

**Copyright**

**by**

**Vivek Vittaldev**

**2015**

The Dissertation Committee for Vivek Vittaldev  
certifies that this is the approved version of the following dissertation:

**Uncertainty Propagation and Conjunction Assessment  
for Resident Space Objects**

Committee:

---

Ryan P. Russell, Supervisor

---

Maruthi R. Akella

---

Srinivas V. Bettadpur

---

R. Scott Erwin

---

Todd E. Humphreys

**Uncertainty Propagation and Conjunction Assessment  
for Resident Space Objects**

**by**

**Vivek Vittaldev, M.Sc.**

**DISSERTATION**

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

**DOCTOR OF PHILOSOPHY**

The University of Texas at Austin  
December 2015

To my parents

## Acknowledgments

I would like to acknowledge and express my gratitude to some of the people who have helped me reach this milestone in my life.

First, I would like to thank Dr. Ryan Russell, my advisor and mentor over the past five years. He gave me the freedom that I needed and wanted to pursue my own ideas while always guiding me towards the end goal. His insight, expertise, and critical thinking have been indispensable. I have also had the privilege of attending two universities and living in two great cities over the course of my PhD because of him.

I would like to thank my dissertation committee: Dr. Maruthi Akella, Dr. Srinivas Bettadpur, Dr. Scott Erwin, and Dr. Todd Humphreys for taking the time to review my dissertation and participating in my defense. I have learned immensely through all of our various interactions. I would also like to thank Dr. Richard Linares and Dr. Nitin Arora for our various collaborations. Their help and friendship is greatly appreciated.

I thank my research group, filled with great researchers, colleagues, and friends. We have overcome many challenges through this common journey and I am glad we traveled together.

I have made many great friends during the course of the PhD both at UT and at Georgia Tech. I thank Tim, Romain, Nuno, Eduardo, Juan, Mock, and Amber for making my short stay at Georgia Tech memorable. Thank

you Etienne, for joining me in the move from Atlanta to Austin. My close friends in Austin: Anand, Ansel, Ben S, Brian, Carly, Divya, Denis, Etienne, Max, Nishant, Ricardo, and Sundeep have made living in this wonderful city a blast. My dear friends from home: Terrence, Ingrid, Ben P, Wesley, Thomas, Abhishek, and Alex F have all been a tremendous support. We may be on different continents, but you are always nearby. I hope our paths keep crossing in the future.

I would like to thank my uncle Ramesh and my aunt Neerja for being my second home so far away from home. I thank Ashley from the bottom of my heart for making all my days better. Finally, I thank my parents for being an endless source of inspiration, love, and support. You will always be my role models.

# Uncertainty Propagation and Conjunction Assessment for Resident Space Objects

by

Vivek Vittaldev, Ph.D.

The University of Texas at Austin, 2015

SUPERVISOR: Ryan P. Russell

Presently, the catalog of Resident Space Objects (RSOs) in Earth orbit tracked by the U.S. Space Surveillance Network (SSN) is greater than 21,000 objects. The size of the catalog continues to grow due to an increasing number of launches, improved tracking capabilities, and in some cases, collisions. Simply propagating the states of these RSOs is a computational burden, while additionally propagating the uncertainty distributions of the RSOs and computing collision probabilities increases the computational burden by at least an order of magnitude.

Tools are developed that propagate the uncertainty of RSOs with Gaussian initial uncertainty from epoch until a close approach. The number of possible elements in the form of a precomputed library, in a Gaussian Mixture Model (GMM) has been increased and the strategy for multivariate problems has been formalized. The accuracy of a GMM is increased by propagating each element by a Polynomial Chaos Expansion (PCE). Both techniques reduce the number of function evaluations required for uncertainty propagation and result

in a sliding scale where accuracy can be improved at the cost of increased computation time. A parallel implementation of the accurate benchmark Monte Carlo (MC) technique has been developed on the Graphics Processing Unit (GPU) that is capable of using samples from any uncertainty propagation technique to compute the collision probability. The GPU MC tool delivers up to two orders of magnitude speedups compared to a serial CPU implementation. Finally, a CPU implementation of the collision probability computations using Cartesian coordinates requires orders of magnitude fewer function evaluations compared to a MC run.

Fast computation of the inherent nonlinear growth of the uncertainty distribution in orbital mechanics and accurately computing the collision probability is essential for maintaining a future space catalog and for preventing an uncontrolled growth in the debris population. The uncertainty propagation and collision probability computation methods and algorithms developed here are capable of running on personal workstations and stand to benefit users ranging from national space surveillance agencies to private satellite operators. The developed techniques are also applicable for many general uncertainty quantification and nonlinear estimation problems.



# Table of Contents

<b>Acknowledgments</b>	<b>v</b>
<b>Abstract</b>	<b>vii</b>
<b>List of Tables</b>	<b>xii</b>
<b>List of Figures</b>	<b>xiii</b>
<b>Chapter 1. Introduction</b>	<b>1</b>
1.1 Space Objects in Earth Object . . . . .	2
1.2 Conjunction Assessment . . . . .	7
1.3 Outline . . . . .	12
1.4 Uncertainty Propagation . . . . .	14
1.4.1 Monte Carlo . . . . .	18
1.4.2 Gaussian Assumption . . . . .	19
1.4.3 Gaussian Mixture Models . . . . .	19
1.4.4 Polynomial Chaos Expansion . . . . .	21
1.5 Collision Probability Computation . . . . .	22
1.5.1 Monte Carlo . . . . .	23
1.5.2 Gaussian Approximation . . . . .	24
1.6 Contributions . . . . .	26
<b>Chapter 2. Multidirectional Gaussian Mixture Models</b>	<b>28</b>
2.1 Introduction . . . . .	29
2.2 Univariate Splitting Library . . . . .	31
2.2.1 Generating a Univariate Split . . . . .	31
2.2.2 Optimization . . . . .	32
2.2.3 Resulting Univariate Libraries . . . . .	37
2.3 Multivariate GMMs . . . . .	42
2.3.1 Splitting along an Arbitrary Direction . . . . .	42

2.3.2	Propagating a Mutivariate Gaussian Distribution . . . . .	45
2.3.3	Importance of Splitting Direction . . . . .	48
2.3.4	Choosing the Splitting Direction . . . . .	50
2.3.5	Sensitivity to Univariate Splitting Library . . . . .	54
2.3.6	Multidirectional Gaussian Mixture Models . . . . .	54
2.4	Numerical Test Cases . . . . .	57
2.4.1	Conversion from Polar Coordinates to Cartesian Coordinates . . . . .	58
2.4.2	Non-Linear ODE . . . . .	59
2.4.3	Orbit Uncertainty Propagation . . . . .	64
2.4.4	High Dimensional Problem . . . . .	65
2.5	Conclusion . . . . .	69
<b>Chapter 3. Gaussian Mixture Models and Polynomial Chaos Expansions</b>		<b>72</b>
3.1	Introduction . . . . .	73
3.2	Polynomial Chaos Expansion . . . . .	74
3.3	Gaussian Mixture Model Polynomial Chaos . . . . .	81
3.4	Results . . . . .	88
3.4.1	Medium Earth Orbit . . . . .	89
3.4.2	Molniya Orbit . . . . .	92
3.4.3	Geosynchronous Transfer Orbit . . . . .	95
3.4.4	Low Earth Orbit . . . . .	96
3.4.5	Weighted Least Squares Technique . . . . .	97
3.5	Conclusion . . . . .	101
<b>Chapter 4. Monte Carlo Collision Probability on the Graphics Processing Unit</b>		<b>103</b>
4.1	Introduction . . . . .	104
4.2	Monte Carlo Collision Probability . . . . .	106
4.2.1	Interpolation . . . . .	109
4.2.2	GPU Implementation . . . . .	114
4.3	Results . . . . .	121
4.3.1	Validation of Close Approach Interpolation . . . . .	122
4.3.2	Performance of the GPU Code . . . . .	127

4.3.3	Effect of Perturbations . . . . .	131
4.4	Conclusion . . . . .	138
<b>Chapter 5. Collision Probability using Multidirectional Gaussian Mixture Models</b>		<b>140</b>
5.1	Introduction . . . . .	141
5.2	Collision Probability . . . . .	143
5.2.1	Monte Carlo . . . . .	144
5.2.2	Collision Probability for Gaussian Distributions in Cartesian Coordinates . . . . .	146
5.2.3	Collision Probability for MGMMs . . . . .	153
5.2.3.1	Chose Splitting Directions . . . . .	153
5.2.3.2	Propagation . . . . .	155
5.2.3.3	Compute Collision Probability . . . . .	155
5.3	Results . . . . .	158
5.3.1	Perturbed Uncertainty Propagation . . . . .	158
5.3.2	GMM Test Case . . . . .	161
5.3.3	MGMMs . . . . .	162
5.4	Conclusion . . . . .	169
<b>Chapter 6. Conclusions</b>		<b>172</b>
6.1	Uncertainty Propagation . . . . .	172
6.2	Collision Probability Computation . . . . .	175
6.3	Future Work . . . . .	177
<b>Appendices</b>		<b>183</b>
<b>Appendix A. Publications</b>		<b>184</b>
<b>Bibliography</b>		<b>189</b>
<b>Vita</b>		<b>208</b>

## List of Tables

2.1	The 7-component univariate splitting library with $\sigma^2 = \left(\frac{1}{N}\right)$ .	41
2.2	Initial state, Gaussian uncorrelated uncertainty, and the nonlinearity measure for a space object in GEO . . . . .	65
2.3	Initial mean, variance, and nonlinearity measure for the Extended Freudenstein & Roth function . . . . .	67
3.1	Initial osculating orbit elements for a MEO object . . . . .	90
3.2	Initial uncertainty expressed in the RIC coordinate frame for a MEO object . . . . .	90
3.3	Initial osculating orbit elements from Jones et. al. [75] for an object in a Molniya orbit . . . . .	92
3.4	Initial uncertainty expressed in Cartesian coordinates in the ECI frame for an object in a Molniya orbit . . . . .	93
3.5	Initial osculating orbit elements for a GTO object . . . . .	95
3.6	Initial uncertainty expressed in Cartesian coordinates in the ECI frame for a GTO object . . . . .	96
3.7	Initial osculating orbit expressed in Equinoctial Elements for a LEO object [68] . . . . .	97
3.8	Initial uncertainty expressed in Equinoctial Elements for a LEO object [68] . . . . .	97
4.1	Test hardware specifications . . . . .	128
4.2	Initial conditions for a long duration LEO encounter . . . . .	135
5.1	Initial conditions for the uncertainty propagation runs with a standard deviation of 10 m and 1 m/s in each of the position and velocity coordinates, respectively . . . . .	160
5.2	State conditions in the ECI frame at epoch ( $t_{CA}$ -2.5 orbital periods of the primary RSO) for the for the GEO (Primary) and GTO (Secondary) RSOs . . . . .	167
5.3	State uncertainty in the RIC frame at epoch ( $t_{CA}$ -3 days) for the GEO (Primary) and GTO (Secondary) RSOs . . . . .	167

## List of Figures

1.1	Location of 15,104 RSOs available as TLEs from the public space catalog . . . . .	3
1.2	Estimated number of objects in LEO [3] . . . . .	4
1.3	Location of the 29 sensors that are part of the Space Surveillance Network [5] . . . . .	5
1.4	Monthly effective mass of objects in Earth orbit by region [6] .	7
1.5	An example Conjunction Summary Message provided by JSpOC [80]	9
1.6	A schematic representation of the role of CAESAR and CARA acting as a middle-man between JSpOC and satellite operators (O/O) in identifying high interest events (HIE) [1] . . . . .	11
1.7	The two phases for the collision probability computation. The ellipsoids represent the Gaussian uncertainty in the states of the RSOs at $t_0$ . The uncertainties no longer have a Gaussian distribution at $t_a$ . . . . .	12
1.8	The evolution of the $3\sigma$ bounds and the mean of an initially IID Gaussian PDF through a linear transformation followed by a nonlinear transformation. Dotted lines represent the $3\sigma$ probability envelope and the black dots represent the mean. . . . .	16
2.1	PDFs of the individual elements subject to the equality and inequality constraints for $N = 7$ . . . . .	35
2.2	The optimization procedure for finding a univariate splitting library . . . . .	36
2.3	The run time required by the trust-region optimization to compute the locations of the means and the corresponding weights for the univariate splitting libraries . . . . .	37
2.4	The resulting univariate GMMs for the three different splitting rules and for various number of elements . . . . .	39
2.5	$L_1$ distance, the maximum and minimum weights, and the maximum mean as a function of the number of elements used in the GMM for various rules for $\sigma$ . . . . .	40
2.6	PDFs of the univariate 9 element GMM of the standard normal distribution using a complete and a recursive splitting technique for $\sigma^2 = (1/N)$ . . . . .	41

2.7	The unit axes of $\mathcal{F}_I$ , $x$ and $y$ , and the desired splitting direction, $\mathbf{a}$ , in the three relevant reference frames for a 2-dimensional case	44
2.8	Sample point cloud from the covariance ellipses of the multivariate GMM	49
2.9	Point clouds of the resulting non-Gaussian distribution using MC and an 11 element GMM	49
2.10	The loglikelihood of the MC simulation and the nonlinearity measure of the function with $5 \times 10^5$ sample points and 5 splits along $\boldsymbol{\nu}_1$ or $\boldsymbol{\nu}_2$ and increasing rotation angle $\theta$	53
2.11	Performance of the various univariate splitting libraries for a nonlinear test case. Higher LL indicates a closer fit.	55
2.12	Splitting a bivariate IID Gaussian into an MGMM with and without a minimum weight threshold and 5 elements along $\boldsymbol{\nu}_1$ and $\boldsymbol{\nu}_2$ . Note that the covariance matrices are multiplied by their weights for ease of visualization	56
2.13	An MC run with $5 \times 10^5$ points for the conversion from Polar to Cartesian coordinates	59
2.14	$\Delta LL$ of a $10^5$ point MC distribution with respect to the MGMM with increasing number of elements and $LL(39, 39) = -7.5789 \times 10^5$ for the conversion from Polar to Cartesian coordinates	60
2.15	$5 \times 10^5$ points sampled from the resulting MGMMs with $N_1$ splits along $\boldsymbol{\nu}_1$ and $N_2$ splits along $\boldsymbol{\nu}_2$ , $(N_1, N_2)$ for the conversion from Polar to Cartesian coordinates	61
2.16	An MC run with $10^4$ points for the bivariate ODE	62
2.17	$\Delta LL$ of a $10^5$ point MC distribution generated with respect to the MGMM with increasing number of elements and $LL(39, 39) = -2.9264 \times 10^5$ for the bivariate ODE	63
2.18	$10^4$ points sampled from the resulting MGMMs with $N_1$ splits along $\boldsymbol{\nu}_1$ and $N_2$ splits along $\boldsymbol{\nu}_2$ , $(N_1, N_2)$ for the bivariate ODE	64
2.19	$\Delta LL$ of a $10^5$ point MC distribution generated with respect to the MGMM with increasing number of elements and $LL(39, 39) = 4.8209 \times 10^6$ for the space object in GEO	66
2.20	$LL$ of a $10^5$ point MC distribution with respect to GMMs and MGMMs for the 10-dimensional case	69
3.1	Terms required for multivariate polynomials and the number of quadrature points for a 6 dimensional input state. The LS regression requires $2L$ points	80
3.2	The difference in a three element refinement of the standard normal distribution between the GMM-PC and the ME-gPC techniques	83

3.3	Performance of approximating a 13 <sup>th</sup> -order Hermite PCE using lower order ( $3 \leq l \leq 10$ ) GMM-PC using 1,000,000 samples . . . . .	86
3.4	Coefficient values for the corresponding univariate polynomial orders for the directions of the IID state for a one element GMM-PC used to propagate uncertainty of an object in MEO. The trend of the maximum values of the coefficients as the order increases is indicated by the arrow. . . . .	91
3.5	Two-sample univariate Cramer-von-Mises metric in the RIC frame for GMM-PC, split along velocity, with respect to an MC simulation of 1,000,000 samples of an object in MEO. $L$ is the total number of coefficients required per direction for the multivariate polynomial and is analogous to the compute cost ( $2L$ ). . . . .	92
3.6	Two-sample univariate Cramer-von-Mises metric in the RIC frame for GMM-PC, split along velocity, with respect to an MC simulation of 1,000,000 samples of an object in MEO. $L$ is the total number of coefficients required per direction for the multivariate polynomial and is analogous to the compute cost ( $2L$ ). . . . .	93
3.7	100,000 samples for velocity in the Radial-Intrack plane from the MC, a PCE, and GMM-PC and GMM simulations with an initial split applied along $\mathbf{v}_v$ for the uncertainty propagation of a MEO object . . . . .	94
3.8	Two-sample univariate Cramer-von-Mises metric in the RIC frame for GMM-PC, split along velocity, with respect to an MC simulation of 1,000,000 samples of an object in a Molniya orbit. $L$ is the total number of coefficients required per direction for the multivariate polynomial and is analogous to the compute cost. . . . .	95
3.9	Histograms in the RIC frame of 1,000,000 samples from the MC, a PCE, and a GMM-PC and GMM with an initial split applied along $\mathbf{v}_v$ for the uncertainty propagation of an object in a Molniya orbit . . . . .	98
3.10	Two-sample univariate Cramer-von-Mises metric for PCE and GMM-PC, split along velocity, with respect to an MC simulation of 1,000,000 samples. $L$ is the total number of coefficients required per direction for the multivariate polynomial and is analogous to the compute cost. . . . .	99
3.11	Two-sample univariate Cramer-von-Mises metric for PCE and GMM-PC, split along the semi-major axis, with respect to an MC simulation of 1,000,000 samples. . . . .	99
3.12	Two-sample univariate Cramer-von-Mises metric for PCE and Weighted Least Squares GMM-PC, split along velocity, with respect to an MC simulation of 1,000,000 samples. . . . .	100

4.1	Samples from the primary (blue) and secondary (red) RSOs required for a MC simulation with $N_{MC}$ samples in the one-on-one and the all-on-all approaches . . . . .	108
4.2	Function $y(\tau)$ with the information available for computing a quartic interpolating polynomial . . . . .	111
4.3	Approximating the quintic polynomial $y(\tau) = 4\tau^5 + \tau^4 + \tau^3 + 4\tau^2 + 1$ by the quartic polynomial $y_*(\tau) = 9\tau^4 - 3\tau^3 + 4\tau^2 + 1$ in $\tau \in [0, 1]$ . . . . .	113
4.4	Effect of changing the $R_{min}$ on the probability of collision for cases from Alfano's paper [17] . . . . .	117
4.5	Memory layout of the GPU . . . . .	120
4.6	Case 7 from Alfano's [17] paper using two-body dynamics with the standard deviation of the in-track velocity of the primary RSO increased by a factor of 30. 10,000,000 samples are generated at $t_{CA}$ using MC, a Gaussian distribution in Cartesian coordinates, a Gaussian distribution in EE, and an 8 <sup>th</sup> order PCE in Cartesian coordinates. . . . .	122
4.7	Spatial and temporal setup of random encounters . . . . .	123
4.8	Normalized histogram of the relative velocities $\Delta v$ at $t_{CA}$ for the 1,000,000 randomly generated encounters at $t_{CA}$ . . . . .	126
4.9	CDF of the error in computing $r_{min}$ for time step sizes $\Delta t = [10, 15, 20, 25, 50, 100]$ s . . . . .	127
4.10	Error in estimating the time of closest approach and minimum separation for $\Delta t = 10$ s . . . . .	128
4.11	True values of the separation and the accuracy of the quartic interpolation for a randomly generated encounter . . . . .	129
4.12	CPU Wall time and GPU speedup, including interpolation, for various dynamics fidelity levels. Every RK4 step requires 8 function evaluations and one interpolation . . . . .	130
4.13	Timing results with two-body dynamics, increasing particles, and decreasing time-step size . . . . .	131
4.14	Timing results with all modeled perturbations and a 120 element mascon model (SH = 9×9), increasing particles, and decreasing time-step size . . . . .	132
4.15	Timing results with all modeled perturbations and a 240 element mascon model (SH = 13×13), increasing particles, and decreasing time-step size . . . . .	133
4.16	Timing results with all modeled perturbations and a 480 element mascon model (SH = 19×19), increasing particles, and decreasing time-step size . . . . .	134



4.17	Timing results with all modeled perturbations and a 960 element mascon model ( $SH = 27 \times 27$ ), increasing particles, and decreasing time-step size . . . . .	135
4.18	Change in the encounter between two RSOs in GEO due to SRP. $CR_P = 1.1$ , $AMR_P = 0.4 \text{ m}^2/\text{kg}$ , $CR_S = 1.5$ , $AMR_S = 1.1 \text{ m}^2/\text{kg}$ . . . . .	136
4.19	Timing results with two-body dynamics and SRP, increasing particles, and decreasing time-step size . . . . .	136
4.20	Change in the encounter between two RSOs in LEO due to drag. $BC_P = 0.05179 \text{ m}^2/\text{kg}$ , $BC_S = 0.00687 \text{ m}^2/\text{kg}$ . . . . .	137
4.21	Timing results with two-body dynamics, J2, and cannonball drag, and increasing particles and decreasing time-step size . . . . .	137
5.1	The two phases for the collision probability computation . . . . .	144
5.2	Initial conditions from the combined covariance matrix at $t_0$ that result in collisions at $t_0$ ( $\star$ ), and at times $t_1$ ( $\bullet$ ), and $t_2$ ( $\blacksquare$ ) during the conjunction . . . . .	148
5.3	The all-on-all approach for collision probability computation using 2 element MGMMs for the primary and the secondary RSOs . . . . .	157
5.4	Sensitivity and accuracy of the splitting directions using a 19 element split . . . . .	160
5.5	Collision probability performance using GMMs for Alfano Case 7 and the quadrature method . . . . .	162
5.6	Collision probability performance using MGMMs for the LEO case with $R = 10 \text{ m}$ and the quadrature method, compared to a MC simulation with $10^8$ cases ( $2 \times 10^8$ propagations) . . . . .	165
5.7	Collision probability performance using MGMMs for the LEO case with $R = 20 \text{ m}$ and the quadrature method, compared to a MC simulation with $10^8$ cases ( $2 \times 10^8$ propagations) . . . . .	166
5.8	Collision probability performance using MGMMs for the GEO case with $R = 40 \text{ m}$ and the linear method, compared to an MC simulation with $2 \times 10^8$ cases ( $4 \times 10^8$ propagations) . . . . .	168

# Chapter 1

## Introduction

Space situational awareness (SSA) is a key area of research for both the civil and military space communities [99]. SSA involves understanding the space environment and its influence on controlled and uncontrolled space assets [110]. The National Space Policy of the United States of America specifically mentions SSA:

Develop, maintain, and use space situational awareness (SSA) information from commercial, civil, and national security sources to detect, identify, and attribute actions in space that are contrary to responsible use and the long-term sustainability of the space environment.

The guidelines for civil space agencies, i.e. NASA, additionally include looking for Near-Earth Objects (NEOs). The European Space Agency (ESA) also has an SSA aim of collecting data about hazards to orbit and ground infrastructure, which includes the potential impacts of NEOs with the Earth<sup>1</sup>. The SSA Initial Capabilities Document (ICD) of the Department of Defense (DoD) from 2011 divides SSA into four functional areas [119]:

---

<sup>1</sup>ESA SSA Programme Overview, [www.esa.int/Our\\_Activities/Operations/Space\\_Situational\\_Awareness](http://www.esa.int/Our_Activities/Operations/Space_Situational_Awareness), [Accessed October 12 2015]

1. Detect/Track/Identification (D/T/ID): Find and catalog objects in space
2. Characterization: Determine strategy and intent of adversaries
3. Threat Warning and Assessment: Differentiate between an attack and space weather
4. Data Integration and Exploitation: Combine data from different sources in order to solve a problem

The work presented here is a part of the D/T/ID phase, although uncertainty propagation and quantification is involved in all phases, which is carried out by the Joint Space Operations Center (JSpOC). JSpOC plans and executes missions by the Joint Functional Command for Space (JFCC Space), which is a part of the US Strategic Command (USSTRATCOM).

## 1.1 Space Objects in Earth Object

The increased population in space of both controllable and uncontrollable Resident Space Objects (RSOs) creates challenges for SSA [7]. The U.S. Space Surveillance Network (SSN), operated by JSpOC, currently tracks over 21,000 RSOs larger than 10 cm [4, 106]. The tracked RSOs include over 1,305 active assets, which are increasing every year. The locations of over 15,000 objects cataloged and made available to the public [81] as NORAD Two-Line Element sets (TLEs) are shown in Figure 1.1. Approximately 500,000 untracked particles exist in the size range of 1 to 10 cm. Additionally, there are

over 100 million particles smaller than 1 cm.<sup>2</sup> The number of objects in LEO estimated using various measurements is shown in Figure 1.2.

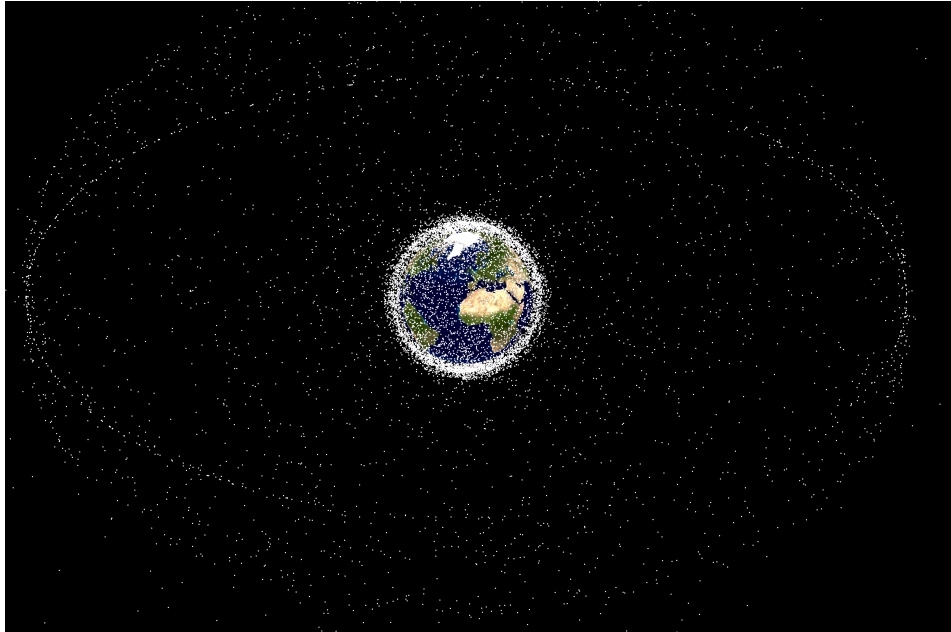


Figure 1.1: Location of 15,104 RSOs available as TLEs from the public space catalog

The difference between the populations of cataloged and uncataloged objects is mostly due to the limitation of the capabilities of sensors [3]. A few RSOs are in the catalog, but the information is not publicly available.

Due to the high relative velocities of RSOs, collisions with relatively small objects, i.e. greater than 10 cm, can be catastrophic for an active satellite. Debris shields can be used to protect vital components of spacecraft from particles with a diameter of up to 1 cm. Collision between any of the RSOs,

---

<sup>2</sup><http://orbitaldebris.jsc.nasa.gov/faqs.html> [Accessed November 17 2015]

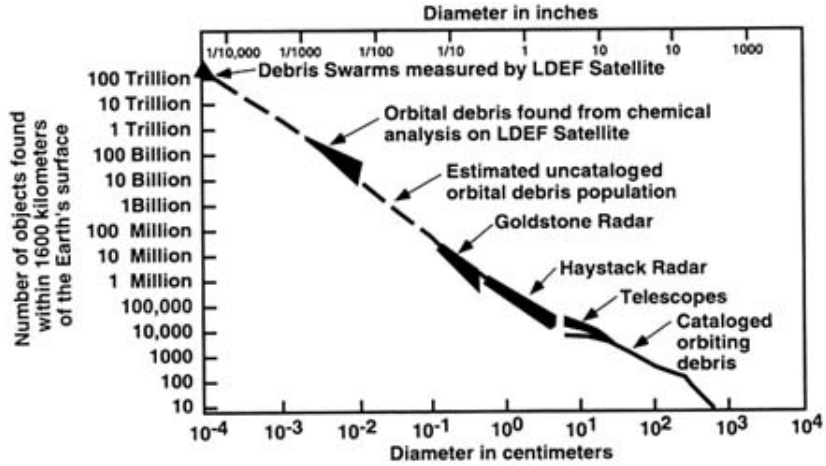


Figure 1.2: Estimated number of objects in LEO [3]

active or inactive, increases the debris count. To date, two major collisions have caused large step functions in the debris population. The most disastrous collision was an intentional anti-satellite test by China in 2007, which increased the tracked debris count by over 2000 [73]. Due to the high altitude, most of the debris is not predicted to burn up in the atmosphere. The second collision was in 2009 between an active Iridium communication satellite and a deactivated Russian military communications satellite in Low Earth Orbit (LEO), which also increased the number of tracked RSOs by approximately 2000. However, many pieces of debris from this collision have since burned up in the atmosphere. Once debris density in LEO reaches a minimum threshold level, extra debris from a collision has the possibility of causing a catastrophic runoff on the number of collisions, which renders the orbital regime unusable. This predicted phenomenon is known as the Kessler syndrome [82].

Various techniques are used for orbital debris measurements because of the wide range of size and orbital regimes. Optical and radar measurements are typically used for high and low altitude debris, respectively. Investigation of spacecraft surfaces is carried out for computing the flux of very small debris that are not large enough for a catastrophic collision. The locations of sensors that are a part of the Space Surveillance Network (SSN) are shown in Figure 1.3.



Figure 1.3: Location of the 29 sensors that are part of the Space Surveillance Network [5]

The limitations on the capability, geographic distribution, availability, and the number of sensors mean that RSOs cannot be tracked continuously by the SSN. Therefore, the locations are first predicted using various uncertainty propagation methods and then measurements are occasionally made when possible [5].

The distribution of LEO debris in the orbital elements phase space is critical for visualization. However, empirically obtaining the data required to adequately characterize the debris population is impossible due to resolution limitations of ground tracking systems. Larger debris can be tracked reliably, but a majority of debris in LEO is beyond the resolution of state-of-the-art tracking facilities, such as the Haystack and HAX radars. These radar facilities can only track debris larger than approximately one centimeter [122]. As mentioned earlier, the SSN currently maintains tracking data for tens of thousands of debris; however, the debris are generally larger than 10 cm in size [81, 122]. Therefore, the debris research community simply does not have the data necessary to fully characterize the debris field over all length scales. Furthermore, this deficiency renders full deterministic simulations of the debris field impossible. As a result, evolutionary studies of the debris field are generally carried out using stochastic models employing Monte Carlo techniques. The effective mass distribution of objects in the various Earth orbit regions is shown in Figure 1.4. The amount of mass decreases as the altitude decreases. The inverse relation is due to the increasing difficulty of launching and tracking for higher altitudes.

A list of important reference documents concerning orbital debris is provided by the NASA Orbital Debris Program Office <sup>3</sup>. Heavily used orbital regions are Low Earth Orbit (LEO), Sun-Synchronous Orbit, and Geosynchronous Orbit (GEO) [3]. LEO is easy to achieve, the low altitude allows

---

<sup>3</sup><http://orbitaldebris.jsc.nasa.gov/library/references.html> [Accessed November 19 2015]

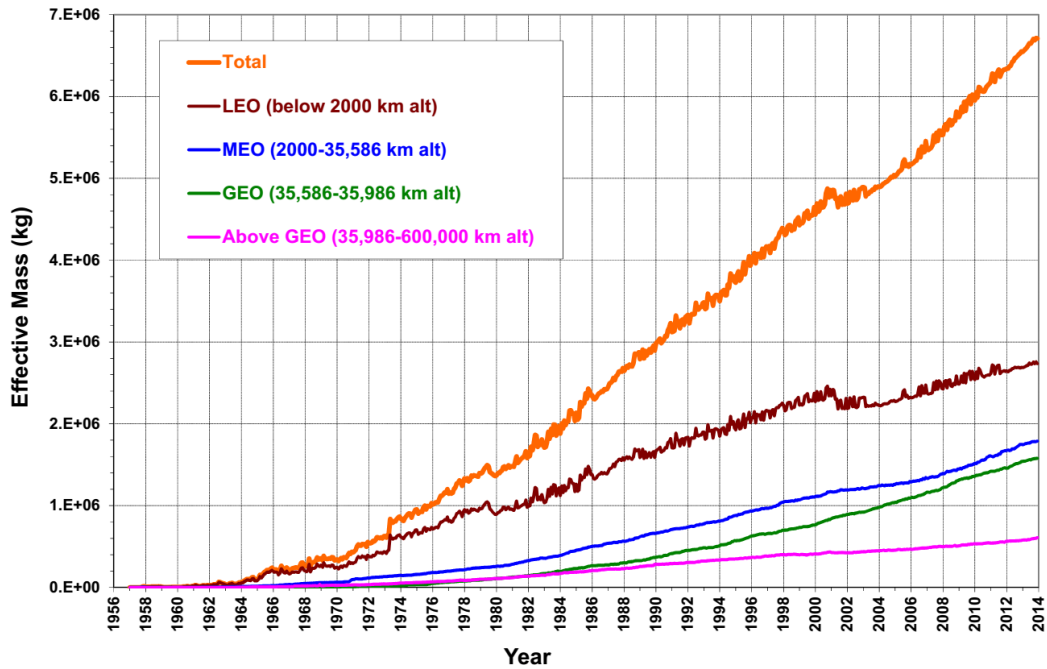


Figure 1.4: Monthly effective mass of objects in Earth orbit by region [6]

for better resolution for Earth sensing applications, and the Earth's magnetic field is a very effective shield against hazards in interplanetary space. The Earth is viewed in almost identical lighting conditions in Sun-Synchronous orbits. Finally, the geostationary orbit, which is a subset of GEO is makes communication with spacecraft easy.

## 1.2 Conjunction Assessment

The space catalog contains the orbital state information and the inherent uncertainties due to tracking and modeling errors, which are used to predict collisions. The active maintenance of the catalog is an international concern



of extraordinary size and complexity, and its scope is unbounded with the potential for exponential growth. Effective automation, scheduling, and management of the decision making process associated with maintaining the U.S. Space Object Catalog is a challenging problem of increasing importance [95].

An extensive catalog is necessary for the safety of space assets; but the information about RSOs in the catalog is meaningful for collision avoidance when collisions are accurately predicted. Only a probability of a collision occurring can be computed because of the uncertainty present in the states of the RSOs. The calculation of collision probabilities is highly computationally intensive and therefore, it is imperative to prune the search space by removing the RSO pairs that have practically no chance of collision. The possible conjunctions are screened using geometric filters such as apogee/perigee heights. The orbital path filters account for the orbit plane orientations. A time filter is subsequently used to check if the two RSOs will come near each other within a time window of interest. The conjunction filters were originally derived by Hoots et. al [65] to use only TLEs. The filters have since been updated for use with a more realistic perturbation model [60, 111, 15].

