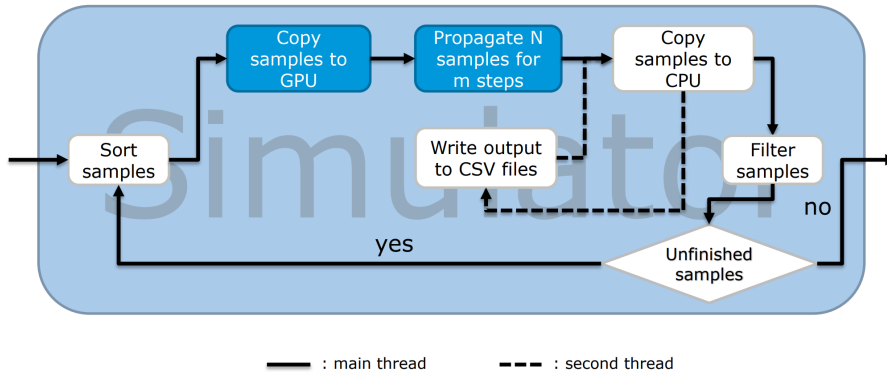


Figure 5.2: ClusterSimulator structure in CUDAjectory. Picture from [57].

- end-of-simulation based on physical time;
- interface with existing event functions.

As an additional user-oriented requirement, the KS dynamics should possibly not be exposed, aiming at preserving the software interface and configuration.

### 5.2.1 High-level implementation

Following the just outlined requirements, CUDAjectory has been extended completely hiding KS variables at the core of the software. All the Simulator classes in CUDAjectory have been extended with a new method that runs KS variables, with the user interface remaining unchanged. CUDAjectory can still be run in the traditional Cartesian setup, calling the usual `evolve()` method, while the KS run is launched calling the `ksevolve()` routine. Figure 5.3 shows the conceptual extension that Simulator classes in CUDAjectory have undergone to run with KS variables.

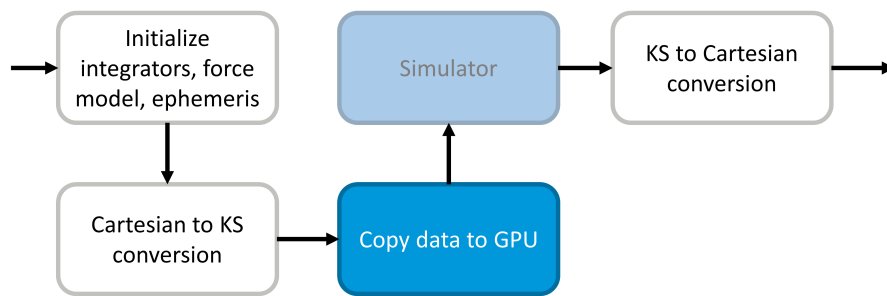


Figure 5.3: CUDAjectory overall extension logic.

Taking a closer look to the Simulator block, the transformation from KS to Cartesian coordinates needs to be implemented also for the storage of intermediate simulation steps, as highlighted in red in Figure 5.4. Internally in the "Propagate N samples for m steps" block,

the given force model is interfaced with KS variables according to Equations (2.30) and (2.35). Events are instead computed by retrieving the Cartesian states, at each step.

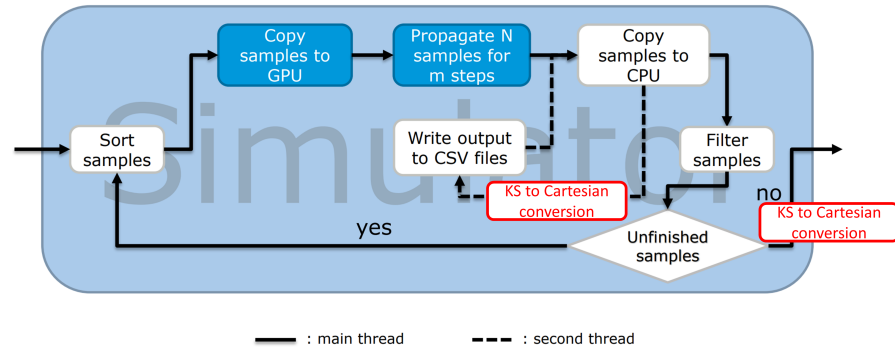


Figure 5.4: CUDAjectory Simulator extension logic. Picture modified from [57].

In this implementation, the end of integration in physical time is implemented by early halting the KS simulation, immediately before stepping over the final epoch. Then, the samples are propagated for the remaining time interval in Cartesian coordinates. This is done to exactly reach the user-supplied final integration time. While approaches to achieve this behavior exist in the literature for generic regularized formulation (e.g. by Amato et al [3]), the CUDAjectory implementation may not be straightforward, because the full trajectory is not stored in general, making the pre end-of-time step data unavailable.

### 5.2.2 Low-level code extension

This section summarizes the extensions that have been implemented in CUDAjectory, referring to the different parts of the code according to the names of the classes involved. The full details can be found in CUDAjectory’s documentation [32].

#### 5.2.2.1 Samples, states, and time

The conversion between the Cartesian and KS formulation, together with the adaptive non-dimensionalization proposed in Chapter 2, have been encapsulated in three newly created classes:

- `Rea14`, that implements four-vector and quaternion algebra (analogous to the `Rea13` class);
- `KSSState` (analogous to the `CartState` class for the Cartesian formulation);
- `KSSample` (analogous to the `CartSample` class for the Cartesian formulation);

The `KSSState` class has been implemented following the paradigms outlined in `CartState`. Other than a few useful methods to wrap quaternion operations and retrieve the Cartesian state, following the `CartState` naming the class properties are defined as in Listing 5.1.

Listing 5.1: KS state class properties.

```
/**
 * @brief KS state containing position, velocity, and
 *        time element.
 */
struct KSSState : public State
{
public:
    // ...

    // Data
    Real4 pos;           /**< position-like */
    Real4 vel;          /**< velocity-like */
    RealPrecision time; /**< time element */

    // ...
}
```

*CUDAjectory wraps single and double precision in the `RealPrecision` class.*

The time element follows the physical time in its relative evolution, i.e. it represents the elapsed time from the given initial condition. The conversion routines between the Cartesian and the KS formulations, as well as the optimization of the KS fibration parameter presented in Chapter 2, are encapsulated in the object constructors: either `CartState` and `KSSState` can be constructed with either object as input, overloading the constructor for different input types and implementing ad-hoc object construction algorithms, all following the procedures presented in Chapter 2.

The management of the adaptive non-dimensionalization and of the fictitious time itself is instead delegated to the `KSSample` class. Sample classes store many properties and useful flags to efficiently run the simulation, other than a state object as the state vector being propagated. The extension does not represent the simple inclusion of `KSSState` instead of `CartState`, rather, two additional properties are created to manage the adaptive non-dimensionalization: `lref` and `tref`, as reference length and time, respectively, as shown in Listing 5.2.

Listing 5.2: Cartesian and KS sample properties.

```
/**
 * @brief Information of a trajectory simulation
 *        sample in Cartesian state representation.
 */
struct CartSample : public Sample
{
```

*The reference dimensions are kilometers and seconds.*

```

        /** Cartesian position and velocity */
        CartState state;

    // ...
}

/**
 * @brief Information of a trajectory simulation
 * sample in KS state representation.
 */
struct KSSample : public Sample
{
    /** KS state */
    KSState state;

    /** Non-dimensionalisaiton quantities*/
    RealPrecision lref;
    RealPrecision tref;

    // ...
}

```

Similarly to the state case, the full conversion process between `CartSample` and `KSSample` is encapsulated in the overloaded definitions of the object constructors, based on the input types, including the non-dimensionalization. Both samples store the initial epoch in MJD2000 in the `startEpoch` property. The adoption of the fictitious time is managed as follows:

- `CartSample` stores the relative simulation time (physical) in seconds in the `relTime` member;
- `KSSample` stores the non-dimensional relative simulation time (fictitious) `relTime` member, that is initialized to 0 unless differently specified in input.

Both classes feature the `getAbsTime()` method to retrieve the absolute simulation time in MJD2000, that is often used to read ephemeris data. `KSSample` implements its own version of the method, retrieving the relative physical simulation time through the time element in `KSState` and the `tref` property, as shown in Listing 5.3. The object constructors fully encapsulate the mapping between `CartSample.relTime` and the time element in `KSState` as well.

Listing 5.3: Absolute time retrieval methods.

```

// getAbsTime() for Cartesian samples
RealPrecision CartSample::getAbsTime() const
{
    return startEpoch + relTime / 86400;
}

```

```

// getAbsTime() for KS samples
RealPrecision KSSample::getRelTime() const
{
    return state.time * tref;
}

RealPrecision KSSample::getAbsTime() const
{
    return startEpoch + getRelTime() / 86400;
}

```

### 5.2.2.2 Dynamics, events, and general interface

CUDAjectory's memory management approach required the creation of a whole new class for the extension of the dynamics function. The Cartesian version is called RHS, shorthand for Right Hand Side, as the right hand side of the dynamics equation, whereas the implemented KS counterpart has been named KSRHS. To optimize the memory usage, RHS is initialized with starting epoch (e.g. `CartSample.startEpoch`), the center of the reference frame, and the force model used in the computation, whereas `CartState` and the relative simulation time `CartState.relTime` are supplied when calling the dynamics evaluation itself. The output location of the evaluated dynamics is set with the dedicated `setOutputLocation()` method, that writes to a pre-allocated memory location with `CartState` type. More than any other interface issue, this aspect has required the computation of the KS dynamics from within a RHS-like object, because major modifications to the numerical schemes functions would be required otherwise. The differences between RHS and KSRHS can be summarized as:

- KSRHS takes a `KSSState` input, as well as the reference dimensions `lref` and `tref`, retrieving the dimensional Cartesian state internally;
- KSRHS computes the model acceleration in the same way as RHS, only modifying the total value according to the non-dimensionalization quantities and converting it to the KS formalism. Eventually, the dynamics result is stored as a `KSSState` made by the position, velocity, and time element derivatives with respect to the fictitious KS time, as presented in Listing 5.4.

Listing 5.4: KS dynamics function interface.

```

void KSRHS::calculate(const RealPrecision relTime,
    KSSState& state, /* other input parameters for the
    acceleration computation are included */ const
    RealPrecision lref, const RealPrecision tref)
{

```

```

// get car state
CartState carState = CartState(state);
// make it dimensional
carState.pos *= lref;
carState.vel *= (lref / tref);

Real3 totalAcc;

/**
 * totalAcc is filled as a normal Cartesian
 * acceleration
 */

// compute KS acceleration
RealPrecision r = state.pos.sumSquare();
totalAcc /= (lref / (tref * tref));
Real4 tmp = Real4(totalAcc);
Real4 ksacc = (state.vel.sumSquare()/r) * state.
    pos;
ksacc += 0.5 * r * tmp.quatMultiply(state.pos.
    quatConjAntiInvolute());

// the derivative of the KS state is stored
// in the stateOutput location
new (stateOutput) KSState(state.vel, ksacc, r);
}

```

The interface for all the event functions follows the simple creation of a `CartSample` object, that is supplied to the usual Cartesian event detection routines. Normal event functions only update some of the flags of `CartSample` to control the simulation: such flags are then all copied in block to the initial `KSSample` to proceed with the propagation. A slightly different approach is implemented for routines that switch the center of the reference frame in case of entrance or exit from a SOI: in this case, the event `CartSample` is switched according to the normal event routine and then used to update the initial `KSSample` with a new constructor call.

The definition of the presented KS-related classes has required the minor adaptation of several functions across `CUDAjectory`. For instance, the RKF78 CUDA<sup>®</sup> kernel and the device function have been overloaded with their KS versions, adapting all the input and internal declarations accordingly.

### 5.3 APPLICATION: JUICE DEBRIS MITIGATION ANALYSIS

The proposed implementation of KS variables in `CUDAjectory` is tested with the propagation of the dispersion of Ariane 5's upper stage of launcher, for the JUICE mission case. Boutonnet and Rocchi used `CUDAjectory` [17] to assess the compliance of different possible launch scenarios with space debris mitigation policies, computing the impact



probability of this dispersion with Earth. The initial dispersion data are given in terms of nominal interplanetary injection state and the related covariance matrix, both covered by a non-disclosure agreement between the ESA and Arianespace. Nevertheless, the results of the CUDAjectory implementation can still be studied with a detailed analysis of the output trajectories.

Figure 5.5 shows the position error evolution for the simulation of one of the samples, performed with  $10^{-12}$  absolute and relative tolerances in Cartesian and KS coordinates. The error is measured against a reference Cartesian simulation, also made with CUDAjectory but with  $10^{-14}$  absolute and relative tolerances. The position error remains mostly the same for the two cases: the chosen values of absolute tolerance make the Cartesian simulation slightly more accurate, as no non-dimensionalization is present in this case. Table 5.1 shows instead the steps required to complete the simulation, achieving a reduction that is even greater than what observed in Chapter 2. This may again be due to the combined effect of absolute tolerance and non-dimensionalization choice, as the Astronomical Unit and the Year have been used to make the benchmark simulation non-dimensional in Chapter 2.