After identifying the candidate RSO pairs that pass the conjunction screening filters, the actual collision probabilities of RSO pairs have to be accurately computed. Presently, JSpOC screens for conjunctions up to 7 days ahead for LEO objects and 3 days for GEO objects. A conjunction summary message (CSM) is then emailed directly to satellite operators with state vectors and covariances to facilitate collision probability computations. An example of a CSM is shown in Figure 1.5. In the past, however, only the miss dis-

tances were used for maneuver decisions, which fails to capture the stochastic aspect of the orbital conjunction problem. To efficiently plan maneuvers, a full collision risk assessment has to be carried out.

```

CONJUNCTION SUMMARY MESSAGE FOR: SATELLITE A
MESSAGE CREATION TIME: 2020 171 (12 JUN) 22:31:12.000 CENTER: JSPOC
MESSAGE VERSION: V2.0
*****
RELATIVE DATA:
TIME OF CLOSEST APPROACH (UTC): 2020 172 (13 JUN) 22:37:52.618
MISS DISTANCE (M): 715
RELATIVE SPEED (M/S): 14762
CLOSEST APP. REL. POSITION (M): 27.4 -70.2 711.8
CLOSEST APP. REL. VELOCITY (M/S): -7.2 -14692.0 -1437.2
*****
ASSET: 12345 INT. DES.: 1917-010A
COMMON NAME: SATELLITE A
TIME OF LAST ACCEPTED OB: <24 HOURS FROM MESSAGE CREATION TIME
LUPI/DC SPAN USED (DAYS): 7.88/ 5.50 RESIDUAL ACCT: 97.8 %
NUM OBS AVAIL/USED: 592/ 418
APOGEE (KM): 779 PERIGEE (KM): 765 INCLINATION (DEG): 86.4
RADAR CROSS SECTION (SCALED): LARGE (>1m sq) WEIGHTED RMS: 0.864
BALLISTIC COEFFICIENT (M2/KG): 0.045663
SOLAR RADIATION PRESSURE COEFFICIENT (M2/KG): 0.000000
ENERGY DISSIPATION RATE (W/KG): 4.54570E-05
GEOPOTENTIAL: EGM-96 36Z,36T DRAG: JACCHIA70DCA LUNAR/SOLAR: ON
SOLAR RAD PRESS: OFF SOLID EARTH TIDES: OFF IN-TRACK THRUST: OFF
*****
ASSET TDR POSITION (M): 2570098.594 2244663.456 6281494.300
ASSET TDR VELOCITY (M/S): 4418.768701 4833.542969 -3526.781960
*****
CONJUNCTING SATELLITE: 12345 INT. DES.: 1917-010B
COMMON NAME: SATELLITE B
TIME OF LAST ACCEPTED OB: <24 HOURS FROM MESSAGE CREATION TIME
LUPI/DC SPAN USED (DAYS): 2.63/ 2.63 RESIDUAL ACCT: 97.8 %
NUM OBS AVAIL/USED: 59/ 58
APOGEE (KM): 786 PERIGEE (KM): 414 INCLINATION (DEG): 98.8
RADAR CROSS SECTION (SCALED): SMALL (<0.1m sq) WEIGHTED RMS: 0.864
BALLISTIC COEFFICIENT (M2/KG): 0.118668
SOLAR RADIATION PRESSURE COEFFICIENT (M2/KG): 0.075204
ENERGY DISSIPATION RATE (W/KG): 5.40900E-03
GEOPOTENTIAL: EGM-96 36Z,36T DRAG: JACCHIA70DCA LUNAR/SOLAR: ON
SOLAR RAD PRESS: ON SOLID EARTH TIDES: OFF IN-TRACK THRUST: OFF
*****
SAT. TDR POSITION (M): 2569542.299 2245102.187 6281596.315
SAT. TDR VELOCITY (M/S): -2888.611701 -6007.242986 3328.778993
*****
ASSET COVARIANCE: (1,1) TO (6,6) (M^2, M^2/S, M^2/S^2)
U V W UD VD WD
4.142E+01 -8.579E+00 -2.312E+01 0.000E+00 0.000E+00 0.000E+00
-8.579E+00 2.533E+03 1.336E+01 0.000E+00 0.000E+00 0.000E+00
-2.312E+01 1.336E+01 7.098E+01 0.000E+00 0.000E+00 0.000E+00
0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
SAT. COVARIANCE: (1,1) TO (6,6) (M^2, M^2/S, M^2/S^2)
U V W UD VD WD
1.337E+03 -4.806E+04 -3.298E+01 0.000E+00 0.000E+00 0.000E+00
-4.806E+04 2.492E+06 -7.588E+02 0.000E+00 0.000E+00 0.000E+00
-3.298E+01 -7.588E+02 7.105E+01 0.000E+00 0.000E+00 0.000E+00
0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
*****
ASSET COVARIANCE (7,1) TO (8,8) (M^3/KG, M^3/KG-S, M^4/KG^2)
(7,1) (7,2) (7,3) (7,4) (7,5) (7,6)
0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
(7,7) (7,8)
0.000E+00 0.000E+00
(8,1) (8,2) (8,3) (8,4) (8,5) (8,6)
0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
(8,7) (8,8)
0.000E+00 0.000E+00
SAT. COVARIANCE (7,1) TO (8,8) (M^3/KG, M^3/KG-S, M^4/KG^2)
(7,1) (7,2) (7,3) (7,4) (7,5) (7,6)
0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
(7,7) (7,8)
0.000E+00 0.000E+00
(8,1) (8,2) (8,3) (8,4) (8,5) (8,6)
0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
(8,7) (8,8)
0.000E+00 0.000E+00

```

Figure 1.5: An example Conjunction Summary Message provided by JSPOC [80]

An accurate collision probability helps the operator of a satellite make more informed decisions about the possible maneuvers. An unnecessary maneuver increases propellant and operational costs, decreases the mission lifetime, and may result in future conjunctions. However, not making a necessary maneuver may result in a collision with catastrophic consequences for the mission. For a possible conjunction, the states with associated uncertainties of the

RSOs at a given time  $t_0$  are known from the space catalog. The screening process identifies a time  $t_1 > t_0$  when there is a close approach of the two RSOs. The essence of the problem is to find the probability that the two RSOs will collide within a time window around  $t_1$ , i.e.  $t \in [t_a = t_1 - \Delta t_-, t_b = t_1 + \Delta t_+]$ . The benefit of having an accurate space catalog and a good screening process is that the conjunction window can be identified far in the future such that  $t_a \gg t_0$ .

The CSM only provides information on the states and the covariances at the predicted close approach. However, the complicated task of analyzing the risk of collision is given to the operators who might not have the necessary tools. To aid operators, the French Space agency (CNES) has implemented a trial public service known as Conjunction Analysis and Evaluation Service, Alerts and Recommendations (CAESAR). CAESAR receives CSMs up to 7 days in advance for all orbit regimes and it helps the subscribing operators in deciding if the collision avoidance maneuver is required. The role of CAESAR as middle-man in effectively informing operators about high interest events (HIE) is shown in Figure 1.6. HIEs are the CSMs that lead to an avoidance action being taken.

NASA has a similar middle-man service collision risk assessment service for all NASA unmanned missions known as the Robotic Conjunction Assessment Risk Analysis (CARA). NASA provides predicted ephemerides daily for all the missions to be screened against a high accuracy catalog available at JSpOC. The screening is 7 days in advance for LEO and up to 10 days in advance for GEO. The high accuracy catalog is unfortunately not publicly

available as part of CelesTrak. CARA then uses the customized higher accuracy CSMs provided by JSpOC to compute risk of collision and effective collision mitigation strategies for HIE [2].

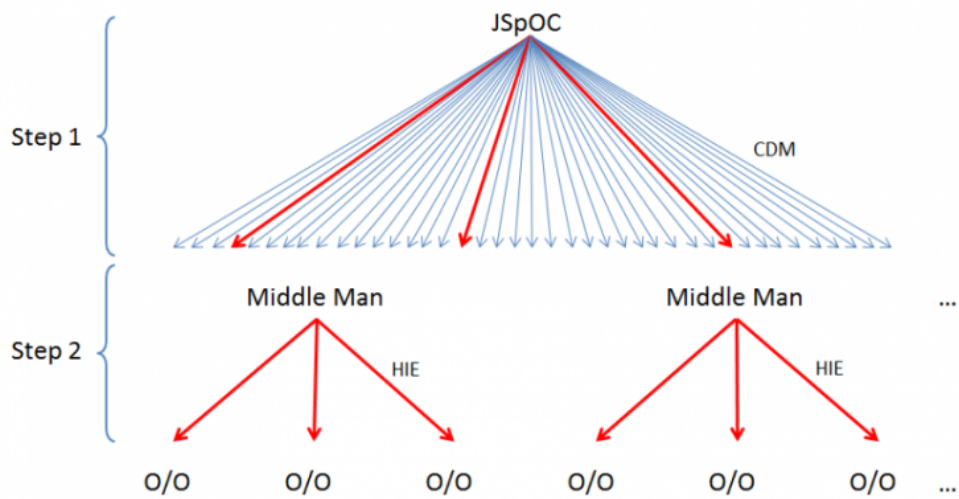


Figure 1.6: A schematic representation of the role of CAESAR and CARA acting as a middle-man between JSpOC and satellite operators (O/O) in identifying high interest events (HIE) [1]

The focus of this dissertation lies in the development of novel approaches for accurate estimation of collision probabilities of RSOs. The problem is divided into two separate phases/subproblems. Section 1.4 presents Phase I, where the states and uncertainties are simulated forward in time from  $t_0$  to  $t_1$ . Section 1.5 presents Phase II, where the collision probability is computed over  $t \in [t_a, t_b]$ . A pictorial explanation of the two phases is shown in Figure 1.7.

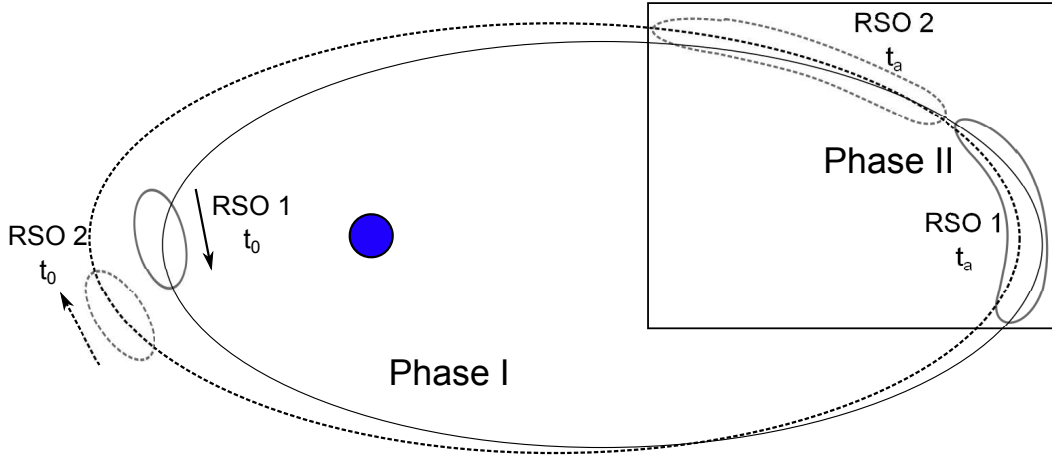


Figure 1.7: The two phases for the collision probability computation. The ellipsoids represent the Gaussian uncertainty in the states of the RSOs at  $t_0$ . The uncertainties no longer have a Gaussian distribution at  $t_a$

### 1.3 Outline

The dissertation is divided into two main parts that delve into Phase I and Phase II, respectively. The uncertainty of RSOs is first propagated from the time of last measurement in Phase I. The numerical collision probability value between the two RSOs is computed in Phase II. Sections 1.4 and 1.5 show the link between uncertainty propagation and collision probability computation, respectively, and their importance for the future of space flight. These concepts form the building blocks of the chapters of this dissertation. The main contributions of the dissertation that advance the state of the art are presented in Section 1.6.

The PDF of the RSO state uncertainty is assumed to be a multivariate Gaussian distribution at time  $t_0$  and finding an accurate representation of the

non-Gaussian distribution at a future time  $t_1$  is shown in Chapters 2 and 3. The computation of the collision probability between two RSOs during the close approach is the focus of Chapters 4 and 5. The conclusions drawn from the work carried out as part of the dissertation and the many avenues of future work are presented in Chapter 6. Finally, publications that are wholly, or partially based on the material in the chapters are listed in Appendix A.

In Chapter 2, a multivariate Gaussian distribution is converted into a Gaussian Mixture Model (GMM), which is a weighted sum of Gaussian distributions. Each element of the GMM is propagated through a nonlinear function. Although each element remains Gaussian after the propagation, the weighted sum is used to approximate the resulting non-Gaussian distribution.

In Chapter 3, each element is no longer constrained to remain a Gaussian distribution after propagation. Instead, orthogonal Hermite polynomials known as Polynomial Chaos Expansions (PCE) are used to compute the behavior of the elements.

After the uncertainty of the RSOs is propagated in Phase I using any of the two uncertainty propagation methods presented in Chapters 2 and 3, the collision probability over Phase II is computed using a Monte Carlo (MC) simulation in Chapter 4. A Graphics Processing Unit (GPU) is used to parallelize the MC computation so that it is feasible on a single workstation.

The computational load of the collision probability calculations is reduced in Chapter 5 by using GMMs in Cartesian coordinates to describe the probability density functions of the two RSOs.

## 1.4 Uncertainty Propagation

Uncertainty propagation is a major factor in the accuracy of the collision probability. This problem is addressed in Phase I as shown in Figure 1.7, i.e. the evolution of the uncertainty of the RSOs from epoch to near the encounter. The state and associated uncertainty of the RSOs are available at time  $t_0$  in the catalog from the Orbit Determination (OD). What will the state and uncertainty be at a future time  $t_a > t_0$ ? This is a general problem that is important for many applications because it is essentially the prediction step of a nonlinear filter without any new measurements going forward.

A commonly made assumption is that the state of the RSO at  $t_0$  has a Gaussian distribution. Most filters such as the Extended Kalman Filter (EKF) and Batch Least Squares (BLS) that are used for OD and other estimation problems output a mean and a covariance estimate of the RSO's state. A mean and a covariance allow for any underlying PDF because they are only the first two moments. The PDFs are, however, assumed to be Gaussian because the Gaussian distribution is the maximum entropy PDF when only the mean and covariance are known. Therefore, the Gaussian distribution is not always representative of the uncertainty distribution of the RSOs. A maximum entropy distribution is the one that corresponds to the least information about the underlying PDF. In the case of a Gaussian distribution, the first two moments, the mean and the covariance are sufficient to exactly describe the PDF. Time  $t_0$  is assumed to be the time of last measurement and the uncertainty in the *a posteriori* state estimate is typically quite small. The better the OD process, i.e. better and more frequent measurements, the smaller the

uncertainty. Therefore, the assumption of a small Gaussian PDF for the state uncertainty of most controlled and uncontrolled space assets is valid for many cases. A Gaussian distribution also has many properties that make it analytically easy to manipulate. If a multivariate random variable  $\mathbf{x}$  has a Gaussian distribution, the PDF is represented as:

$$p(\mathbf{x}) = p_g(\mathbf{x}; \boldsymbol{\mu}_x, \mathbf{P}_{xx}) \quad (1.1)$$

where the  $\boldsymbol{\mu}_x$  is the mean of  $\mathbf{x}$ ; i.e. the first moment.  $\mathbf{P}_{xx}$  is the covariance of  $\mathbf{x}$ , i.e. the second moment about  $\boldsymbol{\mu}_x$ . The result of a linear function of  $\mathbf{x}$ ,  $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{B}$  also has a Gaussian distribution where:

$$p(\mathbf{y}) = p_g(\mathbf{y}; \boldsymbol{\mu}_y, \mathbf{P}_{yy}) \quad (1.2a)$$

$$\boldsymbol{\mu}_y = \mathbf{A}\boldsymbol{\mu}_x + \mathbf{B} \quad (1.2b)$$

$$\mathbf{P}_{yy} = \mathbf{A}\mathbf{P}_{xx}\mathbf{A}^T \quad (1.2c)$$

A general nonlinear function  $\mathbf{z} = f(\mathbf{x})$ , however, does not result in a Gaussian distribution. In fact, this reality is the primary motivation of the uncertainty propagation problem.

$$p(\mathbf{z}) \neq p_g(\mathbf{z}; \boldsymbol{\mu}_z, \mathbf{P}_{zz}) \quad (1.3)$$

A beneficial property of Gaussian distributions is that they can always be related to an independent and identically distributed (IID) standard normal distribution. Let  $\boldsymbol{\xi} = [\xi_1, \dots, \xi_n]^T$  be a multivariate random variable where each of the univariate variables are distributed with the standard normal distribution. The PDF of  $\boldsymbol{\xi}$  is:

$$p(\boldsymbol{\xi}) = p_g(\boldsymbol{\xi}; \mathbf{0}_n, \mathbf{I}_{n \times n}) = \prod_{i=1}^n p_g(\xi_i; 0, 1) \quad (1.4)$$



If  $\mathbf{x}$  is the same dimension as  $\boldsymbol{\xi}$ , a linear transformation exists from  $\boldsymbol{\xi}$  to  $\mathbf{x}$

$$\mathbf{x} = \mathbf{S}_{\mathbf{xx}}\boldsymbol{\xi} + \boldsymbol{\mu}_{\mathbf{xx}} \quad (1.5a)$$

$$\mathbf{P}_{\mathbf{xx}} = \mathbf{S}_{\mathbf{xx}}\mathbf{S}_{\mathbf{xx}}^T \quad (1.5b)$$

where  $\mathbf{S}_{\mathbf{xx}}$  is the square-root of  $\mathbf{P}_{\mathbf{xx}}$  and is computed using Cholesky or spectral decomposition.

The general uncertainty propagation problem is to accurately describe the PDF of  $\mathbf{z}$ , which is a nonlinear transformation acting on  $\mathbf{x}$ , when  $\mathbf{x}$  has Gaussian distribution. The transformation of the PDF of  $\boldsymbol{\xi}$  to the PDF of  $\mathbf{x}$  and finally to the PDF of  $\mathbf{z}$  is shown conceptually in Figure 1.8.

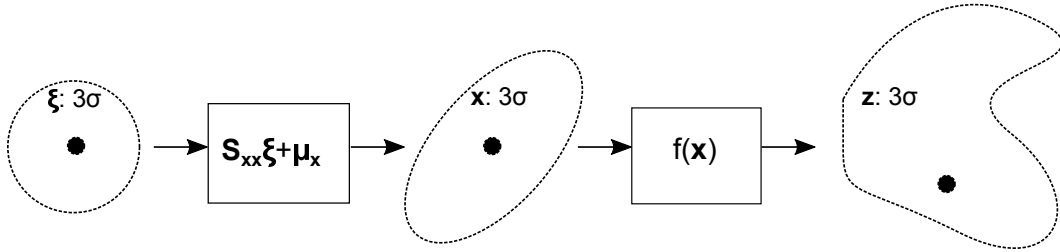


Figure 1.8: The evolution of the  $3\sigma$  bounds and the mean of an initially IID Gaussian PDF through a linear transformation followed by a nonlinear transformation. Dotted lines represent the  $3\sigma$  probability envelope and the black dots represent the mean.

The PDF of  $\mathbf{z}$  is not Gaussian and therefore, more information than the mean and covariance is required to accurately describe the PDF. In case the function  $f(\mathbf{x})$  is not extremely nonlinear and the uncertainty of  $\mathbf{x}$  is small, the true PDF of  $\mathbf{z}$  will be close to a Gaussian distribution:

$$p(\mathbf{z}) \approx p_g(\mathbf{z}; \boldsymbol{\mu}_{\mathbf{z}}, \mathbf{P}_{\mathbf{zz}}) \quad (1.6)$$

For the RSOs,  $f(\mathbf{x})$  is the integration of the equations of motion (EOM) from  $t_0$  to  $t_a$ , which is inherently nonlinear. With increasing nonlinearity in the equations of motion, the validity of the approximation in Eq. (1.6) decreases. The three main contributors to the nonlinearity of the problem are the coordinate set, the size and the fidelity of the perturbations, the orbital regime, and the time of flight ( $t_a - t_0$ ). Expressing the uncertainty in an orbit element set such as Equinoctial Elements (EE) reduces the nonlinearity by absorbing the dominant  $1/r^2$  term [78, 114, 62, 116]. In the same vein, Aristoff et al. [23] have defined a new coordinate set  $J_2$ EE that additionally absorbs some of the perturbation due to the zonal  $J_2$  term. For orbit elements, a new type of distribution called the Gauss von Mises (GvM) is defined by Horwood et al. [66] which is analogous to a multivariate Gaussian on a cylindrical manifold. Wrapping the fast changing angular orbit element on a cylinder better approximates the true distribution for longer times of flight.

An important distinction is now made between *covariance realism*, also known as *covariance consistence* [49], and *uncertainty realism* [67]. Covariance realism is when  $p(\mathbf{z})$  is assumed to be Gaussian:  $p_g(\mathbf{z}; \boldsymbol{\mu}_z, \mathbf{P}_{zz})$ . The goal of uncertainty realism is to instead compute an approximate PDF of  $\mathbf{z}$ ,  $p_*(\mathbf{z})$ , which approximates the true  $p(\mathbf{z})$  better than the Gaussian approximation  $p_g(\mathbf{z}; \boldsymbol{\mu}_z, \mathbf{P}_{zz})$ . The accuracy expressed as an  $L_p$  error is:

$$\left( \int_{\Omega} |p(\mathbf{z}) - p_*(\mathbf{z})|^p d\mathbf{z} \right)^{1/p} < \left( \int_{\Omega} |p(\mathbf{z}) - p_g(\mathbf{z}; \boldsymbol{\mu}_z, \mathbf{P}_{zz})|^p d\mathbf{z} \right)^{1/p} \quad (1.7)$$

where  $1 \leq p < \infty$  and  $\Omega$  is the domain of  $\mathbf{z}$ , i.e.  $\mathbf{z} \in \Omega$ .

Since orbit propagation is the solution of an ODE, the PDF of the state

uncertainty evolves in time according to the Fokker-Plank equation (FPE). The FPE is a PDE that is notoriously difficult to solve [84] and analytical solutions have only been found for linear dynamical systems [54]. Numerically solving the FPE becomes exponentially more difficult with increasing dimension - it suffers from the so-called *curse of dimensionality* for multivariate systems [85]. A method for solving the FPE for a six-dimensional problem in real time using sparse grids has recently been presented [79].

#### 1.4.1 Monte Carlo

Since the FPE is not easily solvable, the standard and the most general approach to approximate the resulting non-Gaussian distribution is via Monte Carlo (MC) simulations. Random initial conditions are sampled from the distribution of the input parameters. The function is evaluated at the initial conditions to create a sample of the output. Information about the PDF of the output, such as the shape and the statistical moments can then be extracted from the sample. With increasing dimension, MC simulation becomes computationally expensive because a large number of function evaluations are required due to the slow convergence rate [42]. A reduction in the required number of MC samples can be achieved by importance sampling in cases when certain input parameters have a large impact on the function output [118]. Two alternative methods for reducing the function evaluations are to use Gaussian Mixture Models and Polynomial Chaos Expansion, which are presented in Chapters 2, and 3, respectively. Function evaluations can also be parallelized on multi-core processors or on Graphics Processing Units [25, 31]

to reduce overall computation time. Chapter 4 shows the MC implementation on a GPU used for uncertainty propagation and computing the collision probability in Phase II.

### 1.4.2 Gaussian Assumption

Assuming the uncertainty in the output parameters follows a Gaussian distribution enables the use of analytical approximations because only the first two statistical moments of the initial PDF have to be propagated through the function. This Gaussian assumption is the basis for many filtering and state estimation applications. The initial covariance can be propagated using a Taylor series approximation of the function [125, 103] or with Fourier-Hermite series [94, 115]. The function can be evaluated at deterministically chosen sigma-point [77, 93, 100, 9], quadrature [71, 34, 28], or cubature [22, 30, 72, 90] based methods. The deterministically chosen points are generated for the IID distributed variable  $\xi$  (Figure 1.8), and can be applied to any Gaussian distribution because of the linear transformation. It should be noted that some of the quadrature and cubature rules can also be used to compute higher order moments of the PDF of  $\mathbf{z}$  using the principle of maximum entropy [8, 9].

### 1.4.3 Gaussian Mixture Models

Even if the mean and covariance are exactly captured by the Gaussian propagation techniques, there is no guarantee that the shape of the PDF is accurate since the isoprobability contours are assumed to be ellipsoids, which may not be the case, as seen in Figure 1.8 on page 16.

A Gaussian Mixture Model (GMM) works under the proposition that any PDF can be approximated in terms of the  $L_1$  norm by using a sum of Gaussian probability distribution functions [19].

$$p(\mathbf{x}) = \sum_{i=1}^N \alpha_i p_g(\mathbf{x}; \boldsymbol{\mu}_i, \mathbf{P}_i) \quad (1.8)$$

where  $N$  is the total number of Gaussian probability distribution functions, and  $\alpha_i$  is a weight, which satisfies  $\forall \alpha_i > 0$  the following constraint:

$$\sum_{i=1}^N \alpha_i = 1 \quad (1.9)$$

GMMs are used to approximate the initial Gaussian PDF by a weighted sum of Gaussian PDFs with a smaller uncertainty. Each element is propagated through the function using any of the methods from Section 1.4.2 to enforce the Gaussian condition on the post-propagation PDF. The weighted sum of the individual propagated Gaussian elements better approximates the final non-Gaussian PDF as the number of elements is increased, while requiring fewer function evaluations than MC. Therefore, Gaussian Mixture Models (GMMs) form a compromise between the analytical approximations of the Gaussian propagation methods and the computationally intensive MC technique. An added benefit of GMMs is the availability of an analytical PDF for the GMM approximation.

Much of the active research in GMMs is in the applied fields of machine learning [88, 29] where a GMM is fitted to a given sample from a non-Gaussian distribution using Expectation Maximization (EM) [47]. The current work, however, deals with the more basic question of how to generalize the number and directions of the splits of the initial Gaussian distribution into a GMM.

The nonlinear orbital equations of motion cause an initial Gaussian state uncertainty distribution of space objects to become non-Gaussian with increasing flight time and perturbations. Capturing the evolution of the state uncertainty is an essential part of operations including orbit determination (OD) and conjunction assessment (CA). Apart from artificial satellites, uncertainty propagation is essential for predicting the threat due to Near Earth Asteroid (NEA) where the uncertainty arises due to observational geometry and modeling [56]. There exists a framework that efficiently, with varying degrees of accuracy, handles uncertainties that are assumed to remain Gaussian [124, 13]. The same framework is readily extended to GMMs since each element remains Gaussian. A Gaussian Sum Filter (GSF) has been used to improve the performance of Gaussian filtering algorithms such as the EKF for nonlinear non-Gaussian problems [19]. GMMs have also been used to propagate state uncertainties through nonlinear ODEs [126, 68, 44, 131]. Psiaki et al. [108] show a resampling strategy for GMMs to efficiently approximate one GMM with another so that the particles in a Particle Filter (PF) are replaced with a GMM. Most recently, GMMs have also been used to improve collision probability computations [133, 45, 134, 53].

#### 1.4.4 Polynomial Chaos Expansion

Polynomial Chaos Expansion (PCE) [136] quantifies non-Gaussian uncertainty by using a linear combination of orthogonal Hermite polynomials (OP) as a surrogate model for the underlying nonlinear function. A surrogate model replaces the actual computationally intensive nonlinear function  $f(\mathbf{x})$

by a function that is computationally less intensive  $f_*(\mathbf{x})$ . The Wiener-Askey scheme [137] generalizes PCE to other initial distributions and is known as generalized PCE (gPCE). It is also possible to compute orthogonal polynomials for arbitrary PDFs that are not part of the Wiener-Askey scheme using arbitrary PCE (aPCE) [101]. After selecting the appropriate OPs, the PCE approximation is determined by the coefficients of the linear combination of these polynomials. Once the coefficients are computed, sampling from the PCE generally has a lower computational cost than MC. PCE has been used in many fields for uncertainty quantification of computationally intensive models [91, 89, 92, 48]. In orbital mechanics, PCE has been previously used for uncertainty propagation [75] and conjunction assessment [74, 76]. Taylor series expansions using State Transition Tensors (STTs) could also be used as surrogate models [103, 53]. However, a PCE with Hermite polynomials has a faster convergence when uncertainty is involved because the polynomials are orthogonal to the Gaussian distribution. A more detailed overview of PCE is provided in Section 3.2.

## 1.5 Collision Probability Computation

After propagating the states of the RSOs from  $t_0$  until  $t_1$ , the probability of collision between  $t_a$  and  $t_b$  from Figure 1.7 on page 12 has to be computed. The collision probability computation of any two RSOs depends on the size, shape, trajectory, and the uncertainty associated with all the aforementioned parameters. In the case of no uncertainty, the collision prediction is a discrete problem with a guaranteed collision or no collision. A common

assumption that makes the problem more tractable and allows for a 3 DOF simulation is that both RSOs are spheres. The reasons are twofold: 1) It is too computationally intensive to do a full 6 DOF simulation and 2) the shape and composition of the RSOs may be unknown. The collision probability is hence calculated as the probability that the two RSOs have a separation less than or equal to the combined radii at any point in time. Therefore, the probability of collision is reduced to a computation of the intersection of the time varying 3D position probability distribution functions (PDFs) of the two RSOs over a specified time window. The introduction to collision probability computation in this section is qualitative in nature, and further detail with extensive equations and references is provided in Chapters 4 and 5.

### 1.5.1 Monte Carlo

The most accurate method for computing the intersection of the PDFs of the two RSOs is a Monte Carlo simulation. However, MC simulations require millions of samples from the PDFs of the primary and secondary RSOs. These samples are propagated over the time window of interest and the collisions are recorded. MC simulations are very general and no additional assumptions are required, but they can be inefficient [16, 113, 31]. MC is applicable to any uncertainty propagation technique in Phase I as long as samples are available for sampling at the beginning of Phase II. The number of samples is reduced, at the cost of statistical rigor, if each sample from the PDF of one RSO is checked against all the samples from the PDF of the other RSO for collisions. The complexity of the dynamics, i.e. the number of modeled perturbation



can be reduced when the PDFs are sampled at, or near, the time of closest approach  $t_{CA}$ . Even two-body dynamics suffice for many encounters [113]. Interpolation, shown in Section 4.2.1, is required to accurately find the close approaches between the samples of the RSOs without reducing the step size of the numerical integration.

### 1.5.2 Gaussian Approximation

Computationally less intensive techniques exist when a few additional assumptions are made. If the RSOs have a Gaussian distribution in Cartesian coordinates during conjunction, and their distributions are independent, the collision probability is calculated with a low computation cost [40] when compared to an MC simulation. The collision probability rate is computed as a quadrature over a sphere. An additional quadrature in time gives the desired probability.

Additionally, if the encounter is brief, there exist many analytical approximations that allow for a fast computation of collision probability [18, 10, 13]. The linear approximation theories do not account for any uncertainty in velocity during the conjunction i.e. the velocities of the RSOs and their covariance matrices are frozen at the time of closest approach. The combined covariance is placed on the secondary RSO. The primary RSO is then assumed to be a sphere with the radius equal to the combined radius of the two RSOs. It should be noted that one of the two RSOs is arbitrarily designated as primary, and the other as secondary. As the sphere passes through the combined position covariance, it traces out a tube. The covariance ellipsoid is a three-

dimensional PDF in the position space and the collision probability is the integral of the PDF over the volume of the tube. The linear encounter equations are the most commonly used in operations and return accurate results when all the assumptions are satisfied.

The linear collision probability computation methods have been extended to include some nonlinear motion [12, 104, 14]. However, these methods are not as general as Coppola's quadrature technique [40]. A thorough discussion with implementation details of the aforementioned two techniques for encounters with Gaussian distributions is presented in Section 5.2.2.

When the PDFs of the RSOs are not Gaussian in Cartesian coordinates during the encounter, a major assumption of the aforementioned methods is no longer valid. However, each of the elements is Gaussian when a sufficient number of GMMs are used for propagating the PDF. Therefore, the quadrature or linear conjunction methods can be used in an all-on-all manner on the GMMs of the involved RSOs [45]. The use of Polynomial Chaos Expansion for computing the collision probability is shown by Jones et. al. [74, 76]. GMMs have also been used with STTs and orbit elements to compute collision probability [53]. Adurthi and Singla [8] use the Conjugate Unscented Transform and the principle of maximum entropy to compute collision probability. Morselli et. al. [97] use a Differential Algebra based Monte Carlo simulation to reduce the computation time during the encounter.

## 1.6 Contributions

The issues present in the uncertainty propagation and collision probability computations have been introduced in the current chapter. The major contributions of the current work in solving some of the issues are summarized in this section:

- The multivariate Gaussian distribution is converted to a Gaussian Mixture Model by applying a univariate splitting library along a direction. Univariate splitting libraries of up to 5 elements exist in the literature. The maximum number of elements is extended in the present work by generating three univariate splitting libraries with up to 39 elements.
- The univariate splitting library can now be applied along an arbitrary direction of the multivariate Gaussian distribution. The splitting directions are no longer limited to the eigen vectors or the columns of the Cholesky decomposition of the covariance matrix.
- A nonlinearity measure is used to find the multiple directions that benefit most from splitting to generate a Multidirectional Gaussian Mixture Model.
- After splitting the initial Gaussian distribution into a GMM, each element is propagated using Polynomial Chaos Expansions. The combination GMM-PC is analogous to the Finite Element Method where the GMM is the mesh refinement ( $h$ -refinement) and varying the order of the PCE is varying the polynomial degree of the basis ( $p$ -refinement).

- A quartic polynomial in the position separation is derived and used in conjunction analysis that uses all the information available nominally from an RK step. The relative position, velocity, and acceleration at the beginning of a time step and only the relative position and velocity at the end of the step. The interpolation ensures that no close approaches between two RSOs are missed.
- The MC collision probability computation is optimized for the GPU such that up to two orders of magnitude speedups are achieved over a serial CPU implementation. The tool is shown to be easy to implement using relatively low fidelity dynamics and is an essential benchmark for comparing other collision probability techniques.
- Multidirectional Gaussian Mixture Models are shown to compute collision probability with fewer function evaluations compared to a Monte Carlo simulation for cases where the traditional Gaussian Mixture Models do not suffice.

## Chapter 2

# Multidirectional Gaussian Mixture Models

Gaussian Mixture Models (GMMs) are used in this chapter for the general uncertainty propagation of initially Gaussian distributions. Multivariate GMMs are typically handled with splits along a single direction where the elements are located, and the variance along that direction is reduced. A regular multidirectional grid is generated over the initial multivariate Gaussian distribution by recursively applying the splitting library along multiple directions. Varying the number of splits in each direction allows for a spectrum of models in the accuracy versus compute time design space, filling the gap between expensive Monte Carlo and fast linearized models. The Multidirectional GMM (MGMM) is demonstrated with four test cases, including an orbit uncertainty propagation case, to illustrate the benefit of properly selecting the splitting directions and splitting along multiple directions. The MGMMs are used to capture the resulting non-Gaussian distribution of the propagated variable through Phase I of Figure 1.7 on page 12.<sup>1</sup> The likelihood of samples from

---

<sup>1</sup>The trust region optimizer used in this work was developed by Ryan P. Russell and the work from this chapter has been submitted as a journal paper:

- V. Vittaldev and R. P. Russell, Multidirectional Gaussian Mixture Models for Non-linear Uncertainty Propagation, Submitted July 2015.

a Monte Carlo simulation being generated by the MGMM is used as a metric for accuracy of the MGMM uncertainty propagation.

## 2.1 Introduction

A Gaussian distribution is a good assumption for the initial uncertainty in parameters for a large number of problems due to the central limit theorem [51]. The outputs of any nonlinear function of the parameters will have an uncertainty distribution that is non-Gaussian. The extent of the non-Gaussian behavior depends on the nonlinearity of the function and the size of the initial uncertainty. Since a Gaussian distribution can completely be characterized by the first two statistical moments, the size of the uncertainty is encapsulated in the covariance. A nonlinear function maps the probability density function (PDF) of the uncertainty of the input parameters onto the uncertainty PDF of the output parameters. The resulting distribution is no longer Gaussian, and if sufficient nonlinearity is present, higher order statistical moments must be considered.

A Gaussian Mixture Models (GMMs) works under the proposition that any PDF can be approximated in terms of the  $L_1$  norm by using a sum of Gaussian probability distribution functions [19]. GMMs are used to discretize the initial Gaussian PDF into weighted Gaussian PDFs with a smaller covariance.

$$p_g(\mathbf{x}) \approx \sum_{i=1}^N \alpha_i p_g(\mathbf{x}; \boldsymbol{\mu}_i, \mathbf{P}_i) \quad (2.1)$$

where  $N$  is the total number of Gaussian probability distribution functions,

and  $\alpha_i$  is a weight, which satisfies  $\forall \alpha_i > 0$  the following constraint:

$$\sum_{i=1}^N \alpha_i = 1 \quad (2.2)$$

Each element is propagated through the function using simplified analytical methods to enforce the Gaussian condition on the post-propagation PDF. The weighted sum of the individual propagated Gaussian elements better approximates the final non-Gaussian PDF as the number of elements is increased, while requiring fewer function evaluations than Monte Carlo.

The weights, means, and covariance matrices from Eq. (2.1) are typically found using univariate splitting libraries, which are explained in Section 2.2. It should be noted that the size of a univariate splitting library available in literature is limited to 5 [44]. For multivariate cases, the univariate libraries are applied along all the  $n$  dimensions for an  $n$ -dimensional problem [68]. Therein lies the motivation of the current chapter: 1) increase the number of splits for the univariate case; 2) extend the GMM concept to apply in multiple directions; and 3) identifying the directions that have the greatest effect on the non-Gaussian behavior of the propagated distribution.

In Section 2.2, a univariate library for splitting the standard normal distribution into a GMM by minimizing the  $L_2$  norm is presented. Solutions for multiple splitting rules are found via nonlinear optimization and archived for up to 39 univariate splits. The new solutions are the largest libraries in the literature to date. The splitting library is then applied along multiple directions of the covariance matrix to extend the theory to multivariate Gaussian distributions in Section 2.3. A second order Stirling's interpolation formula

evaluated at the mean of the original Gaussian distribution is used to rank the candidate directions with respect to the nonlinearity. The number of splits along a given direction is chosen based on the relative degree of nonlinearity. The performance of the MGMMs is shown for multiple numerical test cases, including the uncertainty propagation of an object in Geostationary orbit (GEO), in Section 2.4.

## 2.2 Univariate Splitting Library

Before tackling multivariate scenarios, the univariate case is considered first. To increase the applicability of the univariate GMM, the goal of this section is to develop splitting libraries with larger numbers of splits fit in advance than are currently available in the literature. A tool is created that generates a univariate splitting library based on the standard normal distribution. The library consists of the weights  $\alpha_i$ , means  $\mu_i$ , and the standard deviations  $\sigma_i$  for a desired number of elements. A GMM generated from the standard normal distribution can be shifted and scaled to fit any univariate Gaussian distribution with an arbitrary mean and standard deviation.

### 2.2.1 Generating a Univariate Split

In developing the univariate library, a performance index is necessary to measure the difference between the exact Gaussian distribution and the approximated GMM. The weights, and location and magnitude of the means are optimization variables in a nonlinear programming problem. The performance index ideally should result in a value of zero if and only if the two distribu-



tions are identical. The result should be greater than or equal to zero for any two distributions [61]. There are many metrics that could be used to quantify the dissimilarity between two distributions such as the Kullback-Leibler [83] divergence, the Jensen-Shannon divergence [50], and the  $L_p$  distance [68]. The  $L_p$  distance between 2 distributions  $p_1$  and  $p_2$  is:

$$L_p(p_1, p_2) = \left( \int_{\Omega} |p_1(\mathbf{x}) - p_2(\mathbf{x})|^p d\mathbf{x} \right)^{1/p} \quad (2.3)$$

### 2.2.2 Optimization

Although GMMs approximate PDFs better with respect to the  $L_1$  distance [19], the  $L_2$  distance is chosen as the dissimilarity metric because the solution is analytical and does not require solving a quadrature in the case of a GMM and a Gaussian. The  $L_2$  distance between a GMM PDF  $p$  and a Gaussian distribution  $p_g(\mathbf{x}; \boldsymbol{\mu}_g, \mathbf{P}_g)$  is [44]:

$$L_2(p, p_g) = |4\pi\mathbf{P}_g|^{-1/2} + \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j K(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j, \mathbf{P}_i, \mathbf{P}_j) - 2 \sum_{i=1}^N \alpha_i K(\boldsymbol{\mu}_i, \boldsymbol{\mu}_g, \mathbf{P}_i, \mathbf{P}_g) \quad (2.4)$$

where the operation defined by  $K$  is

$$K(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \mathbf{P}_1, \mathbf{P}_2) = |2\pi(\mathbf{P}_1 + \mathbf{P}_2)|^{-1/2} \times \exp\left(-\frac{1}{2}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T (\mathbf{P}_1 + \mathbf{P}_2)^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)\right) \quad (2.5)$$

The weights are computed by minimizing the  $L_2$  distance between  $p$  and  $p_g$  [68, 44, 70]. Minimizing the  $L_2$  distance becomes a constrained nonlinear optimization problem, with the cost function from Equation 2.4. In the existing literature, univariate libraries have been precomputed for  $N = 4$  [70]

and  $N = 3, 4, 5$  [44]. To compute libraries with a higher  $N$ , a trust-region optimization [39] algorithm is currently implemented, and solutions are found with  $N$  up to 39. There are  $N$   $\sigma_i$ ,  $N$   $\alpha_i$ , and  $N$   $\mu_i$ , totaling  $3N$  variables to ultimately choose, where  $N$  is the desired number of splits.

However, simplifications are made to reduce the number of free parameters in the optimization problem from  $3N$  to  $N - 1$ . All the components of the GMM are given the same standard deviation by specifying a rule for  $\sigma$  as a function of  $N$ . With the assumption of homoscedasticity, the cost function for minimization is a simplified version of the  $L_2$  distance from Equation (2.4):

$$\begin{aligned}
 J = & \frac{1}{2\sqrt{\pi}} + \sum_{i=1}^N \sum_{j=1}^N \frac{\alpha_i \alpha_j}{2\sqrt{\pi}\sigma^2} \exp\left(\frac{-(\mu_i - \mu_j)^2}{4\sigma^2}\right) \\
 & - 2 \sum_{i=1}^N \frac{\alpha_i}{\sqrt{2\pi}(\sigma^2 + 1)} \exp\left(\frac{-\mu_i^2}{2(\sigma^2 + 1)}\right)
 \end{aligned} \tag{2.6}$$

After choosing a rule for  $\sigma$ , the remaining  $2N$  free parameters are the weights  $\alpha_i$  and the means  $\mu_i$ . Assuming that the location and the weights of the elements are symmetric further reduces the number of free parameters to  $N - 1$ . Details on the constraints and the remaining free parameters for odd  $N$  are presented here. The setup of the optimization for even  $N$  is similar but only odd  $N$  are computed to ensure the location of one element coincides with the actual

mean of the normal distribution. The  $N + 1$  equality constraints are:

$$0 = \mu_0 \quad (2.7a)$$

$$0 = \mu_i - \mu_{-i} \quad i = 1, \dots, (N - 1) / 2 \quad (2.7b)$$

$$0 = \alpha_i - \alpha_{-i} \quad i = 1, \dots, (N - 1) / 2 \quad (2.7c)$$

$$0 = 1 - \alpha_0 - 2 \sum_{i=1}^{(N-1)/2} \alpha_i \quad (2.7d)$$

Inequality constraints are imposed to enforce a monotonic increase in the location of the elements and a monotonic decrease in the weights away from the center.

$$\mu_{i-1} - \mu_i < 0 \quad i = 1, \dots, (N - 1) / 2 \quad (2.8a)$$

$$\alpha_{i-1} - \alpha_i < 0 \quad i = 1, \dots, (N - 1) / 2 \quad (2.8b)$$

Figure 2.1 illustrates the effect of the optimization equality and inequality constraints on the PDFs of the individual elements for  $N = 7$ .

A trust-region optimizer is used due to its robustness in solving non-linear optimization problems [32]. Like a line search, the trust-region method enforces that each new iteration reduces the objective function. However, a maximum step distance (the trust-region) is first chosen, followed by the direction of the step. The trust-region is centered on the present iterate and the complicated nonlinear function is replaced by a simpler, quadratic approximation. Choosing the step direction translates to finding the minimum of the quadratic approximation. If the Hessian of the objective function is positive definite and the trust-region is large enough, then the step is identical to the

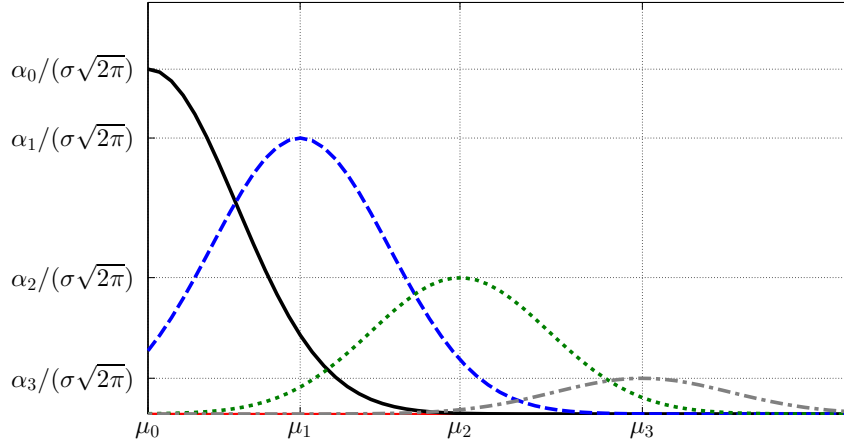


Figure 2.1: PDFs of the individual elements subject to the equality and inequality constraints for  $N = 7$

Newton-Rhapson update. The details of computing the update step and the size of the regularly updated trust-region are beyond the scope of this dissertation. Trust-region optimization is an active area of research and many variations can be found in literature [96].

The trust-region optimization algorithm requires the first and second derivatives of the objective function and the constraints with respect to the state variables. Analytical expressions for the Jacobians and Hessians are computed using the symbolic manipulator, Maple. The optimization routine is coded in Fortran due to the computational speed and the ability to use quadruple precision arithmetic, which is required for convergence when the number of splits,  $N$  is high. During the optimization, the GMM weights and means from the converged solution of the  $N - 2$  case are used as an initial guess for the current  $N$  case. Furthermore, a heuristic predictor step is employed to modestly increase the spread on the means, and lower the weight values.

The problem is highly nonlinear and convergence is increasingly difficult as  $N$  increases. A flowchart of the optimization procedure is shown in Figure 2.2.

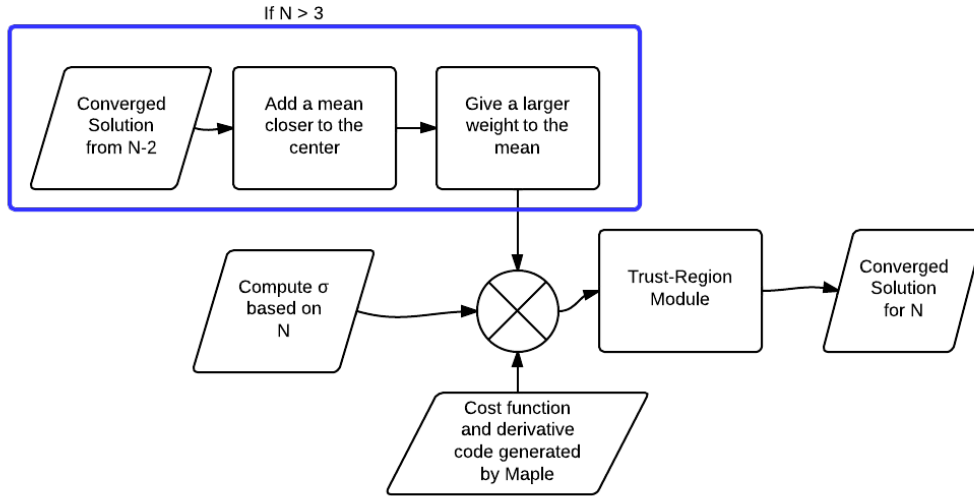


Figure 2.2: The optimization procedure for finding a univariate splitting library

The computation of the trust-region optimization, takes approximately 3 minutes for  $N$  up to 39 on a 3.07 GHz Intel Xeon CPU. The trust-region and function routines are compiled using Intel Visual Fortran Composer XE 2011 and optimization settings `-O3`. Solutions for three different  $\sigma$  rules are found and archived:

1.  $\sigma^2 = (1/N)$  for odd-valued  $N \leq 39$
2.  $\sigma^2 = (1/N)^{3/4}$  for odd-valued  $N \leq 25$
3.  $\sigma^2 = (1/N)^{1/2}$  for odd-valued  $N \leq 15$

The run time for the trust-region optimization is shown in Figure 2.3.

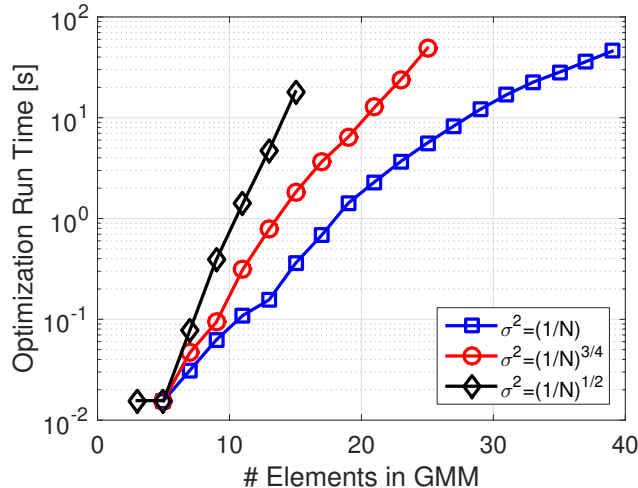


Figure 2.3: The run time required by the trust-region optimization to compute the locations of the means and the corresponding weights for the univariate splitting libraries

### 2.2.3 Resulting Univariate Libraries

An example of a 7 component univariate library is presented in Table 2.1. Some of the splits using the various rules with an increasing number of components are visualized in Figure 2.4. The  $L_1$  distance between the optimized solutions and the standard normal distribution, as a function of the number of the elements are found in Figure 2.5. Although the optimization is carried out using the  $L_2$  distance, the  $L_1$  distance is shown because GMMs can approximate any distribution in the  $L_1$  distance [19]. Libraries with up to 39 elements have been created using an intensive optimization process that solves the highly nonlinear cost function with many local minima. The issue with

increasing the number of elements in the library is mainly numeric sensitivity because the problem becomes too large and sensitive. It could be tackled with new formulations for the optimization problem that use better heuristics for initial guesses and decomposes the problem into smaller, less sensitive, subproblems. A brute force approach is to solve the optimization using higher precision than quadruple. As the problem stands, however, the upper limit on the number of elements has been achieved using the current optimization strategy and quadruple precision.

It is difficult to view the splits when  $N$  is large because the size of individual PDFs in the tails becomes too small. Therefore, a good way to visualize the split is to plot the maximum and minimum weight, and the maximum value of the mean as in Figure 2.5 . Since only an odd number of elements are used, the weights are symmetric about 0 and the means are antisymmetric. Also, the weights monotonically decrease from the center to the tails, and the absolute values of the means increase. The maximum mean value shows how far the furthest split is from the center, indicating how spread out the elements are. The maximum and minimum weights show the difference between the weight of the central element compared to the weight of the element at the tail end, which indicates the importance given to the central elements compared to the ends. For the various rules, Rule 1 with  $\sigma^2 = (1/N)$  results in the smallest  $\sigma$  and Rule 3 with  $\sigma^2 = (1/N)^{1/2}$  results in the largest  $\sigma$  for a given  $N$ . Figure 2.5 shows that Rule 1 has the lowest weight of the different rules for the central mean compared to the other rules. The element with the lowest weight is also closer to the center than for the other rules. For a given  $N$ , as the exponent

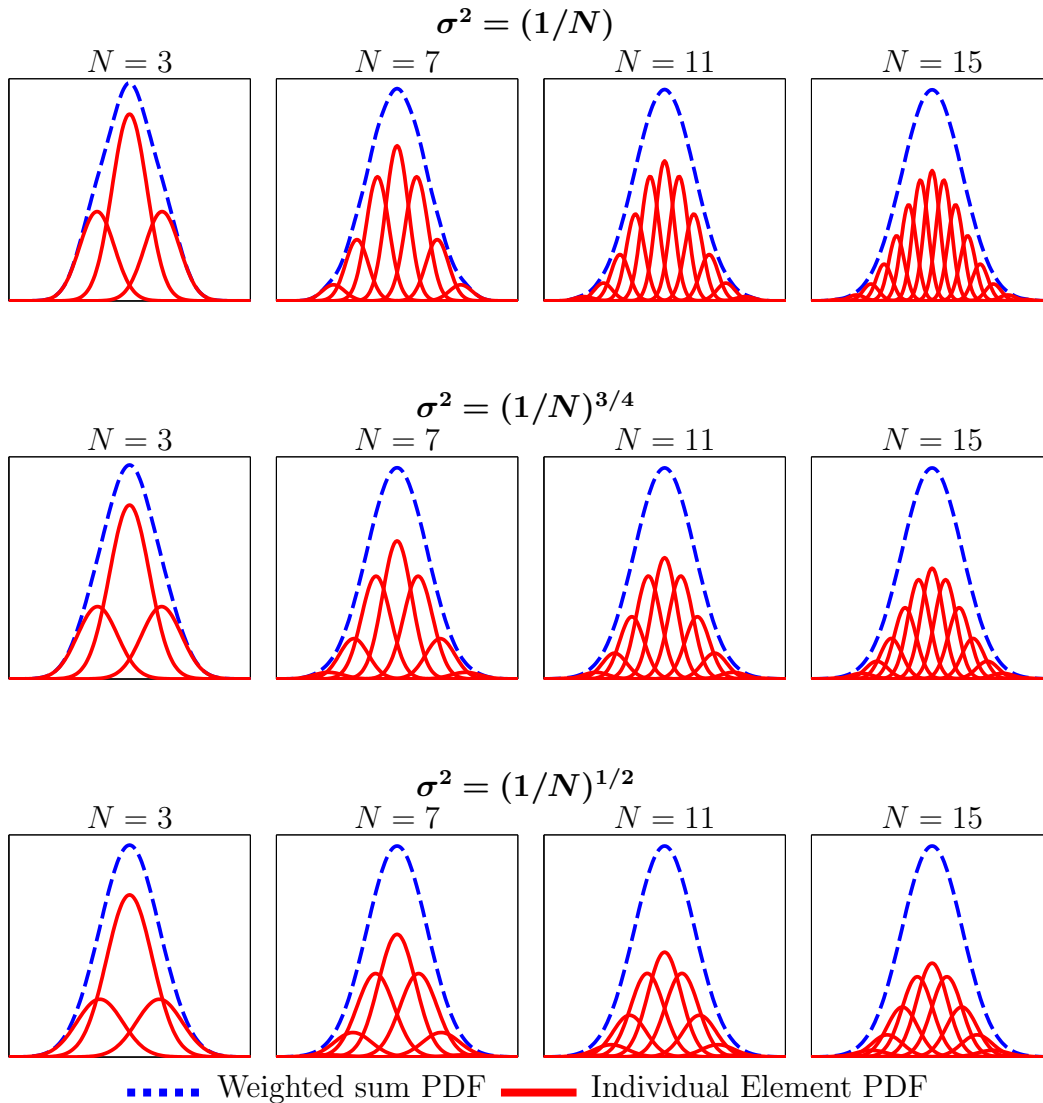


Figure 2.4: The resulting univariate GMMs for the three different splitting rules and for various number of elements

on  $(1/N)$  decreases for the various rules, the  $L_1$  distance decreases. However, the distribution of the weights is sharper since the standard deviation of the computed weights for a given  $N$  increases.

The benefit of generating a complete high dimensional split is seen



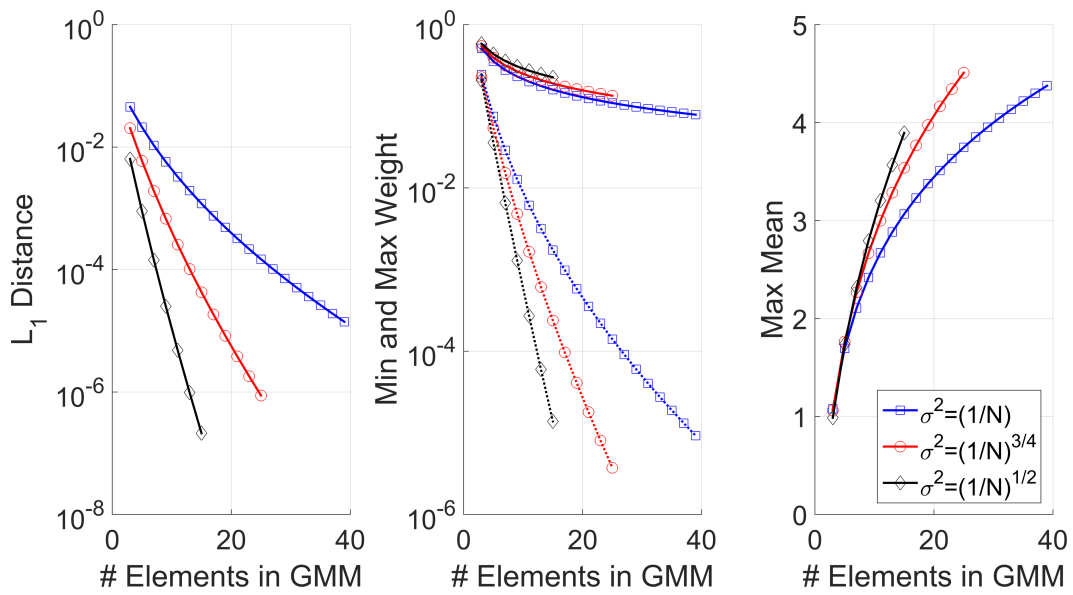


Figure 2.5:  $L_1$  distance, the maximum and minimum weights, and the maximum mean as a function of the number of elements used in the GMM for various rules for  $\sigma$

in Figure 2.6, where a complete 9 element split is compared to a 9 element GMM created by recursively splitting all the elements of a 3 element split [44]. The standard normal distribution is better approximated and the weights and means are distributed more evenly.

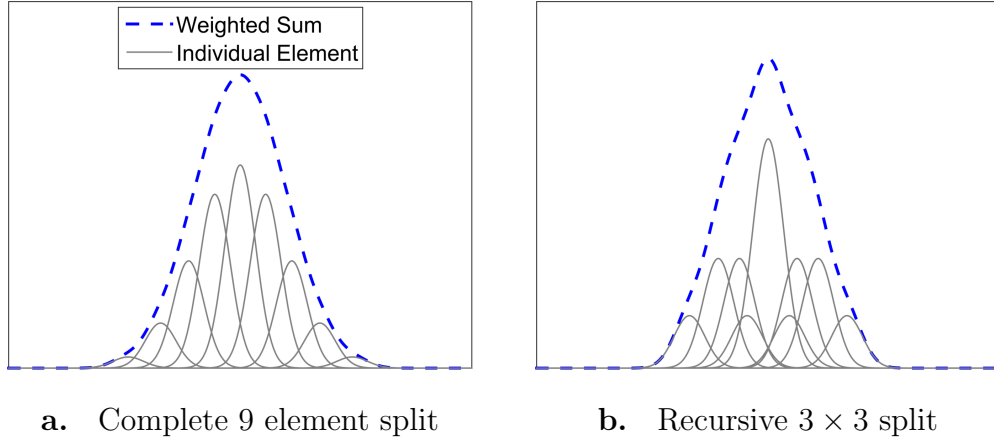


Figure 2.6: PDFs of the univariate 9 element GMM of the standard normal distribution using a complete and a recursive splitting technique for  $\sigma^2 = (1/N)$

Table 2.1: The 7-component univariate splitting library with  $\sigma^2 = (\frac{1}{N})$

Component	$w$	$\mu$
1	0.028799777829539	-2.107361692265483
2	0.109875486136781	-1.329872113204359
3	0.222379075167735	-0.648762460764688
4	0.277891321731891	0
5	0.222379075167735	0.648762460764688
6	0.109875486136781	1.329872113204359
7	0.028799777829539	2.107361692265483

## 2.3 Multivariate GMMs

Uncertainty analysis typically requires a multivariate state. Therefore, it is necessary to split an initial multivariate Gaussian distribution. A univariate library can be applied along a specified direction in order to split a multivariate distribution. The direction choice is along a column of the square-root matrix, which is typically found using Cholesky or spectral decomposition [70]. The following subsection outlines the newly derived method for splitting along an arbitrary direction.

### 2.3.1 Splitting along an Arbitrary Direction

Spectral and Cholesky decomposition are two of the most common methods of computing the square-root  $\mathbf{S}$  of the covariance matrix  $\mathbf{P} = \mathbf{S}\mathbf{S}^T$ .  $\mathbf{S}$  is a coordinate transformation from a new reference frame where the individual variables of the initial multivariate random variable are IID described by the standard normal distribution. The initial reference frame of the multivariate state and the square-root frame where the variables are IID are represented by  $\mathcal{F}_I$  and  $\mathcal{F}_S$ , respectively. The transformation matrix from  $\mathcal{F}_S$  to  $\mathcal{F}_I$  is:

$$R_S^I = \mathbf{S} \tag{2.9}$$

Let  $\mathbf{a}$  be the direction along which the univariate splitting library is applied in  $\mathcal{F}_I$ . When  $\mathbf{a}$  is parallel to the  $k^{\text{th}}$  column of  $\mathbf{S}$  ( $\mathbf{a} \parallel \mathbf{S}(:, k)$ ), and given a univariate splitting library mean  $\mu_i$  and standard deviation  $\sigma_i$ , the

multivariate mean and covariance are [70]:

$$\boldsymbol{\mu}_i = \boldsymbol{\mu} + \mu_i \mathbf{S}(:, k) \quad (2.10a)$$

$$\mathbf{S}_k = \mathbf{S} \quad \mathbf{S}_k(:, k) = \sigma_i \mathbf{S}(:, k) \quad (2.10b)$$

$$\mathbf{P}_i = \mathbf{S}_k \mathbf{S}_k^T \quad (2.10c)$$

where  $\mathbf{S}_k$  is a copy of  $\mathbf{S}$ , but with the  $k^{\text{th}}$  column multiplied by  $\sigma_i$ .

In case  $\mathbf{a}$  is not parallel to any of the columns of the Cholesky or spectrally decomposed square-root matrix, a square-root matrix is constructed where  $\mathbf{S}_\star(:, 1) \parallel \mathbf{a}$  [24]. A new reference frame  $\mathcal{F}_A$  is defined such that the difference between  $\mathcal{F}_A$  and  $\mathcal{F}_S$  is a pure rotation and the first axis of  $\mathcal{F}_A$  is parallel to  $\mathbf{a}$ . Therefore, the variables in  $\mathcal{F}_A$  are also IID. Finally, the new square-root matrix and the mean and covariance matrix are:

$$\mathbf{S}_\star = R_S^I R_A^S \quad (2.11a)$$

$$\boldsymbol{\mu}_i = \boldsymbol{\mu} + \mu_i \mathbf{S}_\star(:, 1) \quad (2.11b)$$

$$\mathbf{P}_i = \mathbf{S}_\star \mathbf{S}_\star^T \quad (2.11c)$$

The rotation matrix  $R_S^A$  from  $\mathcal{F}_S$  to  $\mathcal{F}_A$  is found using Gram-Schmidt orthogonalization. The three reference frames with the confidence bounds for the covariance matrix and reduced variance along  $\mathbf{a}$  are shown in Figure 2.7 for a bivariate case.

A newly derived alternative method of computing the new covariance matrix is now presented where Gram-Schmidt orthogonalization is not required. The mean and covariance of the GMM elements are simple analytical

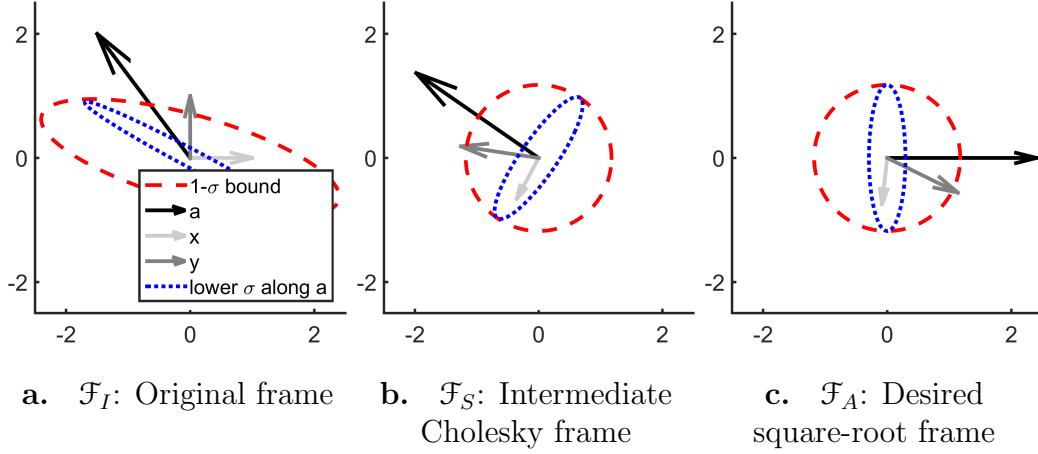


Figure 2.7: The unit axes of  $\mathcal{F}_I$ ,  $x$  and  $y$ , and the desired splitting direction,  $\mathbf{a}$ , in the three relevant reference frames for a 2-dimensional case

equations. In  $\mathcal{F}_S$ , the state is IID. Applying a univariate split along  $\mathbf{a}$  in  $\mathcal{F}_I$  is analogous to changing the standard deviation of the IID. distribution in the  $\hat{\mathbf{a}}^*$  direction from 1 to  $\sigma$  in Figure 2.7b. The unit vector  $\hat{\mathbf{a}}^*$  is simply the direction of  $\mathbf{a}$  expressed in  $\mathcal{F}_S$ :

$$\hat{\mathbf{a}}^* = \frac{\mathbf{S}^{-1}\mathbf{a}}{\|\mathbf{S}^{-1}\mathbf{a}\|_2} \quad (2.12)$$

Reducing the standard deviation along  $\hat{\mathbf{a}}^*$  is a linear transformation:

$$\mathbf{y} = \mathbf{x} - (\mathbf{x}^T \hat{\mathbf{a}}^*) \hat{\mathbf{a}}^* + \sigma (\mathbf{x}^T \hat{\mathbf{a}}^*) \hat{\mathbf{a}}^* = \Phi \mathbf{x} = (\mathbf{I} + (\sigma - 1) \hat{\mathbf{a}}^* \hat{\mathbf{a}}^{*T}) \mathbf{x} \quad (2.13)$$

where  $\mathbf{y}$  is the variable with a standard deviation of  $\sigma$  in the  $\hat{\mathbf{a}}^*$  direction and  $\mathbf{x}$  is the IID state in  $\mathcal{F}_S$  with the covariance matrix  $\mathbf{P}_{\mathbf{xx}} = \mathbf{I}$ . The covariance matrix of  $\mathbf{y}$  expressed in  $\mathcal{F}_S$  is then:

$$\mathbf{P}_{\mathbf{yy}} = \Phi \mathbf{P}_{\mathbf{xx}} \Phi^T = (\mathbf{I} + (\sigma^2 - 1) \hat{\mathbf{a}}^* \hat{\mathbf{a}}^{*T}) \quad (2.14)$$

$\mathbf{P}_{\mathbf{yy}}$  expressed in  $\mathcal{F}_I$  is the covariance matrix of the  $i^{\text{th}}$  element of the multivariate GMM due to applying the univariate splitting library mean  $\mu_i$  and

standard deviation  $\sigma_i$  along the splitting direction  $\mathbf{a}$ . The multivariate mean and covariance matrix are:

$$\boldsymbol{\mu}_i = \boldsymbol{\mu} + \mu_i \mathbf{S} \hat{\mathbf{a}}^* \quad (2.15a)$$

$$\mathbf{P}_i = \mathbf{S} (\mathbf{I} + (\sigma_i^2 - 1) \hat{\mathbf{a}}^* \hat{\mathbf{a}}^{*T}) \mathbf{S}^T \quad (2.15b)$$

where  $\hat{\mathbf{a}}^*$  is computed using Equation (2.12).

### 2.3.2 Propagating a Multivariate Gaussian Distribution

Many methods exist for propagating a Gaussian distribution through a nonlinear function if the resulting distribution is assumed to remain Gaussian. One of the most common methods for propagating the state through a nonlinear function is to use a Taylor series approximation. The first-order method linearizes the nonlinear function and uses the Jacobian to propagate the covariance matrix. The mean of the first-order method is simply the initial mean propagated through the nonlinear function. This first-order Taylor series approximation is the method of mean and covariance propagation for the Extended Kalman Filter (EKF). If the Hessian of the nonlinear function is also used, a second order approximation of the resulting uncertainty is achieved. The second-order propagation method includes some of the initial uncertainty information in computing the mean of the resulting state. The second-order method is the basis for the Second Order Kalman Filter (SOKF) [26] and state transition tensors (STT) may be used for a higher-order expansion [103]. Analytical expressions for the Jacobians, and even more so for the Hessians, may be difficult to derive and compute for complicated functions. In case the

derivatives are not available analytically, they can be also be approximated using numerical differentiation, or with complex [121] and multicomplex [86] step differentiation techniques.

Another set of methods for propagating Gaussian distributions are called sigma point methods. Sigma point methods selectively pick points to propagate through the nonlinear function. These points have associated weights and are chosen deterministically to capture the mean and covariance with an accuracy equivalent to a second-order Taylor series expansion. The post-propagation mean and covariance are computed from the propagated points. Two common methods of computing the sigma points are the Unscented Transform (UT) and second-order divided differences (DD2). For this work, DD2 is used due to the presence of fewer tuning parameters and the default square-root structure of the standard implementation. A square-root implementation only propagates the square-root of the covariance matrix. When the covariance is computed from the square-root, it is guaranteed to not lose positive definiteness. Therefore, the propagation is more numerically stable.

Equations for the DD2 propagation of the mean and the covariance matrix of a nonlinear function  $\mathbf{y} = \mathbf{f}(\mathbf{x})$  with the multivariate state  $\mathbf{x}$  of dimension  $n$  are summarized here. The covariance matrix of the initial uncertainty in  $\mathbf{x}$ ,  $\mathbf{P}_x$ , is first decomposed into a lower triangle square-root matrix,  $\mathbf{S}_x$  where

$$\mathbf{P}_x = \mathbf{S}_x \mathbf{S}_x^T \tag{2.16}$$

The value of  $\mathbf{y}$  is found by using the columns of the square-root matrix

$\mathbf{S}_x(:, i)$

$$\mathbf{y} = \frac{h^2 - n}{h^2} \mathbf{f}(\mathbf{x}) + \frac{1}{2h^2} \sum_{i=1}^n [\mathbf{f}(\mathbf{x} + h\mathbf{S}_x(:, i)) - \mathbf{f}(\mathbf{x} - h\mathbf{S}_x(:, i))] \quad (2.17)$$

where the optimal value for  $h$  is  $\sqrt{3}$  for a Gaussian distribution. The following first and second order square-root matrices are calculated as:

$$\mathbf{S}_{\mathbf{y}\mathbf{x}}^{(1)} = \frac{1}{2h} [\mathbf{f}(\mathbf{x} + h\mathbf{S}_x(:, i)) - \mathbf{f}(\mathbf{x} - h\mathbf{S}_x(:, i))] \quad (2.18)$$

$$\mathbf{S}_{\mathbf{y}\mathbf{x}}^{(2)} = \frac{\sqrt{h^2 - 1}}{2h^2} [\mathbf{f}(\mathbf{x} + h\mathbf{S}_x(:, i)) - \mathbf{f}(\mathbf{x} - h\mathbf{S}_x(:, i)) - 2\mathbf{f}(\mathbf{x})] \quad (2.19)$$

The square-root of the covariance matrix for the uncertainty in  $\mathbf{y}$  is found by computing the Householder triangularization of the following rectangular matrix:

$$\mathbf{S}_y = \text{Householder} \left[ \mathbf{S}_{\mathbf{y}\mathbf{x}}^{(1)}, \mathbf{S}_{\mathbf{y}\mathbf{x}}^{(2)} \right] \quad (2.20)$$

The covariance matrix of  $\mathbf{y}$  is finally computed

$$\mathbf{P}_y = \mathbf{S}_y \mathbf{S}_y^T \quad (2.21)$$

Since  $\mathbf{P}_y$  is computed from its square-root, it is guaranteed to be positive definite. A total of  $2n + 1$  function evaluations are required to compute the mean and the covariance matrix of the propagated function. Variations of sigma point methods exist that allow for a square-root implementation of the UT, or to use UT with only  $n + 2$  points. A comparison of these methods is not considered here since any of the existing techniques for approximately propagating a Gaussian distribution through a nonlinear function, including first-order linearization, can be used with GMMs.



### 2.3.3 Importance of Splitting Direction

It is possible to apply the univariate splitting library along any direction to generate a multivariate GMM. However, the utility of the splits is highly sensitive to the choice of the direction. As an example, the sensitivity of choosing a splitting direction is investigated on a nonlinear transformation of a bivariate Gaussian distribution from [59]:

$$y_1 = ax_1 \tag{2.22a}$$

$$y_2 = x_2/a - b(a^2x_1^2 + a^2) \tag{2.22b}$$

In Eq. (2.22),  $a$  and  $b$  are parameters that control the nonlinearity of the target. The mean and initial covariance for  $(x_1, x_2)$  are chosen to be:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \mathbf{P}_x = \begin{bmatrix} 1 & 0.3 \\ 0.3 & 1 \end{bmatrix} \tag{2.23}$$

To limit the number of direction choices, spectral decomposition is used to create the splits along the two eigenvectors of the covariance matrix found in Equation (2.23). Samples generated from the GMMs with splitting along the minor and major axes of the covariance matrix are seen in the point clouds found in Figure 2.8a and Figure 2.8c. The distributions are plotted as ellipses, as normal 2-dimensional covariance matrices are usually visualized, in Figure 2.8b. To visualize a GMM as a set of ellipses, the covariance matrix of each ellipse element is scaled by its associated weight. Reducing the size of each element's covariance matrix is only a visualization technique. It is emphasized that mathematically, the weight is the probability that a random state is generated by that particular mixture element. Multiplying the weight,

$\alpha_i$ , with the covariance matrix,  $\mathbf{P}_i$  has the effect of reducing the standard deviations by a factor of  $\sqrt{\alpha_i}$  and changing the correlation, which results in a different probability distribution.

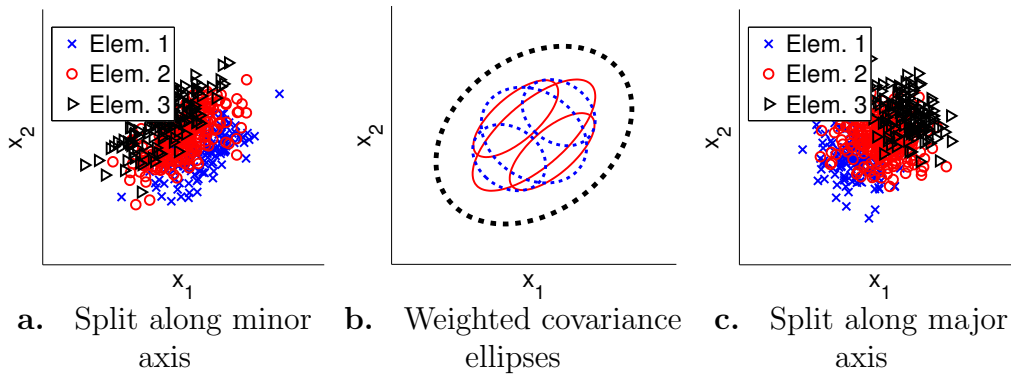


Figure 2.8: Sample point cloud from the covariance ellipses of the multivariate GMM

The resulting distributions from the function in Eq. (2.22) are seen in Figure 2.9. There is a discrepancy between the performance of splitting along the different spectral directions. By visual inspection of the covariance ellipses of the individual GMM elements, better accuracy is seen for this case, by splitting along the major-axis of the covariance ellipse.

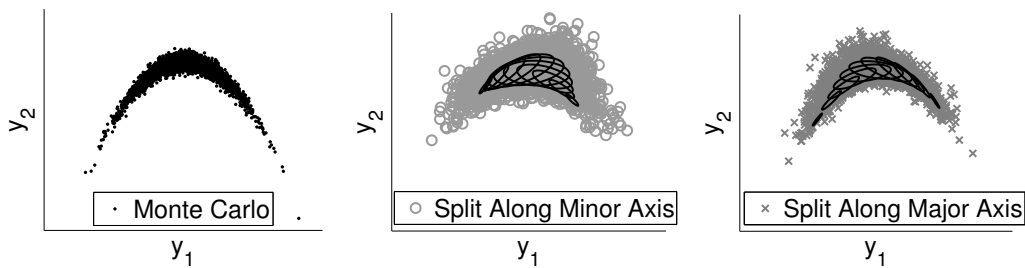


Figure 2.9: Point clouds of the resulting non-Gaussian distribution using MC and an 11 element GMM

### 2.3.4 Choosing the Splitting Direction

A Gaussian distribution propagated through a linear function is completely characterized by another Gaussian distribution. It is the nonlinearity in a function that causes the propagated Gaussian distribution to become non-Gaussian. Using the Hessian or sigma points to propagate the covariance matrix captures some of the higher order terms in the Taylor series expansion. The first two moments of the propagated distribution are captured; however, the effect on the higher order moments due to the nonlinearity in the function is lost because the shape of the PDF is ignored.