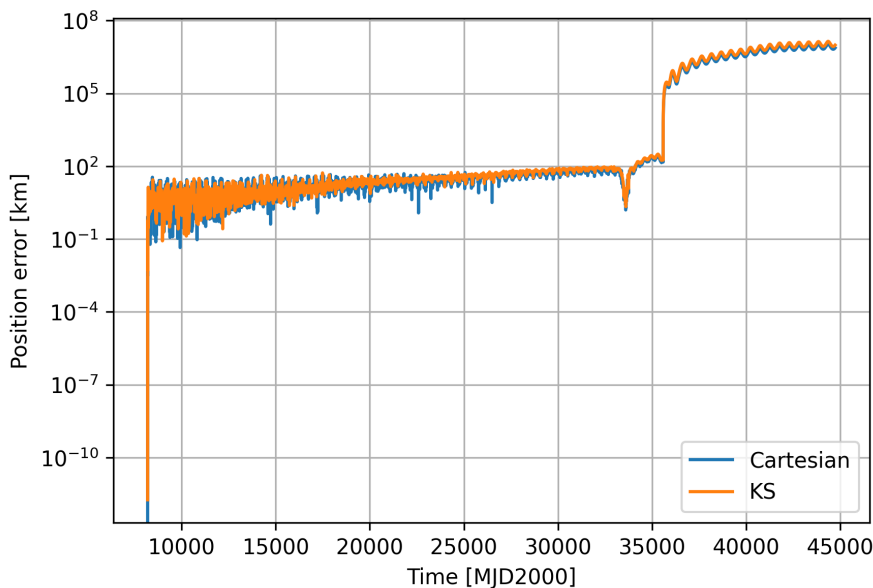


Figure 5.5: Position error evolution example, KS and Cartesian coordinates in CUDAjectory.

The following analysis focuses instead on the achieved runtime reduction, validating the computation of both `evolve()` and `ksevolve()`, propagating the same initial conditions and comparing each sample outcome against a higher precision `evolve()` simulation. The test cases have been run in a Ubuntu<sup>®</sup> Linux<sup>®</sup> 22.04 machine, equipped with two Intel<sup>®</sup> Xeon<sup>®</sup> Platinum 8352V (36 cores, 72 logical @ 2.1 GHz) and a Nvidia<sup>®</sup> RTX A6000 graphics card. Absolute and relative tolerances for

*Assessing if each sample impacts with Earth or not.*

Table 5.1: CUDAjectory steps for Cartesian and KS cases.

STEPS	
evolve()	5272
ksevolve()	2399
Reduction	58.3%

the numerical schemes have been set equal to  $10^{-12}$ , and the samples have been propagated 100 years forward in time.

Tables 5.2 and 5.3 show the runtimes for the different CUDAjectory calls, with and without storage of the full propagated trajectories, respectively. KS variables visibly improve the efficiency of the software, which increases with the number of samples. In case of stored trajectory, the improvements become even more relevant, achieving a runtime reduction of over 60% in the case of 20k propagated samples. Other than the reduction of the steps taken presented in Table 5.1, the reason for this additional performance enhancement is explained by the lower amount of states that CUDAjectory’s output handlers need to manage, resulting in an overall reduced memory and output processing overhead. Nonetheless, the runtime reduction without trajectory storage remains consistently around 40%, a much lower value compared to the observed step reduction, particularly if comparing it with the relation between step and runtime reduction presented in Chapter 2. This suggests that the current implementation may be optimized, minimizing the number of conversion operations to compute events and reducing the memory usage.

Table 5.2: CUDAjectory runtimes for different configurations, with trajectory storage. Average of 5 different runs.

	SAMPLES AND RUNTIME [s]						
	1	10	100	1k	2k	10k	20k
evolve()	3.03	3.33	4.93	5.44	7.17	29.00	68.56
ksevolve()	1.91	2.09	2.90	3.13	3.24	12.20	23.95
Reduction	36.9%	37.3%	41.3%	42.5%	54.9%	57.9%	65.1%

Table 5.3: CUDAjectory runtimes for different configurations, without trajectory storage. Average of 5 different runs.

	SAMPLES AND RUNTIME [s]						
	1	10	100	1k	2k	10k	20k
evolve()	2.97	3.22	4.69	5.01	5.12	6.10	8.60
ksevolve()	1.83	2.05	2.83	3.02	3.06	3.60	5.24
Reduction	38.5%	36.4%	39.7%	39.7%	40.2%	41.0%	39.0%

Table 5.4 shows the number of different sample outcomes between the different CUDAjectory calls, against a higher precision `evolve()` call with  $10^{-14}$  relative tolerance. The two simulation strategies appear to be equivalent in terms of statistical significance. While in general KS simulations are more precise and numerically stable, as shown in Chapter 2, the low value of absolute tolerance may have allowed the Cartesian simulation to bridge the accuracy gap with the KS variables, in this particular setup. In any case, the lower runtime makes KS simulations the most suitable choice, given that the overall statistical accuracy is preserved.

Table 5.4: CUDAjectory outcome difference against a  $10^{-14}$  relative tolerance `evolve()` simulation.

	SAMPLES						
	1	10	100	1k	2k	10k	20k
<code>evolve()</code>	0	0	0	0	0	2	6
<code>ksevolve()</code>	0	0	0	0	0	2	5

#### 5.4 SUMMARY

The implementation of KS variables successfully introduces notable performance improvements in CUDAjectory. One of the research results of this dissertation is therefore made available on operational software, allowing ESA to nearly halve the computational cost of massively parallel propagation-related applications, such as PP/SDM compliance analyses.

Future implementation works should mainly extend the software flexibility in KS variables. Given the regularity that KS variables introduce in UP (shown in Chapter 2), storing the full trajectories in KS variables would allow future researchers to benefit from the renewed computational efficiency, while having a full KS output at their disposal. Upon optimization and generalization, it may be worth attempting to implement PC-based propagators in CUDAjectory, comparing the efficiency against step-based schemes. In this case, the major limitation is represented by event detection algorithms: while the flyby detection concept has been validated in Chapter 4, the different programming paradigms required with GPUs may require a CUDA<sup>®</sup> implementation with remarkably different algorithms.

CUDAjectory remains a propagation-only related software, thereby limiting the extension possibility to improved input/output management, propagation techniques, and other dynamical formulations. Nonetheless, CUDAjectory may become the simulator at the core of more advanced UP software tools, that may exploit the potential of GPU computing even beyond the trajectory propagation block.



---

## CONCLUSION

---

The research presented in this dissertation has focused on different aspects of uncertainty propagation (UP), proposing novel dynamical and computational approaches to make PP/SDM compliance analyses more efficient. In addition, the influence of the dynamics formulation on the statistical description of the uncertainty has been studied, highlighting that the choice made on the equations of motion may be the enabling factor for the use of certain probabilistic approaches. Despite notable performance and accuracy improvements have been found in all the explored directions, only the combination of all the research areas affecting UP can eventually bring computational feasibility to the problem. So far, the either exclusively astrodynamical or computational approaches have been capable of providing good compromise solutions, fitting most practical needs, yet all feature clear conceptual limitations and are not applicable in all cases. Be that the long-term simulation of the debris environment or the mission compliance assessment with given policies, only the synergy between the research fields affecting UP can lead to stable and effective solutions to these computationally intense modeling challenges.

### 6.1 SUMMARY AND FINDINGS

#### 6.1.1 *KS variables for uncertainty propagation*

Chapter 2 explored the use of KS variables in high-fidelity orbit propagation and PP/SDM compliance analysis. The re-formulation of the orbital dynamics naturally introduces regularity to the problem, removing the non-linearity of the Kepler problem, as well as the near-singularity experienced at close distance from celestial bodies. Hence, KS variables build a more efficient trajectory propagation framework, regardless the perturbation sources introduced in the dynamical model. Nearly halved runtimes come in combination with an increased simulation accuracy of a factor 10, thereby establishing KS variables as a flexible yet robust formulation choice for any orbital problem, flybys included. These results reflect what also observed in the CUDAjectory implementation of KS variables in Chapter 5,

providing an excellent example of the suitability of this formulation of the dynamics with GPU computing architectures.

Chapter 2 also presents the preparatory work to run UP completely in KS variables. In particular, several mapping techniques have been studied for the conversion between Cartesian and KS uncertainties, proposing an analytical solution for the conversion of the position covariance of Gaussian distributions. The stronger non-linearity of the KS-Cartesian transformation prevents finding closed-form solution. Nonetheless, an Unscented Transform-based conversion approach has been devised, reaching highly accurate uncertainty mapping also for cross and velocity covariance terms. Next, using Monte Carlo simulated data, KS variables have been observed to introduce regularity and predictability even to whole uncertainty distributions. Here, the role played by the adaptive non-dimensionalization combined with the synchronization upon the fictitious KS time allowed the reconstruction of a continuum-like behavior of the uncertainty, even after flyby scattering events. Modeling the uncertainty directly in KS variables gives even further improvement, as the orbital dynamics reduces to a quasi-linear harmonic oscillator.

### 6.1.2 *Jacobian spheroids and the Keyhole Map*

Chapter 3 tested different approaches to detect and model shallow and distant interactions. Perturbation approaches applied on the hyperbolic two-body solution are a relatively unexplored topic, in contrast with the abundance of studies on Earth bound orbits. The lack of periodicity makes it difficult to replicate analog techniques: whilst in the elliptic case similar approaches work for both Earth's oblateness and the third body perturbation, the same consideration does not hold for hyperbolic trajectories.

Chapter 3 also proposes a novel concept of SOI, based on the Jacobian eigenvalues of the Barycentric Restricted Three Body Problem. This criterion bounds an extended region of space, compared to the traditional SOI and Hill's sphere definitions. This criterion perfectly complies with numerical analyses, that identified this zone as the most numerically stable switch distance to change the center of integration between Sun and the planet flown by. The dynamical meaning of Jacobian eigenvalues highlights that this criterion is better suited to track local changes in the dynamics function, compared to the "static" traditional definitions linked to gravitational forces and potential, hence fitting the fast and dynamical nature of flybys.

Finally, Chapter 3 parametrizes the Jacobian spheroids to study the effects of shallow and distant interactions in Earth-impacting trajectories. A simple surrogate model for the dynamics in the proximity of Earth allowed the construction of the Keyhole Map: high impact probability regions can be detected using this graphical statistical

tool, that finds threatening trajectories even in the case of nominally non-resonant orbital periods.

### 6.1.3 *Picard-Chebyshev for uncertainty propagation*

Chapter 4 analyzed strengths and weaknesses of iteration-based numerical schemes from an applied viewpoint. While step-based integrators are in general a more robust and flexible choice, some selective applications may obtain a relevant performance boost with a different simulation approach. The PC scheme has been applied to the optimization of a Solar Orbiter-like multi-flyby trajectory: the fixed-point logic of the integrator minimized the need of reading ephemeris data, making it feasible to use of a full-perturbation model of the interplanetary environment.

An augmented version of the PC numerical scheme is also proposed, to maximize the performance thereof when propagating large sets of initial conditions. The method builds an augmented system containing all the trajectories, and applies the PC iterations to this system as a whole. This approach optimizes the use of high-performance clusters and GPUs, as the parallelization of computation acts at the finest possible level, resulting in a highly scalable approach. The results shown in Chapter 4 highlight the large computational potential of the PC integration already at the current development level, for applications that do not require intensive event detection procedures.

## 6.2 LIMITATIONS AND APPLICABILITY

Rather than a true limitation, the major difficulty related to the extensive use of KS variables, or of advanced formulation of the dynamics in general, is perhaps the not so clear physical meaning thereof. Possibly, this is also the reason why KS variables and other formulations like DROMO have not captured yet a wide interest in the astrodynamics engineering community, mentioning for instance the abstraction operated on the very concept of time. Whilst application-only users may be hidden of KS variables, in line to what implemented in *CUDAjectory*, researchers working on uncertainty propagation would necessarily need to get acquainted with KS variables to exploit their full potential. This task is not so straightforward, as the orbital dynamics is basically removed of any physical meaning, becoming a purely mathematical construct devoted to increasing the regularity and predictability of the analyzed system. Nonetheless, true limitations to KS variables have not been found in the research discussed in this dissertations, although only additional testing on other environments can either confirm or dispute this claim, e.g. with space debris-related applications.

Jacobian spheroids proved to match the open question of finding a robust definition of SOI, even coming with an analytical solutions.

Nonetheless, despite the definition derivation based on the general three-body problem, systems with higher mass ratios, such as the Earth-Moon case, may not behave as Sun-planet systems. While these environments are in general studied on different dynamical models, this aspect represents an open point to the validation of the Jacobian spheroid concept, that could either confirm the suitability for Sun-planet-like systems only, or even open new research pathways related to the three-body dynamics.

The Keyhole Map has been obtained for a one encounter-only frame of analysis and only for the planar case. This may not be a true limitation for PP/SDM, as the trajectories to be analyzed remain approximately on the planet's orbital plane, however, adding the inclination variable would increase by one the problem dimensionality, leading to three-dimensional maps. The Keyhole Map is a statistical construct, only spotting high impact probability regions relying on a surrogate model of the planetary dynamics. The accuracy strongly depends on the choices made for this model and the complexity of each planetary environment: the  $\gamma$  parametrization ensures that the effects of other planets are statistically irrelevant inside the chosen Jacobian spheroid, whether and how much other internal effects influence the predicted keyholes needs however to be assessed.

The major limitation of the PC scheme is represented by the intrinsic lack of flexibility. Event routines need to be completely re-thought and made compliant with the algorithm logic, i.e. traditional approaches relying on persistent checks of event functions cannot be relied upon. Slicing the trajectory until a preliminary event is detected may be a viable approach, similarly to what presented in this dissertation for the detection of close approaches. However, whether running the slice identification process on the initial trajectory guess is accurate enough remains an open research point, and a major limitation to the adoption of PC-based tools in modern astrodynamics applications. The augmented version suffers of this condition as well, whose solution is unfortunately made even more complex by the need of managing the augmented system. Still, the computational potential of the augmented approach may provide sufficient push for future works to address this aspect.

### 6.3 REMARKS FOR FUTURE WORKS

The research presented in this dissertation has covered the various topics in sufficient detail to enable the combination of some of the outlined techniques. First of all, the PC scheme has already been tested with KS variables and resulted in great increase of the simulation efficiency. For Earth-related applications, such as modeling the evolution of space debris clouds or of the whole environment, the augmented version of the PC method may already be used, perhaps slicing the



trajectories when the object atmosphere re-entry is detected, perhaps already using KS variables. A dedicated work may be however needed for interplanetary and PP/SDM application, as robust and precise flyby detection routines for PC have not been developed yet.