The Taylor series approximation improves as the interval  $(\mathbf{x} - \mathbf{x}_0)$  is decreased. The uncertainty of the state is a measure of the spread, so applying a GMM split decreases the size of the covariance matrix along the split direction and thus, each element covers a smaller domain  $(\mathbf{x} - \mathbf{x}_0)$ . However, the splits only decrease the domain along a single spectral direction. The choice of which direction should consider the degree of nonlinearity each direction exhibits. However, the eigenvalues of the covariance matrix should also play a role in the direction choice, as they are a measure of the amount of uncertainty along the corresponding spectral directions. The eigenvectors form a new reference frame where the individual variables are uncorrelated and have a variance equal to the eigenvalue. Some of the nonlinearity at the standard deviation locations can be quantified by using the second-order divided difference form of Stirling's interpolation formula [52]. Stirling's interpolation formula is used instead of the Taylor series because the Taylor series approximation is less accurate further from the expansion location [100], i.e. the

mean of the original Gaussian distribution. The second-order approximation of a univariate function  $f(x)$  about  $\bar{x}$  is:

$$f(x) \approx f(\bar{x}) + \frac{f(\bar{x} + h) - f(\bar{x} - h)}{2h}(x - \bar{x}) + \frac{f(\bar{x} + h) + f(\bar{x} - h) - 2f(\bar{x})}{2h^2}(x - \bar{x})^2 \quad (2.24)$$

where  $h$  is the step size used for the interpolation. The nonlinearity along any direction  $\mathbf{a}$  of the multivariate function is quantified using the second term:

$$\phi = \frac{f(\bar{\mathbf{x}} + \tilde{h}\sigma_{\|\hat{\mathbf{a}}}\hat{\mathbf{a}}) + f(\bar{\mathbf{x}} - \tilde{h}\sigma_{\|\hat{\mathbf{a}}}\hat{\mathbf{a}}) - 2f(\bar{\mathbf{x}})}{2\tilde{h}^2} \quad (2.25)$$

where  $\tilde{h} = \sqrt{3}$  is recommended so that the function evaluations are the sigma points for the DD2. The standard deviation of the cut along the arbitrary direction  $\mathbf{a}$  passing through the mean is:

$$\sigma_{\|\hat{\mathbf{a}}} = \|\mathbf{S}^{-1}\hat{\mathbf{a}}\|_2^{-1} \quad (2.26)$$

where  $\mathbf{S}$  is the square-root factor (Cholesky or Spectral decomposition) of the covariance matrix  $\mathbf{P}$ . It should be noted that  $\phi$  has the same units as  $f(\mathbf{x})$ . If  $H$  is the Hessian, i.e. the second order derivative of  $f(\mathbf{x})$  with respect to  $\mathbf{x}$ , the nonlinearity measure for a univariate problem is:

$$\phi \approx \left| \frac{1}{2}H\sigma^2 \right| \quad (2.27)$$

After evaluating  $\phi$  for all the spectral directions, the maximum of some measure, such as the  $p$ -norm, is used to rank the spectral directions in order of nonlinearity. The divided difference can be taken of any other function

of interest of  $\mathbf{y} = f(\mathbf{x})$ . The optimal direction of splitting is the spectral direction which, considering the uncertainty in that direction, undergoes the most nonlinearity for a desired objective. Higher order divided differences of the function can also be used, at a higher computation cost, to include more information about the nonlinearity of the function and its sensitivity to the spectral directions. If the difference between two  $\phi_i$  is not large, both directions undergo similar amounts of nonlinearity up to the second order.

There exists another measure of nonlinearity that has been developed by Junkins et. al. [78]:

$$\varphi = \frac{\|J(\bar{\mathbf{x}} + 3\sigma_{\|\hat{\mathbf{a}}}\hat{\mathbf{a}}) - J(\bar{\mathbf{x}})\|_2}{\|J(\bar{\mathbf{x}})\|_2} \quad (2.28)$$

where

$$J(\bar{\mathbf{x}}) = \left. \frac{\partial f}{\partial \mathbf{x}} \right|_{\bar{\mathbf{x}}} \quad (2.29)$$

The nonlinearity measure in Eq. (2.28) measures the change in the first derivative of the function at the  $3\sigma$  bounds of the various directions. The nonlinearity measure derived by Junkins et. al. is nondimensionalized and for a univariate function approximates:

$$\varphi \approx \left| \frac{1}{2}kH\sigma \right| \quad (2.30)$$

where  $k$  is a scaling quantity. The nonlinearity measure from Junkins et. al. [78] Eq. (2.28) and  $\phi$  from Eq. (2.25) are similar, however, the nonlinearity measure derived and used in this chapter places a larger emphasis on the magnitude of the variance as seen in Equation (2.27). An added benefit is that the Jacobian is not explicitly required in Equation (2.25).

The nonlinearity measure is illustrated on the test case from Section 2.3.3. The accuracy of the GMM approximated non-Gaussian distribution is quantified using the log-likelihood ( $LL$ ) of the MC sample:

$$LL = \sum_{i=1}^M \log \left( \sum_{j=1}^N \alpha_j p_g(\mathbf{x}_i; \boldsymbol{\mu}_j, \mathbf{P}_j) \right) \quad (2.31)$$

where  $M$  is the total number of MC runs and  $\mathbf{x}_i$  is the  $i^{\text{th}}$  MC sample point. The covariance matrix from Eq. (2.23) is first transformed into a new coordinate frame by using a two-dimensional rotation matrix with the angle  $\theta$ . The initial state and uncertainty are then propagated through Eq. (2.22). The splitting direction are, however, the spectral directions of the original non-rotated covariance matrix. Figure 2.10b shows the nonlinearity measure  $|\phi_i|_2$ , as a function of the changing  $\theta$ . Comparing Figure 2.10a and Figure 2.10b shows the importance of splitting along the spectral direction with the higher nonlinearity.

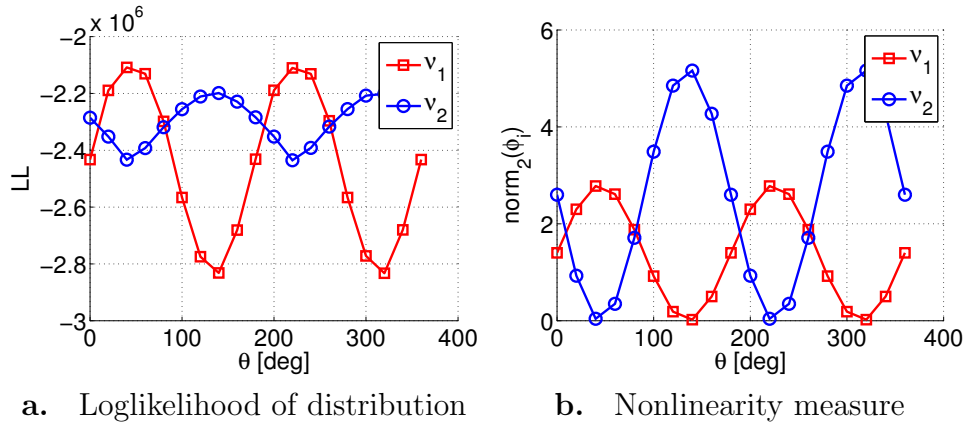


Figure 2.10: The loglikelihood of the MC simulation and the nonlinearity measure of the function with  $5 \times 10^5$  sample points and 5 splits along  $\boldsymbol{\nu}_1$  or  $\boldsymbol{\nu}_2$  and increasing rotation angle  $\theta$

### 2.3.5 Sensitivity to Univariate Splitting Library

Three different splitting libraries are generated in Section 2.2. Since the three libraries produce different means and weights, the resulting accuracy after the propagation through the nonlinear function differs with the rule.

The test case from Eq. (2.22) in Section 2.3.3 is run with GMMs generated using the different univariate libraries. The accuracy with which the GMMs capture the non-Gaussian distribution is found in Figure 2.11. The rule with  $\sigma^2 = 1/N$  performs the best by consistently having a higher  $LL$  than the other splitting rules for the cases tested here. A similar behavior is found for the test cases presented in Section 2.4. The rule with  $\sigma^2 = 1/N$  gives each element more authority due to a more uniform weight distribution across all elements, and this leads to stronger performance for the highly nonlinear examples considered. The higher accuracy of the library with the smallest standard deviations per element is likely to be problem dependent, however, the rule with  $\sigma^2 = 1/N$  is used exclusively in the following sections and is recommended for future use. Figure 2.11 shows the expected trade-off between the accuracy and the number of required elements. Future work includes exploring other sigma rules not considered currently.

### 2.3.6 Multidirectional Gaussian Mixture Models

Splitting along only one direction cannot fully describe the resulting non-Gaussianity in the propagated distribution after a nonlinear transformation. Therefore, in cases where one direction is not sufficient, the natural extension is to apply the univariate split along multiple spectral directions. The

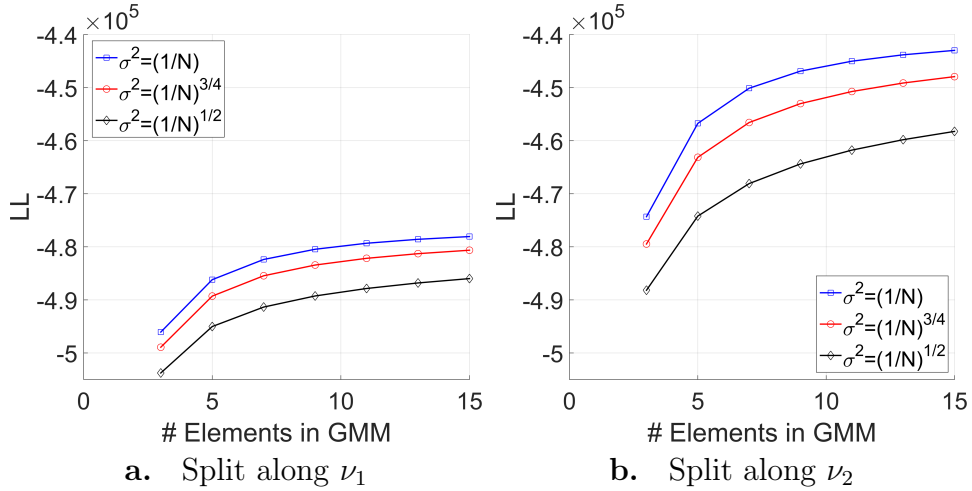


Figure 2.11: Performance of the various univariate splitting libraries for a nonlinear test case. Higher LL indicates a closer fit.

resulting Multidirectional GMMs (MGMMs) are presented in this section.

Splitting along multiple directions is carried out in a recursive manner. After the initial multivariate Gaussian distribution is split along the first direction, each of the resulting multivariate mixture elements is split along the next specified direction. The elements are again all split as a full tensor product until all specified directions have been covered. For spectral directions, the order of the directions to be split along is not important, as the spectral directions are all orthogonal, and applying the univariate library only reduces the eigenvalue of the covariance matrix along the specified eigenvector direction.

The multidirectional multivariate splitting technique essentially forms a regular grid in probabilistic space. Since the initial conditions are stochastic, there is a probability associated with an initial condition. A nonlinear function increases the non-Gaussianity of the initial distribution based on the



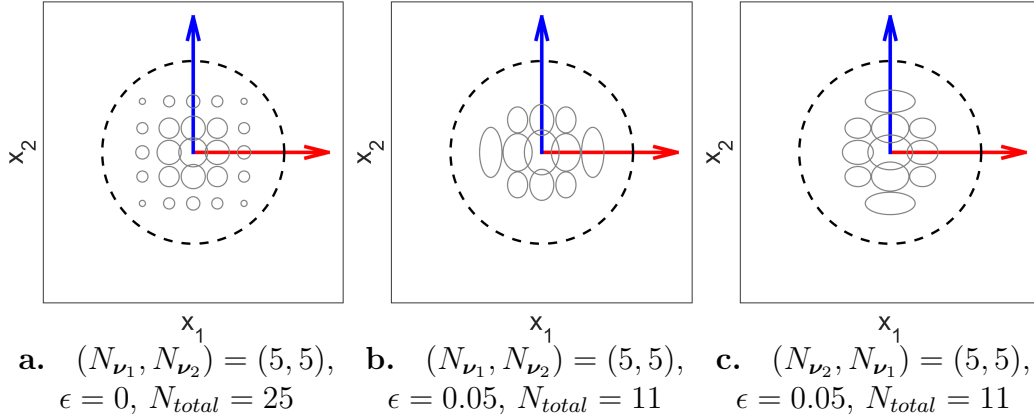


Figure 2.12: Splitting a bivariate IID Gaussian into an MGMM with and without a minimum weight threshold and 5 elements along  $\nu_1$  and  $\nu_2$ . Note that the covariance matrices are multiplied by their weights for ease of visualization

nonlinearity of the function and the size of the covariance matrix. By using the MGMM grid, the variance along the spectral directions is reduced. Also, the probability that a given region of the initial covariance matrix is described by the smaller variance GMM element is given by its weight. Thus, an MGMM becomes analogous to a Finite Element Method (FEM) technique where the basis functions are Gaussian distributions. As illustrated in Figure 2.12a, this grid is uniform since there is a symmetry about the mean of the initial multivariate Gaussian distribution.

The elements near the tails may have a very low weight when the number of elements is large along a certain direction, or due to the tensor product in multiple dimensions. Therefore, it is beneficial to specify a minimum weight threshold  $\epsilon$  such that the weights of all the elements in the MGMM are greater than  $\epsilon$ . Since the computational burden of propagating an element through the nonlinear function is not dependent on the weight, increasing  $\epsilon$  can reduce

the total computation cost with only a slight degradation in accuracy. The multidirectional splitting with a threshold is illustrated for a bivariate case in Figure 2.12b and Figure 2.12c. The order of splitting is important when the threshold is specified because the weight budget decreases with each recursive split.

## 2.4 Numerical Test Cases

The MGMM splitting technique is now applied to two two-dimensional test problems, one six-dimensional orbit uncertainty propagation, and one ten-dimensional problem. The two-dimensional problems are chosen in order to visualize the results. An initial bidirectional MGMM is expressed as  $(N_1, N_2)$ , where  $N_1$  and  $N_2$  are the number of splits along the eigenvectors  $\boldsymbol{\nu}_1$  and  $\boldsymbol{\nu}_2$ , respectively. For the bivariate cases, a GMM is simply an MGMM where either  $N_1$  or  $N_2$  are 1. The univariate library used to generate the MGMMs is Rule 1,  $\sigma^2 = (1/N)$ . The methodology and conclusions are applicable to higher dimensions as demonstrated by the orbit propagation and ten-dimensional problems.

The Gaussian distribution of the initial state is converted into an MGMM. Each element is then propagated through the nonlinear function. The true final non-Gaussian distribution after the nonlinear transformation is assumed to be the same as the result of using MC runs. Each of the Gaussian elements is propagated through the nonlinear function using the DD2 method found in Section 2.3.2. However, any other technique for propagating a Gaussian distribution, such as the UT or a Taylor series based method can be used. The

weights of the GMM elements are not updated after propagating. Updating weights improve performance at the cost of decreasing ease of implementation when a first order Taylor series approximation is used for propagating the elements [126]. However, no improvement is seen when higher order propagation methods such as the UT and DD2 are used [68].

#### 2.4.1 Conversion from Polar Coordinates to Cartesian Coordinates

The first test case is the conversion from Polar coordinates to Cartesian coordinates [77]. The initial state, covariance of the uncertainty, and the eigenvalues and vectors of the covariance matrix are:

$$\begin{aligned} \begin{bmatrix} r \\ \theta \end{bmatrix} &= \begin{bmatrix} 70 \\ \pi/3 \end{bmatrix} & \mathbf{P}_x &= \begin{bmatrix} 16 & 1.3 \\ 1.3 & (\pi/6)^2 \end{bmatrix} \\ \lambda_1 &= 0.1674 & \boldsymbol{\nu}_1 &= [ 0.0818 \quad -0.9966 ]^T \\ \lambda_2 &= 16.1067 & \boldsymbol{\nu}_2 &= [ -0.9963 \quad -0.0818 ]^T \end{aligned} \quad (2.32)$$

The resulting non-Gaussian distribution is illustrated in Figure 2.13. When Eq. (2.25), is used along with the 2–norm of  $\boldsymbol{\phi}$ , the values along the 2 spectral directions are:  $|\boldsymbol{\phi}_1|_2 = 5.5823$ ,  $|\boldsymbol{\phi}_2|_2 = 3.8793$ . Since the nonlinearity is higher along  $\boldsymbol{\nu}_1$  than along  $\boldsymbol{\nu}_2$ , a higher improvement is expected to be seen for splits along  $\boldsymbol{\nu}_2$ . However, since they are still similar in value, splitting along both spectral directions is likely beneficial.

Distributions sampled from the propagated MGMMs are seen in Figure 2.15. The likelihood of an MC run with respect to an MGMM is always smaller than 1 and therefore, the log-likelihood from Eq. (5.26) is a negative

quantity. A new quantity is now defined as:

$$\Delta LL(n_1, n_2) = LL(n_{1max}, n_{2max}) - LL(n_1, n_2) \quad (2.33)$$

$\Delta LL$  is the difference between the  $LL$  of the most accurate GMM case (39, 39), and the  $LL$  of an arbitrary case, and should always be positive. Figure 2.14 shows the  $\Delta LL$  as a function of the number of splits per direction in the MGMM. As predicted by the nonlinearity metric, splitting along  $\nu_1$  is more effective than splitting along  $\nu_2$ . However, the increase in performance due to the splitting along multiple spectral directions is clearly seen in Figure 2.14. Note that the performance of an MGMM (2,2) split, requiring 4 elements, is better than either the (1,39) or (39,1) GMMs that each require 15 elements.

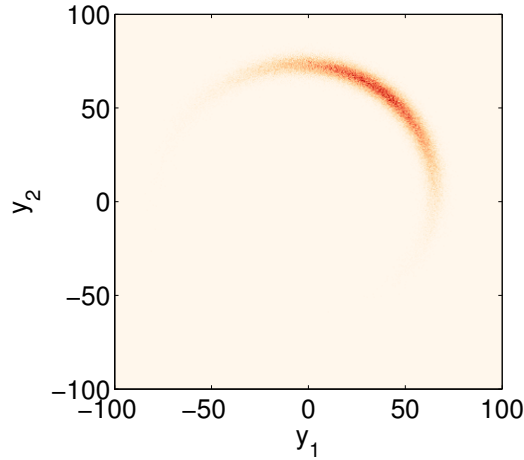


Figure 2.13: An MC run with  $5 \times 10^5$  points for the conversion from Polar to Cartesian coordinates

#### 2.4.2 Non-Linear ODE

A more complicated test function is now presented, where increasing the number of splits along only one direction does not suffice. The nonlinear

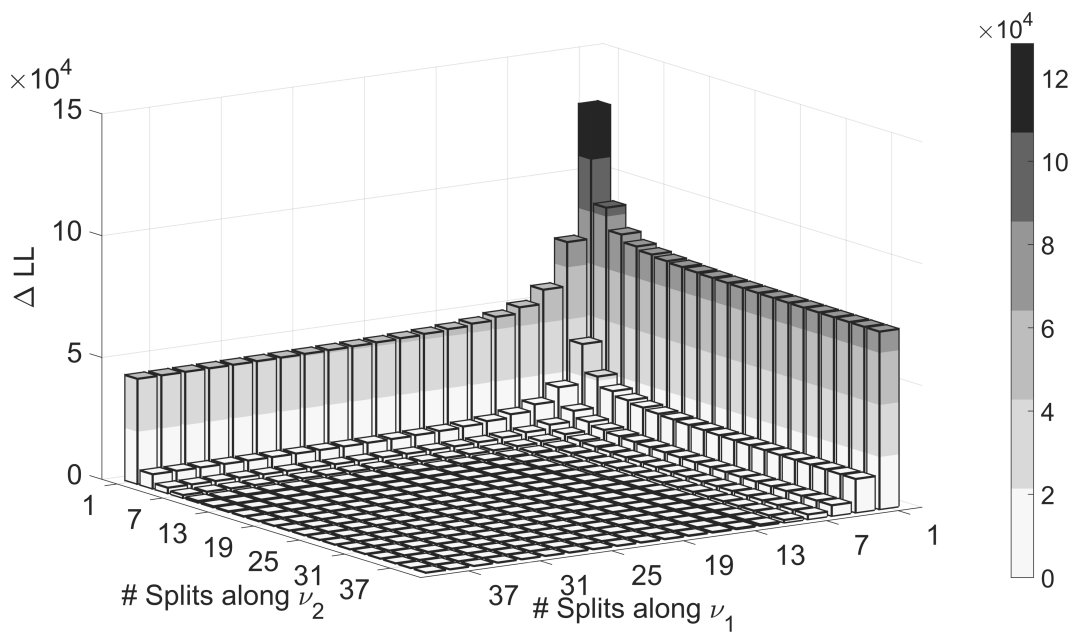


Figure 2.14:  $\Delta LL$  of a  $10^5$  point MC distribution with respect to the MGMM with increasing number of elements and  $LL(39, 39) = -7.5789 \times 10^5$  for the conversion from Polar to Cartesian coordinates

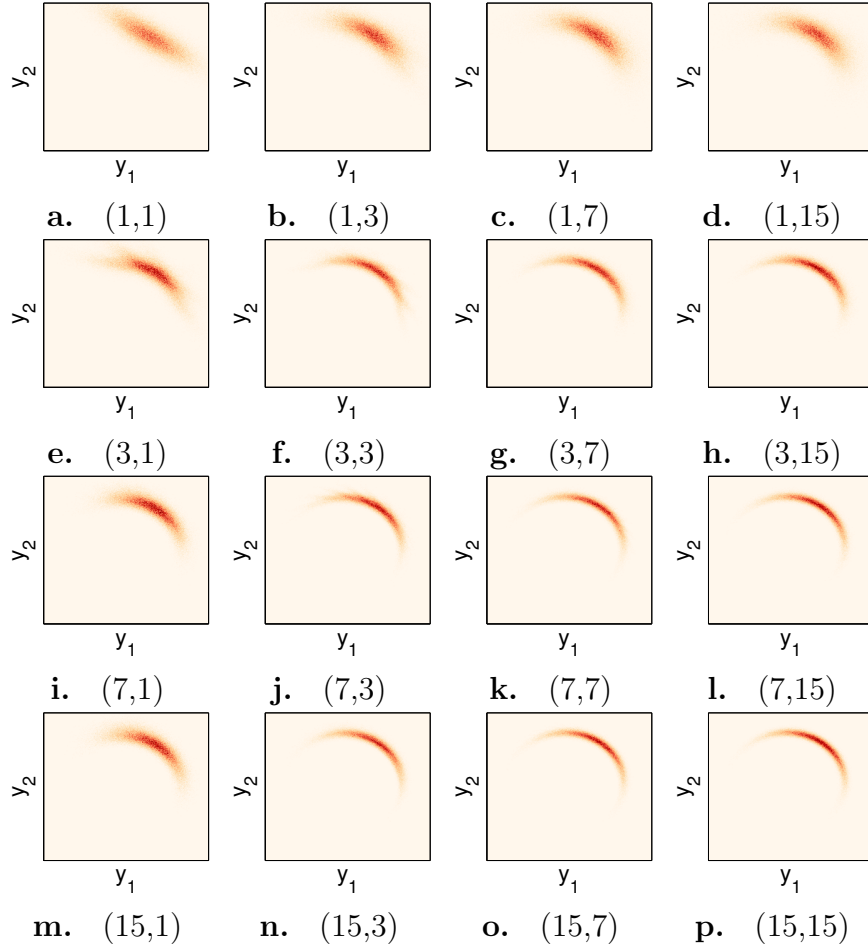


Figure 2.15:  $5 \times 10^5$  points sampled from the resulting MGMMs with  $N_1$  splits along  $\nu_1$  and  $N_2$  splits along  $\nu_2$ ,  $(N_1, N_2)$  for the conversion from Polar to Cartesian coordinates

function is the solution to the following differential equation:

$$\begin{aligned} \dot{x}_1 &= \cos(x_2) \sin(x_1) \\ \dot{x}_2 &= -\cos(x_1) \sin(x_2) \end{aligned} \tag{2.34}$$

Equation (2.34) is numerically integrated from  $t = 0$  to  $t = 3$  and the resulting values of  $x_1$  and  $x_2$  are used as the final values after the nonlinear transformation. The integrated ODE solution is considered to be a discrete black box

function. The initial state and covariance matrix are:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \mathbf{P}_x = \begin{bmatrix} 1 & 0.1 \\ 0.1 & 1 \end{bmatrix} \quad (2.35)$$

The eigenvalues and eigenvectors of the covariance matrix are:

$$\begin{aligned} \lambda_1 &= 0.9 & \boldsymbol{\nu}_1 &= [-1 \quad 1]^T \\ \lambda_2 &= 1.1 & \boldsymbol{\nu}_2 &= [1 \quad 1]^T \end{aligned} \quad (2.36)$$

The result of an MC run with  $10^4$  sample points is shown in Figure 2.16. The resulting distribution is highly non-Gaussian and bimodal, which favors the MGMM technique.

The nonlinearity metric using Eq. (2.25) results in  $|\phi_1|_2 = |\phi_2|_2 = 0$ . A  $\mathbf{0}$ -valued result using the second-order divided difference implies that the function is either linear, or it has higher-order nonlinearity at the mean of the initial Gaussian distribution. A conservative approach in this case is to split along both spectral directions.

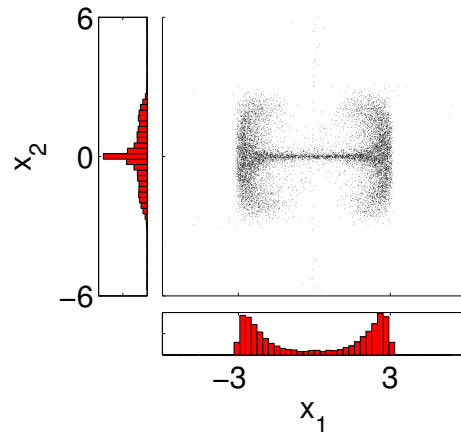


Figure 2.16: An MC run with  $10^4$  points for the bivariate ODE

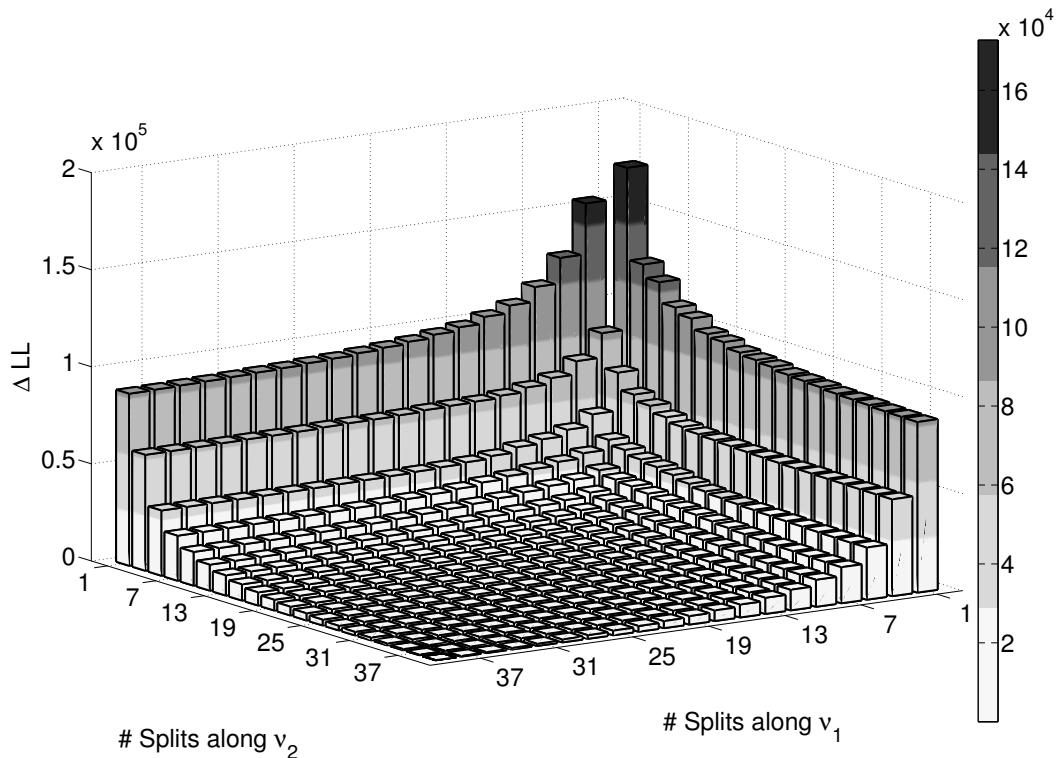


Figure 2.17:  $\Delta LL$  of a  $10^5$  point MC distribution generated with respect to the MGMM with increasing number of elements and  $LL(39, 39) = -2.9264 \times 10^5$  for the bivariate ODE

The samples from the propagated MGMMs are shown in Figure 2.18. The  $\Delta LL$  values for  $10^5$  MC sample points with respect to the MGMMs are seen in Figure 2.17. The benefit of multiple splitting directions arises due to nonlinearity existing along both eigenvectors. Figure 2.18 and Figure 2.17 show that an MGMM with an equal number of splits in both directions performs better than a GMM with the total number of splits applied along only one spectral direction.



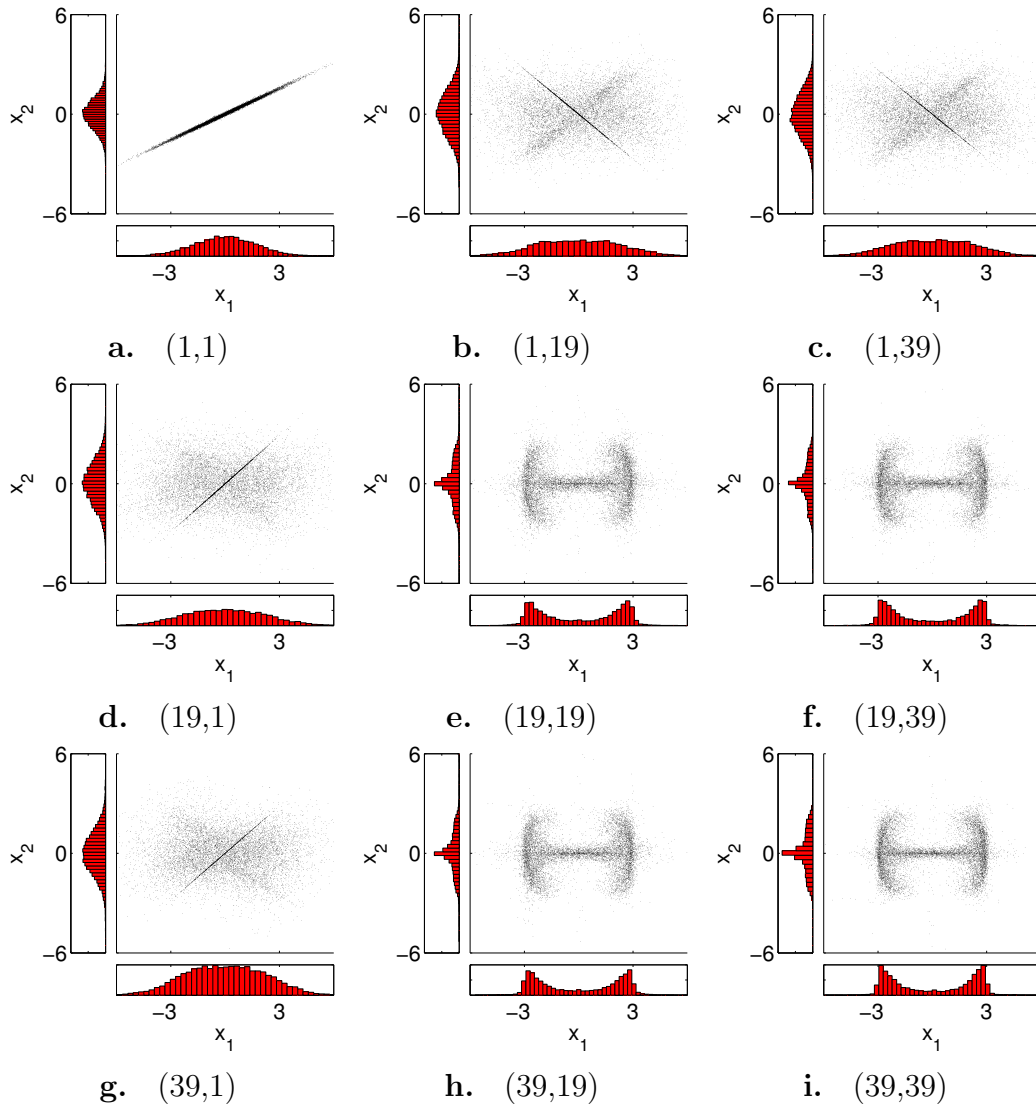


Figure 2.18:  $10^4$  points sampled from the resulting MGMMs with  $N_1$  splits along  $\nu_1$  and  $N_2$  splits along  $\nu_2$ ,  $(N_1, N_2)$  for the bivariate ODE

### 2.4.3 Orbit Uncertainty Propagation

The state uncertainty for a space object in GEO is propagated using two-body dynamics for a flight time of 3 days. The initial state and Gaussian

uncertainty are shown in Table 2.2. The orbit is assumed to be derived from optical observations, right ascension (RA) and declination (DEC), which in general create larger errors in the range direction for GEO objects. The largest nonlinearity directions, found in Table 2.2, are along  $x$  and  $v_y$ , which are the radial and tangential directions, respectively. The remaining directions have nonlinearity values that are orders of magnitude lower and therefore, do not benefit from splitting.

Table 2.2: Initial state, Gaussian uncorrelated uncertainty, and the nonlinearity measure for a space object in GEO

Variable	Mean	$\sigma$	$\ \phi\ _2$
$x$ [km]	42057.9	10.0	$6.1 \times 10^{-4}$
$y$ [km]	0	0.1	$1.1 \times 10^{-9}$
$z$ [km]	0	0.1	$1.1 \times 10^{-9}$
$v_x$ [km/s]	0	$1.2 \times 10^{-4}$	$2.8 \times 10^{-7}$
$v_y$ [km/s]	3.0800809759824	$0.6 \times 10^{-4}$	$4.1 \times 10^{-4}$
$v_z$ [km/s]	0	$0.25 \times 10^{-4}$	$1.2 \times 10^{-8}$

The  $\Delta LL$  values for  $10^5$  MC sample points with respect to the MGMMs are seen in Figure 2.19. As predicted by the nonlinearity measure, splitting along the  $x$  direction is more beneficial than splitting along the  $v_y$  direction. However, an MGMM along both  $x$  and  $v_y$  directions provides a more accurate result than a GMM along only one of the directions.

#### 2.4.4 High Dimensional Problem

The MGMM splitting technique is now applied to the ten-dimensional Extended Freudenstein & Roth function [21] to demonstrate the benefit of

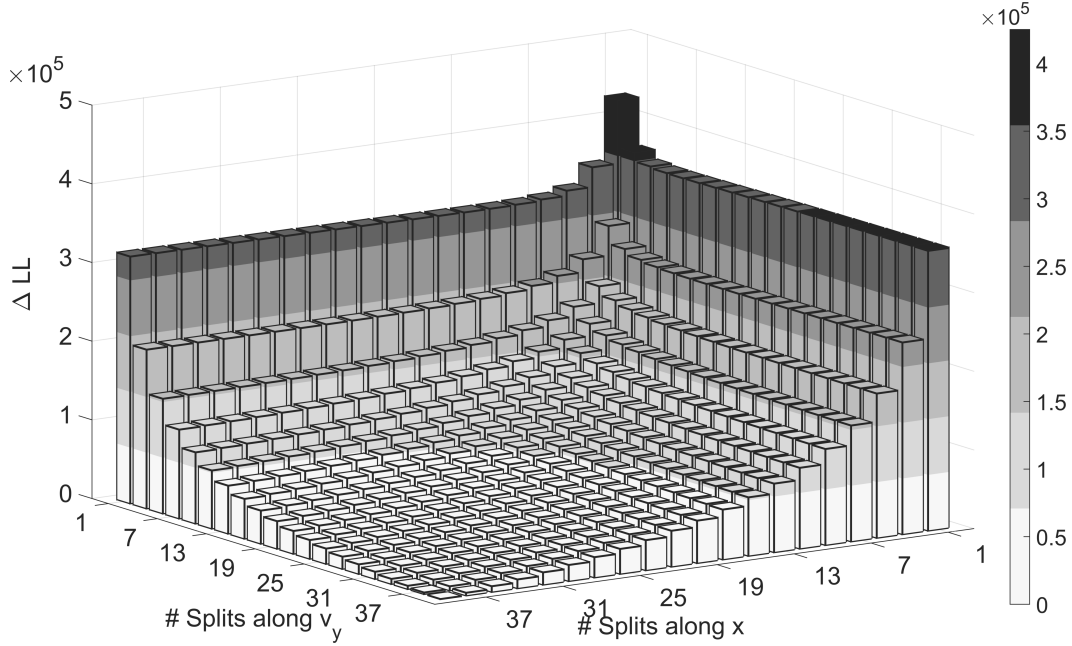


Figure 2.19:  $\Delta LL$  of a  $10^5$  point MC distribution generated with respect to the MGMM with increasing number of elements and  $LL(39, 39) = 4.8209 \times 10^6$  for the space object in GEO

using MGMMs and the nonlinearity test from Equation (2.25).

$$f(\mathbf{x}) = \sum_{i=1}^5 (-13 + x_{2i-1} + ((5 - x_{2i}) x_{2i} - 2) x_{2i})^2 + (-29 + x_{2i-1} + ((x_{2i} + 1) x_{2i} - 14) x_{2i})^2 \quad (2.37)$$

The ten-dimensional state  $\mathbf{x}$  has an initial mean and a diagonal covariance matrix for the uncertainty with the exact values found in Table 2.3. The initial means and variances are generated using a random number generator and then rounded to two decimal places. The absolute value of the nonlinearity measure, Eq. (2.25), along the various spectral directions is also found, and computed values are given in the last row of Table 2.3.

Spectral decomposition of the covariance matrix results in the eigenvec-

Table 2.3: Initial mean, variance, and nonlinearity measure for the Extended Freudenstein & Roth function

Variable	Mean	$\sigma^2$	$\phi$
$x_1$	6.19	1.67	3.34
$x_2$	3.76	1.81	$4.80 \times 10^3$
$x_3$	1.94	1.27	2.54
$x_4$	0.21	1.01	12.42
$x_5$	1.53	1.67	3.34
$x_6$	3.36	1.08	904.84
$x_7$	6.67	2.40	4.80
$x_8$	4.93	1.67	$1.68 \times 10^4$
$x_9$	2.33	1.35	2.70
$x_{10}$	5.73	1.06	$1.99 \times 10^4$

tors being the unit vectors of the individual univariate variables and  $\lambda_i = \sigma_i^2$ . The last row of Table 2.3 shows the largest nonlinearity directions are 10, 8, 2, and 6, in descending order. The nonlinearity along the other directions are considered to be insignificant. The values of the nonlinearity measure along a given direction correspond to the improvement in performance, shown in Figure 2.20, gained by generating splits along that direction. Since nonlinearity is present along multiple spectral directions, an MGMM further improves accuracy compared to using a GMM. The nonlinearity measure is a useful ranking method to find the number of splits to implement along the spectral directions. The number of splits along direction 10  $N_{10}$  is used to calibrate the number of splits along the other directions 8, 2, and 6 because highest nonlinearity is along  $i = 10$ . The number of splits along the remaining spectral directions are

computed using the following equation:

$$N_i = \text{odd} \left( \frac{\log(\phi_i)}{\log(\phi_{10})} N_{10} \right) \quad i = 8, 2, 6 \quad (2.38)$$

where  $\text{odd}$  computes the closest odd integer to a number:

$$\text{odd}(x) = 2 \times \text{ceil}(x/2) - 1 \quad (2.39)$$

The performance of the GMMs along directions 10, 8, 2, and 6 are seen in Figure 2.20, along with the performance of the 4-dimensional MGMM. The nonlinearity measure along a spectral direction performs well as an indicator of increase in performance gained by splitting along that direction. Direction 10 has the highest value for the nonlinearity measure and therefore, splitting along this direction results in the largest increase in performance. Splitting only along any one direction has a plateau in performance after a certain number of splits is reached. Using an MGMM, however, results in a higher value of the log-likelihood compared to a GMM along any of the directions.

The benefit of using a minimum threshold for the MGMM weights is seen in Figure 2.20. When a threshold of  $\epsilon = 10^{-4}$  is used, approximately 6 times fewer elements are required to achieve the same accuracy ( $LL \approx -1.2 \times 10^6$ ) for the MGMM without a threshold. For the highest accuracy presented here with  $N_{10} = 17$ , thresholds of  $\epsilon = 10^{-6}$  and  $\epsilon = 10^{-5}$  require approximately 2 times and 4 times fewer elements, respectively.

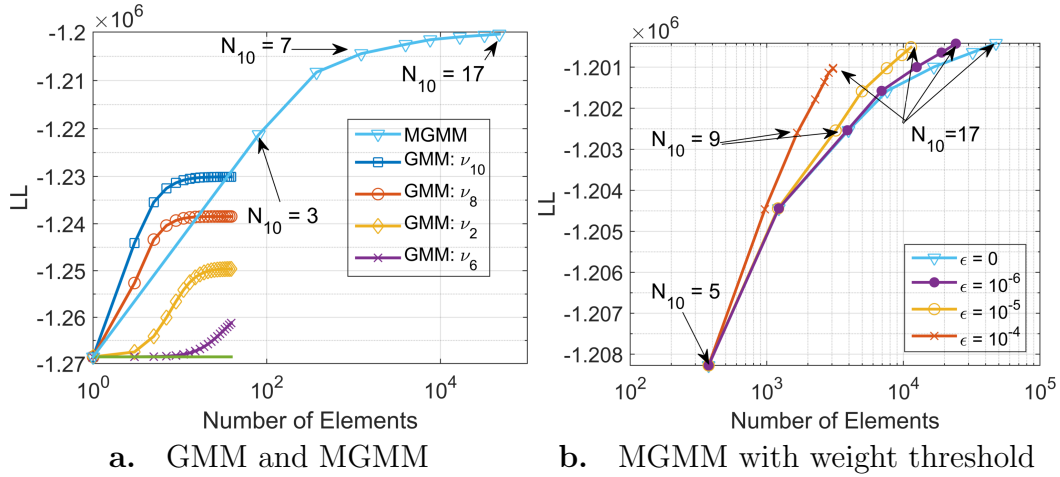


Figure 2.20:  $LL$  of a  $10^5$  point MC distribution with respect to GMMs and MGMMs for the 10-dimensional case

## 2.5 Conclusion

Non-linear functions acting on an initial multivariate state with a Gaussian uncertainty are common scenarios because a Gaussian state uncertainty is the result of most Orbit Determination and state estimation algorithms for satellites and Near Earth Asteroids (NEAs). Accurately capturing the resulting non-Gaussian distribution after the nonlinear transformation without resorting to computationally expensive techniques such as MC simulations remains an attractive ongoing area of research. The primary contributions of this chapter are 1) the extension of the univariate GMM library to 39 elements; 2) a formal extension of the GMM concept to multiple directions; and 3) a method to choose any split direction and introduction of a nonlinearity metric to rank the importance of splitting along a certain direction.

Univariate splitting libraries of up to 39 elements that are carefully fit

in advance are generated by minimizing the  $L_2$  distance with respect to a standard normal distribution. Instead of applying the univariate library along only a single direction, the univariate library is applied recursively along multiple directions forming a regular grid over multiple dimensions. These Multidimensional GMMs (MGMMs) approximate the propagated multivariate non-Gaussian distribution more accurately than if the split is made along only one direction. A second-order divided difference is used to measure nonlinearity in each of the directions. Thus, the directions that benefit most from splitting are identified and can be exploited. This nonlinearity merit provides a ratio to relate the number of splits along each direction. Therefore, a practitioner can implement the MGMM technique with the selection of just one fidelity tuning parameter. A user enforces a number of splits along a single direction, then the number of splits in the other directions can be chosen according to their relative nonlinearity. A threshold for the weight further reduces the computational load by ensuring that the MGMM only contains elements above a certain weight.

Combining the univariate library with up to 39 elements with the multidirectional splitting results in a deterministic choice of initial states for propagation through the nonlinear function. The number of necessary function evaluations is ultimately a trade-off between accuracy and computational cost. The MGMM allows users to access all regions of this spectrum, filling in the gap that typically exists between Monte Carlo and simple linearized covariance analyses. The MGMM representation of the state uncertainty of space objects, including potentially hazardous Near Earth Asteroids, is capable of

improving the accuracy of commonly used estimation and conjunction assessment techniques with minor considerations in the implementation as shown in Chapter 5.



## Chapter 3

# Gaussian Mixture Models and Polynomial Chaos Expansions

Polynomial Chaos Expansion (PCE) and Gaussian Mixture Models (GMMs) are combined in a hybrid fashion (GMM-PC) to propagate state uncertainty for Resident Space Objects with initial Gaussian errors in this chapter <sup>1</sup>. GMM-PC reduces the overall polynomial order required to reach a desired accuracy. The initial Gaussian distribution is converted to a GMM, and PCE is used to propagate the state uncertainty represented by each of the elements through the nonlinear dynamics. Splitting the initial distribution into a GMM reduces the size of the covariance associated with each new element thereby reducing the domain of approximation and allowing for lower order polynomials to be used. Several RSO state uncertainty propagation examples, Phase I of Figure 1.7 on page 12, are shown using GMM-PC. The resulting distributions are shown to efficiently capture the full shape of the true

---

<sup>1</sup>The work from this chapter has been presented at a conference:

- V. Vittaldev, R. Linares, and R. P. Russell, Uncertainty Propagation using the Gaussian Mixture Model Polynomial Chaos Approach, Paper AAS 15-448, AAS/AIAA Space Flight Mechanics Meeting, Williamsburg, VA, 2015.

A manuscript has also been submitted to a peer reviewed journal:

- V. Vittaldev, R. Linares, and R. P. Russell, Spacecraft Uncertainty Propagation using Gaussian Mixture Models and Polynomial Chaos Expansion, Submitted August 2015.

non-Gaussian distribution better than a standard GMM or a PCE. The two-sample Cramér-von Mises parameter is used to compare the samples from the GMM-PC uncertainty propagation and the samples from a benchmark Monte Carlo simulation to quantify the accuracy of PCE and the newly developed GMM-PC.

### 3.1 Introduction

A spectrum of techniques exists that propagate the state and uncertainty of an initially Gaussian distribution through a nonlinear function, such as orbit propagation [107]. Reduction in computation cost comes with a sacrifice in the accuracy of the final probability density function (PDF). Considering the runtime vs. accuracy spectrum, first order Taylor series propagation lies on the low computation and accuracy extremity whereas MC lies on the high computation and accuracy extremity. Two techniques that lie in between the first order Taylor series and MC are the Gaussian Mixture Models (GMMs) shown in Chapter 2 and Polynomial Chaos expansions (PCEs). PCEs are presented in detail in Section 3.2 and use a linear combination of orthogonal polynomials (OP) as a surrogate model for the underlying nonlinear function.

The motivation for combining PCE and GMMs (GMM-PC) for uncertainty propagation arises from a comparison with the finite element method (FEM). In this analogy the initial uncertainty is considered to be an object of interest and the nonlinear function is an applied load. In FEM, a mesh grid is generated over the object to discretize it into smaller and simpler geometries. The exact solution is obtained at the nodes, but between the nodes

inside each element some polynomial interpolant is used to approximately obtain the functional form of displacements. Increasing the number of elements by reducing the size of each element is known as  $h$ -refinement and increasing the order of the interpolant is  $p$ -refinement. The order of the interpolant for uncertainty propagation is increased by using higher order Taylor Series Expansions [103], using more sigma points [9], or using a PCE [75]. Selectively modifying both the size and number of elements, and the order of the polynomial interpolant is  $hp$ -refinement [27, 46]. Splitting the initial Gaussian distribution into Gaussian distributions with a smaller differential entropy is analogous to  $h$ -refinement. Using a higher order accurate method of propagating the uncertainty of a single Gaussian element is  $p$ -refinement. A proof of concept is presented here to demonstrate that a  $hp$ -refinement method of using PCE for the propagation of a GMM is applicable for orbit uncertainty propagation by increasing accuracy while having a lower computational cost than an MC simulation.

The organization of this chapter is as follows: first a summarized background is provided for PCE in Section 3.2. Next, the combined GMM-PC is presented in Section 3.3. Then the performance of GMM-PC is compared to PCE for three highly nonlinear orbit uncertainty propagation cases in Section 3.4. Finally, conclusions are provided from these results in Section 3.5.

## 3.2 Polynomial Chaos Expansion

For a PCE, the uncertainty in variables through a transformation is represented by a linear combination of orthogonal polynomials. The main rea-

son for using PCE is to efficiently generate samples from the post-propagated uncertainty distribution. For extremely non-Gaussian distributions, millions of samples are required from a Monte Carlo simulation in order to accurately capture the PDF. A PCE, or any other surrogate model, replaces the true non-linear function by a function that is less computationally intensive to evaluate. Therefore, the combined computation time required for creating the surrogate model, i.e. computing the coefficients of the orthogonal polynomials, and for generating the required number of samples is less than the computation time of a Monte Carlo simulation with the original function.

Let  $u(\xi, t)$  be a second order random process which is a function of time and a random variable  $\xi \in \Omega$  where  $u(\xi, t) : (\Omega \times \mathbb{R}^+) \rightarrow \mathbb{R}^1$ , then  $u(\xi, t)$  can be expressed by a gPCE as:

$$u(\xi, t) = \sum_{i=0}^{\infty} c_i(t) \Psi_i(\xi) \quad (3.1a)$$

$$\Psi_i(\xi) = \sum_{j=0}^{j=i} a_{i,j} \xi^j \quad (3.1b)$$

Where  $c_i(t) \in \mathbb{R}$  are the  $i^{\text{th}}$  coefficient associated with the  $i^{\text{th}}$  orthogonal polynomials  $\Psi_i$ . The orthogonal polynomials  $\Psi_i$  and their coefficients  $a_{i,j}$  are defined as being orthogonal w.r.t. the following inner product:

$$\int_{\Omega} \Psi_m(\xi) \Psi_n(\xi) p(\xi) d\xi = 0, \quad m \neq n \quad (3.2)$$

where  $p(\xi)$  satisfies the properties of a PDF over the domain  $\Omega$  i.e.  $\xi \in \Omega$ . In the Weiner-Askey [137] scheme the orthogonal polynomial type is determined

by  $p(\xi)$  for classical PDFs. In general, the aPCE scheme can account for cases where  $p(\xi)$  are not classical but arbitrary PDFs by determining the orthogonal polynomials using the raw moments of the distribution [101]. In this work the initial distribution is assumed to be Gaussian and therefore, probabilists' Hermite polynomials are chosen according to the Wiener-Askey [137] scheme. The weighting function for the probabilists' Hermite polynomials is proportional to the standard normal distribution:

$$p(\xi) = e^{-\xi^2/2} \quad (3.3)$$

For Hermite polynomials, the three term relation is as follows [123]:

$$\psi_{n+1}^*(\xi) = \xi\psi_n^*(\xi) - n\psi_{n-1}^*(\xi) \quad (3.4)$$

Equation (3.4) is used to derive  $n^{\text{th}}$ -order orthogonal polynomials given  $\psi_0^* = 1$  and  $\psi_1^* = \xi$ . The polynomials used in this chapter are further made orthonormal:

$$\psi_n(\xi) = \psi_n^*(\xi)/\sqrt{n!} \quad (3.5)$$

The orbit uncertainty propagation problem is a multivariate problem and therefore, requires orthogonal multivariate polynomials. A multivariate Gaussian distribution can be converted to a product of IID standard normal distributions. Multivariate polynomials are generated from a set of independent random variables and created using the multi-index notation. The multi-index vectors,  $\boldsymbol{\alpha}_i \in \mathbb{R}^n$ , contain the orders of the  $d$  univariate polynomials that are combined in a tensor product to produce multivariate polynomials.

The number of monomials,  $L$ , of order  $\ell$  or less in  $d$  variable is computed by

$$L = \frac{(d + \ell)!}{d! \ell!} \quad (3.6)$$

Using multi-index notation, the multivariate polynomial is the product:

$$\Psi_{\alpha i}(\boldsymbol{\xi}) = \prod_{k=1}^d \psi_{\alpha_i(k)}(\xi_k) \quad (3.7)$$

If the output is a vector function of dimension  $m$ ,  $m \times L$  coefficients  $c_i(t)$  have to be solved for.

In practice, Eq. (3.1a) is approximated by truncating the infinite series for a fixed number of polynomial functions. The approximate PCE for multivariate polynomials is:

$$\bar{u}_L(\boldsymbol{\xi}, t) = \sum_{i=0}^L c_i(t) \Psi_{\alpha i}(\boldsymbol{\xi}) \quad (3.8)$$

It has been shown that for analytic functions the truncated approximate PCE,  $\bar{u}(\boldsymbol{\xi}, t)$ , given above converges exponentially to the second order process  $u(\boldsymbol{\xi}, t)$  in the  $L_2$ -norm sense [137], i.e.,  $\mathbb{E} [(\bar{u}_L(\boldsymbol{\xi}, t) - u(\boldsymbol{\xi}, t))^2] \rightarrow 0$  as  $L \rightarrow \infty$ .

The coefficients  $c_i(t)$  determine the response surface of the surrogate model consisting of the Hermite polynomials. Two methods of finding these coefficients are the *intrusive* and the *non-intrusive* methods. The intrusive method requires knowledge of the full nonlinear function that determines the evolution of the random vector of inputs. The truncated polynomial expansions are introduced into the model equations and solved using a Galerkin projection of the equations on the polynomial space. A system of equations is then solved for  $c_i(t)$ . The intrusive method cannot be used with black-box

dynamics because existing codes have to be rewritten to form additional differential equations for  $c_i(t)$  and therefore, is not considered in this work. The intrusive method is analogous to integrating the analytical Jacobian of the dynamics to compute the first-order STM. The non-intrusive method does not require any knowledge of the propagation function and is analogous to using one of the numerical differencing methods for computing the STM. Given that the system can be solved for a sample initial condition, the projection property (Galerkin Projection) for approximating the coefficients in Eq. (3.8) can be used:

$$c_i(t) = \int_{\Omega} u(\boldsymbol{\xi}, t) \Psi_{\alpha i}(\boldsymbol{\xi}) p(\boldsymbol{\xi}) d\boldsymbol{\xi} \quad (3.9)$$

where  $p(\boldsymbol{\xi})$  is the PDF of  $\boldsymbol{\xi} \in \Omega$ . The coefficients can be computed by a quadrature numerical approximation of Eq. (3.9), MC sampling, or with Least Squares (LS) regression. The results in this chapter use LS regression for computing the coefficients. Both LS and MC methods for numerically approximating Eq. (3.9) are based on random sampling. The quadrature methods on the other hand, require deterministic precomputed nodes and weights. All non-intrusive methods generate sample points from  $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}_d, \mathbf{I}_{d \times d})$ . A square-root factor of the covariance matrix linearly converts  $\boldsymbol{\xi}$  from an IID variable to the initial multivariate Gaussian distribution.

For LS regression,  $N$  random samples are generated for  $\boldsymbol{\xi}$ . The coefficients minimize the squared difference between the propagated sample points and the PC expansion.

$$c_i(t) \approx \arg \min_{\hat{c}_i(t)} \frac{1}{N} \sum_{j=1}^N \left( u(\boldsymbol{\xi}_j, t) - \sum_{i=1}^L \hat{c}_i(t) \Psi_{\alpha i}(\boldsymbol{\xi}_j) \right)^2 \quad (3.10)$$

Equation (3.10) is solved by rewriting into the traditional LS form using the following matrix containing the  $L$  multivariate polynomials evaluated at the  $N$  nodes:

$$\mathbf{\Psi} = \begin{bmatrix} \Psi_{\alpha 1}(\boldsymbol{\xi}_1) & \dots & \Psi_{\alpha L}(\boldsymbol{\xi}_1) \\ \vdots & \ddots & \vdots \\ \Psi_{\alpha 1}(\boldsymbol{\xi}_N) & \dots & \Psi_{\alpha L}(\boldsymbol{\xi}_N) \end{bmatrix} \quad (3.11)$$

Arranging the coefficients and function evaluations into vectors results in the following linear system:

$$(\mathbf{\Psi}^T \mathbf{\Psi}) \begin{bmatrix} c_1(t) \\ \vdots \\ c_L(t) \end{bmatrix} = \mathbf{\Psi}^T \begin{bmatrix} u(\boldsymbol{\xi}_1, t) \\ \vdots \\ u(\boldsymbol{\xi}_N, t) \end{bmatrix} \quad (3.12)$$

The LS method suffers from the curse of dimensionality where the combination of increasing problem dimension and order of the polynomial scale the number of required evaluations in a factorial manner. For an  $n$ -dimensional input state with univariate polynomials of maximum order  $\ell$ , the number of terms  $L$  in the multivariate polynomial is computed using Equation (3.6). The number of terms in a six-dimensional multivariate polynomial as a function of  $\ell$  is shown in Figure 3.1.

The LS regression method requires approximately  $2L$  sample points to solve for the coefficients [69]. However, if many coefficients have a small value, compressive sensing techniques can generate a sparse representation of  $\mathbf{c}(t) = [c_1(t), \dots, c_L(t)]^T$  so that  $N < L$  function evaluations suffice. To generate a true sparse representation of  $\mathbf{c}(t)$ , the  $L_0$  norm of  $\mathbf{c}(t)$  has to be minimized subject to the  $L_2$  norm conditions between the function evaluations and the PC solution. However, minimizing the  $L_1$  norm instead of the  $L_0$  norm



of  $\mathbf{c}(t)$  converts the problem into a convex optimization problem that can be solved with common solvers [58, 57].

A full tensor product for the multivariate quadrature requires  $N = l^d$  function evaluations and therefore, also suffers from the curse of dimensionality. Sparse grids compute quadrature nodes using sparse tensor product and therefore, reduce the number of function evaluations to  $N < l^d$  for high dimensions. The number of function evaluations required for using a full tensor product quadrature rule and two different sparse grids, Smolyak [120] and Genz-Keister [55], is shown in Figure 3.1. LS regression is used in the current chapter, which uses  $N = 2L$  function evaluations, where  $L$  is computed using Equation (3.6).

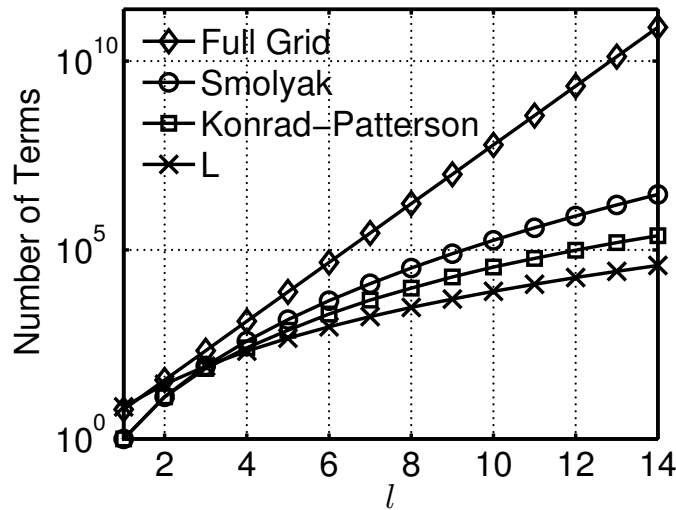


Figure 3.1: Terms required for multivariate polynomials and the number of quadrature points for a 6 dimensional input state. The LS regression requires  $2L$  points

### 3.3 Gaussian Mixture Model Polynomial Chaos

The Gaussian Mixture Model Polynomial Chaos (GMM-PC) approach is used to approximate the distribution of the following nonlinear function:

$$\mathbf{y} = \mathbf{f}(\mathbf{x}), \quad \mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{P}) \quad (3.13)$$

where  $\mathbf{x} \in \mathbb{R}^d$  and  $\mathbf{y} \in \mathbb{R}^{d_2}$ . For a pure state propagation problem with only state uncertainty, the nonlinear problem is:

$$\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k, t_k, t_{k+1}), \quad \mathbf{x}_k \sim \mathcal{N}(\boldsymbol{\mu}_k, \mathbf{P}_k) \quad (3.14)$$

where  $\mathbf{x}_k$  denotes the state of the system at time index  $k$  or time  $t_k$  and  $\mathbf{f}_k(\cdot)$  is a nonlinear function. This system is assumed to have Gaussian initial errors with mean and covariance  $\boldsymbol{\mu}_k$  and  $\mathbf{P}_k$  respectively. A PCE forms a surrogate model, which is sampled to approximate the distribution of  $\mathbf{x}_k$ . From Eq. (3.6) the number of terms  $L$  that are required to achieve a given order  $\ell$  for the PCE approach grows factorially with dimension  $d$ , therefore it becomes increasingly difficult to increase the order of accuracy for higher dimensional problems. This provides a strong incentive to use the lowest order PCE possible, subject to an acceptable accuracy level. The GMM-PC reduces the order of the PCE by splitting the initial Gaussian distribution into a GMM. Each weighted Gaussian element is further propagated through the nonlinear function using PCE. The non-Gaussian behavior of the PDF post-propagation depends primarily on the nonlinearity of the function over the domain of the initial uncertainty. Splitting the initial Gaussian distribution into smaller distributions, decreases the size of the initial uncertainty of each element. Splitting reduces the nonlinearity of the function by reducing the domain of each element, requiring

lower order polynomials to achieve desired accuracy. However, increasing the number of splits in the initial distribution increases the required function evaluations linearly, whereas a reduction in order decreases the number of function evaluations factorially.

The Multi-Element generalized Polynomial Chaos (ME-gPC) method [135] also decomposes the PDF into smaller subdomains by partitioning the support of the initial PDF. For a Gaussian initial distribution, the subdomains of the ME-gPC method have a non-classical distribution seen in Figure 3.2. Therefore, Gram-Schmidt orthogonalization is required to form the polynomials for this non-classical distribution, this is in effect an aPCE technique over the subdomains [101]. Because each element in the GMM-PC technique is itself a Gaussian distribution, standard Hermite polynomials are used, which greatly reduces the complexity of the code over the ME-gPC because analytical recursive relations exist for Hermite polynomials. In the ME-gPC method, the initial PDF is exactly represented by the elements, but the numerically computed polynomials are approximations. However, in the GMM-PC method, the initial PDF is approximated by a GMM but the form of the orthogonal polynomials (coefficients  $a_i$  from Equation (3.1b)) are available in closed form. It should be noted that the actual PCE coefficients, which determine the linear combination of the orthogonal polynomials (coefficients  $c_i$  from Equation (3.1a)), are computed using regression.

To construct an  $N$ -element GMM-PC for the nonlinear problem from Eq. (3.13), the initial multivariate Gaussian distribution is first approximated

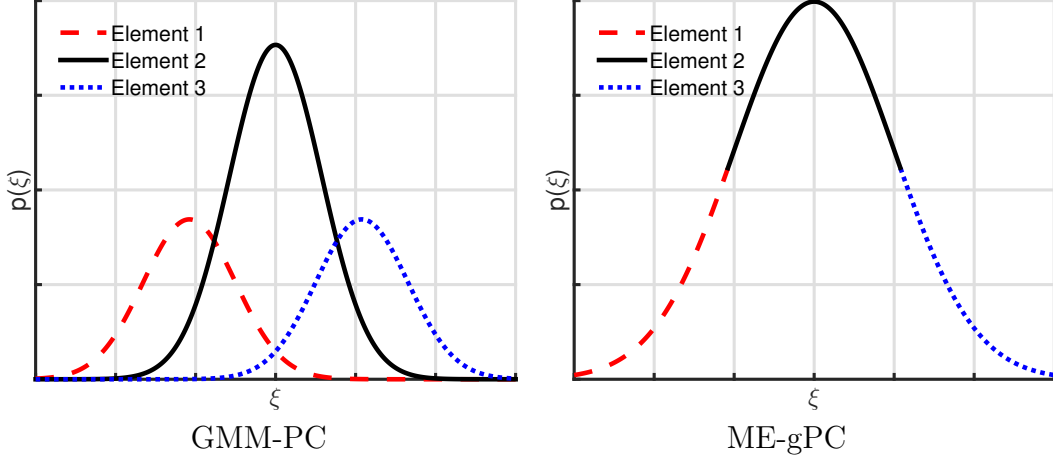


Figure 3.2: The difference in a three element refinement of the standard normal distribution between the GMM-PC and the ME-gPC techniques

by a GMM.

$$p_g(\mathbf{x}; \boldsymbol{\mu}, \mathbf{P}) \approx \sum_{i=1}^N \alpha_i p_g(\mathbf{x}_i; \boldsymbol{\mu}_i, \mathbf{P}_i) \quad (3.15)$$

Note that the subscript  $i$  denotes the GMM element number. Samples can be drawn from the GMM in Eq. (3.15) and PCE is used to propagate the uncertainty of each element through the nonlinear function. For each element, the square-root matrix  $\mathbf{S}_i$  is a transformation from a reference frame where the variables are IID.

$$\mathbf{x}_i = \mathbf{S}_i \boldsymbol{\xi}_i + \boldsymbol{\mu}_i, \quad \mathbf{P}_i = \mathbf{S}_i \mathbf{S}_i^T, \quad \boldsymbol{\xi}_i \sim \mathcal{N}(\mathbf{0}_d, \mathbf{I}_{d \times d}) \quad (3.16)$$

where  $i$  represents the GMM element and  $d$  is the dimension of the input parameters  $\mathbf{x}$ . A sample from the GMM-PC is generated randomly by the  $i^{\text{th}}$  element, where  $i$  is determined by a random variable  $X$  with a uniform distribution,  $X \sim \text{U}(0, 1)$ . Given a random sample  $X$ ,  $i$  is determined such

that:

$$\sum_{k=0}^{i-1} \alpha_k < X \leq \sum_{k=1}^i \alpha_k, \quad \alpha_0 = 0 \quad (3.17)$$

To sample from the GMM-PC, a set of samples  $X$  are drawn from  $U(0, 1)$  and a set of  $i$  are found from these samples using Eq. (3.17). Then the number of samples that are drawn from each element in the GMM is determined by the set of  $i$ . Finally, with a desired univariate order  $\ell_i$  for each element, the surrogate model given by the GMM-PC approximation for Eq. (3.13) is:

$$\mathbf{y}_i = \mathbf{C}_i \Psi_i^* (\boldsymbol{\xi}_i) \quad (3.18)$$

where

$$\Psi_i^* = [ \Psi_{\alpha_1} (\boldsymbol{\xi}_i) \quad \dots \quad \Psi_{\alpha_{L_i}} (\boldsymbol{\xi}_i) ]^T \quad (3.19a)$$

$$\mathbf{C}_i \in \mathbb{R}^{d_2 \times L_i} \quad (3.19b)$$

The matrix of coefficients  $\mathbf{C}_i$  are computed for each element using Least Squares.

A univariate example is now considered where a 13<sup>th</sup>-order normalized Hermite polynomial expansion is used as the nonlinear function,  $y = f(x)$ , is . The polynomials are computed using Equations (3.4) and (3.5).

$$y = \sum_{k=0}^{13} c_k \psi_k(x), \quad x \sim \mathcal{N}(0, 1) \quad (3.20)$$

$$c_k = (-1)^k \times e^{-\beta \times k} \quad (3.21)$$

The assigned value of  $\beta$  controls the influence of the higher order coefficients and a larger value indicates greater nonlinearity. The result of the uncertainty

propagation through Eq. (3.20) is approximated by PCE (1 element GMM-PC) and GMM-PC of orders 3 - 10. The two-sample univariate Cramér-von Mises parameter is used to compare the performance of the PCE and GMM-PC (3, 5, and 7 elements) with respect to an MC simulation [20]:

$$CvM = \frac{N_1 N_2}{N_1 + N_2} \int_{-\infty}^{\infty} [F_{N_1}(x) - G_{N_2}(x)]^2 dH_{N_1+N_2}(x) \quad (3.22)$$

where  $F_{N_1}(x)$  is the empirical distribution of the surrogate model with  $N_1$  samples,  $G_{N_2}(x)$  is the empirical distribution of the MC simulation with  $N_2$  samples, and  $H_{N_1+N_2}(x)$  is the empirical distribution function of two samples together. A lower value of  $CvM$  implies higher accuracy. The  $CvM$  values for  $\beta = 1$  and  $\beta = 0.1$  are computed for increasing number of elements and PCE order for GMM-PC in Figures 3.3a and 3.3b, respectively. When  $\beta = 1$ , a one element GMM-PC (standard PCE) is able to accurately capture the uncertainty of the 13<sup>th</sup>-order PCE for orders greater than 3 and there is no benefit of using more elements. The GMM-PC with 3 elements does not provide accurate results ( $CvM < 0.3$ ) for any order because of the difference between the original Gaussian distribution and the GMM approximation. The behavior of the  $CvM$  values resembles a random walk in Figure 3.3a because the test cannot discriminate between two distributions when they are virtually identical. When  $\beta = 0.1$ , the benefit of increasing the number of elements is clearly seen and a GMM-PC with  $N > 1$  outperforms the PCE for  $\ell < 8$ . The nonlinearity of the function is large enough that the difference between the GMM distribution and the original Gaussian distribution is not noticeable after propagation.

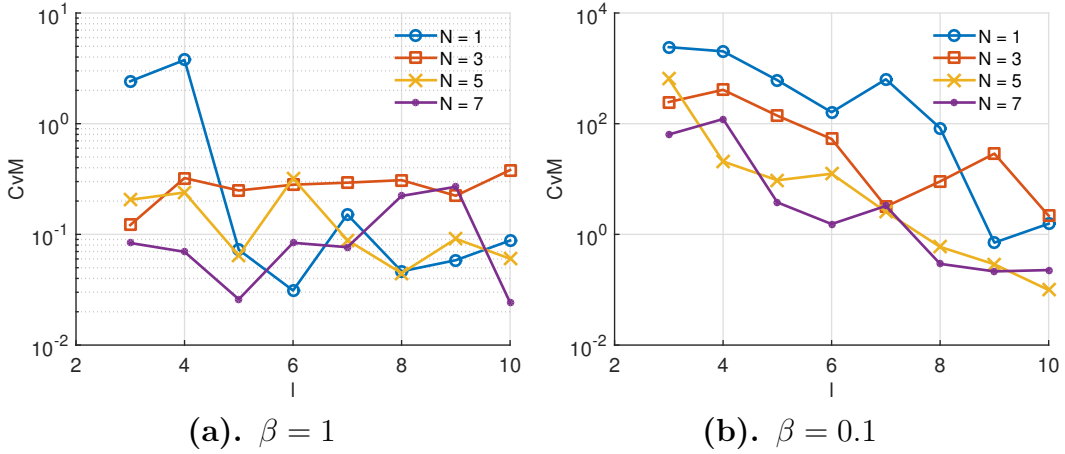


Figure 3.3: Performance of approximating a 13<sup>th</sup>-order Hermite PCE using lower order ( $3 \leq l \leq 10$ ) GMM-PC using 1,000,000 samples

Approximating the uncertainty propagation through an arbitrary function using PCE is analogous to approximating a higher order PCE with a lower order PCE. In case the function is not highly nonlinear, a lower order PCE is able to approximate the resulting uncertainty with high accuracy and there is no benefit of using GMM-PC. However, when the nonlinearity is strong, the benefit of GMM-PC is clearly seen. The computational benefit is greater as the dimension of the problem increases because the rate of growth of the number of multivariate polynomials  $L$  is factorial with respect to the dimension. The computational load of GMM-PC can be further reduced if the PCE order of each element in the GMM-PC is varied proportionally to the element weight. If the quadrature or cubature methods are used to compute the PCE coefficients in GMM-PC, the black box function has to be run for the deterministic nodes for each element. In this case, the computational load increases linearly with the number of elements if each GMM-PC element uses the same

order of polynomials  $\ell$ .

A novel strategy is now proposed, and later tested in the Results, Section 3.4.5, when using the LS method for GMM-PC. The computational budget for the number of function evaluations is first fixed. Using the same function evaluations, the coefficients for each GMM-PC are found using Weighted Least Squares (WLS). For each element, the initial conditions for the function calls are converted to the uncorrelated reference frame of the element's covariance matrix using the square-root.

$$\boldsymbol{\xi} = \mathbf{S}_i^{-1}(\mathbf{x} - \boldsymbol{\mu}_i), \quad \mathbf{P}_i = \mathbf{S}_i \mathbf{S}_i^T \quad (3.23)$$

The coefficients are then computed using the standard normal equations for WLS with an  $n \times n$  diagonal weight matrix  $\mathbf{W}$ :

$$\mathbf{W}(j, j) = (2\pi)^{-\frac{d}{2}} e^{-\frac{1}{2} \boldsymbol{\xi}_j^T \boldsymbol{\xi}_j} \quad (3.24)$$

The diagonal elements are the PDF value of normalized  $d$ -dimensional input points with respect to a Gaussian distribution with  $\mathbf{0}_d$  mean and a covariance of identity  $\mathbf{I}_{d \times d}$ . To keep the total number of function evaluations fixed, the order of the PCE for all the elements is no longer the same. The multivariate order of each element  $L_i$  is chosen such that  $L_i = \text{floor}(M/2)$  and  $M$  is the number of points  $\boldsymbol{\xi}_i$  that are within a  $3\sigma$  bound for the current GMM element.

The final issue relates to optimally constructing the GMM approximation of the initial Gaussian distribution prior to converting each element into a PCE. In this study, univariate splitting libraries with an odd number of elements are used so that the locations of the means all lie along a single line



in the  $d$ -dimensional space [133]. An initial, optionally lower order, PCE is fit to find the direction with the highest nonlinearity i.e., the direction in which the value of the PCE coefficients decays the slowest. If the absolute values of the coefficients decay rapidly in all directions, there is no reason to switch to a GMM-PC; otherwise. If the GMM-PC is required due to the nonlinearity, the univariate splitting library is applied along the identified direction. The WLS technique can be used to possibly reuse the function evaluations for at least the central element, which is the most similar to the original Gaussian distribution. *A priori* knowledge about the problem can also be used to identify the directions with highest nonlinearity (e.g. the velocity uncertainty for orbit propagation).