For PP/SDM applications, the Keyhole Map introduces an approach with the potential to avoid large Monte-Carlo-based strategies altogether. The already accurate keyhole detection capabilities justify expanding the concept, aiming at including more cases and scenarios. First of all, a chaining strategy should be addressed to "collapse" the map of different encounters, possibly with multiple bodies as well, into a single one, completely representing the impact probability domain in a single graphical tool. Then, the computation of the overall impact probability could be remarkably simplified: all it would take is the superposition of the given uncertainty dispersion over the initial map, as a multi-dimensional probability density function, whose surface integral would finally give the global impact probability.

The work on CUDAJjectory could continue, expanding the running modes to allow users to retrieve trajectories written in KS variables. Although this update would not affect PP/SDM-related analyses, research users could benefit from trajectories written in this framework to continue the work on uncertainty propagation and modeling with KS variables, exploiting at the same time the computational potential of GPUs. This update may require a structural revision of CUDAJjectory, that is still strongly reliant to Cartesian coordinates, affecting the memory management.

KS variables have yielded notable results in any explored direction. On the numerical viewpoint, some other formulations outperform KS coordinates, DROMO-based for instance. Still, the straightforward interface between the Cartesian and the KS realms is an advantage in many applications, e.g. for the computation of perturbing accelerations and events. Hence, future works should compare the effective performance of other orbital formulation with KS variables, and assess whether the improved numerical stability outweighs the increased cost of retrieving the Cartesian states.

KS variables led to a highly regular and predictable evolution of the orbital uncertainty, even after being scattered by flyby events. These data outline a promising potential for KS variables to improve uncertainty propagation in general: for instance, fully probabilistic and continuum-based approaches should be implemented and tested in KS variables. Not only would KS variables raise the computational efficiency of such techniques, also the possibly present surrogate models should become more accurate and reliable.



# A

---

## QUATERNION AND TENSOR ALGEBRA

---

### A.1 QUATERNION ALGEBRA

Quaternions can be seen as a generalization of complex numbers, the formalism was first proposed by Hamilton back in the 1800s. The here provided short outline, necessary to introduce the concepts presented in Section 2.1.2, follows what presented in [154] and [156].

Quaternions can be seen as hypercomplex numbers, formed by a scalar and a vector part:

$$\mathbf{q} = q_0 + i_1 q_1 + i_2 q_2 + i_3 q_3 = q_0 + \begin{Bmatrix} q_1 \\ q_2 \\ q_3 \end{Bmatrix} \quad (\text{A.1})$$

with  $q_0, \dots, q_3$  real numbers. Similarly to complex numbers and vectors, only components of the same type can be summed together:

$$\begin{aligned} \mathbf{a} &= a_0 + i_1 a_1 + i_2 a_2 + i_3 a_3, \quad \mathbf{b} = b_0 + i_1 b_1 + i_2 b_2 + i_3 b_3 \\ \mathbf{a} + \mathbf{b} &= (a_0 + b_0) + i_1(a_1 + b_1) + i_2(a_2 + b_2) + i_3(a_3 + b_3) \end{aligned} \quad (\text{A.2})$$

The quaternion product (not commutative) is ruled by

$$\begin{aligned} i_1^2 &= i_2^2 = i_3^2 = i_1 i_2 i_3 = -1 \\ i_1 i_2 &= -i_2 i_1 = i_3 \\ i_2 i_3 &= -i_3 i_2 = i_1 \\ i_3 i_1 &= -i_1 i_3 = i_2 \end{aligned} \quad (\text{A.3})$$

The product by a real number is defined as:

$$\mathbf{c}\mathbf{q} = (cq_0) + i_1(cq_1) + i_2(cq_2) + i_3(cq_3) \quad (\text{A.4})$$

Given three quaternions  $(\mathbf{a}, \mathbf{b}, \mathbf{c})$  the associative law holds:

$$(\mathbf{ab})\mathbf{c} = \mathbf{a}(\mathbf{bc}) \quad (\text{A.5})$$

The conjugate  $\bar{\mathbf{q}}$  of  $\mathbf{q}$  is defined as

$$\bar{\mathbf{q}} = q_0 - i_1 q_1 - i_2 q_2 - i_3 q_3 = q_0 - \begin{Bmatrix} q_1 \\ q_2 \\ q_3 \end{Bmatrix} \quad (\text{A.6})$$

which then leads to the definition of modulus  $|\mathbf{q}|$  of a quaternion:

$$|\mathbf{q}|^2 = \mathbf{q}\bar{\mathbf{q}} = \bar{\mathbf{q}}\mathbf{q} = \sum_{n=0}^3 q_n^2 \quad (\text{A.7})$$

The conjugate of a quaternion has the further properties

$$\overline{\bar{\mathbf{q}}} = \mathbf{q}; \quad |\bar{\mathbf{q}}|^2 = |\mathbf{q}|^2; \quad \overline{(\mathbf{q}_1\mathbf{q}_2)} = \bar{\mathbf{q}}_2\bar{\mathbf{q}}_1 \quad (\text{A.8})$$

The definition of quaternion conjugate makes it possible to introduce the division between two quaternions. In particular

$$\mathbf{a}/\mathbf{b} = \mathbf{a}\mathbf{b}^{-1} \quad (\text{A.9})$$

with

$$\mathbf{b}^{-1} = \frac{\bar{\mathbf{b}}}{|\mathbf{b}|^2} \quad (\text{A.10})$$

Waldvogel proposed the definition of the star conjugate [156]  $\mathbf{q}^*$  of a quaternion, based on the anti-involute definition of Vivarelli [154]:

$$\mathbf{q}^* = q_0 + i_1 q_1 + i_2 q_2 - i_3 q_3 \quad (\text{A.11})$$

which features the properties

$$(\mathbf{q}^*)^* = \mathbf{q}; \quad |\mathbf{q}^*|^2 = |\mathbf{q}|^2; \quad (\mathbf{q}_1\mathbf{q}_2)^* = \mathbf{q}_2^*\mathbf{q}_1^* \quad (\text{A.12})$$

## A.2 TENSOR STATISTICS

The conformal squaring embedded in the KS transformation [139, 156] requires the introduction of higher order statistical moments when dealing with the conversion of covariance matrices. This section provides some basic concepts and some tensor notation for statistical moments, useful to write the compact equations presented in Section 2.3.

### A.2.1 Tensor algebra

Tensors can be thought as a generalization of matrices and vectors. A zero-dimensional tensor corresponds to a scalar, a one-dimensional tensor corresponds to a vector, and a two-dimensional tensor corresponds to a matrix. In the following lines,  $\mathbf{a}$  and  $\mathbf{b}$  denote two different tensors, of dimension  $m$  and  $n$  respectively. The *tensor* product, or *outer* product, denoted with  $\otimes$ , is defined by elements as:

$$(\mathbf{a} \otimes \mathbf{b})_{i,j} = a_i b_j \quad (\text{A.13})$$

The output is a tensor of dimension  $m + n$ .

The *inner* product, or *dot* product, denoted by  $\cdot$ , is defined by elements as:

$$(\mathbf{a} \cdot \mathbf{b})_{i,j} = \sum_k a_{ik} b_{kj} \tag{A.14}$$

The output is a tensor with dimension  $m + n - 2$ . The inner product reduces to the traditional scalar product in the case of vectors, and the matrix-matrix products for dimensions equal to one or to two. Both inner and outer product collapse to the classical product for scalars.

Finally, the superscript  $k \otimes$  denotes the  $k$ -th *tensor power*, defined as:

$$\mathbf{a}^{k \otimes} = \underbrace{\mathbf{a} \otimes \dots \otimes \mathbf{a}}_k \tag{A.15}$$

and it holds that

$$(\mathbf{a} \cdot \mathbf{b})^{k \otimes} = \mathbf{a}^{k \otimes} \cdot \mathbf{b}^{k \otimes} \tag{A.16}$$

### A.2.2 Tensor description of statistical moments

The key operator used in Section 2.3 is the *expectation* operator. Given a multi-dimensional random variable  $\mathbf{x}$  and its probability density function  $\mathbf{f}(\mathbf{x})$ , the expectation of the generic function  $\mathbf{g}(\mathbf{x})$  is defined as:

$$\mathbb{E}[\mathbf{g}(\mathbf{x})] = \int_{-\infty}^{+\infty} \mathbf{g}(\mathbf{x}) \mathbf{f}(\mathbf{x}) d\mathbf{x} \tag{A.17}$$

The expectation is a linear operator.

The expectation allows the definition of multivariate statistical moments. Using tensor notation, the  $k$ -th order multivariate raw moment of the first order tensor statistical variable  $\mathbf{x}$  becomes :

$$\mathbf{m}'_{\mathbf{x},k} = \mathbb{E}[\mathbf{x}^{k \otimes}] \tag{A.18}$$

The first order raw moment is also called *mean* and traditionally denoted by  $\mathbf{m}_{\mathbf{x}} = \mathbb{E}[\mathbf{x}]$ , omitting the "prime" symbol with a little notation abuse. The second order raw moment is also called *autocorrelation matrix*  $\mathbf{R}_{\mathbf{xx}}$ :

$$\mathbf{R}_{\mathbf{xx}} = \mathbf{m}'_{\mathbf{x},2} = \mathbb{E}[\mathbf{x}^{2 \otimes}] = \mathbb{E}[\mathbf{xx}^T] \tag{A.19}$$

The  $k$ -th order multivariate central moment of  $\mathbf{x}$  is defined with respect to the mean:

$$\mathbf{m}_{\mathbf{x},k} = \mathbb{E}[(\mathbf{x} - \mathbf{m}_{\mathbf{x}})^{k \otimes}] \tag{A.20}$$

The second order central moment is commonly called *covariance matrix*  $\mathbf{P}_{\mathbf{xx}}$ :

$$\mathbf{P}_{\mathbf{xx}} = \mathbf{m}_{\mathbf{x},2} = \mathbb{E}[(\mathbf{x} - \mathbf{m}_{\mathbf{x}})^{2 \otimes}] = \mathbb{E}[(\mathbf{x} - \mathbf{m}_{\mathbf{x}})(\mathbf{x} - \mathbf{m}_{\mathbf{x}})^T] = \mathbf{R}_{\mathbf{xx}} - \mathbf{m}_{\mathbf{x}} \mathbf{m}_{\mathbf{x}}^T \tag{A.21}$$

*The superscript ' marks the raw moments, whereas the central ones are written as clean symbols.*



# B

---

## PERTURBATIONS BY LIE TRANSFORMS

---

For generic Hamiltonian systems, perturbation approaches aim to find a canonical transformation that makes the problem integrable, and then retrieves the perturbation effects by a change of variables. The following lines summarize the Lie transform-based perturbation approach, following the works of Hori [65] and Deprit [35]. In the following lines, the pair of generalized coordinates and conjugate momenta are denoted by  $(\xi, \Xi)$ .

### B.1 LIE TRANSFORMS SUMMARY

Given the canonical coordinate transformation  $\mathcal{X} : (\xi, \Xi, \varepsilon) \mapsto (\xi', \Xi')$  generated by

$$\mathcal{W} = \mathcal{W}(\xi', \Xi', \varepsilon) \equiv \sum_{j=0}^{+\infty} \frac{\varepsilon^j}{j!} \mathcal{W}_{j+1}(\xi', \Xi') \quad (\text{B.1})$$

the Lie transform of the composite function

$$G = G(\xi, \Xi, \varepsilon) \equiv F(\xi'(\xi, \Xi, \varepsilon), \Xi'(\xi, \Xi, \varepsilon), \varepsilon) \quad (\text{B.2})$$

for the first order derivative with respect to  $\varepsilon$  is defined as

$$\frac{dG}{d\varepsilon} = \frac{\partial F}{\partial \varepsilon} + \{F; \mathcal{W}\} \quad (\text{B.3})$$

$\{F; \mathcal{W}\}$  is the Poisson brackets operator

$$\{F; \mathcal{W}\} = \frac{\partial F}{\partial \mathbf{x}} \cdot \mathbf{K} \frac{\partial \mathcal{W}}{\partial \mathbf{x}} \quad (\text{B.4})$$

where, denoting with  $(\mathbf{o}, \mathbf{I})$  the  $N \times N$  null and identity matrices respectively and  $N$  the number of generalized coordinates  $\xi$ , the coordinates-momenta vector  $\mathbf{x}$  and the symplectic matrix  $\mathbf{K}$  are:

$$\mathbf{x} = \begin{Bmatrix} \xi \\ \Xi \end{Bmatrix} \quad \text{and} \quad \mathbf{K} = \begin{bmatrix} \mathbf{o} & \mathbf{I} \\ -\mathbf{I} & \mathbf{o} \end{bmatrix} \quad (\text{B.5})$$

If  $F$  is given as a Taylor series expansion

$$F \equiv \sum_{j=0}^{+\infty} \frac{\varepsilon^j}{j!} F_{j,0}(\xi', \Xi') \quad (\text{B.6})$$

with  $F_{j,0} = d^j F / d\varepsilon^j |_{\varepsilon=0}$ , writing explicitly the expansions for both  $F$  and  $\mathcal{W}$  gives

$$\frac{dG}{d\varepsilon} = \sum_{j=0}^{+\infty} \frac{\varepsilon^j}{j!} F_{j,1}(\xi', \Xi') \quad (\text{B.7})$$

where

$$F_{j,1}(\xi', \Xi') = F_{j+1,0}(\xi', \Xi') + \sum_{l=0}^j \frac{j!}{(j-l)! l!} \{F_{j-l,0}; \mathcal{W}_{l+1}\} \quad (\text{B.8})$$

and in particular

$$\left. \frac{dG}{d\varepsilon} \right|_{\varepsilon=0} = F_{0,1}(\xi', \Xi') |_{\varepsilon=0} = F_{0,1}(\xi, \Xi) \quad (\text{B.9})$$

By induction, the  $q$ -th order for  $G$  becomes

$$\frac{d^q G}{d\varepsilon^q} = \sum_{j=0}^{+\infty} \frac{\varepsilon^j}{j!} F_{j,q}(\xi', \Xi') \quad (\text{B.10})$$

and similarly to the first order term

$$\left. \frac{d^q G}{d\varepsilon^q} \right|_{\varepsilon=0} = F_{0,q}(\xi', \Xi') |_{\varepsilon=0} = F_{0,q}(\xi, \Xi) \quad (\text{B.11})$$

which is computed with the aid of the Deprit's triangle recursion:

$$F_{j,q+1} = F_{j+1,q} + \sum_{l=0}^j \frac{j!}{(j-l)! l!} \{F_{j-l,q}; \mathcal{W}_{l+1}\} \quad (\text{B.12})$$

Summarizing, the Lie transform method allows to write  $G$  as a Taylor series expansion on the new variables  $(\xi, \Xi)$ :

$$G = \sum_{j=0}^{+\infty} \frac{\varepsilon^j}{j!} G_{j,0}(\xi, \Xi) = \sum_{j=0}^{+\infty} \frac{\varepsilon^j}{j!} F_{0,j}(\xi, \Xi) \quad (\text{B.13})$$

with the notation  $G_{j,0} = d^j G / d\varepsilon^j |_{\varepsilon=0}$  having the same definition of  $F_{j,0}$ .