### 3.4 Results

Four orbital dynamics test cases that benefit from using GMM-PC are presented in this section. The first two, and the fourth cases are subject to Kelperian motion only: 1) an eccentric Medium Earth Orbit (MEO), 2) a Molniya orbit, and 4) a circular LEO. The third test case is an object in a Geosynchronous Transfer Orbit (GTO) under the influence of perturbations. Cartesian coordinates in the ECI frame are used to express the uncertainties for the first three cases and EE are used for the last case. The univariate splitting library used to generate the initial GMMs is from Chapter 2. The PCE coefficients are computed using LS where the number of function evaluations is twice the number of multivariate coefficients,  $L$ . The final subsection uses the WLS GMM-PC technique to reuse function evaluations.

The RSO uncertainty propagation cases shown here all have post-propagation PDFs that are spread over a large angular portion of the orbit. Such large uncertainty spreads might result in a RSO not being trackable by ground sensors, or in a mislabeling with another RSO. Even if the  $3\sigma$  PDF is too large, knowledge of the mean and other statistical information, which is more accurately available with GMM-PC, only improves the tracking capability. The accurate PDF information can be combined with state of the art tracking and Initial Orbit Determination (IOD) techniques such as Bayesian inference and admissible regions [64], and dynamic sensor steering [63] to regain an accurate estimate of the RSO. Multiple hypothesis filters can then be used to distinguish between the possibly mislabeled RSOs [38]. In reality, targets that have a high importance will have frequent measurements that will ensure the PDF is not large. The SSN is, however, typically not capable of making ad hoc measurements and therefore, other orbit determining assets should be used [102].

### 3.4.1 Medium Earth Orbit

The first test case is a MEO with two-body dynamics propagated for 3 days. The initial state and uncertainty at epoch are found in Tables 3.1 and 3.2. The highest nonlinearity and therefore, the non-convergence is found along the initial velocity direction for the MEO case.

The coefficient values corresponding to the order of univariate polynomials within the multivariate polynomials from Eq. (3.7) for a one element GMM-PC are plotted in Figure 3.4. The square-root of the initial covariance

$a$	$e$	$i$	$\Omega$	$\omega$	$\nu$
24,475km	0.5	0°	0°	0°	0°

Table 3.1: Initial osculating orbit elements for a MEO object

$\sigma_{\mathbf{r}_R}$	$\sigma_{\mathbf{r}_I}$	$\sigma_{\mathbf{r}_C}$	$\sigma_{\mathbf{v}_R}$	$\sigma_{\mathbf{v}_I}$	$\sigma_{\mathbf{v}_C}$
1 km	1 m	1 m	0.1 m/s	0.1 m/s	0.1 m/s

Table 3.2: Initial uncertainty expressed in the RIC coordinate frame for a MEO object

is constructed such that  $\xi_1$  is along the velocity vector  $\mathbf{v}_v$ . Figure 3.4a shows that this case is highly non-linear along the velocity direction because increasing the order of the PCE does not result in a fast reduction in the values of the coefficients. The expansion along the other directions converges, as shown by the reducing coefficient magnitude in Figure 3.4b. Applying a univariate split along  $\xi_1$  reduces the size of the uncertainty and therefore, polynomials of a lower order can be used to achieve convergence.

The initial Gaussian distribution is split into a GMM with up to 9 elements along the  $\mathbf{v}_v$  direction. The sum of  $L$  of all the elements is used as a proxy for the computation cost. The GMM-PC is propagated for the desired flight time of 3 days and is sampled. The  $CvM$  values for the samples in the Radial-Intrack-Crosstrack (RIC) frame with respect to a MC simulation is shown in Figure 3.5 for a one element GMM-PC and a nine element GMM-PC. Using only PCE, which is a one element GMM-PC, does not result in a converged solution with low error for all the univariate directions except for  $\mathbf{v}_I$ . As the number of splits along the velocity direction is increased, a lower

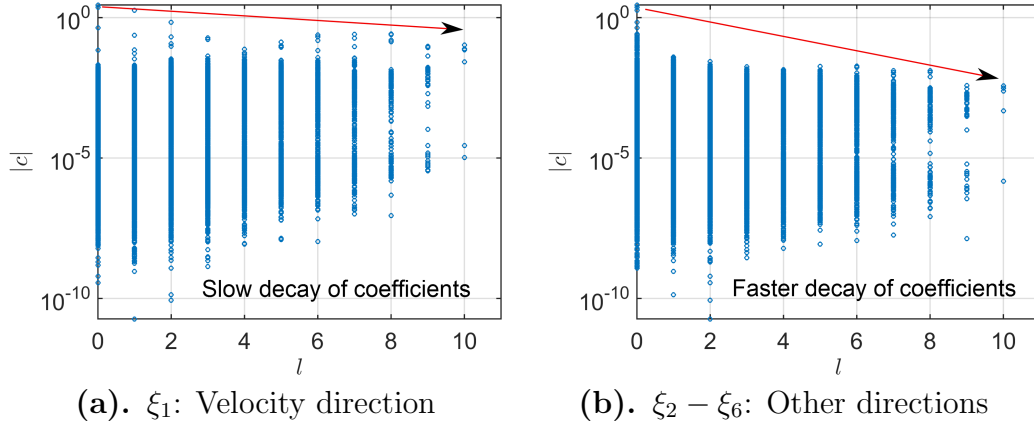


Figure 3.4: Coefficient values for the corresponding univariate polynomial orders for the directions of the IID state for a one element GMM-PC used to propagate uncertainty of an object in MEO. The trend of the maximum values of the coefficients as the order increases is indicated by the arrow.

computation cost results in a more accurate representation of the uncertainty. The norm of the  $CvM$  values in all the position and velocity directions of the RIC frame is found in Figure 3.6.

Figure 3.7 clearly shows the accurate representation of the MC simulation by the GMM-PC. Point clouds have a tendency to exaggerate the visual effect amount of outliers because point density is not easily represented. There are many outliers present in the pure PCE and GMM solutions, especially in the radial direction. The GMM approximation is jagged because it is similar to plotting a circle with only 9 points. Each element approximates a curve with a line, so the semi minor axis of the post-propagation confidence ellipse gets inflated to accommodate for curvature of the true sub-distribution each element is fitting.

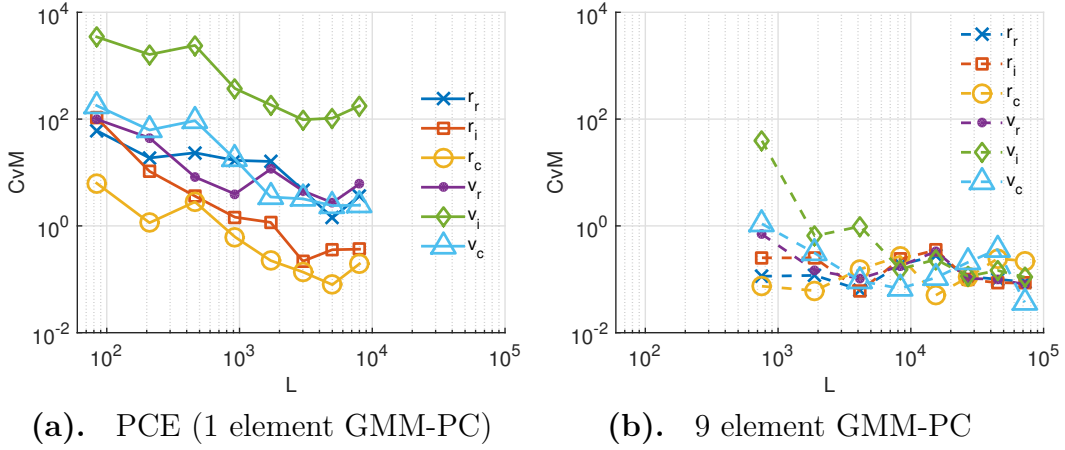


Figure 3.5: Two-sample univariate Cramer-von-Mises metric in the RIC frame for GMM-PC, split along velocity, with respect to an MC simulation of 1,000,000 samples of an object in MEO.  $L$  is the total number of coefficients required per direction for the multivariate polynomial and is analogous to the compute cost ( $2L$ ).

### 3.4.2 Molniya Orbit

The second test case is an orbit from Jones et. al. [75]. The initial state and uncertainty for the Molniya orbit are found in Tables 3.3 and 3.4. The orbit is propagated for a flight time of 10 days to compare the final distributions. This case is known to not converge when Cartesian coordinates are used to represent the uncertainty [75].

$a$	$e$	$i$	$\Omega$	$\omega$	$\nu$
26,562km	0.741	63.4°	90°	-90°	0°

Table 3.3: Initial osculating orbit elements from Jones et. al. [75] for an object in a Molniya orbit

This case is again highly nonlinear along velocity  $\xi_1$ , direction because increasing the order of the PCE does not result in reducing values of the

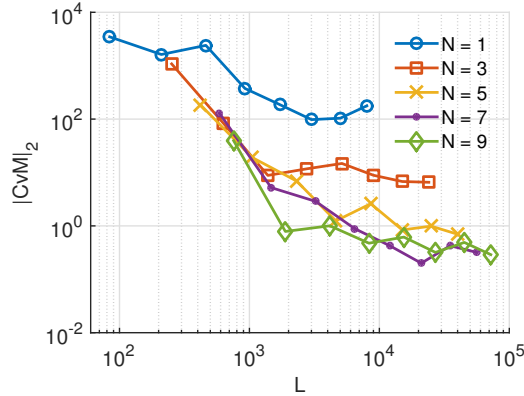


Figure 3.6: Two-sample univariate Cramer-von-Mises metric in the RIC frame for GMM-PC, split along velocity, with respect to an MC simulation of 1,000,000 samples of an object in MEO.  $L$  is the total number of coefficients required per direction for the multivariate polynomial and is analogous to the compute cost ( $2L$ ).

$\sigma_x$	$\sigma_y$	$\sigma_z$	$\sigma_{v_x}$	$\sigma_{v_y}$	$\sigma_{v_z}$
10 m	10 m	10 m	1 m/s	1 m/s	1 m/s

Table 3.4: Initial uncertainty expressed in Cartesian coordinates in the ECI frame for an object in a Molniya orbit

coefficients and the behavior is similar to the behavior in Figure 3.4. The expansion along other directions converges.

The initial Gaussian distribution is split into a GMM with up to 9 elements along the  $\mathbf{v}_v$  direction. The GMM-PC is propagated for the desired flight time of 10 days and is sampled. The  $CvM$  values for the samples in the RIC frame with respect to a MC simulation is shown in Figure 3.8. Using only PCE does not result in a converged solution with low error for all the univariate directions except for  $\mathbf{r}_C$ . As the number of splits along the velocity direction is increased, a lower computation cost results in an increased accuracy.

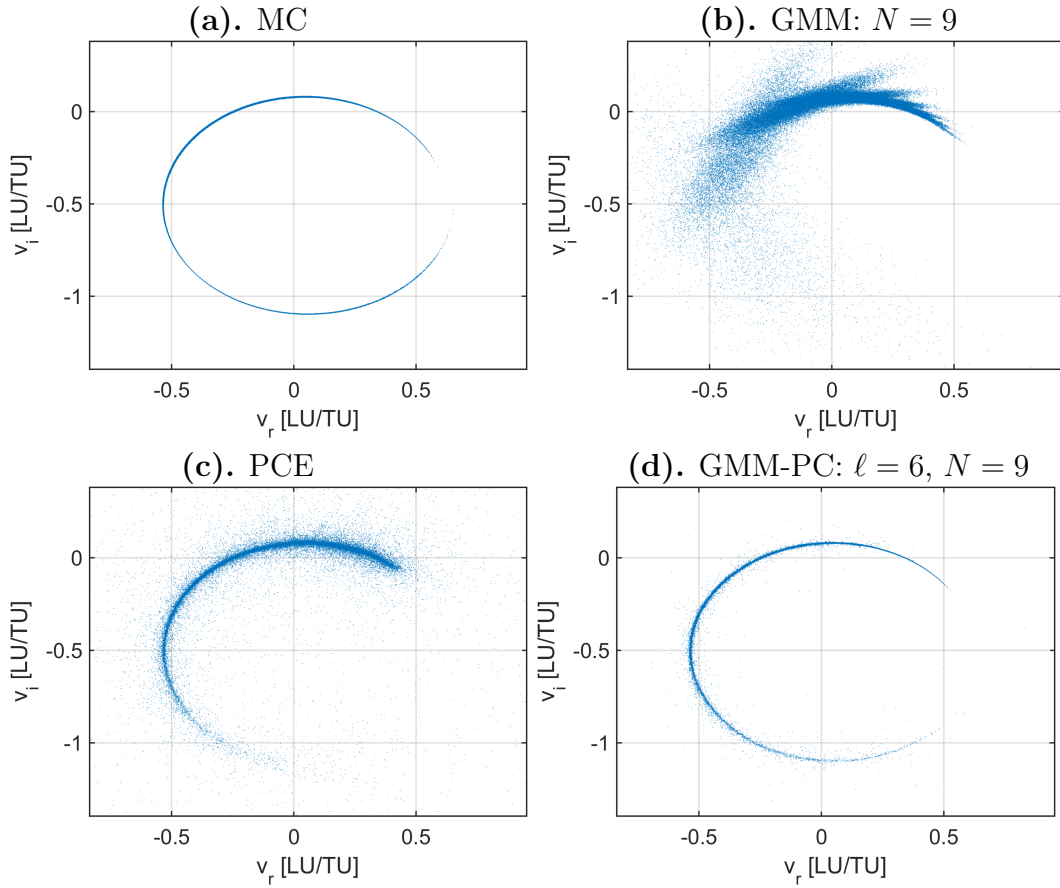


Figure 3.7: 100,000 samples for velocity in the Radial-Intrack plane from the MC, a PCE, and GMM-PC and GMM simulations with an initial split applied along  $\mathbf{v}_v$  for the uncertainty propagation of a MEO object

Because only two-body dynamics are used and the initial uncertainty in the position coordinates is only 10 m, the final distribution is thinly spread along the orbit. Therefore, the resulting distributions of the MC, PCE, GMM, and GMM-PC are shown as histograms in the RIC frame of the MC mean in Figure 3.9. The samples from the 9 element 6<sup>th</sup>-order GMM-PC clearly form a good approximation of the MC histograms in all the univariate directions of the RIC frame. PCE and the pure GMM fail to approximate the tails of the

distributions.

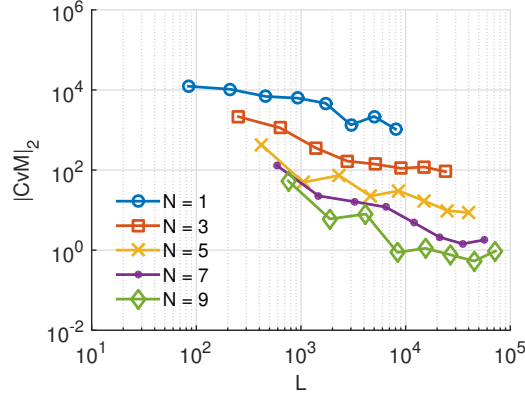


Figure 3.8: Two-sample univariate Cramer-von-Mises metric in the RIC frame for GMM-PC, split along velocity, with respect to an MC simulation of 1,000,000 samples of an object in a Molniya orbit.  $L$  is the total number of coefficients required per direction for the multivariate polynomial and is analogous to the compute cost.

### 3.4.3 Geosynchronous Transfer Orbit

The third test case is a GTO with perturbations due to atmospheric drag, non-spherical Earth of degree and order 8, Solar Radiation Pressure (SRP), and third-body attraction of the Sun and the Moon. The initial state and uncertainty for the GTO are found in Tables 3.5 and 3.6. The highest nonlinearity and therefore, the non-convergent source for the PCE is again the initial velocity uncertainty and the behavior is similar to Figure 3.4.

$a$	$e$	$i$	$\Omega$	$\omega$	$\nu$
24,475km	0.731	7°	250°	8°	0°

Table 3.5: Initial osculating orbit elements for a GTO object



$\sigma_x$	$\sigma_y$	$\sigma_z$	$\sigma_{v_x}$	$\sigma_{v_y}$	$\sigma_{v_z}$
5 m	5 m	5 m	0.5 m/s	0.5 m/s	0.5 m/s

Table 3.6: Initial uncertainty expressed in Cartesian coordinates in the ECI frame for a GTO object

The  $CvM$  values for the samples in the RIC frame with respect to the MC simulation is shown in Figure 3.10. Using a one element GMM-PC, does not result in a converged solution in the  $\mathbf{v}_I$  direction. As the number of splits along the velocity direction is increased, a lower computation cost results yields a higher accuracy. The behavior of the point clouds and the histograms are similar to the MEO and Molniya cases, respectively and are therefore not presented.

#### 3.4.4 Low Earth Orbit

The final test case uses orbit elements to express the state uncertainty. The initial state and state uncertainty in EE are taken from Horwood et al. [68]. The initial conditions are found in Tables 3.7 and 3.8. The time of flight, however, is increased to 65.6 hours. The increased flight time makes the uncertainty distribution of the SO wrap around the whole orbit. The major source of the final uncertainty is the initial semi-major axis uncertainty [68] and the plot of the coefficients is similar the velocity direction of the previous cases as seen in Figure 3.4. Splitting the initial distribution along the semi-major axis direction ensures that the angular spread in the true anomaly of each element is less than  $2\pi$ . The orbit uncertainty occupying the full range from 0 to  $2\pi$  in the true anomaly is a common occurrence for debris field propagation cases .

The GMM-PC fitting is carried out in EE but the  $CvM$  metric in Figure 3.11 is computed in Cartesian coordinates by converting the EE GMM-PC samples to Cartesian coordinates. With number of elements  $N > 1$ , the GMM-PC solution converges because the uncertainty in the true anomaly corresponding to each element is contained within the  $[0, 2\pi]$  range.

$a$	$h$	$k$	$p$	$q$	$l$
6,980km	0.0	0.0	0.0	0.0	0.0°

Table 3.7: Initial osculating orbit expressed in Equinoctial Elements for a LEO object [68]

$\sigma_a$	$\sigma_h$	$\sigma_k$	$\sigma_p$	$\sigma_q$	$\sigma_l$
20 km	$10^{-3}$	$10^{-3}$	$10^{-3}$	$10^{-3}$	0.1°

Table 3.8: Initial uncertainty expressed in Equinoctial Elements for a LEO object [68]

### 3.4.5 Weighted Least Squares Technique

Apart from the naive GMM-PC approach, the WLS approach is used for the first three orbit uncertainty propagation cases, which use Cartesian Coordinates. The number of function evaluations are held constant and the number of elements in the GMM-PC is increased and the PCE order of each GMM-PC element is adapted based on the number of evaluation nodes available in its  $3\sigma$  range. The WLS approach is suboptimal compared to the standard LS method because the function is not reevaluated for each element and the order decreases further from the central element. However, the accuracy is better than the standard PCE approach as seen in Figure 3.12.

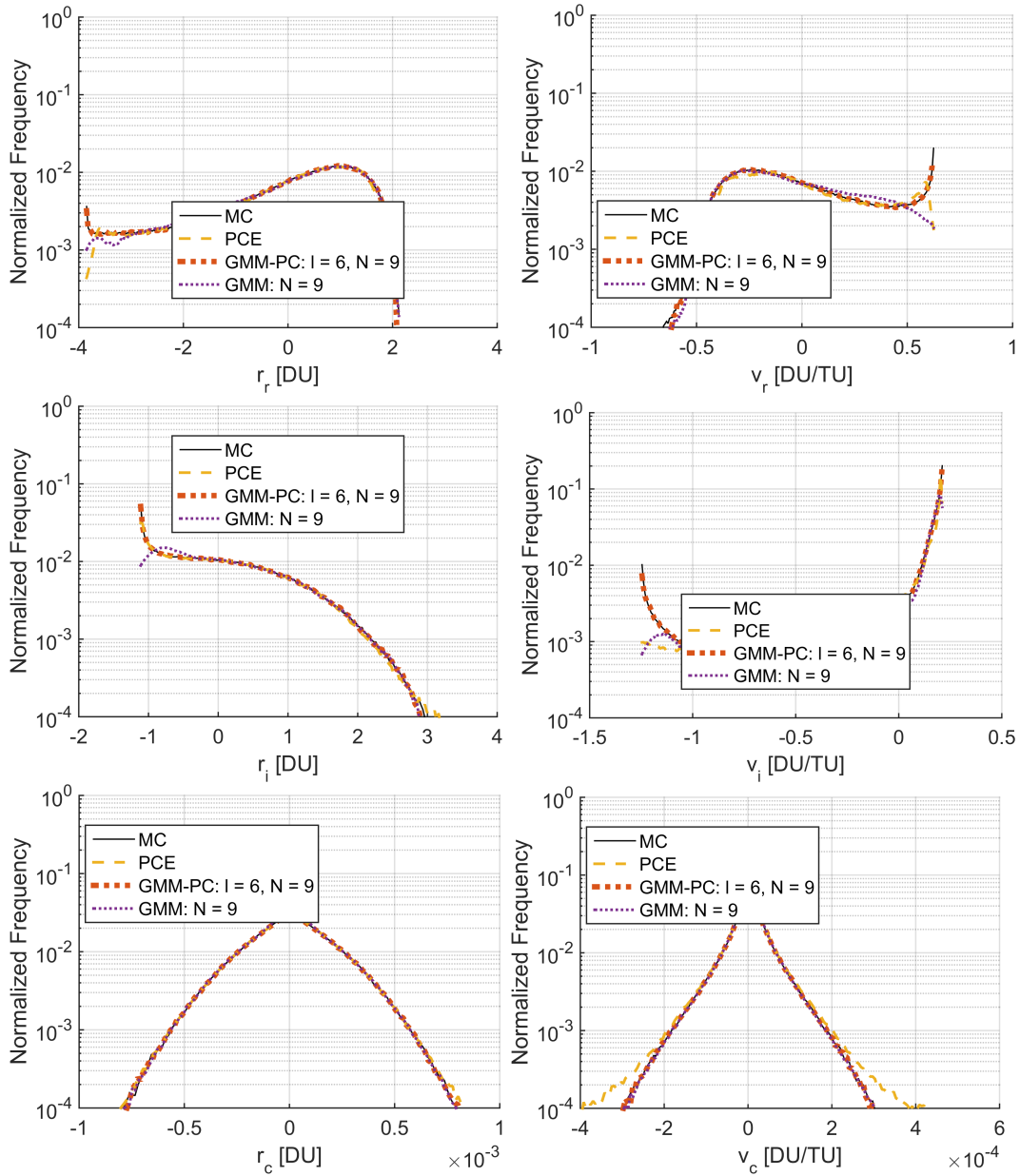


Figure 3.9: Histograms in the RIC frame of 1,000,000 samples from the MC , a PCE, and a GMM-PC and GMM with an initial split applied along  $\mathbf{v}_v$  for the uncertainty propagation of an object in a Molniya orbit

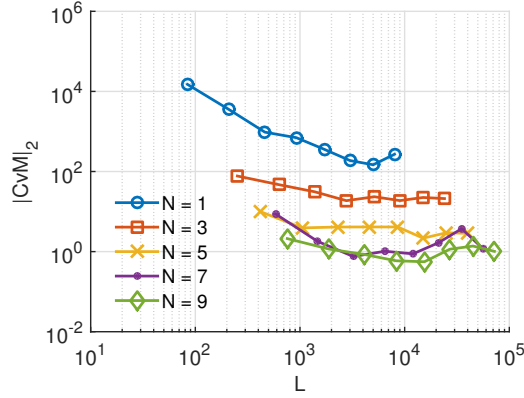


Figure 3.10: Two-sample univariate Cramer-von-Mises metric for PCE and GMM-PC, split along velocity, with respect to an MC simulation of 1,000,000 samples.  $L$  is the total number of coefficients required per direction for the multivariate polynomial and is analogous to the compute cost.

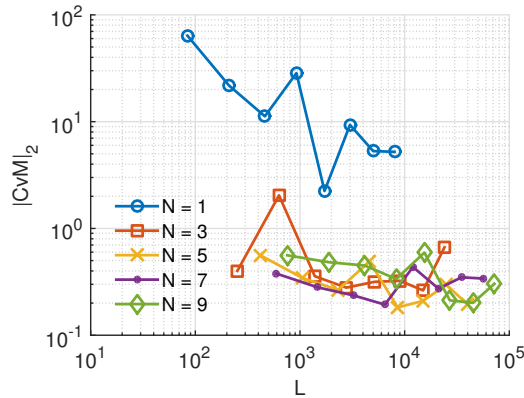


Figure 3.11: Two-sample univariate Cramer-von-Mises metric for PCE and GMM-PC, split along the semi-major axis, with respect to an MC simulation of 1,000,000 samples.

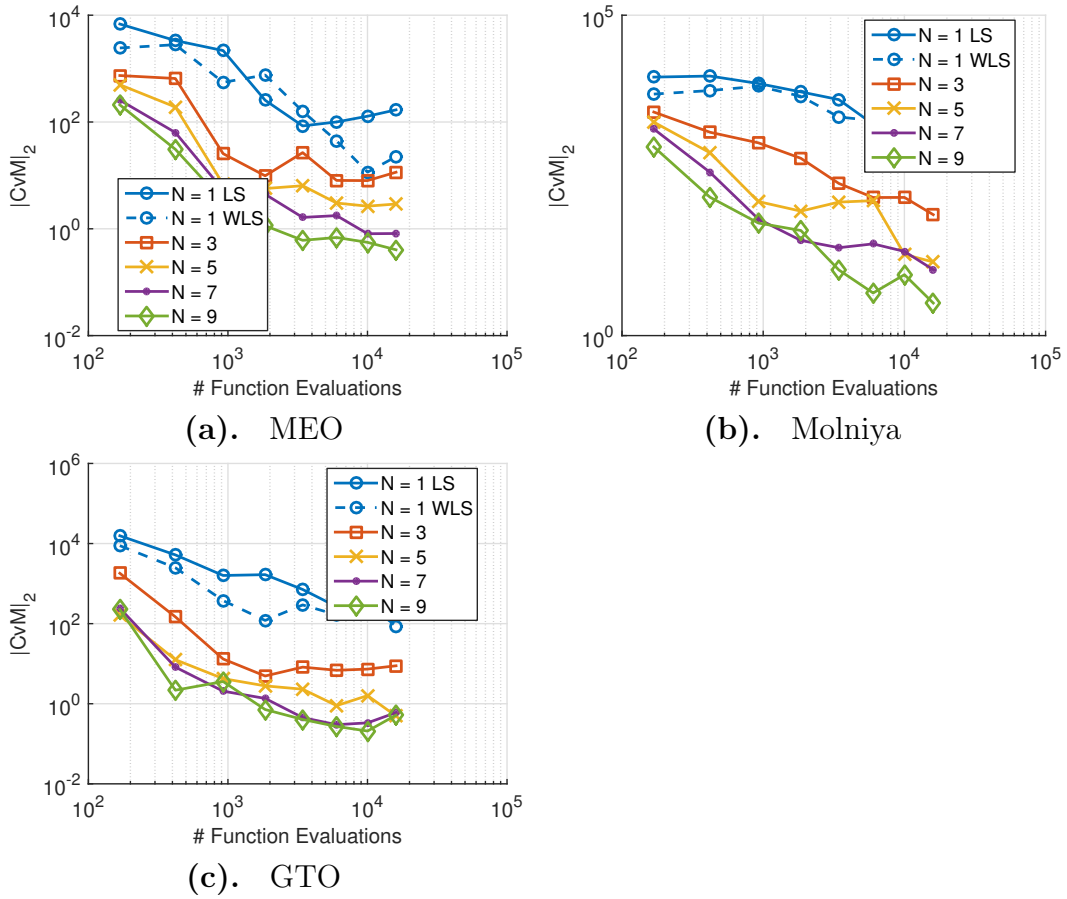


Figure 3.12: Two-sample univariate Cramer-von-Mises metric for PCE and Weighted Least Squares GMM-PC, split along velocity, with respect to an MC simulation of 1,000,000 samples.

### 3.5 Conclusion

A Polynomial Chaos Expansion forms a surrogate model for uncertainty propagation through a nonlinear function, which is computationally more efficient to sample from, compared to a full Monte Carlo simulation. The performance of the PCE method depends on the nonlinearity of the function and the size of the initial uncertainty. For the four cases of uncertainty propagation shown here, increasing the order of the PCE results in slow convergence and a large computation cost. Splitting the initial Gaussian distribution into a GMM with the means of the weighted Gaussian distributions along the most influential direction (initial velocity vector for the Cartesian Coordinates cases and semi-major axis for the EE case presented here) reduces the uncertainty along that direction. Therefore, a more accurate description of the propagated uncertainty is found at a lower computation cost, compared to simply increasing the order of the PCE. However, the GMM-PC technique is mostly recommended for scenarios where a PCE does not converge, as is the case for orbit problems with large propagation times, large initial uncertainties, and highly eccentric orbits. The weighted least squares method for computing the coefficients of the PCE is capable of reusing the function evaluations so that accuracy of the GMM-PC method can be increased without increasing the computational load.

A limitation of the current implementation of the PCE and GMM-PC techniques is that additional uncertainty cannot easily be introduced into the propagation problem. Therefore, the current implementation is not amenable to adding process noise or modeling unknown maneuvers during the predic-

tion phase in RSO tracking. An unknown maneuver could be modeled by appending the maneuver to the state vector and increasing the dimension of the problem. With PCE, different types of PDFs can be easily combined as a tensor product if the distributions are independent. Therefore, even a uniform distribution can be used to model the maneuver. The types of polynomials are proven to exponentially converge for selected initial distributions, i.e. Hermite for Gaussian distributions [137]. Using a different type of polynomial should only result in a slower convergence. Therefore, for cases with low to moderate process noise, the GMM-PCE using Hermite polynomials might still converge, albeit at a slower rate.

Comparing uncertainty propagation to FEM makes the GMM-PC method an *hp*-refinement method where the order of the PCE and the number of splits are independently adapted. The GMM-PC technique is easier than the ME-gPC to implement because the PDF of each element remains Gaussian. Therefore, Hermite polynomials are used instead of developing a gPC framework to find orthogonal multivariate polynomials with respect to arbitrary PDFs. In ME-gPC, errors may creep into the computation of the orthogonal polynomials, while the error in the GMM-PC technique arises due to the approximation of the initial PDF with a finite number of GMM elements. Hence, the best possible performance of the GMM-PC has an upper bound which is equivalent to the approximation accuracy of the GMM splitting library. For extending a PCE implementation to the GMM-PC, only the weights, means, and standard deviations of a univariate splitting library are required, which are previously archived and stored in a tabular manner in the work presented in Chapter 2.

## Chapter 4

# Monte Carlo Collision Probability on the Graphics Processing Unit

Chapters 2 and 3 show the propagation of uncertainty in Phase I of Figure 1.7 on page 12. The focus of this chapter is on computing the collision probability in Phase II. Fast and accurate collision probability computations are essential for protecting space assets. Monte Carlo (MC) simulation is the most accurate but computationally intensive method. A Graphics Processing Unit (GPU) is used to parallelize the computation and reduce the overall runtime. Using MC techniques to compute the collision probability is common in literature because it is the benchmark. An optimized implementation on the GPU, however, is a challenging problem and is the novel contribution of this chapter. The MC simulation takes samples from the uncertainty distributions of the Resident Space Objects (RSOs) at any time during a time window of interest and outputs the separation at closest approach. Therefore, any uncertainty propagation method may be used and the collision probability is automatically computed as a function of RSO collision radii. Integration using a fixed time step and a quartic interpolation after every Runge Kutta step ensures that no close approaches are missed. Two orders of magnitude speedups over a serial CPU implementation are shown, and speedups improve



moderately with higher fidelity dynamics. The tool makes the MC approach tractable on a single workstation, and can be used for a final product or for verifying surrogate or analytical collision probability methods.

## 4.1 Introduction

The collision probability computation of two RSOs depends on the size, shape, trajectory, and the uncertainty associated with all the aforementioned parameters. A common assumption that makes the problem more tractable is that both RSOs are spheres and therefore, using a 3 degree of freedom (DOF) simulation is sufficient. The collision probability is then the probability that the two RSOs have a separation less than or equal to the combined radii at any point in time. Therefore, an integral part of the collision probability computation is parameterizing the intersection of the time varying 3D position probability distribution functions (PDFs) of two RSOs over a specified time window.

The most accurate, but computationally intensive method of computing the intersection of the PDFs of the two RSOs is to use Monte Carlo (MC) simulations. MC simulations require millions of samples from the PDFs of the primary and secondary RSOs. These samples are propagated over the time window of interest and the collisions are recorded. MC simulations are very general and no assumptions are required, but they can be very inefficient. Computationally less intensive techniques exist when a few assumptions are made about the conjunction. If the RSOs are spherical with a Gaussian uncertainty distribution in Cartesian coordinates, and the RSOs have independent

PDFs, then the collision probability is calculated with a low computation cost [40].

The collision probability is computed in this chapter using the accurate MC method on the GPU. The speedups due to GPU parallelization for trajectory propagation have been demonstrated for a PF [117] and for trajectory propagation [98, 25]. Each sample in the MC simulation is 12 dimensional and consists of one six-state from the PDF of the primary RSO and one six-state from the PDF of the secondary RSO. The output of the simulation is the closest approach between a primary and a secondary RSO sample during the time period of interest. Any PDF representation of the RSOs, such as PCE from Chapter 3 orbit elements, may be used. The GPU tool only requires the samples of both RSOs at the same instant in time within the time window of interest. The MC simulation is then an embarrassingly parallel problem. The massive parallelism present in the current problem is well-suited for the GPU architecture. MC collision probability computation has been presented by Alfano [16] and the effect of perturbations has been studied by Sabol et al. [113]. Brown et al. [31] used GPUs to compute conjunctions in a space catalog, but not for a collision probability computation. The advantage of GPU based parallelism lies in its single user capability without the need for expensive CPU clusters. Attractive compute speeds are realized on the GPU without sacrificing accuracy. However, GPU programming requires more development time and is less portable when compared to CPU programming due to the more complicated memory hierarchy [127]. The main contribution of the MC implementation presented in this chapter is the optimization for the

GPU.

Computing the collision probability with the MC method is first presented in Section 4.2. An emphasis is placed on the GPU implementation of a new interpolation method used to find the closest separation between the RSO trajectories. In Section 4.3, the interpolation is validated and tested, the speedups gained by using the GPU are presented, and the effect of perturbations on two conjunction scenarios is shown. Finally conclusions are drawn from these results in Section 4.4.

## 4.2 Monte Carlo Collision Probability

The collision probability is considered here to be the probability that two RSOs will collide over a given time period of interest  $t \in [t_a, t_b]$ . To make the computation tractable, the two RSOs are assumed to be spherical. Therefore, a 3 degree of freedom (DOF) simulation suffices. The probability of collision is the probability that the two RSOs have a separation less than or equal to their combined hard body radius,  $R_{min}$  at any instant of time within the time window of interest. It is assumed that the 6 dimensional PDFs of state uncertainties of both RSOs are available at the same time  $t_*$  within the time period of interest, i.e.  $t_* \in [t_a, t_b]$ , and that these PDFs can be sampled. Therefore, the PDF can be represented by any technique: mean and covariance matrix, PCE, GMMs, etc. Orbit elements can also be used to express the PDF as long as the samples are converted Cartesian coordinates.

The probability is approximated with MC techniques by generating

millions of point pairs where the number of samples is denoted as  $N_{MC}$ . One six-state from the primary and one six-state from the secondary RSO's PDFs constitute a point pair. The output is a vector containing all the minimum separations between each point pair  $r_{min}$ . A counter,  $n_c$  is first initialized at 0. Each point pair is propagated over the time window,  $t \in [t_a, t_b]$ , and if the separation at any time is less than or equal to  $R_{min}$ ,  $n_c(r_{min} \leq R_{min})$  is incremented by 1. Finally, the probability of collision is:

$$P_{c_{MC}}(R_{min}) = n_c(R_{min})/N_{MC} \quad (4.1)$$

A simple loop over the vector of minimum separations of all the pairs is required to recompute the probability of collision when  $R_{min}$  is changed, which is a negligible computational effort compared to computing the close approaches. The generality of the MC method with respect to changing  $R_{min}$  is beneficial because the change in the collision risk as a function of the hardbody radius is readily available. When other collision probability methods such as linear collision are used, the entire simulation is repeated.

There are two options for generating the  $N_{MC}$  pairs:

- One-on-one:  $N_{MC}$  samples from the primary RSO PDF and  $N_{MC}$  samples from the secondary RSO PDF
- All-on-all:  $\sqrt{N_{MC}}$  samples from the primary RSO PDF and  $\sqrt{N_{MC}}$  samples from the secondary RSO PDF

A diagram emphasizing the difference between the one-on-one and the all-on-all approaches is shown in Figure 4.1. The one-on-one approach is preferred

on the GPU and is the only method implemented here. None of the samples are repeated and are completely independent for the one-on-one approach. In the second approach, a sample from one of the RSOs is compared to all the samples from the other RSO. The first method is more accurate because an outlier only affects one pair. In the second method, however, the same outlier affects  $\sqrt{N_{MC}}$  pairs. The all-on-all approach has a lower computational cost and a larger memory footprint in case the ephemerides of the samples over the time window of interest are stored. The cost of this exchange is an increase in the complexity of the code and the loss of statistical rigor. If an MC simulation is used to capture the evolution of the PDF prior to the time window of interest, the all-on-all approach can be beneficial for reducing the computation time by drastically reducing the number of samples required for propagation. The all-on-all approach can be used with the present GPU implementation if the combinations are prepared by another tool and provided as a one-on-one input to the GPU.

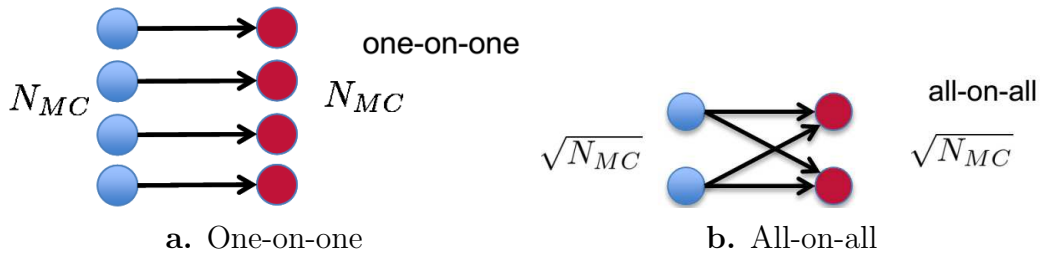


Figure 4.1: Samples from the primary (blue) and secondary (red) RSOs required for a MC simulation with  $N_{MC}$  samples in the one-on-one and the all-on-all approaches

Explicit Runge Kutta (RK) integration is used to propagate the trajectories. Due to the possibility of high relative velocities during conjunctions,

close approaches are easily missed. Primary and secondary RSOs have relative velocities greater than 1 km/s, however,  $R_{min}$  is generally a few meters. Generating dense ephemerides using small integrator step sizes is a computationally inefficient method of finding the close approaches. A more efficient approach is to interpolate on the separation between the two RSOs  $r$ . The interpolation technique and the MC simulation implementation on the GPU are now presented.

#### 4.2.1 Interpolation

Each step of the RK integration takes the state of the RSO from  $t = t_k$  to  $t = t_{k+1}$ , where the time step size is  $\Delta t = t_{k+1} - t_k$ . The relative position, velocity, and acceleration for the two RSOs is defined as:

$$\mathbf{r}(t) = \mathbf{r}_S(t) - \mathbf{r}_P(t) \quad (4.2a)$$

$$\dot{\mathbf{r}}(t) = \dot{\mathbf{r}}_S(t) - \dot{\mathbf{r}}_P(t) \quad (4.2b)$$

$$\ddot{\mathbf{r}}(t) = \ddot{\mathbf{r}}_S(t) - \ddot{\mathbf{r}}_P(t) \quad (4.2c)$$

The aim is to find the minimum of the separation  $r(t)$  with  $t \in [t_k, t_{k+1}]$  where:

$$r = \|\mathbf{r}\|_2 \quad (4.3)$$

Instead of iteratively finding the minimum by using methods such as bisection, the minimum separation is computed using interpolation. The interpolation methodology is largely based on the method presented in Alfano's paper [16] for finding close approaches. Due to the GPU implementation, a different interpolating polynomial is used here and the interpolation is carried out after each RK step.

At each RK step, the position, velocity, and acceleration at  $t_k$  are known. However, only the position and velocity are known at  $t_{k+1}$ . The acceleration could also be computed at the end of the step and used to initialize the next RK step. The transfer of the acceleration, however, requires additional expensive bookkeeping and memory transactions (on the GPU). The five known quantities before and after the RK step are sufficient for a quartic interpolating polynomial for the separation.

Instead of constructing the polynomial for  $r(t)$  with  $t \in [t_k, t_{k+1}]$ , a normalized polynomial for  $r^2(\tau)$  is constructed with  $\tau \in [0, 1]$ . The transformation between  $t$  and  $\tau$  is:

$$\tau(t) = \frac{t - t_k}{t_{k+1} - t_k} \quad (4.4a)$$

$$t(\tau) = \tau(t_{k+1} - t_k) + t_k \quad (4.4b)$$

Due to the linear mapping between  $t$  and  $\tau$ , the functions of  $t$  and their derivatives are:

$$g(\tau) = f(\tau\Delta t + t_k) \quad (4.5a)$$

$$g'(\tau) = \frac{dg(\tau)}{d\tau} = \frac{df(\tau\Delta t + t_k)}{dt} \Delta t = \dot{f}(\tau\Delta t + t_k) \Delta t \quad (4.5b)$$

$$g''(\tau) = \frac{d^2g(\tau)}{d\tau^2} = \frac{d^2f(\tau\Delta t + t_k)}{dt^2} (\Delta t)^2 = \ddot{f}(\tau\Delta t + t_k) (\Delta t)^2 \quad (4.5c)$$

The linear mapping into the normalized domain is also valid for  $\Delta t < 0$ , i.e. integration backwards in time. Values for the square of the separation and its

derivatives are [16]:

$$r^2 = \mathbf{r}^T \mathbf{r} \quad (4.6a)$$

$$\frac{d(r^2)}{dt} = 2 (\dot{\mathbf{r}}^T \mathbf{r}) \quad (4.6b)$$

$$\frac{d^2(r^2)}{dt^2} = 2 (\ddot{\mathbf{r}}^T \mathbf{r} + \dot{\mathbf{r}}^T \dot{\mathbf{r}}) \quad (4.6c)$$

Equation (4.5) is used to normalize the function and derivatives to get  $d(r^2)/d\tau$  and  $d^2(r^2)/d\tau^2$  with  $\tau \in [0, 1]$ .

A new normalized interpolating quartic polynomial for a function  $y(\tau)$  with the domain  $\tau \in [0, 1]$  is now constructed. The known quantities are  $y(0)$ ,  $y'(0)$ ,  $y''(0)$ ,  $y(1)$ , and  $y'(1)$ , which is the information about the position that is readily available from an RK step. A diagram of an arbitrary function and the boundary conditions is shown in Figure 4.2. The quartic interpolating

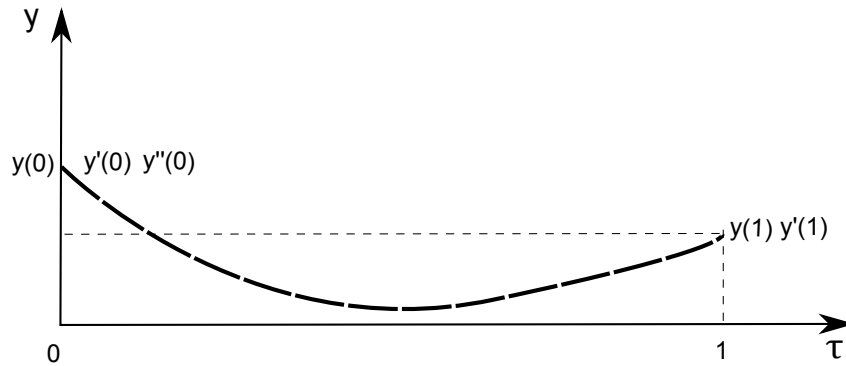


Figure 4.2: Function  $y(\tau)$  with the information available for computing a quartic interpolating polynomial



polynomial that satisfies all the boundary conditions is:

$$y_*(t) = a_4\tau^4 + a_3\tau^3 + a_2\tau^2 + a_1\tau + a_0 \quad (4.7a)$$

$$a_4 = 3y(0) - 3y(1) + 2y'(0) + y'(1) + y''(0)/2 \quad (4.7b)$$

$$a_3 = 4y(1) - 4y(0) - 3y'(0) - y'(1) - y''(0) \quad (4.7c)$$

$$a_2 = y''(0)/2 \quad (4.7d)$$

$$a_1 = y'(0) \quad (4.7e)$$

$$a_0 = y(0) \quad (4.7f)$$

The polynomial is similar to a Hermite interpolation polynomial, however it is one-sided since more information about the original function is used at  $\tau = 0$  compared to  $\tau = 1$ . The performance of the quartic interpolation polynomial  $y_*(\tau)$  from Eq. (4.7) on a quintic polynomial  $y(\tau)$  is shown in Figure 4.3. The error at  $y(0)$ ,  $y(1)$ ,  $y'(0)$ , and  $y'(1)$  are all 0 because of the boundary conditions imposed on  $y_*(\tau)$ . The boundary conditions are more accurate at  $\tau = 0$  than at  $\tau = 1$  and therefore, the approximation error is usually higher in the range  $\tau \in [0.5, 1]$  than in  $\tau \in [0, 0.5]$ . A similar behavior is seen in the approximation of  $y'(\tau)$  by  $y'_*(\tau)$ , however, the error is usually 0 somewhere in  $\tau \in (0, 1)$  and typically when  $\tau > 0.5$ .

Using Eq. (4.7), a quartic interpolation of  $r^2(\tau)$  is constructed using the boundary conditions from Eqs. (4.2-4.6) and the normalizing conditions of Equation (4.5). Finding the closest approach between a time step, i.e.  $\min_{t \in [t_k, t_{k+1}]} r(t)$ , is analogous to finding the minimum of  $r^2(\tau)$  in the range  $\tau = 0$  to  $\tau = 1$ , i.e.  $\min_{\tau \in [0, 1]} r^2(\tau)$ . The quartic interpolation polynomial for

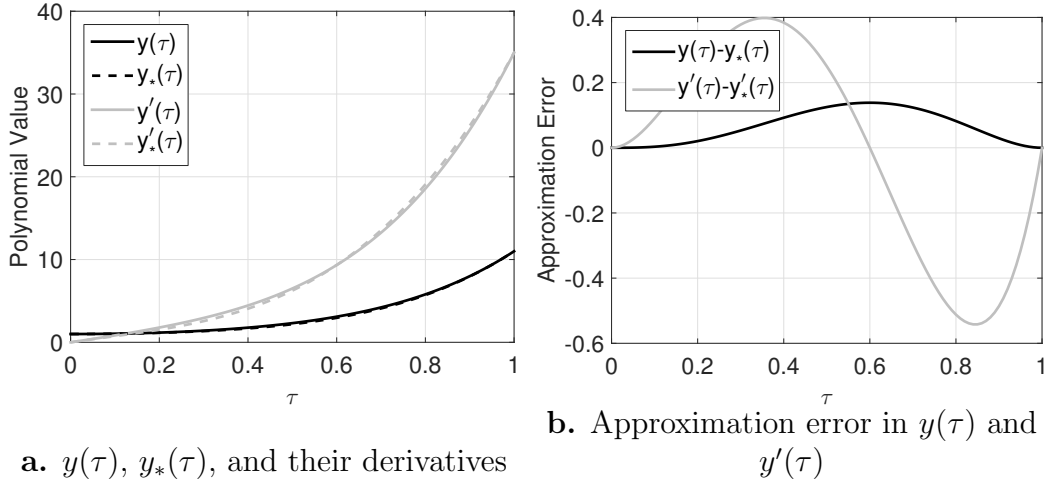


Figure 4.3: Approximating the quintic polynomial  $y(\tau) = 4\tau^5 + \tau^4 + \tau^3 + 4\tau^2 + 1$  by the quartic polynomial  $y_*(\tau) = 9\tau^4 - 3\tau^3 + 4\tau^2 + 1$  in  $\tau \in [0, 1]$

$r^2(\tau)$  is denoted  $r_*^2(\tau)$ . The minimum of  $r^2(\tau)$  occurs when the first derivative is 0 and the second derivative is positive. Therefore, the roots of the cubic polynomial for  $(r_*^2)'$  are computed. Only real roots in the range  $\tau_{root} \in [0, 1]$  are possible candidates because the polynomial is constructed to only approximate the true  $r^2(\tau)$  in the range  $\tau \in [0, 1]$ . According to Alfano [16] there is no real root for the cubic polynomial  $y(\tau) = a_3\tau^3 + a_2\tau^2 + a_1\tau + a_0$  when:

$$\min(a_1, a_1 + a_2, a_1 + a_2 + a_3) > -a_0, \quad a_0 > 0 \quad (4.8a)$$

$$\max(a_1, a_1 + a_2, a_1 + a_2 + a_3) < -a_0, \quad a_0 < 0 \quad (4.8b)$$

For cubic polynomials, there is a possibility of having only one real root and a possibility of having three real roots. For each real root in  $\tau_{RR} \in [0, 1]$ , the quadratic polynomial for  $(r_*^2)''$  can be computed to check if it is a maximum or a minimum. Another option is to simply compute  $r_*^2(\tau_{RR})$  at all the roots and compare them with boundary condition on  $r^2(\tau)$  to determine if it is indeed a

minimum. The latter option is used in this chapter.

Once the roots are found, there are two options for computing  $r_{min}$ . The obvious option is to evaluate the previously constructed quartic polynomial  $r_*^2(\tau)$  at  $\tau = \tau_{RR}$  and then take the square-root. The more accurate option, which is used here is to construct three additional quartic interpolating polynomials  $x_*(\tau)$ ,  $y_*(\tau)$ , and  $z_*(\tau)$  where:

$$x = x_S - x_P \tag{4.9a}$$

$$y = y_S - y_P \tag{4.9b}$$

$$z = z_S - z_P \tag{4.9c}$$

Finally the value for  $r_{min}$  is:

$$r_{min} \approx \sqrt{(x_*(\tau_{RR}))^2 + (y_*(\tau_{RR}))^2 + (z_*(\tau_{RR}))^2} \tag{4.10}$$

The entire interpolation procedure is outlined in Algorithm 1.

### 4.2.2 GPU Implementation

A GPU has hundreds of processors that run simultaneously. Each processor is weak compared to a single CPU, but as a whole, the GPU has the capacity for a significant increase in floating-point operations per second (FLOPS). Each GPU processor must carry out the same instructions in order to achieve noticeable speedups due to the architecture. Memory transactions are also more expensive than on CPUs and need to be thoroughly optimized. Therefore, some design choices have to be made differently for a code that efficiently runs on the GPU. In order to maintain the same computation load

---

**Algorithm 1** Interpolation over 1 RK step

---

- 1: **Input:**  $\mathbf{r}_P(t_k), \dot{\mathbf{r}}_P(t_k), \mathbf{r}_S(t_k), \dot{\mathbf{r}}_S(t_k), \Delta t$
  - 2: **Compute**  $\mathbf{r}_P(t_{k+1}), \dot{\mathbf{r}}_P(t_{k+1}), \ddot{\mathbf{r}}_P(t_k), \mathbf{r}_S(t_{k+1}), \dot{\mathbf{r}}_S(t_{k+1}), \ddot{\mathbf{r}}_S(t_k), t_{k+1}$  using RK step
  - 3: **Compute**  $r^2(t_k), \left. \frac{d(r^2)}{dt} \right|_{t_k}, \left. \frac{d^2(r^2)}{dt^2} \right|_{t_k}, r^2(t_{k+1}), \left. \frac{d(r^2)}{dt} \right|_{t_{k+1}}$  using Equation (4.6)
  - 4: **Compute**  $r_{min}^2 = \min(r^2(t_k), r^2(t_{k+1}))$
  - 5: **Normalize**  $r^2(\tau = 0), \left. \frac{d(r^2)}{d\tau} \right|_0, \left. \frac{d^2(r^2)}{d\tau^2} \right|_0, r^2(\tau = 1), \left. \frac{d(r^2)}{d\tau} \right|_1$  using Equations (4.4) and (4.5)
  - 6: **Compute** Normalized quartic interpolating polynomial  $r_*^2(\tau)$  with  $\tau \in [0, 1]$  using Equation (4.7)
  - 7: **Compute** Cubic polynomial  $\frac{d(r_*^2)}{d\tau}$  (i.e. differentiate  $r_*^2(\tau)$ )
  - 8: **if** Real root exists in  $\tau \in [0, 1]$  (Equation (4.8)) **then**
  - 9:     **Compute** Number of real roots  $N_{RR}$ :  $N_{RR} = 1$  or  $N_{RR} = 3$
  - 10:    **Compute** Real roots:  $\tau_{RR_i}, i = 1, \dots, N_{RR}$
  - 11:    **Compute** Quartic polynomials for  $x_*, y_*$ , and  $z_*$  using Equations (4.5), (4.7), and (4.9)
  - 12:    **for**  $i \leftarrow 1, N_{RR}$  **do**
  - 13:      **if**  $0 < \tau_{RR_i} < 1$  **then**
  - 14:        **Compute**  $r^2(\tau_{RR_i})$  using Equation (4.10)
  - 15:         $r_{min}^2 = \min(r_{min}^2, r^2(\tau_{RR_i}))$
  - 16: **Compute**  $r_{min} = \sqrt{r_{min}^2}$
  - 17: **Output:**  $\mathbf{r}_P(t_{k+1}), \dot{\mathbf{r}}_P(t_{k+1}), \mathbf{r}_S(t_{k+1}), \dot{\mathbf{r}}_S(t_{k+1}), t_{k+1}, r_{min}$
-

for each of the GPU threads and to minimize the memory transactions, the following design choices are implemented:

1. Each thread propagates its trajectory over the entire time window  $t \in [t_a, t_b]$  and only outputs the closest separation  $r_{min}$ , i.e.  $\min_{t \in [t_a, t_b]} r(t)$
2. A fixed step RK4 integration is used with the same time step for all samples
3. The one-on-one approach to PDF sampling is used

The first choice ensures that all the samples are propagated from  $t_a$  to  $t_b$  and the samples from the PDFs of both RSOs can be provided at any time  $t_a \geq t_* \leq t_b$ . An added benefit of searching over the entire space and only returning the closet separation is that the combined hard body radius  $R_{min}$  is not a necessary input. Hence, the probability of collision is easily computed for various values of  $R_{min}$  in a small post processing step as shown in Figure 4.4. On a CPU implementation, the computation time could be reduced by supplying the algorithm with  $R_{min}$  and moving to the next sample as soon as  $r(t) < R_{min}$ . Propagating over the entire time window also ensures that multiple close approaches are easily considered.

Using a fixed step integration ensures that the same number of function evaluations are carried out for all samples. For a GPU employing millions of threads, the wall clock time for the entire batch is determined by the thread with the highest load. Therefore, if the MC is performed with a single batch there is no advantage of some threads finishing sooner than others. The fixed

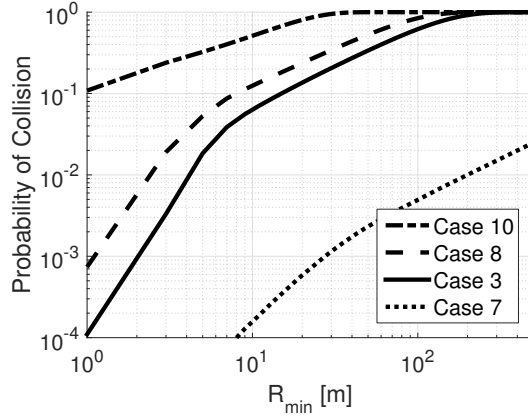


Figure 4.4: Effect of changing the  $R_{min}$  on the probability of collision for cases from Alfano’s paper [17]

step integration is also essential for the interpolation technique as the RK steps for both RSOs should be identical in time. A sample is considered to be a 12 dimensional state where:

$$\mathbf{X} = [\mathbf{x}_P^T, \mathbf{x}_S^T]^T \quad (4.11)$$

The time step size for the integration depends on the step sizes (dictated by the dynamics) required for the primary and secondary RSOs, and the step size necessary for the accurate interpolation. The optimal step size is the smallest of the three candidate sizes.

Finally, the one-on-one approach is used because it is more exact and the samples can be discarded without storing any ephemerides. It should be noted that the all-on-all approach can always be set up as the one-on-one approach by appropriately arranging the samples.

Generating the samples is handled by the CPU and therefore, the GPU only requires the pairs. The benefit of separating the sampling algorithm from

the GPU MC routine is that any surrogate model can be used to generate the samples of the two RSOs without changing the GPU implementation.

For RSOs in Earth orbit it is typical to take into account a static gravity field model, atmospheric drag, third body attraction of the sun and the moon, and solar radiation pressure (SRP). Therefore, the acceleration model for the RSO investigated in this study is:

$$\mathbf{a}_{Full} = \mathbf{a}_{2Body} + \mathbf{a}_{J2} + \mathbf{a}_{mascon} + \mathbf{a}_{Drag} + \mathbf{a}_{Sun3B} + \mathbf{a}_{Moon3B} + \mathbf{a}_{SRP} \quad (4.12)$$

Mass concentrations (mascons) are an alternate method to Spherical Harmonics (SH) for capturing the static non-spherical gravity effects of the central body. Mascon approximations of varying fidelity have been previously developed for the Earths gravity field ranging from 120 mascons (8 x 8 SH field) to 30,720 mascons (56 x 156 SH field) [112]. Each mascon has a position in the rotating Earth-fixed reference frame, and an associated gravitational parameter. The perturbing acceleration is the sum of the acceleration of the RSO due to each mascon. For numerical accuracy, the mascon acceleration does not include the main two-body and J2 accelerations. On a CPU, SH and mascons have approximately the same computational load [112]. On the GPU, however, they are much simpler to implement. The other perturbation sources implemented are low fidelity for the current implementation. Note that the significance or necessity of including higher fidelity dynamics is lessened as the time window of interest is shortened. It is therefore important to note that typical encounter windows are minutes or hours. Furthermore, any perturbation added acts on both RSOs. In this study, a cannonball drag model is used

with an exponential atmosphere. The lunisolar ephemerides are third order analytical equations from Vallado's book [128]. The SRP computations also use a simple cannonball model.

The GPU memory is separate from the CPU and the initial conditions for the MC simulations have to first be transferred to the global memory of the GPU. The transfer speed is limited by the CPU-GPU Peripheral Component Interconnect Express (PCIe) bandwidth. The number of MC initial conditions that fit on the GPU  $N_{MCGPU}$  is limited by the global memory of the GPU. In case  $N_{MCGPU} < N_{MC}$ , the transfers and computations are split into batches of  $N_{MCGPU}$  so that the initial conditions fit on the GPU. A simple memory layout of GPU in Figure 4.5 shows only the memory locations used for the MC computation. For the computations on the GPU, the threads accessing the global memory is the slowest of the internal transfers. The fastest memory access is with the registers available to each thread. The registers are extremely small and can only fit 63 32-bit words in a GPU with CUDA compute capability 2.0. Newer GPUs with compute capability 3.5 and higher can fit 255 32-bit words. If the internal computations of the threads require more memory than can fit on the registers, additional local memory is used. Using the local memory, which is private per thread but resides on the global memory and passes through a L2 and L1 cache, is known as register spilling. Register spillage can hurt performance due to the increased memory traffic and instruction count. The final relevant memory type is constant memory, which is cached memory that is read only for the threads. In the current GPU implementation, the lookup tables for the exponential atmosphere model are



stored in the constant memory. The memory requirement for the RK step

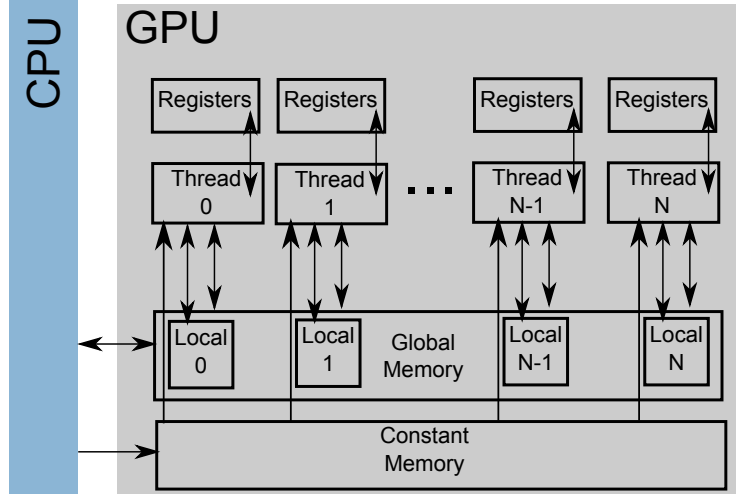


Figure 4.5: Memory layout of the GPU

including the intermediate stages and the interpolation always causes register spillage. To reduce the performance hit, the  $L_1$  cache size is increased and the shared memory size is decreased by using the command: `cudaDeviceSetCacheConfig(cudaFuncCachePreferL1)`. Shared memory is not used in this implementation and therefore, reducing its size is not a problem. Another constraint of the register spillage is that a high order RK integration becomes prohibitively slow. The classical fourth order Runge-Kutta (RK4) method with four intermediate stages has a low memory load because only one intermediate stage has to be stored. For higher order RK methods, however, all the intermediate stages have to be stored because they are used multiple times, which results in a high memory spillage and a large hit on performance. Using a lower order RK integration is not detrimental to the overall application,

however, because interpolation sets a upper limit on the size of the maximum time step, as seen in the Results section.

### 4.3 Results

The performance of the GPU implementation of the collision probability computation is now tested. Any uncertainty propagation method can be used with the GPU MC tool as long as samples are generated. Figure 4.6 shows the collision probability computed by generating samples from the uncertainty distribution propagated using a Gaussian distribution in Cartesian coordinate, a Gaussian distribution in Equinoctial Elements, and an 8<sup>th</sup> order Polynomial Chaos Expansion in Cartesian coordinates. The PDFs of both the Primary RSO at  $t_{CA}$  is non-Gaussian in Cartesian coordinates and therefore, the accuracy of the propagation method has a noticeable effect on the collision probability.

The accuracy of the quartic interpolation for computing the close approaches is first investigated. The speedups due to the parallel processing on the GPU are then presented. The exact same collision probability algorithm is computed on the GPU and the CPU to calculate the speedups for the sake of consistency. It should be noted that CPU implementation could be sped up by using a different interpolation technique, variable step integration, etc. Finally, the effect of perturbations on the collision probability is investigated for a LEO and a GEO case.

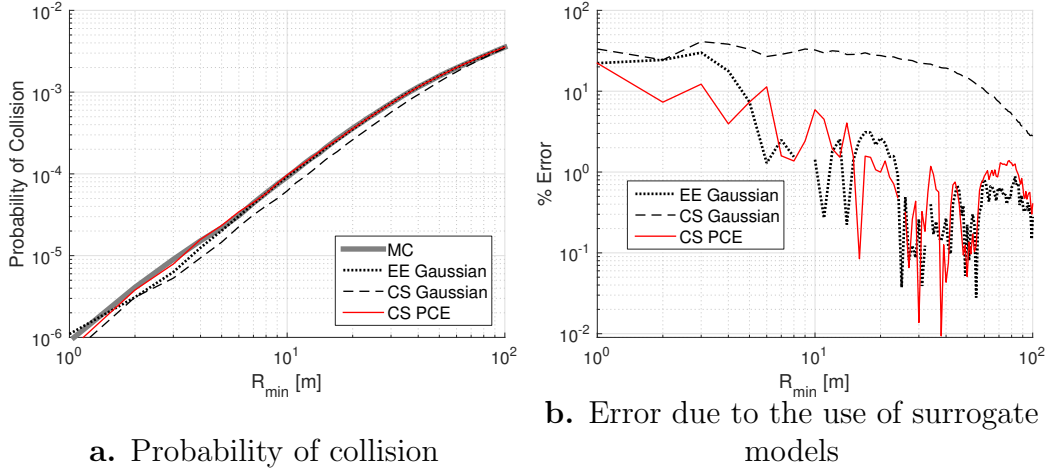


Figure 4.6: Case 7 from Alfano’s [17] paper using two-body dynamics with the standard deviation of the in-track velocity of the primary RSO increased by a factor of 30. 10,000,000 samples are generated at  $t_{CA}$  using MC, a Gaussian distribution in Cartesian coordinates, a Gaussian distribution in EE, and an 8<sup>th</sup> order PCE in Cartesian coordinates.

#### 4.3.1 Validation of Close Approach Interpolation

The interpolation is tested to find the optimal RK step sizes for the interpolation. One million encounter geometries are randomly generated by first creating the state of the primary RSO at the time of closest approach  $t_{CA}$ , i.e.  $r(t_{CA}) = r_{min}$ . The secondary RSO’s state relative to the primary is then created in spherical coordinates. The resulting geometry is shown in Figure 4.7a. Given a time step size  $\Delta t$  and the time fraction  $t_{frac}$ , the states are propagated back in time to  $t_0$  using two-body dynamics:

$$t_{CA} = t_0 + t_{frac} \times \Delta t \quad (4.13)$$

Two-body dynamics are used to ensure that the exact motion of the RSOs over the time step is known. The accuracy of the interpolation is calculated

by computing the error in  $r_{min}$  for each sample. The time fraction and its relation to the time step size is shown in Figure 4.7b. The acceleration due to perturbations is orders of magnitude lower than the two-body acceleration and therefore perturbations are not included in this test.

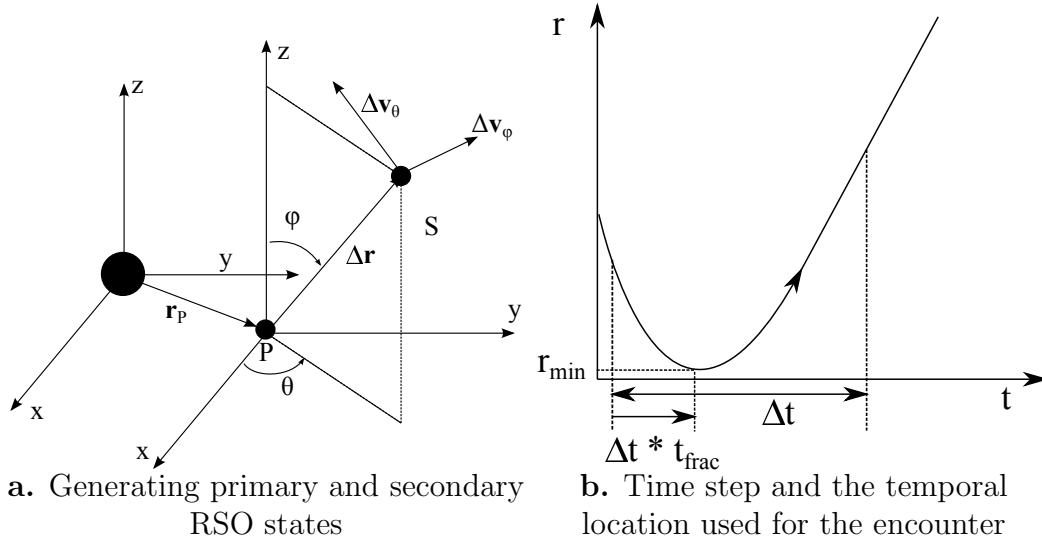


Figure 4.7: Spatial and temporal setup of random encounters

The state of the primary RSO is sampled from a Uniform distribution of Keplerian orbit elements with a few constraints shown in Equation (4.14). The state of the secondary RSO is generated by creating a random relative state in spherical coordinates, centered at the primary RSO as shown in Equation (4.15). The radial velocity is chosen to be 0 and the radial acceleration  $d^2(r^2)/dt^2$  is constrained to be positive to ensure that the secondary and pri-

mary RSO are at their closest approach.

$$r_{P1} \sim \mathcal{U}(6.525 \times 10^3 \text{ km}, 5 \times 10^4 \text{ km}) \quad (4.14a)$$

$$r_{P2} \sim \mathcal{U}(6.525 \times 10^3 \text{ km}, 5 \times 10^4 \text{ km}) \quad (4.14b)$$

$$r_{P-} = \min(r_{P1}, r_{P2}) \quad (4.14c)$$

$$r_{P+} = \max(r_{P1}, r_{P2}) \quad (4.14d)$$

$$i_P \sim \mathcal{U}(0, 2\pi) \quad (4.14e)$$

$$\Omega_P \sim \mathcal{U}(0, 2\pi) \quad (4.14f)$$

$$\omega_P \sim \mathcal{U}(0, 2\pi) \quad (4.14g)$$

$$\nu_P \sim \mathcal{U}(0, 2\pi) \quad (4.14h)$$

$$\mathbf{r}_S = \mathbf{r}_P + \Delta \mathbf{r} \quad (4.15a)$$

$$\mathbf{v}_S = \mathbf{v}_P + \Delta \mathbf{v} \quad (4.15b)$$

$$\Delta \mathbf{r} = \Delta r \begin{bmatrix} \cos \theta \sin \varphi \\ \sin \theta \sin \varphi \\ \cos \varphi \end{bmatrix} \quad (4.15c)$$

$$\Delta \mathbf{v} = \Delta v \left\{ \alpha \begin{bmatrix} -\sin \theta \\ \cos \theta \\ 0 \end{bmatrix} + (1 - \alpha) \begin{bmatrix} \cos \theta \cos \varphi \\ \sin \theta \cos \varphi \\ -\sin \varphi \end{bmatrix} \right\} \quad (4.15d)$$

$$\Delta r \sim \mathcal{U}(1 \text{ m}, 220 \text{ m}) \quad (4.15e)$$

$$\Delta v \sim \mathcal{U}(0, 2v_{esc}(\mathbf{r}_P)) \quad (4.15f)$$

$$\theta \sim \mathcal{U}(0, 2\pi) \quad (4.15g)$$

$$\varphi \sim \mathcal{U}(0, 2\pi) \quad (4.15h)$$

Additional constraints on the state of the secondary RSO are:

$$0 \leq e_S < 1 \quad (4.16a)$$

$$a_S(1 - e_S) \geq 6525 \text{ km} \quad (4.16b)$$

$$a_S(1 + e_S) \leq 5 \times 10^4 \text{ km} \quad (4.16c)$$

To ensure that the conditions created are a conjunction, the second time derivative of  $r^2$  from Equation (4.6c) should be positive. The encounter is constructed using the two-body dynamics and therefore, the relative acceleration is:

$$\ddot{\mathbf{r}} = -\mu_E \left( \frac{\mathbf{r}_S}{r_S^3} - \frac{\mathbf{r}_P}{r_P^3} \right) \quad (4.17)$$

All the encounters generated have RSOs in closed orbits with a minimum perigee altitude of 150 km and a maximum apogee of 50,000 km. The minimum separation  $r_{min}$  ranges from 1 m to 220 m. The latter is twice the hardbody radius of the International Space Station (ISS). The normalized histogram of the relative velocities of the 1,000,000 randomly generated encounters at  $t_{CA}$  is shown in Figure 4.8.

Finally, the interpolation algorithms are used to estimate the closest approach for various time step sizes. The encounter is assumed to occur at a fraction of the step size. The initial conditions generated using Eqs. (4.14) and (4.15) are at time  $t_{CA} = t_0 + t_{frac} \times \Delta t$  and the interpolation is over  $t \in [t_0, t_0 + \Delta t]$ . A plot of the cumulative distribution function (CDF) of the error in  $r_{min}$  for the interpolation with various step sizes is shown in Figure 4.9. With a time step size of 10 seconds, computing the closest approach using the individual state interpolations results in all the errors being smaller than 1 m

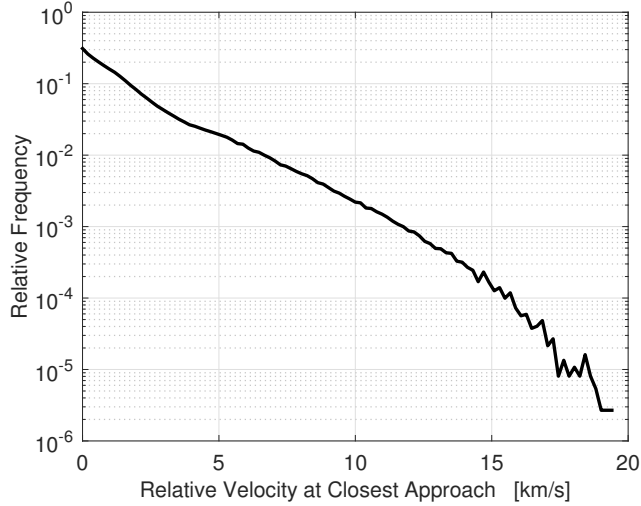


Figure 4.8: Normalized histogram of the relative velocities  $\Delta v$  at  $t_{CA}$  for the 1,000,000 randomly generated encounters at  $t_{CA}$

and the maximum error of approximately 35 cm. Using this test, the individual interpolations with a maximum time step size of 10 s is recommended for use. For a time step size of 15 s, the maximum error is 1.2 m and 99.998% of the samples tested have an error less than 1 m. For a time step size of 20 s, the maximum error is 2.8 m and 99.96% of the samples tested have an error less than 1 m. Finally, for a time step size of 25 s, the maximum error is approximately 5.4 m and 99.80% of the samples tested have an error less than 1 m. The error in computing  $t_{min}$  and  $r_{min}$  are shown in Figure 4.10.

The value of  $r_{min}$  is also computed directly from  $r_*^2(\tau)$  instead of forming the quartic polynomials of  $x_*(\tau)$ ,  $y_*(\tau)$ , and  $z_*(\tau)$ . As expected, the interpolation error is larger and the error profile for an encounter is shown in Figure 4.11. The separation between the RSOs has a V-shaped behavior over the

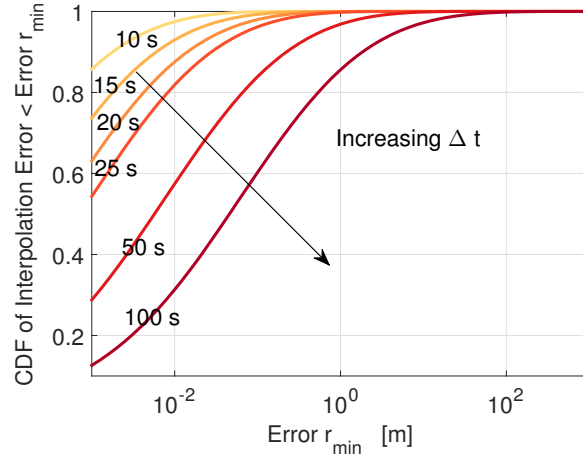


Figure 4.9: CDF of the error in computing  $r_{min}$  for time step sizes  $\Delta t = [10, 15, 20, 25, 50, 100]$  s

encounter. The behavior is only nonlinear in the vicinity of  $t_{CA}$  and thus, the error in  $r_*^2(\tau)$  increases dramatically. Far from  $t_{CA}$ , however, the behavior is quite linear and the error in  $r_*^2(\tau)$  is lower than the error in  $x_*^2(\tau) + y_*^2(\tau) + z_*^2(\tau)$ .