## B.2 PERTURBATIONS APPROACH

In general, given:

$$\mathcal{H} = \sum_{j=0}^{+\infty} \frac{\varepsilon^j}{j!} \mathcal{H}_{j,0}(\xi, \Xi) \quad (\text{B.14})$$

the aim is to find the transformation  $\mathcal{X} : (\xi, \Xi, \varepsilon) \mapsto (\xi', \Xi')$  generated by

$$\mathcal{W} = \sum_{j=0}^{+\infty} \frac{\varepsilon^j}{j!} \mathcal{W}_{j+1,0}(\xi', \Xi') \quad (\text{B.15})$$



so that the transformed Hamiltonian is defined as

$$\mathcal{H} \circ \mathcal{X} = \sum_{j=0}^{+\infty} \frac{\varepsilon^j}{j!} \mathcal{H}_{0,j}(\xi', \Xi') \quad (\text{B.16})$$

requiring the terms  $\mathcal{H}_{0,j}$  to have some desired properties.

The analytical procedure based on Deprit's recursion (Equation (B.12)) at the order  $n$  can be written as the following steps:

1. The recursion gives  $\mathcal{H}_{0,n} = \mathcal{H}_{1,n-1} + \{\mathcal{H}_{0,n-1}; \mathcal{W}_1\}$  at the  $n$ -th step, which can be re-arranged to obtain the homological equation:

$$\{\mathcal{W}_n; \mathcal{H}_{0,0}\} + \mathcal{H}_{0,n} = \tilde{\mathcal{H}}_n \quad (\text{B.17})$$

or, highlighting the Lie derivative operator  $\mathcal{L}_0(-) = \{-, \mathcal{H}_{0,0}\}$

$$\mathcal{L}_0(\mathcal{W}_n) = \tilde{\mathcal{H}}_n - \mathcal{H}_{0,n} \quad (\text{B.18})$$

2.  $\mathcal{H}_{0,n}$  is chosen according to the desired properties of the final system, whereas  $\tilde{\mathcal{H}}_n$  can be computed from the terms already computed in the previous steps of the recursion.
3.  $\mathcal{W}_n$  is obtained from the solution of a partial differential equation.

The coordinates  $(\xi, \Xi)$  can be expressed as a series expansion using the computed function  $\mathcal{W}$  as well:

$$\begin{aligned} \xi &= \sum_{j=0}^{+\infty} \frac{\varepsilon^j}{j!} \xi_{0,j}(\xi', \Xi') \\ \Xi &= \sum_{j=0}^{+\infty} \frac{\varepsilon^j}{j!} \Xi_{0,j}(\xi', \Xi') \end{aligned} \quad (\text{B.19})$$

where the computation of the terms  $(\xi_{0,j}, \Xi_{0,j})$  follows once again the Deprit's recursion (Equation (B.12)) for the known generating function  $\mathcal{W}$ .

The initial condition for the integration in prime variables  $(\xi', \Xi')$  at the starting time  $t = t_0$  is simply given by  $(\xi', \Xi')|_{t=t_0} = (\xi, \Xi)|_{t=t_0}$ .





---

## PARALLEL AND GPU COMPUTING FUNDAMENTALS

---

Parallelizable programs are characterized by some of their parts that can be executed simultaneously. Conceptually, task-parallel and data-parallel routines may exist: the former features completely independent tasks that do not need to be executed one before the other, the latter is identified by a common and repeated task that should be executed on different data. This work focuses on this second aspect, whose features enable efficient GPU computing when massive parallelization is possible. For instance, taking the product of two matrices is a highly parallelizable task, as any element of the result matrix could be computed in parallel. Similarly, in the orbital propagation of large sets of initial conditions<sup>1</sup> the dynamics function could be evaluated in parallel for all the states.

### C.1 MULTI-CPU SHARED MEMORY PARALLELISM: OPENMP

Prior to discussing the rationale of GPU computing, the most straightforward parallelization concept involves the use of multiple CPUs (Central Processing Units). Shared memory parallelism is the simplest scenario, where all the machine compute cores have access to the same and common memory locations. More complex supercomputers use however a distributed memory logic, where group of CPUs access their own independent memory. Such systems also include a communication network, to distribute and collect the computed data on the different nodes, and require their own programming paradigm that includes the data and message passing routines [46].

OpenMP is a set of pre-processor instructions that enable simple shared memory parallelization of C, C++ and Fortran programs [28], which take action during the code compilation. Only minor modifications are required to accelerate the most intensive parts of the program, and OpenMP instructions are simply ignored and treated as comments if the OpenMP compilation flag is disabled. Common parallelization instructions involve for loops, which can include some extra functionality: for instance, perfectly nested loops can be collapsed into

---

<sup>1</sup> Assuming they are not interacting with each other and have negligible mass.

a single larger loop, increasing the program efficiency, and specific instructions can aid the memory management. Despite the simple interface, most programs require at least some variables to be declared private to each worker, since OpenMP treats all the variables as shared by default. The typical case of variables that should be made private are the loop counters: each thread must work with its own variable, particularly for those cases where the loop counter also identifies the position in a shared array where to access data. More complex programming scenarios often arise and require the programmer to explicitly control concurrent updates of shared variables, it is beyond the scopes of this work to tackle them all. The reader can refer to the OpenMP programming guide for more complete and detailed information [28]. This section serves as a simple introduction to the tool that is used to parallelize the C version of the augmented PC method proposed in Chapter 4.

## C.2 PARALLEL REDUCTION

The parallelization potential is also exploited in the computation of the PC iteration error, through the parallel reduction mechanism. Reduction tasks are a broad category of compute operations, whose aim is to extract a single scalar value from an array of elements. Some examples are the sum of array elements, finding the maximum or minimum element in an array, "and" and "or" logical operators. Despite appearing intrinsically sequential tasks, parallelization possibilities do exist even in the reduction case: in principle, the whole array is split into several chunks on the parallel workers, which cooperate to perform the reduction task on their own chunk of elements. The cooperated process continues, until a single scalar reduced value is obtained. Figure C.1 shows a graphical example of the parallel reduction logic: on an array of 16 elements, only 5 sequential steps are eventually required with 16 parallel workers. OpenMP implements the reduction clause among its functions, the programmer is only asked to define the final reduced scalar as a shared variable [28]. The compiler then ensures that all the array values are scanned through, and avoids simultaneous overwriting of the reduced scalar.

## C.3 GPU COMPUTING

The features of GPU computing arise from the hardware architecture of graphics cards, which is profoundly different from the traditional compute units. Figure C.2 shows a graphical representation of such differences: in summary, more transistors are devoted to pure data processing on GPUs, instead of flow control and memory management as in the CPU case [112]. Corresponding colors refer to chip elements of the same type, i.e. green for compute cores, yellow for control units,

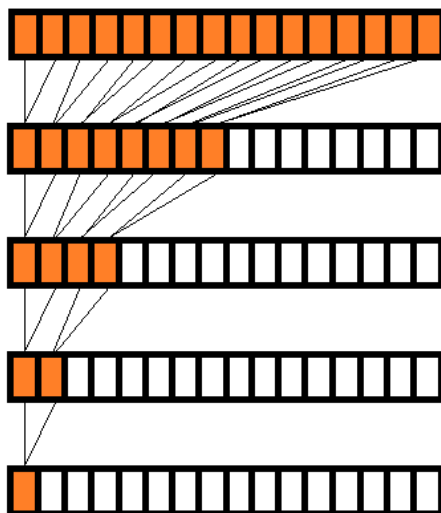


Figure C.1: Parallel reduction graphical scheme.

purple for the core-level cached memory, blue for shared cache, and orange for global memory.

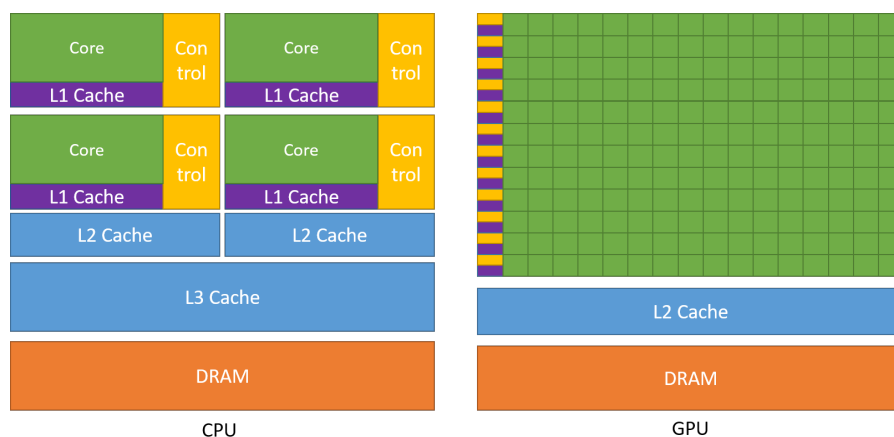


Figure C.2: CPU vs GPU architecture difference graphical scheme. Picture from [112].

The processing units are grouped in blocks (typically 32 processing units per block), each controlled by one controller. All processing units in the same block all execute the same instruction, issued only once by the controller. This aspect, together with the normally hundreds to thousands of processing units available in modern graphics cards, makes GPUs prone to implement massive parallelism, although with lower flexibility and higher programming effort compared to CPU applications. Some key concepts are given in the following subsection, a comprehensive view can be found in the CUDA<sup>®</sup> C++ programming guide [112].

## C.4 MAIN PROGRAMMING PARADIGMS AND THE CUDA API

This section is intended to provide a brief overview and nomenclature of the CUDA<sup>®</sup> language and API. Italic font is used to introduce CUDA<sup>®</sup>-specific names and concepts. A complete description can be found in the CUDA<sup>®</sup> user manual [112].

The fundamental execution unit is called *thread*. Threads can be grouped in *blocks*, and some *shared memory* (dozens of kilobytes) is available to all threads in a common block. The execution of a single instruction is always performed by groups of 32 threads at the same time, called a *warp*, regardless the number of threads in a block. Therefore, blocks with less than 32 threads do not exploit the full hardware resources. Complex configurations can be achieved combining multiple CPUs and GPUs to run the same program. The program flow is always controlled by the CPU, which also controls the execution of the GPU. A function that is invoked by the CPU but executed on the GPU is called *kernel*.

The first difference when comparing GPU codes to standard programming regards the memory access: the GPU cannot read the usual compute memory, but data must be loaded on the GPU memory prior to running kernels. In a similar manner, data must be retrieved to the CPU after the kernel has completed its execution, before executing other CPU tasks on the same data. Consequently, the flow of a GPU program/sub-program always follows a first initialization on the CPU, a subsequent data movement from the CPU to the GPU, the kernel execution, and finally the data retrieval from the GPU to the CPU, as represented in Figure C.3.

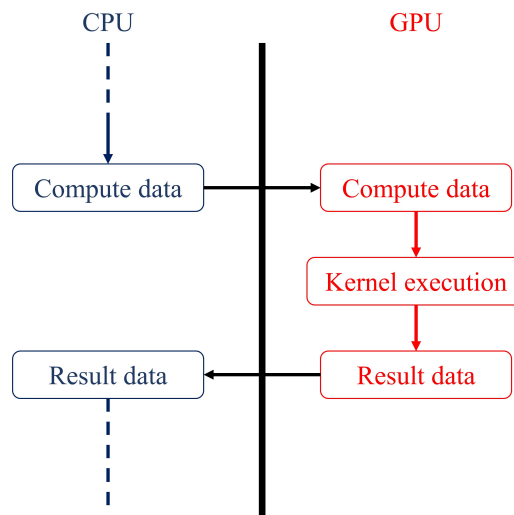


Figure C.3: GPU program basic flow.

Because all threads execute the same instruction at the same time, kernels must be programmed in a warp-oriented manner. This also includes explicitly managing the data access by the various threads,

and complex functions typically require the programmer to optimize memory access and cache utilization by hand: while it is implicitly controlled by the compiler for CPUs, graphics card store data by default in the so called *global* memory, large in size (some gigabytes) and referred to by default when GPU variables are initialized. This eases data movements between CPU and GPU, even in case of large arrays. However, the access latency is much higher compared to shared memory: this makes global memory not efficient for repeated read/write operations with GPU variables. To overcome this limitation, shared memory can be exploited for compute purposes to keep read-only values on lower latency locations, instead of holding common values for all the threads in a block only. Few cached bytes are also available on *registers*, thread-private locations to store up to 256 single precision floating point values. The small size and the compute intensity of the relativistic dynamics function makes it difficult to use registers instead of the shared memory for the proposed application. Their use is limited to temporary and handle variables that aid the final acceleration computation.

The way array elements are sorted is also fundamental for optimized data movements across global and shared GPU memory. While CPU-GPU data transfers are specified by the array size, the most efficient intra-GPU memory access happens when threads read/write values on adjacent locations. If this condition is satisfied, data are moved as a single memory transaction for all the threads, resulting in minimized cycles spent reading or writing on global memory. This memory access pattern is called *coalesced*, and represents a fundamental performance driver in complex GPU programs: even if a program has massive parallelization possibilities, non-optimal memory access may result in the memory access latency not compensated at all by the parallelized computational tasks.

NVIDIA<sup>®</sup> developed and maintains the CUDA<sup>®</sup> programming language, which remarkably simplifies the use of NVIDIA<sup>®</sup> GPUs in computer programs. It is built as a C++ extension, with a set of keywords and API functions that allow programmers to build their own kernels and control the GPU execution flow. A set of optimized libraries is also available, for instance the basic linear algebra cuBLAS<sup>®</sup> implemented for the PC matrix multiplications of this work [112].

## C.5 CONCURRENCY AND ADVANCED FEATURES

The CPU-GPU duality and cooperation exposes more possibilities, other than the simple acceleration of intensive parts of the program. In general, kernel calls are asynchronous and do not block the work of the CPU, which allows the CPU to execute other tasks while a GPU kernel is still running. Furthermore, modern GPUs can manage time two memory transfers at the same (one per direction, CPU to GPU

and GPU to CPU), while also saturating its compute units for one or more concurrent kernel execution [112].

CUDA<sup>®</sup> allows the queuing of a series of sequentially dependent GPU function calls with the use of *streams*: for instance, an application may require some data to be transferred to the GPU before the execution of a custom kernel, which must necessarily be completed before calling a cuBLAS<sup>®</sup> function, at the end of which the processed data should be transferred back to the CPU. All it takes is assigning the sequentially dependent GPU function calls to the same stream. Multiple streams can be created and used at the same time, the obtained behaviour mimics batch job submissions to super-computing facilities. CUDA<sup>®</sup> guarantees the correct execution line and synchronization within the same stream, whereas different streams must instead be synchronized with each other by hand. The compiler typically schedules executions and memory transactions so that the GPU use is maximized, superposing different GPU function calls and data transfers from separate streams [112].

Despite the lower latency, optimized programs are designed to also control the way threads access shared memory locations. If two adjacent threads are asked to access two non-contiguous array elements, then the memory access is performed on two cycles instead of a single one, slightly slowing down the program execution. This issue is called *bank conflict*, and can be avoided ensuring thread-varying elements to be stored in the leading dimension of shared memory arrays. Bank conflicts do not happen if all threads access the same single memory location.



---

## BIBLIOGRAPHY

---

- [1] S. J. Aarseth and K. Zare, "A regularization of the three-body problem," *Celestial mechanics*, vol. 10, no. 2, pp. 185–205, Oct. 1974, ISSN: 1572-9478. DOI: [10.1007/BF01227619](https://doi.org/10.1007/BF01227619).
- [2] C. H. Acton, "Ancillary data services of NASA's Navigation and Ancillary Information Facility," *Planetary and Space Science*, vol. 44, no. 1, pp. 65–70, Jan. 1996, ISSN: 0032-0633. DOI: [10.1016/0032-0633\(95\)00107-7](https://doi.org/10.1016/0032-0633(95)00107-7).
- [3] D. Amato, G. Baù, and C. Bombardelli, "Accurate orbit propagation in the presence of planetary close encounters," *Monthly Notices of the Royal Astronomical Society*, vol. 470, no. 2, pp. 2079–2099, Sep. 2017, ISSN: 0035-8711. DOI: [10.1093/mnras/stx1254](https://doi.org/10.1093/mnras/stx1254).
- [4] H. Arakida and T. Fukushima, "Long-term integration error of Kustaanheimo-Stiefel regularized orbital motion. ii. method of variation of parameters," *The Astronomical Journal*, vol. 121, no. 3, pp. 1764–1767, Mar. 2001, ISSN: 0004-6256. DOI: [10.1086/319408](https://doi.org/10.1086/319408).
- [5] R. Armellin, P. Di Lizia, F. Bernelli-Zazzera, and M. Berz, "Asteroid close encounters characterization using differential algebra: The case of Apophis," *Celestial Mechanics and Dynamical Astronomy*, vol. 107, no. 4, pp. 451–470, Aug. 2010, ISSN: 1572-9478. DOI: [10.1007/s10569-010-9283-5](https://doi.org/10.1007/s10569-010-9283-5).
- [6] P. Arzt and A. Rocchi, "Efficient Atmospheric Drag Prediction for Trajectory Propagation on GPUs," Master's thesis, Darmstadt University of Technology, 2023.
- [7] A. M. Atallah, R. M. Woollands, T. A. Elgohary, and J. L. Junkins, "Accuracy and efficiency comparison of six numerical integrators for propagating perturbed orbits," *Journal of the Astronautical Sciences*, vol. 67, no. 2, pp. 511–538, Jun. 2020, ISSN: 2195-0571. DOI: [10.1007/s40295-019-00167-2](https://doi.org/10.1007/s40295-019-00167-2).
- [8] X. Bai and J. L. Junkins, "Solving initial value problems by the Picard-Chebyshev method with NVIDIA GPUs," in *Advances in the Astronautical Sciences*, 2010, ISBN: 9780877035602.
- [9] X. Bai and J. L. Junkins, "Modified Chebyshev-Picard iteration methods for orbit propagation," *Journal of the Astronautical Sciences*, vol. 58, no. 4, pp. 583–613, Oct. 2011, ISSN: 2195-0571. DOI: [10.1007/BF03321533](https://doi.org/10.1007/BF03321533).

- [10] G. Baù, C. Bombardelli, J. Peláez, and E. Lorenzini, “Non-singular orbital elements for special perturbations in the two-body problem,” *Monthly Notices of the Royal Astronomical Society*, vol. 454, no. 3, pp. 2890–2908, Dec. 2015, ISSN: 0035-8711. DOI: [10.1093/mnras/stv2106](https://doi.org/10.1093/mnras/stv2106).
- [11] G. Baù and J. Roa, “Uniform formulation for orbit computation: The intermediate elements,” *Celestial Mechanics and Dynamical Astronomy*, vol. 132, no. 2, p. 10, Feb. 2020, ISSN: 1572-9478. DOI: [10.1007/s10569-020-9952-y](https://doi.org/10.1007/s10569-020-9952-y).
- [12] M. Berz, “Modern map methods in particle beam physics,” in *Dynamics of Particles and Fields*, Elsevier, 1999, pp. 1–79. DOI: [10.1016/s1076-5670\(08\)70227-1](https://doi.org/10.1016/s1076-5670(08)70227-1).
- [13] D. Blair, *Inversion Theory and Conformal Mapping*. The Student Mathematical Library, 2000, vol. 9, ISBN: 978-1-4704-1816-8. DOI: [10.1090/stml/009](https://doi.org/10.1090/stml/009).
- [14] D. Boccaletti and G. Pucacco, *Theory of Orbits*. Astronomy and Astrophysics Library, 1996, vol. 1, ISBN: 978-3-662-03319-7. DOI: [10.1007/978-3-662-03319-7](https://doi.org/10.1007/978-3-662-03319-7).
- [15] V. R. Bond, “The uniform, regular differential equations of the KS transformed perturbed two-body problem,” *Celestial mechanics*, vol. 10, no. 3, pp. 303–318, Sep. 1974, ISSN: 1572-9478. DOI: [10.1007/BF01586860](https://doi.org/10.1007/BF01586860).
- [16] S. Boone and J. McMahan, “Directional state transition tensors for capturing dominant nonlinear effects in orbital dynamics,” *Journal of Guidance, Control, and Dynamics*, vol. 46, no. 3, pp. 431–442, 2023. DOI: [10.2514/1.G006910](https://doi.org/10.2514/1.G006910).
- [17] A. Boutonnet and A. Rocchi, “Taxonomy of Earth Impacting Trajectories: the Case of JUICE Launcher Upper Stage,” in *AIAA SciTech 2022 Forum*, San Diego (CA), USA, 2022. DOI: [10.2514/6.2022-2460](https://doi.org/10.2514/6.2022-2460).
- [18] S. Breiter and K. Langner, “Kustaanheimo-Stiefel transformation with an arbitrary defining vector,” *Celestial Mechanics and Dynamical Astronomy*, vol. 128, no. 2, pp. 323–342, Jun. 2017, ISSN: 1572-9478. DOI: [10.1007/s10569-017-9754-z](https://doi.org/10.1007/s10569-017-9754-z).
- [19] S. Breiter and K. Langner, “The extended Lissajous–Levi-Civita transformation,” *Celestial Mechanics and Dynamical Astronomy*, vol. 130, no. 10, p. 68, Oct. 2018, ISSN: 1572-9478. DOI: [10.1007/s10569-018-9862-4](https://doi.org/10.1007/s10569-018-9862-4).
- [20] S. Breiter and K. Langner, “The Lissajous–Kustaanheimo-Stiefel transformation,” *Celestial Mechanics and Dynamical Astronomy*, vol. 131, no. 2, p. 9, Feb. 2019, ISSN: 1572-9478. DOI: [10.1007/s10569-019-9887-3](https://doi.org/10.1007/s10569-019-9887-3).

- [21] S. Campagnola, A. Boutonnet, J. Schoenmaekers, D. J. Grebow, A. E. Petropoulos, and R. P. Russell, "Tisserand-leveraging transfers," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 4, pp. 1202–1210, Jul. 2014. DOI: [10.2514/1.62369](https://doi.org/10.2514/1.62369).
- [22] S. Campagnola and Y. Kawakatsu, "Three-dimensional resonant hopping strategies and the Jupiter magnetospheric orbiter," *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 1, pp. 340–344, Jan. 2012. DOI: [10.2514/1.53334](https://doi.org/10.2514/1.53334).
- [23] S. Campagnola and R. P. Russell, "Endgame problem part 2: Multibody technique and the Tisserand-Poincare graph," *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 2, pp. 476–486, Mar. 2010. DOI: [10.2514/1.44290](https://doi.org/10.2514/1.44290).
- [24] G. Campiti, A. Masat, and C. Colombo, "Resonant Flybys in the B-plane: extension of the theory to elliptical planetary orbits and dynamic programming application," Master's thesis, Politecnico di Milano, 2021. [Online]. Available: <http://hdl.handle.net/10589/179968>.
- [25] A. Carusi, G. B. Valsechi, and R. Greenberg, "Planetary close encounters: Geometry of approach and post-encounter orbital parameters," *Celestial Mechanics and Dynamical Astronomy*, vol. 49, no. 2, pp. 111–131, Jun. 1990, ISSN: 1572-9478. DOI: [10.1007/BF00050709](https://doi.org/10.1007/BF00050709).
- [26] I. Cavallari, C. Grassi, G. F. Gronchi, G. Baù, and G. B. Valsecchi, "A dynamical definition of the sphere of influence of the earth," *Communications in Nonlinear Science and Numerical Simulation*, vol. 119, p. 107091, 2023, ISSN: 1007-5704. DOI: [10.1016/j.cnsns.2023.107091](https://doi.org/10.1016/j.cnsns.2023.107091).
- [27] M. Ceriotti and C. R. McInnes, "Design of ballistic three-body trajectories for continuous polar earth observation in the Earth-Moon system," *Acta Astronautica*, vol. 102, pp. 178–189, Sep. 2014, ISSN: 0094-5765. DOI: [10.1016/j.actaastro.2014.06.001](https://doi.org/10.1016/j.actaastro.2014.06.001).
- [28] R. Chandra, L. Dagum, D. Kohr, R. Menon, D. Maydan, and J. McDonald, *Parallel programming in OpenMP*. Morgan kaufmann, 2001.
- [29] C. Colombo, F. Letizia, and J. Van Der Eynde, "SNAPPshot ESA planetary protection compliance verification software Final report V1.0, Technical Report ESA-IPL-POM-MB-LE-2015-315," University of Southampton, Tech. Rep., 2016.
- [30] C. Colombo, M. Romano, and A. Masat, "SNAPPshot ESA planetary protection compliance verification software Final report V 2.0, Technical Report ESA-IPL-POM-MB-LE-2015-315," Politecnico di Milano, Tech. Rep., 2020.

- [31] COSPAR - Committee on Space Research, "Cospar policy on planetary protection," *Space Research Today*, vol. 208, pp. 10–22, Aug. 2020, ISSN: 1752-9298. DOI: [10.1016/j.srt.2020.07.009](https://doi.org/10.1016/j.srt.2020.07.009).
- [32] *CUDAjectory documentation*, Last access: May 2023. [Online]. Available: <https://ad.space-codev.org/cudajjectory/>.
- [33] F. Debatin, A. Tilgner, and F. Hechler, "Fast numerical integration of interplanetary orbits," in *ESA Special Publication*, ser. ESA Special Publication, vol. 255, Dec. 1986, pp. 329–333.
- [34] K. J. DeMars, R. H. Bishop, and M. K. Jah, "Entropy-based approach for uncertainty propagation of nonlinear dynamical systems," *Journal of Guidance, Control, and Dynamics*, vol. 36, no. 4, pp. 1047–1057, 2013. DOI: [10.2514/1.58987](https://doi.org/10.2514/1.58987).
- [35] A. Deprit, "Canonical transformations depending on a small parameter," *Celestial Mechanics*, vol. 1, no. 1, pp. 12–30, Mar. 1969. DOI: [10.1007/BF01230629](https://doi.org/10.1007/BF01230629).
- [36] A. Deprit, "Ideal Elements for Perturbed Keplerian Motions," *Journal of Research of the Notional Bureau of Standards - B. Mathematical Sciences*, vol. 79B, pp. 1–15, Jan. 1975. [Online]. Available: [https://nvlpubs.nist.gov/nistpubs/jres/79B/jresv79Bn1-2p1\\_A1b.pdf](https://nvlpubs.nist.gov/nistpubs/jres/79B/jresv79Bn1-2p1_A1b.pdf).
- [37] A. Deprit, "The Lissajous transformation i. basics," *Celestial Mechanics and Dynamical Astronomy*, vol. 51, no. 3, pp. 201–225, Sep. 1991, ISSN: 1572-9478. DOI: [10.1007/BF00051691](https://doi.org/10.1007/BF00051691).
- [38] A. Deprit and A. Elipe, "The Lissajous transformation ii. normalization," *Celestial Mechanics and Dynamical Astronomy*, vol. 51, no. 3, pp. 227–250, Sep. 1991, ISSN: 1572-9478. DOI: [10.1007/BF00051692](https://doi.org/10.1007/BF00051692).
- [39] A. Deprit, A. Elipe, and S. Ferrer, "Linearization: Laplace vs. Stiefel," *Celestial Mechanics and Dynamical Astronomy*, vol. 58, no. 2, pp. 151–201, Feb. 1994, ISSN: 1572-9478. DOI: [10.1007/BF00695790](https://doi.org/10.1007/BF00695790).
- [40] A. Deprit and C. A. Williams, "The Lissajous transformation iv. Delaunay and Lissajous variables," *Celestial Mechanics and Dynamical Astronomy*, vol. 51, no. 3, pp. 271–280, Sep. 1991, ISSN: 1572-9478. DOI: [10.1007/BF00051694](https://doi.org/10.1007/BF00051694).
- [41] EADS-Astrium, "Solar orbiter," *Journal of Physics: Conference Series*, vol. 271, p. 011 004, Jan. 2011, ISSN: 1742-6596. DOI: [10.1088/1742-6596/271/1/011004](https://doi.org/10.1088/1742-6596/271/1/011004).
- [42] European Space Agency, *ESA Space Debris Mitigation guidelines*, Last access: June 2023. [Online]. Available: [https://www.esa.int/Space\\_Safety/Space\\_Debris/Mitigating\\_space\\_debris\\_generation](https://www.esa.int/Space_Safety/Space_Debris/Mitigating_space_debris_generation).

- [43] European Space Agency (ESA), "Solar Orbiter Definition Study Report (Red Book)," Tech. Rep. July, 2011. [Online]. Available: <https://sci.esa.int/s/w7y04P8>.
- [44] European Space Agency (ESA), "Jupiter ICy moons Explorer Exploring the emergence of habitable worlds around gas giants. Definition Study Report.," Tech. Rep. 1.0, 2014, p. 128. [Online]. Available: <https://sci.esa.int/web/juice/-/54994-juice-definition-study-report>.
- [45] L. Floria, "A Simple Derivation of the Hyperbolic Delaunay Variables," *The Astronomical Journal*, vol. 110, p. 940, Aug. 1995. DOI: [10.1086/117575](https://doi.org/10.1086/117575).
- [46] M. P. Forum, "Mpi: A message-passing interface standard," USA, Tech. Rep., 1994.
- [47] S. Frey, "Evolution and hazard analysis of orbital fragmentation continua," Ph.D. dissertation, Politecnico di Milano, Supervisors: Colombo, Camilla and Lemmens, Stijn and Krag, Holger, Jul. 2020. [Online]. Available: <http://hdl.handle.net/10589/165144>.
- [48] T. Fukushima, "Picard iteration method, Chebyshev polynomial approximation, and global numerical integration of dynamical motions," *The Astronomical Journal*, vol. 113, pp. 1909–1914, May 1997. DOI: [10.1086/118404](https://doi.org/10.1086/118404).
- [49] T. Fukushima, "Vector Integration of Dynamical Motions by the Picard-Chebyshev Method," *The Astronomical Journal*, vol. 113, p. 2325, Jun. 1997. DOI: [10.1086/118443](https://doi.org/10.1086/118443).
- [50] T. Fukushima, "Efficient integration of highly eccentric orbits by quadruple scaling for Kustaanheimo-Stiefel regularization," *The Astronomical Journal*, vol. 128, no. 6, pp. 3108–3113, Dec. 2004, ISSN: 0004-6256. DOI: [10.1086/425630](https://doi.org/10.1086/425630).
- [51] T. Fukushima, "Efficient integration of highly eccentric orbits by scaling methods applied to Kustaanheimo-Stiefel regularization," *The Astronomical Journal*, vol. 128, no. 6, pp. 3114–3122, Dec. 2004, ISSN: 0004-6256. DOI: [10.1086/425553](https://doi.org/10.1086/425553).
- [52] T. Fukushima, "Efficient orbit integration by Kustaanheimo-Stiefel regularization using time element," *The Astronomical Journal*, vol. 129, no. 3, pp. 1746–1754, Mar. 2005, ISSN: 0004-6256. DOI: [10.1086/427718](https://doi.org/10.1086/427718).
- [53] T. Fukushima, "Efficient orbit integration by linear transformation for Kustaanheimo-Stiefel regularization," *The Astronomical Journal*, vol. 129, no. 5, pp. 2496–2503, May 2005, ISSN: 0004-6256. DOI: [10.1086/429546](https://doi.org/10.1086/429546).

- [54] T. Fukushima, "Efficient orbit integration by manifold correction methods," *Annals of the New York Academy of Sciences*, vol. 1065, no. 1, pp. 37–43, Dec. 2005, ISSN: 0077-8923. DOI: [10.1196/annals.1370.026](https://doi.org/10.1196/annals.1370.026).
- [55] T. Fukushima, "Efficient orbit integration by orbital longitude methods with Sundmann transformation on the time variable," *The Astronomical Journal*, vol. 129, no. 2, pp. 1171–1177, Feb. 2005, ISSN: 0004-6256. DOI: [10.1086/427139](https://doi.org/10.1086/427139).
- [56] A. T. Fuller, "Analysis of nonlinear stochastic systems by means of the fokker-planck equation," *International Journal of Control*, vol. 9, no. 6, pp. 603–655, 1969. DOI: [10.1080/00207176908905786](https://doi.org/10.1080/00207176908905786).
- [57] M. Geda, R. Noomen, and F. Renk, "Massive Parallelization of Trajectory Propagations using GPUs," Master's thesis, Delft University of Technology, 2019. [Online]. Available: <http://resolver.tudelft.nl/uuid:1db3f2d1-c2bb-4188-bd1e-dac67bfd9dab>.
- [58] J. S. Greaves *et al.*, "Phosphine gas in the cloud decks of Venus," *Nature Astronomy*, vol. 5, no. 7, pp. 655–664, Jul. 2021, ISSN: 2397-3366. DOI: [10.1038/s41550-020-1174-4](https://doi.org/10.1038/s41550-020-1174-4).
- [59] R. Greenberg, A. Carusi, and G. Valsecchi, "Outcomes of planetary close encounters: A systematic comparison of methodologies," *Icarus*, vol. 75, no. 1, pp. 1–29, 1988, ISSN: 0019-1035. DOI: [10.1016/0019-1035\(88\)90125-X](https://doi.org/10.1016/0019-1035(88)90125-X).
- [60] I. Griva, S. G. Nash, and A. Sofer, *Linear and Nonlinear Optimization*. 2009. DOI: [10.1137/1.9780898717730](https://doi.org/10.1137/1.9780898717730).
- [61] E. Hairer, G. Wanner, and S. P. Nørsett, *Solving Ordinary Differential Equations I*, 2nd. Springer Berlin, 1993. DOI: [10.1007/978-3-540-78862-1](https://doi.org/10.1007/978-3-540-78862-1).
- [62] D. C. Heggie, "A global regularisation of the gravitational N-body problem," *Celestial mechanics*, vol. 10, no. 2, pp. 217–241, Oct. 1974, ISSN: 1572-9478. DOI: [10.1007/BF01227621](https://doi.org/10.1007/BF01227621).
- [63] S. Hernandez and M. R. Akella, "Energy preserving low-thrust guidance for orbit transfers in KS variables," *Celestial Mechanics and Dynamical Astronomy*, vol. 125, no. 1, pp. 107–132, May 2016, ISSN: 1572-9478. DOI: [10.1007/s10569-016-9677-0](https://doi.org/10.1007/s10569-016-9677-0).
- [64] J. Hernando-Ayuso, C. Bombardelli, and G. Baù, "Uncertainty propagation in the n-body problem using dromo elements," *Acta Astronautica*, vol. 156, pp. 252–261, 2019, ISSN: 0094-5765. DOI: [10.1016/j.actaastro.2017.12.030](https://doi.org/10.1016/j.actaastro.2017.12.030).
- [65] G. Hori, "Theory of General Perturbation with Unspecified Canonical Variable," *Publications of the Astronomical Society of Japan*, vol. 18, p. 287, Jan. 1966.



- [66] G.-I. Hori, "The motion of a hyperbolic artificial satellite around the oblate earth," *The Astronomical Journal*, vol. 66, p. 258, Aug. 1961. DOI: [10.1086/108405](https://doi.org/10.1086/108405).
- [67] A. F. Inno, L. Bucci, A. Masat, and C. Colombo, "Orbital interference, planetary close-approaches detection and memory handling on GPUs," M.S. thesis, Politecnico di Milano, 2022. [Online]. Available: <http://hdl.handle.net/10589/188499>.
- [68] L. Isserlis, "On a formula for the product-moment coefficient of any order of a normal frequency distribution in any number of variables," *Biometrika*, vol. 12, no. 1-2, pp. 134-139, Nov. 1918, ISSN: 0006-3444. DOI: [10.1093/biomet/12.1-2.134](https://doi.org/10.1093/biomet/12.1-2.134).
- [69] E. J. Opik, *Interplanetary Encounters: Close-Range Gravitational Interactions*. Amsterdam, Jan. 1976, vol. 2, ISBN: 978-0444413710.
- [70] R. Jehn, "Estimating the impact probability of ariane upper stages," MAS Working paper 601, European Space Agency, Tech. Rep., Dec. 2014.
- [71] B. A. Jones, A. Doostan, and G. H. Born, "Nonlinear propagation of orbit uncertainty using non-intrusive polynomial chaos," *Journal of Guidance, Control, and Dynamics*, vol. 36, no. 2, pp. 430-444, 2013. DOI: [10.2514/1.57599](https://doi.org/10.2514/1.57599).
- [72] S. Jordan, O. Shorttle, and P. B. Rimmer, "Proposed energy-metabolisms cannot explain the atmospheric chemistry of Venus," *Nature Communications*, vol. 13, no. 1, p. 3274, Jun. 2022, ISSN: 2041-1723. DOI: [10.1038/s41467-022-30804-8](https://doi.org/10.1038/s41467-022-30804-8).
- [73] *JSON file format*, Last access: May 2023. [Online]. Available: <https://www.json.org/json-en.html>.
- [74] S. Julier and J. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401-422, 2004. DOI: [10.1109/JPROC.2003.823141](https://doi.org/10.1109/JPROC.2003.823141).
- [75] S. Julier, J. Uhlmann, and H. Durrant-Whyte, "A new approach for filtering nonlinear systems," in *Proceedings of 1995 American Control Conference - ACC'95*, vol. 3, 1995, 1628-1632 vol.3. DOI: [10.1109/ACC.1995.529783](https://doi.org/10.1109/ACC.1995.529783).
- [76] J. L. Junkins, A. Bani Younes, R. M. Woollands, and X. Bai, "Picard iteration, Chebyshev polynomials and Chebyshev-Picard methods: Application in Astrodynamics," *Journal of the Astronautical Sciences*, vol. 60, no. 3, pp. 623-653, Dec. 2013, ISSN: 2195-0571. DOI: [10.1007/s40295-015-0061-1](https://doi.org/10.1007/s40295-015-0061-1).
- [77] R. E. Kalman and R. S. Bucy, "New Results in Linear Filtering and Prediction Theory," *Journal of Basic Engineering*, vol. 83, no. 1, pp. 95-108, Mar. 1961, ISSN: 0021-9223. DOI: [10.1115/1.3658902](https://doi.org/10.1115/1.3658902).

- [78] G. Kminek, "ESA planetary protection requirements, Technical Report ESSB-ST-U-001," European Space Agency, Tech. Rep., 2012.
- [79] D. Koblick, M. Poole, and P. Shankar, "Parallel high-precision orbit propagation using the Modified Picard-Chebyshev Method," in *ASME International Mechanical Engineering Congress and Exposition, Proceedings (IMECE)*, 2012, ISBN: 9780791845172. DOI: [10.1115/IMECE2012-87878](https://doi.org/10.1115/IMECE2012-87878).
- [80] D. Koblick and P. Shankar, "Evaluation of the Modified Picard-Chebyshev Method for high-precision orbit propagation," *Journal of Aerospace Engineering*, vol. 28, no. 5, p. 04014125, Sep. 2015. DOI: [10.1061/\(ASCE\)AS.1943-5525.0000463](https://doi.org/10.1061/(ASCE)AS.1943-5525.0000463).
- [81] D. Koblick, S. Xu, J. Fogel, and P. Shankar, "Low thrust minimum time orbit transfer nonlinear optimization using impulse discretization via the Modified Picard-Chebyshev Method," *Computer Modeling in Engineering & Sciences*, vol. 111, no. 1, 2016, ISSN: 1526-1506. DOI: [10.3970/cmcs.2016.111.001](https://doi.org/10.3970/cmcs.2016.111.001).
- [82] Y. Kozai, "Secular perturbations of asteroids with high inclination and eccentricity," *The Astronomical Journal*, vol. 67, pp. 591–598, Nov. 1962. DOI: [10.1086/108790](https://doi.org/10.1086/108790).
- [83] P. Kustaanheimo, H. Schinzel, and E. Stiefel, "Perturbation theory of Kepler motion based on spinor regularization," *Journal für die reine und angewandte Mathematik*, vol. 1965, no. 218, pp. 204–219, 1965. DOI: [10.1515/crll.1965.218.204](https://doi.org/10.1515/crll.1965.218.204).
- [84] P. Kustaanheimo, *Spinor Regularization of the Kepler Motion*. Turun Yliopisto, 1964. [Online]. Available: <https://books.google.it/books?id=ZhQ9ywAACAAJ>.
- [85] K. Langner and S. Breiter, "KS variables in rotating reference frame. application to cometary dynamics," *Astrophysics and Space Science*, vol. 357, no. 2, p. 153, May 2015, ISSN: 1572-946X. DOI: [10.1007/s10509-015-2384-6](https://doi.org/10.1007/s10509-015-2384-6).
- [86] G. Lantoine, R. P. Russell, and S. Campagnola, "Optimization of low-energy resonant hopping transfers between planetary moons," *Acta Astronautica*, vol. 68, no. 7, pp. 1361–1378, Apr. 2011, ISSN: 0094-5765. DOI: [10.1016/j.actaastro.2010.09.021](https://doi.org/10.1016/j.actaastro.2010.09.021).
- [87] B. Lazare, "The French Space Operations Act: Technical Regulations," *Acta Astronautica*, vol. 92, no. 2, pp. 209–212, 2013, 2nd IAA Symposium on Private Human Access to Space, ISSN: 0094-5765. DOI: [10.1016/j.actaastro.2012.07.031](https://doi.org/10.1016/j.actaastro.2012.07.031).
- [88] F. Letizia, C. Colombo, J. Van den Eynde, R. Armellin, and R. Jehn, "SNAPSHOT: Suite for the numerical analysis of planetary protection," in *6th International Conference on Astrodynamics Tools and Techniques (ICATT)*, 2016, pp. 14–17.



- [89] F. Letizia, C. Colombo, J. Van den Eynde, and R. Jehn, "B-plane visualisation tool for uncertainty evaluation," in *Proceedings of the 26th AAS/AIAA Space Flight Mechanics Meeting*, vol. AAS 16-438, Napa, CA, USA, Feb. 2016.
- [90] M. Lidov, "Evolution of artificial planetary satellites under the action of gravitational perturbations due to external bodies," *Iskusstviennye Sputniki Zemli*, vol. 8, pp. 5–45, 1961.
- [91] Y.-z. Luo and Z. Yang, "A review of uncertainty propagation in orbital mechanics," *Progress in Aerospace Sciences*, vol. 89, pp. 23–39, Feb. 2017, ISSN: 0376-0421. DOI: [10.1016/j.paerosci.2016.12.002](https://doi.org/10.1016/j.paerosci.2016.12.002).
- [92] B. Macomber, A. B. Probe, R. Woollands, J. Read, and J. L. Junkins, "Enhancements to modified chebyshev-picard iteration efficiency for perturbed orbit propagation," *Computer Modeling in Engineering & Sciences*, vol. 111, no. 1, pp. 29–64, 2016, ISSN: 1526-1506. DOI: [10.3970/cmesci.2016.111.029](https://doi.org/10.3970/cmesci.2016.111.029).
- [93] B. D. Macomber, "Enhancements to chebyshev-picard iteration efficiency for generally perturbed orbits and constrained dynamical systems," Ph.D. dissertation, Texas A & M University, Supervisor: Junkins, John L., Aug. 2015. [Online]. Available: <https://hdl.handle.net/1969.1/155745>.
- [94] A. Masat, M. Romano, and C. Colombo, "B-plane orbital resonance analysis and applications," Master's thesis, Politecnico di Milano, 2019. [Online]. Available: <http://hdl.handle.net/10589/151660>.
- [95] A. Masat, M. Romano, and C. Colombo, "Different perspectives on the b-plane: Perturbation effects and use for resonant flyby design," *Celestial Mechanics and Dynamical Astronomy*, vol. 134, no. 2, p. 17, Apr. 2022, ISSN: 1572-9478. DOI: [10.1007/s10569-022-10072-w](https://doi.org/10.1007/s10569-022-10072-w).
- [96] A. M. Mathai and S. B. Provost, *Quadratic forms in random variables*. 1992, ISBN: 978-0824786915.
- [97] W. McClain and D. Vallado, *Fundamentals of Astrodynamics and Applications* (Space Technology Library). Springer Netherlands, 2001, ISBN: 9780792369035. [Online]. Available: <https://books.google.it/books?id=PJLlWzMBKjkC>.
- [98] B. A. McElhoe, "An assessment of the navigation and course corrections for a manned flyby of mars or venus," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-2, no. 4, pp. 613–623, 1966. DOI: [10.1109/TAES.1966.4501892](https://doi.org/10.1109/TAES.1966.4501892).

- [99] M. McGuire, L. Burke, S. McCarty, K. Hack, R. Whitley, D. Davis, and C. Ocampo, "Low Thrust Cis-Lunar Transfers Using a 40 kW-Class Solar Electric Propulsion Spacecraft," in *AAS/AIAA Astrodynamics Specialist Conference*, Stevenson (WA), USA, Aug. 2017, pp. 1–21.
- [100] S. Mikkola and S. Aarseth, "A time-transformed leapfrog scheme," *Celestial Mechanics and Dynamical Astronomy*, vol. 84, no. 4, pp. 343–354, Dec. 2002, ISSN: 1572-9478. DOI: [10.1023/A:1021149313347](https://doi.org/10.1023/A:1021149313347).
- [101] S. Mikkola and S. J. Aarseth, "A chain regularization method for the few-body problem," *Celestial Mechanics and Dynamical Astronomy*, vol. 47, no. 4, pp. 375–390, Dec. 1989, ISSN: 1572-9478. DOI: [10.1007/BF00051012](https://doi.org/10.1007/BF00051012).
- [102] S. Mikkola and S. J. Aarseth, "An implementation of N-body chain regularization," *Celestial Mechanics and Dynamical Astronomy*, vol. 57, no. 3, pp. 439–459, Nov. 1993, ISSN: 1572-9478. DOI: [10.1007/BF00695714](https://doi.org/10.1007/BF00695714).
- [103] S. Mikkola and S. J. Aarseth, "A slow-down treatment for close binaries," *Celestial Mechanics and Dynamical Astronomy*, vol. 64, no. 3, pp. 197–208, Sep. 1996, ISSN: 1572-9478. DOI: [10.1007/BF00728347](https://doi.org/10.1007/BF00728347).
- [104] S. Mikkola and S. J. Aarseth, "An efficient integration method for binaries in N-body simulations," *New Astronomy*, vol. 3, no. 5, pp. 309–320, Jul. 1998, ISSN: 1384-1076. DOI: [10.1016/S1384-1076\(98\)00018-9](https://doi.org/10.1016/S1384-1076(98)00018-9).
- [105] S. Mikkola and D. Merritt, "Algorithmic regularization with velocity-dependent forces," *Monthly Notices of the Royal Astronomical Society*, vol. 372, no. 1, pp. 219–223, Oct. 2006, ISSN: 0035-8711. DOI: [10.1111/j.1365-2966.2006.10854.x](https://doi.org/10.1111/j.1365-2966.2006.10854.x).
- [106] S. Mikkola and D. Merritt, "Implementing few-body algorithmic regularization with post-Newtonian terms," *The Astronomical Journal*, vol. 135, no. 6, pp. 2398–2405, May 2008, ISSN: 0004-6256. DOI: [10.1088/0004-6256/135/6/2398](https://doi.org/10.1088/0004-6256/135/6/2398).
- [107] A. Milani, "The asteroid identification problem: I. recovery of lost asteroids," *Icarus*, vol. 137, no. 2, pp. 269–292, 1999, ISSN: 0019-1035. DOI: [10.1006/icar.1999.6045](https://doi.org/10.1006/icar.1999.6045).
- [108] A. Milani, S. R. Chesley, P. W. Chodas, and G. B. Valsecchi, "Asteroid Close Approaches: Analysis and Potential Impact Detection," *Asteroids III*, pp. 55–69, 2002.
- [109] A. Milani, S. R. Chesley, M. E. Sansaturio, G. Tommei, and G. B. Valsecchi, "Nonlinear impact monitoring: Line of variation searches for impactors," *Icarus*, vol. 173, no. 2, pp. 362–384, 2005, ISSN: 0019-1035. DOI: [10.1016/j.icarus.2004.09.002](https://doi.org/10.1016/j.icarus.2004.09.002).

- [110] Milani, A., Sansaturio, M. E., Tommei, G., Arratia, O., and Chesley, S. R., "Multiple solutions for asteroid orbits: Computational procedure and applications," *A&A*, vol. 431, no. 2, pp. 729–746, 2005. DOI: [10.1051/0004-6361:20041737](https://doi.org/10.1051/0004-6361:20041737).
- [111] *Numpy Python package*, Last access: May 2023. [Online]. Available: <https://numpy.org/>.
- [112] NVIDIA corporation, *CUDA C++ Programming Guide*, Nov. 2021. [Online]. Available: <https://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html>.
- [113] M. Palacios and C. Calvo, "Ideal frames and regularization in numerical orbit computation," *Journal of the Astronautical Sciences*, vol. 44, no. 1, pp. 63–77, 1996.
- [114] P. L. Palmer, S. J. Aarseth, S. Mikkola, and Y. Hashida, "High precision integration methods for orbit propagation," *The Journal of the Astronautical Sciences*, vol. 46, no. 4, pp. 329–342, Dec. 1998, ISSN: 2195-0571. DOI: [10.1007/BF03546385](https://doi.org/10.1007/BF03546385).
- [115] R. S. Park and D. J. Scheeres, "Nonlinear mapping of gaussian statistics: Theory and applications to spacecraft trajectory design," *Journal of Guidance, Control, and Dynamics*, vol. 29, no. 6, pp. 1367–1375, 2006. DOI: [10.2514/1.20177](https://doi.org/10.2514/1.20177).
- [116] J. Peláez, J. M. Hedo, and P. R. de Andrés, "A special perturbation method in orbital dynamics," *Celestial Mechanics and Dynamical Astronomy*, vol. 97, no. 2, pp. 131–150, Feb. 2007, ISSN: 1572-9478. DOI: [10.1007/s10569-006-9056-3](https://doi.org/10.1007/s10569-006-9056-3).
- [117] *PyBind Python/C++ interface*, Last access: May 2023. [Online]. Available: <https://github.com/pybind/pybind11>.
- [118] A. Quarteroni, R. Sacco, and F. Saleri, *Numerical Mathematics Texts in Applied Mathematics*. 2007, ISBN: 978-3-540-49809-4. DOI: [10.5555/1212166](https://doi.org/10.5555/1212166).
- [119] T. J. Rivlin, "The Chebyshev Polynomials," *Mathematics of Computation*, vol. 30, Apr. 1976, ISSN: 00255718. DOI: [10.2307/2005983](https://doi.org/10.2307/2005983).
- [120] J. Roa and N. J. Kasdin, "Alternative set of nonsingular quaternionic orbital elements," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 11, pp. 2737–2751, Nov. 2017. DOI: [10.2514/1.G002753](https://doi.org/10.2514/1.G002753).
- [121] J. Roa and J. Peláez, "Orbit propagation in Minkowskian geometry," *Celestial Mechanics and Dynamical Astronomy*, vol. 123, no. 1, pp. 13–43, Sep. 2015, ISSN: 1572-9478. DOI: [10.1007/s10569-015-9627-2](https://doi.org/10.1007/s10569-015-9627-2).
- [122] J. Roa and J. Peláez, "Efficient trajectory propagation for orbit determination problems," in *2015 AAS/AIAA Astrodynamics Specialist Conference*, Vail (CO), USA, 2015.

- [123] J. Roa, M. Sanjurjo-Rivo, and J. Peláez, "Singularities in DROMO formulation. analysis of deep flybys," *Advances in Space Research*, vol. 56, no. 3, pp. 569–581, Aug. 2015, ISSN: 0273-1177. DOI: [10.1016/j.asr.2015.03.019](https://doi.org/10.1016/j.asr.2015.03.019).
- [124] J. Roa, H. Urrutxua, and J. Peláez, "Stability and chaos in Kustaanheimo–Stiefel space induced by the hopf fibration," *Monthly Notices of the Royal Astronomical Society*, vol. 459, no. 3, pp. 2444–2454, Apr. 2016, ISSN: 1365-2966. DOI: [10.1093/mnras/stw780](https://doi.org/10.1093/mnras/stw780).
- [125] M. Romano, "Orbit propagation and uncertainty modelling for planetary protection compliance verification," Ph.D. dissertation, Politecnico di Milano, Supervisors: Colombo, Camilla and Sánchez Pérez, José Manuel, Feb. 2020. DOI: [10.13140/RG.2.2.19692.80001](https://doi.org/10.13140/RG.2.2.19692.80001).
- [126] M. Romano, M. Losacco, C. Colombo, and P. Di Lizia, "Impact probability computation of near-Earth objects using Monte Carlo line sampling and subset simulation," *Celestial Mechanics and Dynamical Astronomy*, vol. 132, no. 8, p. 42, Aug. 2020, ISSN: 1572-9478. DOI: [10.1007/s10569-020-09981-5](https://doi.org/10.1007/s10569-020-09981-5).
- [127] P. Saha, "Interpreting the kustaanheimo–Stiefel transform in gravitational dynamics," *Monthly Notices of the Royal Astronomical Society*, vol. 400, no. 1, pp. 228–231, Nov. 2009, ISSN: 1365-2966. DOI: [10.1111/j.1365-2966.2009.15437.x](https://doi.org/10.1111/j.1365-2966.2009.15437.x).
- [128] F. Schrammel, F. Renk, A. Mazaheri, and F. Wolf, "Efficient ephemeris models for spacecraft trajectory simulations on gpus," in *Euro-Par 2020: Parallel Processing*, M. Malawski and K. Rządca, Eds., Cham: Springer International Publishing, 2020, pp. 561–577, ISBN: 978-3-030-57675-2.
- [129] P. K. Seidelmann, *Explanatory Supplement To The Astronomical Almanac*. 1305 Walt Whitman Road, Suite 110 Melville, NY 11747 USA: University Science Books, 1992, ch. 5, pp. 281–288, ISBN: 1-891389-45-9.
- [130] H. Sellamuthu and R. K. Sharma, "Orbit theory with lunar perturbation in terms of Kustaanheimo–Stiefel regular elements," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 5, pp. 1272–1277, 2017. DOI: [10.2514/1.G002342](https://doi.org/10.2514/1.G002342).
- [131] H. Sellamuthu and R. K. Sharma, "Hybrid orbit propagator for small spacecraft using Kustaanheimo–Stiefel elements," *Journal of Spacecraft and Rockets*, vol. 55, no. 5, pp. 1282–1288, 2018. DOI: [10.2514/1.A34076](https://doi.org/10.2514/1.A34076).
- [132] H. Sellamuthu and R. K. Sharma, "Regularized luni-solar gravity dynamics on resident space objects," *Astrodynamics*, vol. 5, no. 2, pp. 91–108, Jun. 2020, ISSN: 2522-0098. DOI: [10.1007/s42064-020-0085-6](https://doi.org/10.1007/s42064-020-0085-6).

- [133] J. S. Shaver, "Formulation and evaluation of parallel algorithms for the orbit determination problem," Ph.D. dissertation, Massachusetts Institute of Technology, Supervisors: Walter M. Hollister and John P. Vinti, 1980. [Online]. Available: <https://hdl.handle.net/1721.1/130994>.
- [134] C. Short, K. Howell, A. Haapala, and D. Dichmann, "Mode analysis for long-term behavior in a resonant Earth-Moon trajectory," *The Journal of the Astronautical Sciences*, vol. 64, no. 2, pp. 156–187, Jun. 2017, ISSN: 2195-0571. DOI: [10.1007/s40295-016-0098-9](https://doi.org/10.1007/s40295-016-0098-9).
- [135] J. R. Silvester, "Determinants of block matrices," *The Mathematical Gazette*, vol. 84, no. 501, pp. 460–467, May 2000, Full publication date: Nov., 2000, ISSN: 00255572. DOI: [10.2307/3620776](https://doi.org/10.2307/3620776).
- [136] S. Singh, J. Junkins, B. Anderson, and E. Taheri, "Eclipse-conscious transfer to lunar gateway using ephemeris-driven terminal coast arcs," *Journal of Guidance, Control, and Dynamics*, vol. 44, no. 11, pp. 1972–1988, 2021. DOI: [10.2514/1.G005920](https://doi.org/10.2514/1.G005920).
- [137] S. K. Singh, R. Woollands, E. Taheri, and J. Junkins, "Feasibility of quasi-frozen, near-polar and extremely low-altitude lunar orbits," *Acta Astronautica*, vol. 166, pp. 450–468, 2020, ISSN: 0094-5765. DOI: [10.1016/j.actaastro.2019.10.037](https://doi.org/10.1016/j.actaastro.2019.10.037).
- [138] G. Smith, S. Schmidt, L. McGee, and U. States, "Application of statistical filter theory to the optimal estimation of position and velocity on board a circumlunar vehicle," NASA (National Aeronautics and Space Administration), Tech. Rep., 1962. [Online]. Available: <https://ntrs.nasa.gov/citations/20190002215>.
- [139] E. L. Stiefel and G. Scheifele, *Linear and Regular Celestial Mechanics* (Grundlehren der mathematischen Wissenschaften). Springer, Berlin, 1971, ISBN: 978-3-642-65029-1.
- [140] K. F. Sundman, "Mémoire sur le problème des trois corps," *Acta Mathematica*, vol. 36, 105–179, 1913. DOI: [10.1007/BF02422379](https://doi.org/10.1007/BF02422379).
- [141] T. Swenson, R. Woollands, J. Junkins, and M. Lo, "Application of Modified Chebyshev Picard Iteration to Differential Correction for Improved Robustness and Computation Time," *Journal of the Astronautical Sciences*, vol. 64, no. 3, pp. 267–284, Sep. 2017. DOI: [10.1007/s40295-016-0110-4](https://doi.org/10.1007/s40295-016-0110-4).
- [142] G. Terejanu, P. Singla, T. Singh, and P. D. Scott, "Uncertainty propagation for nonlinear dynamic systems using gaussian mixture models," *Journal of Guidance, Control, and Dynamics*, vol. 31, no. 6, pp. 1623–1633, 2008. DOI: [10.2514/1.36247](https://doi.org/10.2514/1.36247).

- [143] F. Topputo, M. Vasile, and F. Bernelli-Zazzera, "Earth-to-Moon low energy transfers targeting L<sub>1</sub> hyperbolic transit orbits," *Annals of the New York Academy of Sciences*, vol. 1065, no. 1, pp. 55–76, Dec. 2005, ISSN: 0077-8923. DOI: [10.1196/annals.1370.025](https://doi.org/10.1196/annals.1370.025).
- [144] V. Tunjov and A. Rocchi, "Simplified Radiation Model of Jupiter for Planetary Protection Applications," Delft University of Technology, Tech. Rep., 2022.
- [145] H. Urrutxua, M. Sanjurjo-Rivo, and J. Peláez, "DROMO propagator revisited," *Celestial Mechanics and Dynamical Astronomy*, vol. 124, no. 1, pp. 1–31, Jan. 2016, ISSN: 1572-9478. DOI: [10.1007/s10569-015-9647-y](https://doi.org/10.1007/s10569-015-9647-y).
- [146] D. Vallado and S. Alfano, "Updated analytical partials for covariance transformations and optimization," in *2015 AAS/AIAA Astrodynamics Specialist Conference*, Vail (CO), USA, Aug. 2015. [Online]. Available: [https://www.researchgate.net/publication/282151888\\_Updated\\_Analytical\\_Partial\\_for\\_Covariance\\_Transformations\\_and\\_Optimization](https://www.researchgate.net/publication/282151888_Updated_Analytical_Partial_for_Covariance_Transformations_and_Optimization).
- [147] G. B. Valsecchi, E. M. Alessi, and A. Rossi, "An analytical solution for the swing-by problem," *Celestial Mechanics and Dynamical Astronomy*, vol. 123, no. 2, pp. 151–166, Oct. 2015, ISSN: 1572-9478. DOI: [10.1007/s10569-015-9631-6](https://doi.org/10.1007/s10569-015-9631-6).
- [148] G. B. Valsecchi, A. Milani, G. F. Gronchi, and S. R. Chesley, "Resonant returns to close approaches: Analytical theory," *A&A*, vol. 408, no. 3, pp. 1179–1196, 2003. DOI: [10.1051/0004-6361:20031039](https://doi.org/10.1051/0004-6361:20031039).
- [149] G. B. Valsecchi, "Geometric conditions for quasi-collisions in Öpik's theory," in *Dynamics of Extended Celestial Bodies and Rings*, J. Souchay, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 145–158, ISBN: 978-3-540-32455-3. DOI: [10.1007/3-540-32455-0\\_6](https://doi.org/10.1007/3-540-32455-0_6).
- [150] M. Vaquero and K. C. Howell, "Transfer design exploiting resonant orbits and manifolds in the Saturn-Titan system," *Journal of Spacecraft and Rockets*, vol. 50, no. 5, pp. 1069–1085, Sep. 2013, ISSN: 0022-4650. DOI: [10.2514/1.A32412](https://doi.org/10.2514/1.A32412).
- [151] W. Velte, "Concerning the regularizing KS-transformation," *Celestial mechanics*, vol. 17, no. 4, pp. 395–403, May 1978, ISSN: 1572-9478. DOI: [10.1007/BF01228959](https://doi.org/10.1007/BF01228959).
- [152] V. Vittaldev and R. P. Russell, "Space object collision probability using multidirectional gaussian mixture models," *Journal of Guidance, Control, and Dynamics*, vol. 39, no. 9, pp. 2163–2169, 2016. DOI: [10.2514/1.G001610](https://doi.org/10.2514/1.G001610).



- [153] M. D. Vivarelli, "The KS-transformation in hypercomplex form and the quantization of the negative-energy orbit manifold of the Kepler problem," *Celestial mechanics*, vol. 36, no. 4, pp. 349–364, Aug. 1985, ISSN: 1572-9478. DOI: [10.1007/BF01227489](https://doi.org/10.1007/BF01227489).
- [154] M. D. Vivarelli, "Geometrical and physical outlook on the cross product of two quaternions," *Celestial mechanics*, vol. 41, no. 1, pp. 359–370, Mar. 1987, ISSN: 1572-9478. DOI: [10.1007/BF01238771](https://doi.org/10.1007/BF01238771).
- [155] M. D. Vivarelli, "The Kepler problem: A unifying view," *Celestial Mechanics and Dynamical Astronomy*, vol. 60, no. 3, pp. 291–305, Nov. 1994, ISSN: 1572-9478. DOI: [10.1007/BF00691898](https://doi.org/10.1007/BF00691898).
- [156] J. Waldvogel, "Quaternions and the perturbed Kepler problem," *Celestial Mechanics and Dynamical Astronomy*, vol. 95, no. 1, pp. 201–212, May 2006, ISSN: 1572-9478. DOI: [10.1007/s10569-005-5663-7](https://doi.org/10.1007/s10569-005-5663-7).
- [157] M. Wallace, "A massively parallel bayesian approach to planetary protection trajectory analysis and design," in *Proceedings of the 2015 AAS/AIAA Astrodynamics Specialist conference*, vol. AAS 15-535, Vail, CO, USA, Aug. 2015. [Online]. Available: <https://trs.jpl.nasa.gov/handle/2014/45859>.
- [158] Q. Wang, X. Zhang, Y. Zhang, and Q. Yi, "AUGEM: Automatically generate high performance Dense Linear Algebra kernels on x86 CPUs," in *SC '13: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, 2013, pp. 1–12. DOI: [10.1145/2503210.2503219](https://doi.org/10.1145/2503210.2503219).
- [159] N. Wiener, "The homogeneous chaos," *American Journal of Mathematics*, vol. 60, no. 4, pp. 897–936, 1938, ISSN: 00029327, 10806377. DOI: [10.2307/2371268](https://doi.org/10.2307/2371268). (visited on 05/26/2023).
- [160] C. M. Will, *Theory and Experiment in Gravitational Physics*, 2nd ed. Cambridge University Press, 2018, pp. 78–155. DOI: [10.1017/9781316338612](https://doi.org/10.1017/9781316338612).
- [161] E. B. Wilson, "Probable inference, the law of succession, and statistical inference," *Journal of the American Statistical Association*, vol. 22, no. 158, pp. 209–212, 1927, ISSN: 01621459. [Online]. Available: <http://www.jstor.org/stable/2276774>.
- [162] A. Wittig, P. Di Lizia, R. Armellin, K. Makino, F. Bernelli-Zazzera, and M. Berz, "Propagation of large uncertainty sets in orbital dynamics by automatic domain splitting," *Celestial Mechanics and Dynamical Astronomy*, vol. 122, no. 3, pp. 239–261, Jul. 2015, ISSN: 1572-9478. DOI: [10.1007/s10569-015-9618-3](https://doi.org/10.1007/s10569-015-9618-3).

- [163] R. Woollands and J. L. Junkins, "Nonlinear differential equation solvers via Adaptive Picard-Chebyshev iteration: Applications in Astrodynamics," *Journal of Guidance, Control, and Dynamics*, vol. 42, no. 5, pp. 1007–1022, May 2019. DOI: [10.2514/1.G003318](https://doi.org/10.2514/1.G003318).
- [164] R. Woollands, E. Taheri, and J. L. Junkins, "Efficient computation of optimal low thrust gravity perturbed orbit transfers," *Journal of the Astronautical Sciences*, vol. 67, no. 2, pp. 458–484, Jun. 2020, ISSN: 2195-0571. DOI: [10.1007/s40295-019-00152-9](https://doi.org/10.1007/s40295-019-00152-9).
- [165] R. M. Woollands, A. Bani Younes, and J. L. Junkins, "New solutions for the perturbed Lambert problem using regularization and Picard iteration," *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 9, pp. 1548–1562, Sep. 2015. DOI: [10.2514/1.G001028](https://doi.org/10.2514/1.G001028).
- [166] R. M. Woollands, J. Read, K. Hernandez, A. Probe, and J. L. Junkins, "Unified Lambert tool for massively parallel applications in space situational awareness," *Journal of the Astronautical Sciences*, vol. 65, no. 1, pp. 29–45, Mar. 2018, ISSN: 2195-0571. DOI: [10.1007/s40295-017-0118-4](https://doi.org/10.1007/s40295-017-0118-4).
- [167] R. M. Woollands, J. L. Read, A. B. Probe, and J. L. Junkins, "Multiple revolution solutions for the perturbed Lambert problem using the Method of Particular Solutions and Picard iteration," *Journal of the Astronautical Sciences*, vol. 64, no. 4, pp. 361–378, Dec. 2017, ISSN: 2195-0571. DOI: [10.1007/s40295-017-0116-6](https://doi.org/10.1007/s40295-017-0116-6).
- [168] Z. Xianyi, W. Qian, and Z. Yunquan, "Model-driven Level 3 BLAS Performance Optimization on Loongson 3A Processor," in *2012 IEEE 18th International Conference on Parallel and Distributed Systems*, 2012, pp. 684–691. DOI: [10.1109/ICPADS.2012.97](https://doi.org/10.1109/ICPADS.2012.97).
- [169] *Xianyi/openblas*, Last access: November 2021. [Online]. Available: <https://github.com/xianyi/OpenBLAS#supported-cpus-and-operating-systems>.
- [170] *YML file format*, Last access: May 2023. [Online]. Available: <https://yaml.org/>.
- [171] E. M. Zimovan, K. C. Howell, and D. C. Davis, "Near rectilinear halo orbits and their application in cis-lunar space," in *3rd IAA Conference on Dynamics and Control of Space Systems*, vol. 20, Moscow, Russia, May 2017.




---

DECLARATION

---

I, Alessandro Masat, hereby declare that whatever presented in this dissertation and not specifically referenced is all work of my own.

*Milano, September 2023*

A handwritten signature in black ink that reads "Alessandro Masat". The signature is written in a cursive style with a horizontal line underneath it.

Alessandro Masat