### 4.3.2 Performance of the GPU Code

The speedups for the collision probability computation on the GPU is shown here. The test hardware specifications for all the timing results are shown in Table 4.1 and speedups on the GPU are with respect to an identical serial implementation on a single core CPU. The NVIDIA GPU has a compute capability of 2.0 and therefore, register spilage heavily influences the speedups achieved.

The wall time for taking 125,000 RK4 steps on the CPU is shown



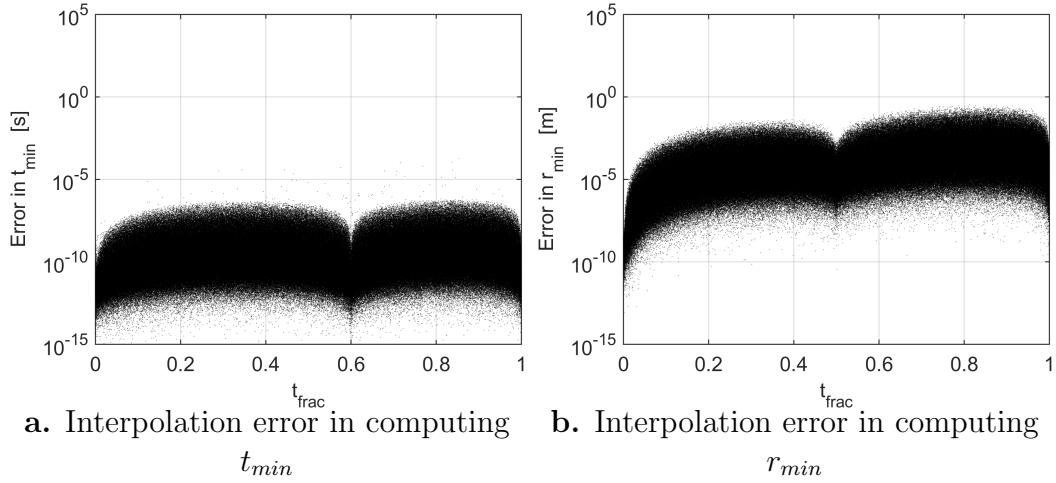


Figure 4.10: Error in estimating the time of closest approach and minimum separation for  $\Delta t = 10s$

in Figure 4.12a for various dynamics fidelities. Each RK4 step consists of 8 dynamics function evaluations (four for each RSO) followed by interpolation to find  $r_{min}$ . For the two-body dynamics, the interpolation counts for a noticeable portion of the total wall time. As the dynamics become more computationally intensive, however, the added wall time due to the interpolation is negligible. Figure 4.12b shows the speedups that are attained on the GPU. As will be

Table 4.1: Test hardware specifications

Component type	Component
CPU	Intel Xeon X5650 @ 2.67 Ghz
Operating System	Linux X86_64
GPU	1 TESLA M2090
Cuda cores	512
CPU memory	48.0 GB
GPU global memory	6.0 GB
CPU compiler	GCC
GPU compiler	NVCC 6.5 & 7.0

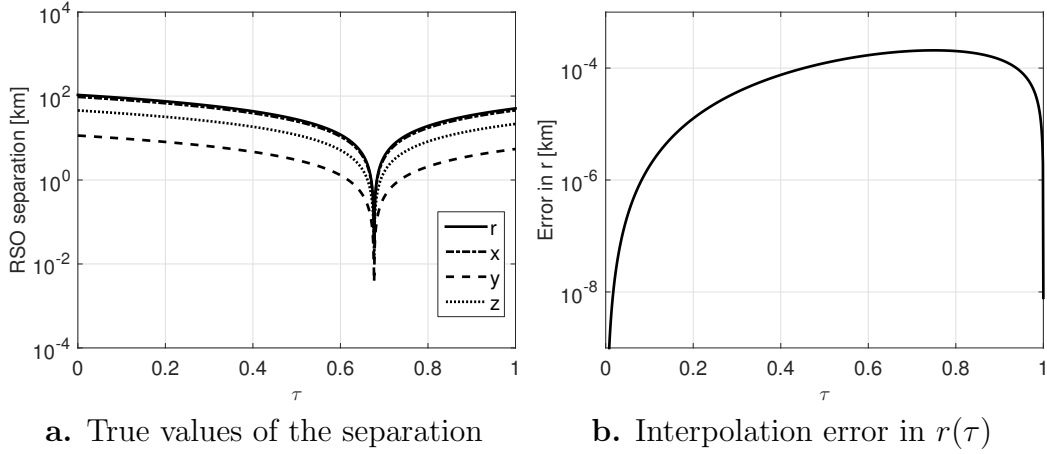


Figure 4.11: True values of the separation and the accuracy of the quartic interpolation for a randomly generated encounter

shown in Figures 4.13 - 4.17, the speedups on the GPU increase with the number of particles and the number of function evaluations till a maximum level is reached. The speedups increase with increasing computation load until the computation compensates for the slower memory transfers on the GPU. The speedups reported in Figure 4.12b are in the vicinity of the steady state speedup.

The GPU implementation is highly sensitive to the memory requirement of the dynamics and the interpolation. There is also a difference between the specific versions of CUDA: CUDA 6.5 is faster than 7.0 for the dynamics with a higher computation load. It should be noted that the speedups for the dynamics with mascons decrease with increasing number of mascons although the computational load increases. Each mascon has four double precision numbers associated with it: the 3D position and the gravitational parameter. Increasing number of mascons results in more reads from the global memory

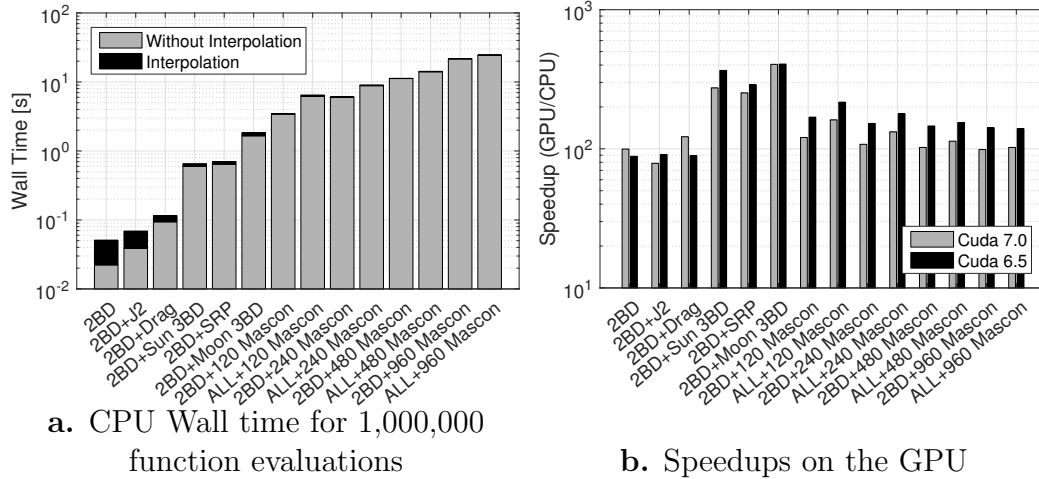


Figure 4.12: CPU Wall time and GPU speedup, including interpolation, for various dynamics fidelity levels. Every RK4 step requires 8 function evaluations and one interpolation

per kernel, which has an overall slowing effect on the computation.

Speedups for using only the two-body dynamics on the GPU is shown in Figure 4.13. The same code running on the GPU is up to 89 times faster than on the CPU. Figures 4.14 - 4.17 show the speedups when all the dynamics are included. For mascon field sizes of 120 ( $9 \times 9$  SH) and 960 ( $27 \times 27$  SH) maximum speedups of  $224\times$  and  $145 \times$  are achieved, respectively. Note that memory transfers from the CPU to the GPU and vice-versa are included in the speedup calculation. As an example wall clock reading, Figures 4.13 and 4.14 show that 10 million particles taking 1,000 RK4 steps requires approximately 5 minutes and half an hour of wall clock time for the low and high fidelity dynamics considered, respectively. The speedups presented are for the exact same collision probability computation method on the CPU and GPU. The speedups reported are agnostic to the geometry of the encounter because of

the uniformity of the computations on the GPU. The computation load only depends on the number of RK4 steps (dynamics function evaluations), number of particles, and the dynamics function.

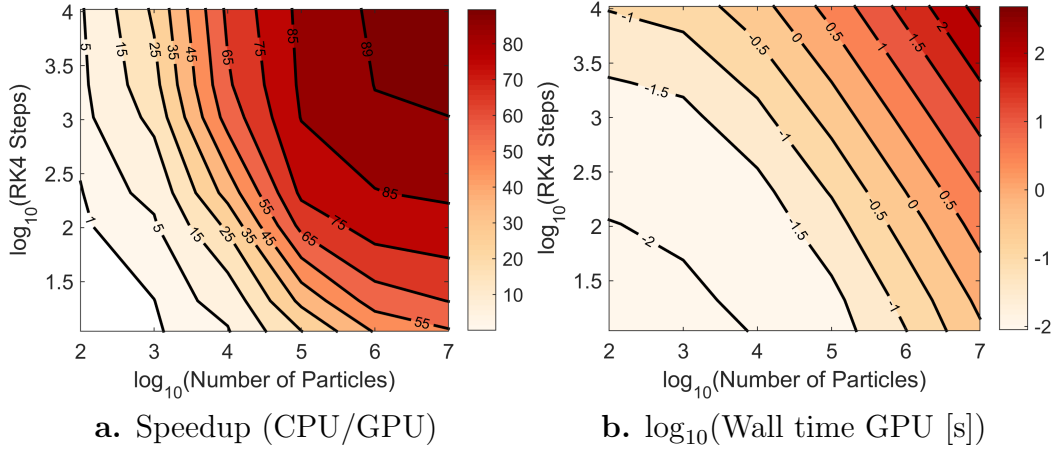


Figure 4.13: Timing results with two-body dynamics, increasing particles, and decreasing time-step size

### 4.3.3 Effect of Perturbations

The main factor that determines the fidelity of the perturbations required for an accurate collision probability estimate is the encounter window used for the simulation. When the encounter window is short, the samples are generated close to  $t_{CA}$  and the effect of perturbations can become negligible. Therefore, a two body MC simulation is likely sufficient [113].

For large encounter windows inaccurate collision probabilities may result if low fidelity dynamics are used when the samples are provided at  $t_* \in [t_a, t_b]$ , and  $|t_{CA} - t_*|$  and  $|r(t_{CA}) - r(t_*)|$  are large. In this scenario, the samples are not provided close to  $t_{CA}$ , and the mean separation is large for most

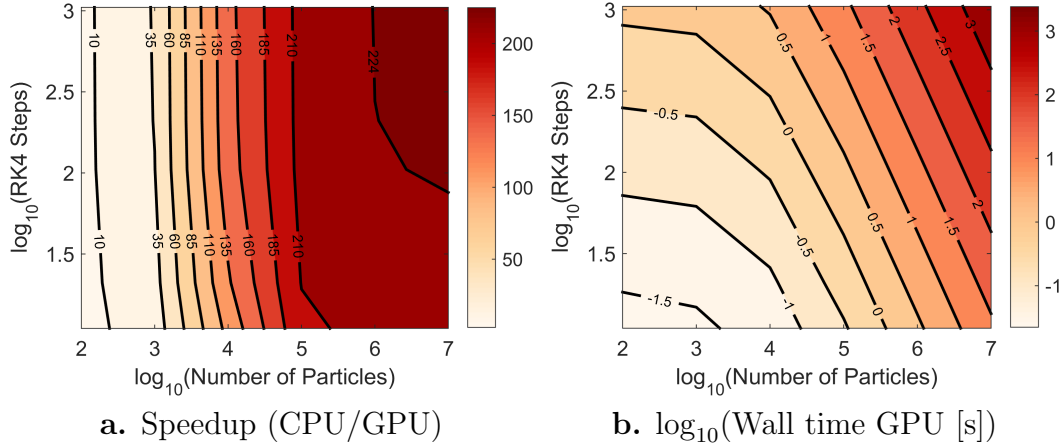


Figure 4.14: Timing results with all modeled perturbations and a 120 element mascon model ( $SH = 9 \times 9$ ), increasing particles, and decreasing time-step size

of the encounter simulation. Not using the full fidelity dynamics can result in a large difference between the true and lower dynamics fidelity conjunction.

Another important scenario is when  $|r(t_{CA}) - r(t_*)|$  is small. The encounter window is long but the two RSOs are in the general vicinity of each other, i.e. the mean separation is small during the encounter simulation. In this domain where relative motion approximations are valid, differential dynamics between the two RSOs becomes the main source of the low fidelity trajectories diverging from the high fidelity trajectories. For LEO orbits, the difference in dynamics when two RSOs are in similar orbits is largely caused when the primary and secondary RSOs have different ballistic coefficients. Differing SRP coefficients, product of the coefficient of reflection and the area to mass ratio (AMR), have a similar effect in GEO.

Low fidelity dynamics can be used when the encounter window is small, i.e. much smaller than the orbital period. For the two examples of long

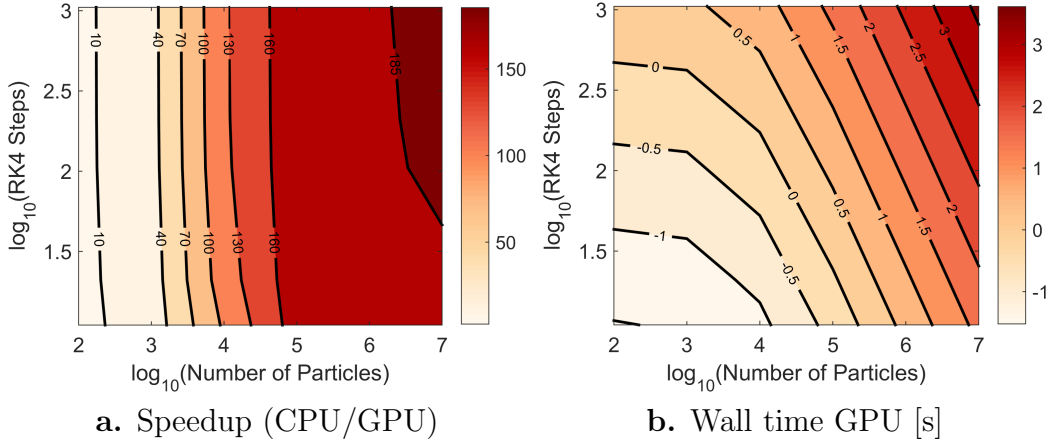


Figure 4.15: Timing results with all modeled perturbations and a 240 element mascon model ( $SH = 13 \times 13$ ), increasing particles, and decreasing time-step size

duration encounters shown here, including only J2 and drag for LEO, and SRP and third body perturbation due to the sun for GEO are sufficient to provide fast and accurate collision probability values.

A long duration GEO encounter is constructed using Case 1 from Alfano’s paper [17]. The initial conditions at epoch are sampled and then propagated using only two-body dynamics for 4630 minutes. The collision probability over the next 16,000 s, i.e.  $t_* = 0, t \in [0, 16000]$ , is computed using only two body dynamics and with all modeled perturbations. The collision probability and the separation between the nominal trajectories of the RSOs is shown in Figure 4.18. The relative trajectories and the collision probability differ from the two body case due to the differing SRP coefficients. Once the SRP is accounted, however, there is no need for modeling the other perturbations as they do not have a noticeable effect on the relative trajectories. In computing the SRP, the location of the Sun has to be computed and the third-body per-

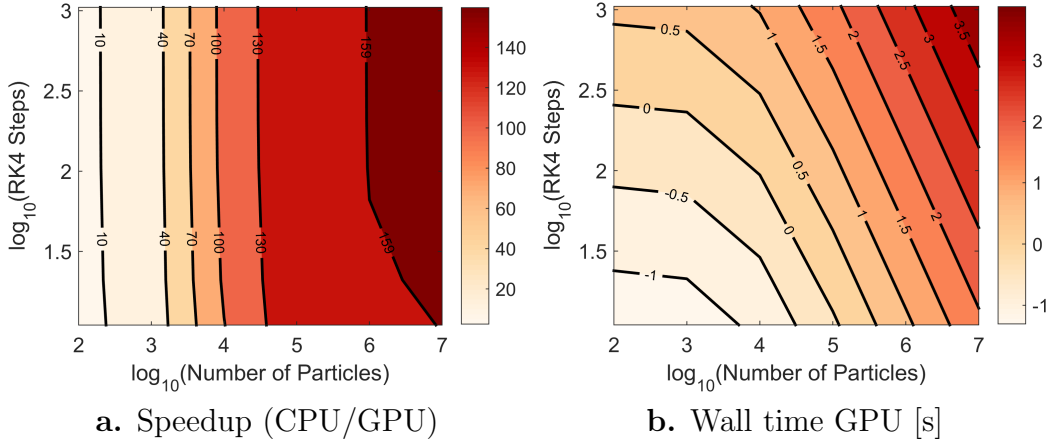


Figure 4.16: Timing results with all modeled perturbations and a 480 element mascon model ( $SH = 19 \times 19$ ), increasing particles, and decreasing time-step size

turbation due to the sun requires only a small extra computation. Therefore, a GEO perturbation model that has the two body dynamics, SRP, and the solar third-body perturbation is created for fast and accurate computations for long duration encounters. The speedup of the GEO perturbation model is shown in Figure 4.19 with a maximum speed up of  $270 \times$ .

A long duration LEO encounter is constructed based on Case 11 from Alfano's paper [17]. The initial uncertainty for both RSOs at epoch is the same as Case 11 [17]. The initial position and velocity, however, are presented in Table 4.2. The conditions at epoch are sampled and then propagated using only two body dynamics for 24 hours. The collision probability over the next 5000 s, i.e.  $t_* = 0, t \in [0, 5000]$ , is computed using only two-body dynamics and also with all modeled perturbations. The collision probability and the separation between the nominal trajectories of the RSOs is shown in Figure 4.20. The relative trajectories and the collision probability differ from the two

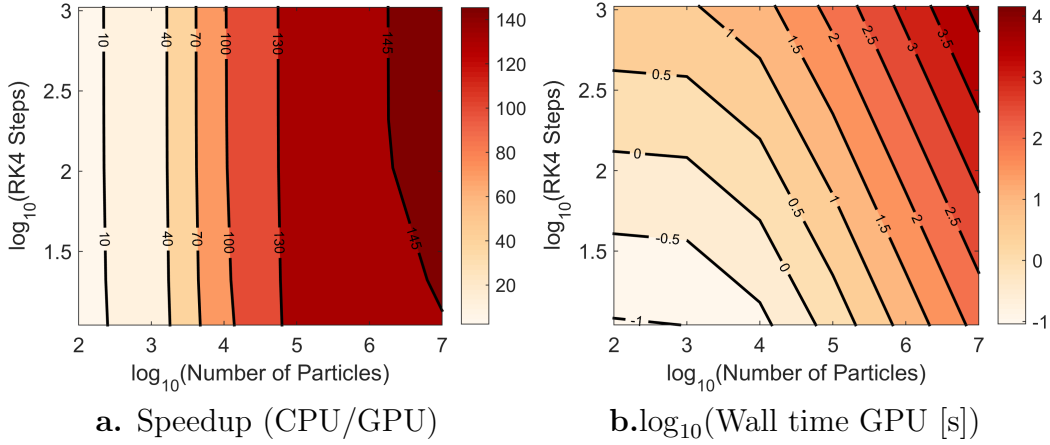


Figure 4.17: Timing results with all modeled perturbations and a 960 element mascon model ( $SH = 27 \times 27$ ), increasing particles, and decreasing time-step size

Table 4.2: Initial conditions for a long duration LEO encounter

	Primary RSO	Secondary RSO
$x$ [km]	-1984.685063909913	-1984.724564629808
$y$ [km]	6465.093096520082	6465.062782202386
$z$ [km]	0.0	0.0
$v_x$ [km/s]	-7.274454471564	-7.274446635145
$v_y$ [km/s]	-2.290234878974	-2.290325932204
$v_z$ [km/s]	0.0	0.0

body case due to the differing ballistic coefficients. As long as the drag and J2 are accounted, however, there is no need for modeling the other perturbations as they do not have a noticeable effect on the relative trajectories. Therefore, a LEO perturbation model that has the two-body dynamics, J2, and drag is created for fast and accurate computations for long duration encounters. The speedup of the LEO perturbation model is shown in Figure 4.21 with a maximum speed up of  $92 \times$ .



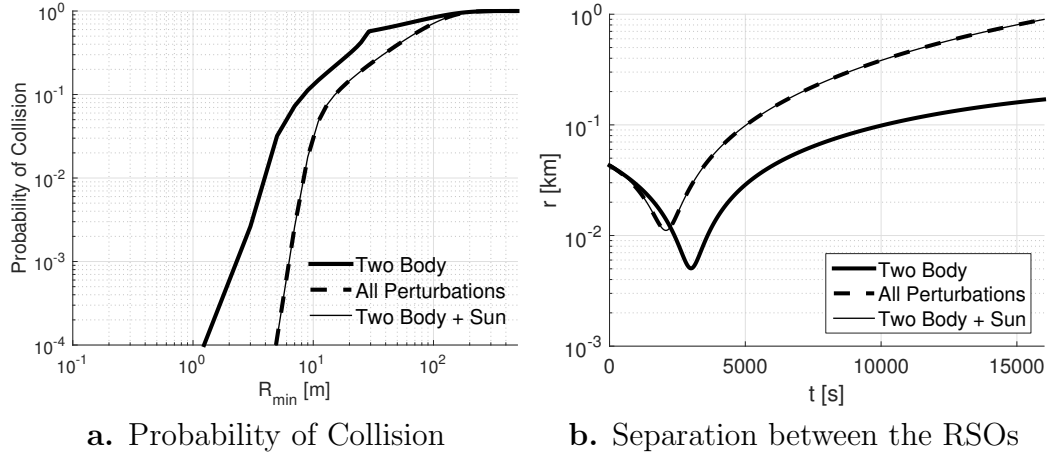


Figure 4.18: Change in the encounter between two RSOs in GEO due to SRP.  $CR_P = 1.1$ ,  $AMR_P = 0.4 \text{ m}^2/\text{kg}$ ,  $CR_S = 1.5$ ,  $AMR_S = 1.1 \text{ m}^2/\text{kg}$ .

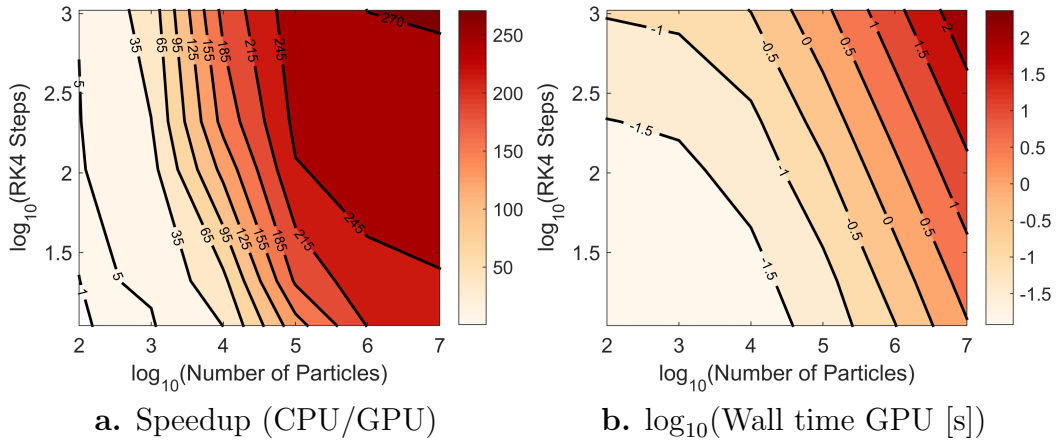


Figure 4.19: Timing results with two-body dynamics and SRP, increasing particles, and decreasing time-step size

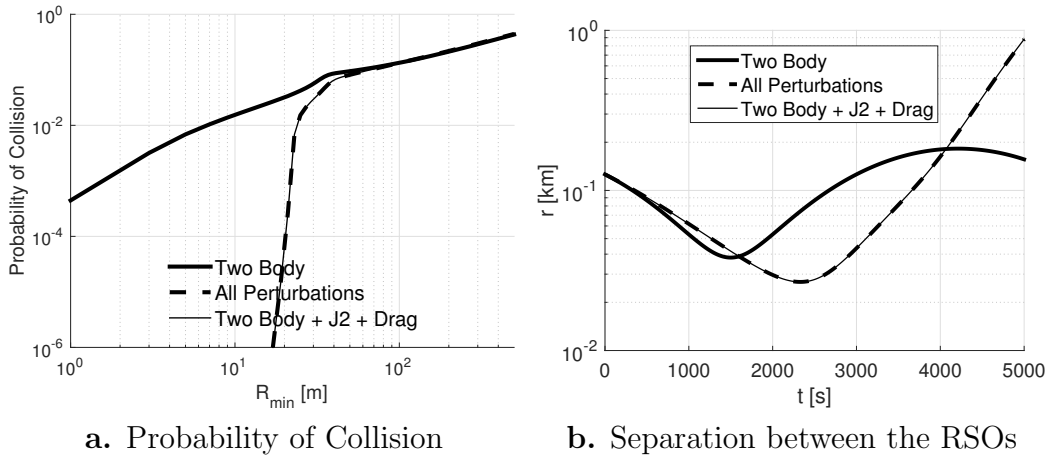


Figure 4.20: Change in the encounter between two RSOs in LEO due to drag.  $BC_P = 0.05179 \text{ m}^2/\text{kg}$ ,  $BC_S = 0.00687 \text{ m}^2/\text{kg}$ .

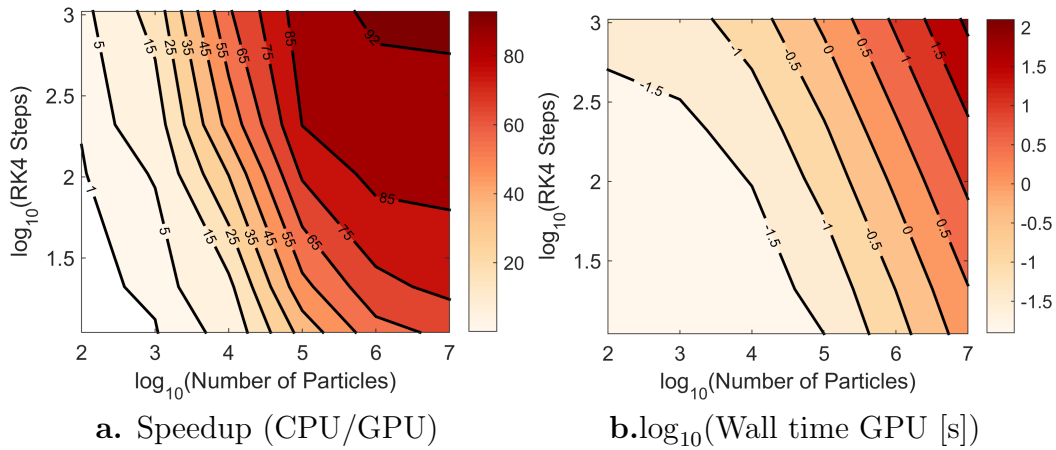


Figure 4.21: Timing results with two-body dynamics, J2, and cannonball drag, and increasing particles and decreasing time-step size

## 4.4 Conclusion

Using a Monte Carlo simulation to compute the probability of collision provides accurate results for any scenario and is always required as a benchmark for testing other techniques. Although millions of samples are required to compute the collision probability, the use of a GPU provides approximately two orders of magnitude speedups when compared to a serial CPU implementation for all the scenarios, making the technique feasible on a single workstation.

The conjunction assessment problem is split into two phases (Figure 1.7 page 12) with the first phase focusing on the uncertainty propagation of the RSOs and the second phase focusing on computing the probability of collision. The GPU MC simulation is used only on the second phase and therefore any of the uncertainty propagation methods, including Multidirectional Gaussian Mixture Models from Chapter 2, and the combination of Polynomial Chaos Expansions and Gaussian Mixture Models from Chapter 3, can be used as long as samples can be generated from them. The output of the MC simulation is the closest approach between the samples of the RSOs over the encounter window. Therefore, recomputing the probability of collision for different values of the hard body radius is a quick post processing step and rerunning the MC simulation is not required.

Fixed step RK4 integration ensures that the computational load is uniform for all the samples in the MC simulation and reduces the memory required per run. A one sided Hermite interpolation is used to find the closest separation between the realization of the two RSOs during each step. A trade study

with 1,000,000 randomly generated encounters covering most of the relevant orbit element space was implemented to find the time step for accurate interpolation. A maximum time step size of 10 s was shown to accurately compute the minimum separation to an accuracy of 1 m or less. With a time step size of 20 s, the maximum interpolation error is less than 3 m and over 99.9% of the test cases have an interpolation error less than 1 m.

For a short duration encounter, i.e. the RSOs are close to each other only for a short period of time, a low dynamics fidelity MC simulation is sufficient for an accurate collision probability if the samples are provided near the time of closest approach. If the two RSOs are in proximity to each other at the time of sampling and the relative velocity is low, the relative motion is determined by the differential dynamics. Therefore, it is important to include drag for LEO and SRP for GEO cases, but the other computationally intensive perturbations like the gravity field can be omitted. Many encounter problems fall under one of the two aforementioned scenarios, and simpler low fidelity dynamics can be used for the MC collision probability computation. Higher fidelity dynamics, which are more computationally intensive, only become more important for longer duration windows with a larger mean RSO separation over the window. Development on the GPU is expensive in terms of effort. Therefore, it is useful to have a static GPU code with as few dynamics as possible for practical purposes, especially if there is no accuracy benefit of including higher fidelity.

## Chapter 5

# Collision Probability using Multidirectional Gaussian Mixture Models

The benchmark Monte Carlo method for collision probability (Phase II from Figure 1.7 on page 12) has been parallelized on a Graphics Processing Unit in Chapter 4. Mutidirectional Gaussian Mixture Models (MGMMs) provide a compromise between accuracy and runtime by better approximating the true non-Gaussian distributions during conjunction. The selection of the directions and number of splits along relevant directions from Chapter 2 provides a dial that spans from Monte Carlo to the classic Gaussian approximation. MGMMs are used in this chapter<sup>1</sup> to successfully compute the collision probability for several test cases with non-Gaussian uncertainty distributions for both short and long duration encounters while requiring orders of magnitude fewer dynamics function evaluations.

---

<sup>1</sup>The work from this chapter has been presented at a conference:

- V. Vittaldev and R. P. Russell, Collision Probability using Multidirectional Gaussian Mixture Models, Paper AAS 15-394, AAS/AIAA Space Flight Mechanics Meeting, Williamsburg, VA, 2015.

A manuscript has also been submitted to a peer reviewed journal:

- V. Vittaldev and R. P. Russell, Space Object Collision Probability Using Multidirectional Gaussian Mixture Models, *Journal of Guidance, Navigation, and Control*, Submitted August 2015.

## 5.1 Introduction

For collision probability computation, the intersection of the time varying 3-dimensional position PDFs of two Resident Space Objects (RSOs) over a specified time range is required. The extent of the non-Gaussian behavior depends on the state parameterization, orbital regime, initial uncertainty magnitudes, and the time of flight. The most accurate but computationally expensive method is to use a full-blown Monte Carlo (MC) technique [17], which was presented in detail in Chapter 4, requiring millions of samples.

When two RSOs are assumed to be spherical, convolution integrals can be used to compute the collision probability between them when the distributions are non-Gaussian [36]. Additionally, if the RSOs have a Gaussian distribution in Cartesian coordinates during conjunction, and their distributions are independent, the collision probability is calculated with a low computation cost [40] when compared to an MC simulation. The collision probability rate is computed as a quadrature over a sphere. An additional quadrature in time gives the desired probability. Additionally, if the encounter is brief, there exist many analytical approximations that allow for a fast computation of collision probability [13]. The linear approximation theories do not account for any uncertainty in velocity during the conjunction i.e. the velocities of the RSOs and their covariance matrices are frozen at the time of closest approach. The combined covariance is placed on the secondary RSO. The primary RSO is then assumed to be a sphere with the radius equal to the combined radius of the two RSOs. As the sphere passes through the combined position covariance, it traces out a tube. The covariance ellipsoid is a three-dimensional PDF in

the position space and the collision probability is the integral of the PDF over the volume of the tube. When the PDFs of the RSOs are not Gaussian during the encounter, a major assumption of the aforementioned methods is no longer valid. However, each of the elements is Gaussian when a sufficient number of GMMs are used to propagate the uncertainty. Therefore, the quadrature or linear conjunction methods can be used in an all-on-all manner on the GMMs of the involved RSOs.

The first segment of the collision probability computation concerns the accurate propagation of the non-Gaussian PDFs of the two involved RSOs from epoch to the encounter time. The PDF of the uncertainty of an RSO has been propagated using GMMs in orbit elements [68], by adaptively splitting and merging elements based on their differential entropy during propagation [44] in Cartesian coordinates, and by updating the weights of the GMMs during propagation [130]. The univariate library is applied to the largest eigenvalue spectral direction of the covariance matrix in the adaptive splitting technique [44]. Either the linear or the quadrature based computations are carried out in an all-on-all manner for all elements in the GMMs of both RSOs. The linear collision method has been shown to work with GMMs [132] for non-Gaussian distributions, and the adaptively split GMMs have been used successfully with the quadrature method [45]. The strategy implemented in this chapter is to split the initial Gaussian PDFs of both RSOs into MGMMs. The MGMM elements are propagated through the nonlinear orbit dynamics. The number of elements and the associated weights are not varied during, or after the propagation. The propagation is therefore treated as a black box,

discrete function.

Without weight updates and adaptive splitting, non-linearity in Cartesian coordinates becomes present in multiple directions [68] and could require a 6-dimensional recursive initial split. A full 6-dimensional tensor product grid of GMMs has a prohibitive computational cost i.e. the curse of dimensionality, comparable to an MC simulation. Hence, the nonlinearity parameter is used to identify and split along only the most important directions. For computing the collision probability, the MC simulation is replaced by increasing the number of elements along multiple important arbitrary directions of the initial Gaussian distribution of the RSOs. As the number of elements is increased, the probability converges to the true value at a lower computational cost than the tens of millions of propagations required for an MC simulation.

The quadrature and linear collision probability methods, and the extension using MGMMs are presented in Section 5.2. The MGMM collision probability method is successfully applied to various encounters in Section 5.3. Finally, the conclusions drawn are presented in Section 5.4.

## 5.2 Collision Probability

Computing the probability of collision is divided into two distinct phases, seen in Figure 1.7 on page 12. The figure is recreated in Figure 5.1 with the PDFs consisting of GMMs.

No measurements are assumed to occur during Phase I in order to decouple uncertainty propagation from the state estimation. The uncertainty



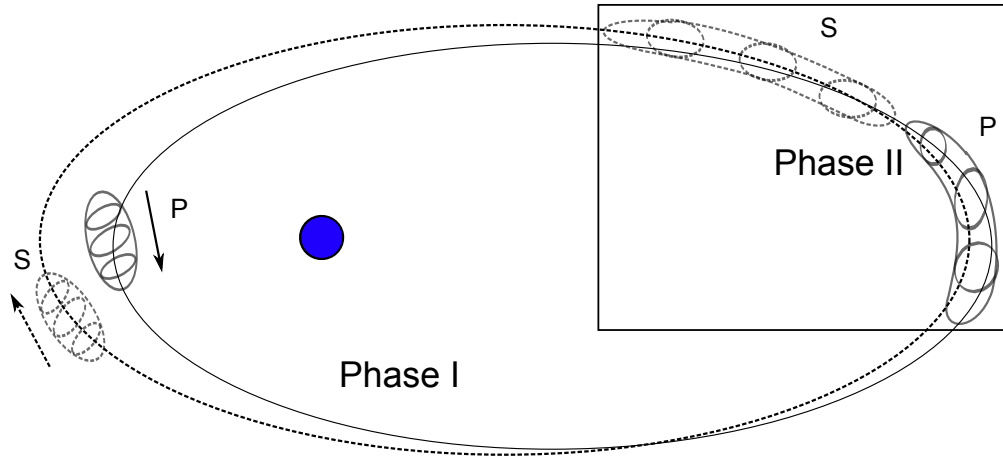


Figure 5.1: The two phases for the collision probability computation

expressed as MGMMs is propagated using sigma-point methods. The coordinate system of choice is important because orbit element sets such as EE are known to remain Gaussian for longer flight times than Cartesian coordinates. Cartesian coordinates are chosen in this chapter because the computations in Phase II require the uncertainty in Cartesian coordinates.

### 5.2.1 Monte Carlo

The most accurate and computationally intensive method is an MC simulation. Millions of point pairs are generated, where each point pair consists of one sample from the primary RSO's distribution and one from the secondary RSO's distribution. If the separation between the samples in the pair is less than the minimum separation distance at any time during the search window, a collision is counted. The total number of counted collisions is divided by the total number of pairs used to generate the probability of collisions. Details on the implementation of the MC method and speedups gained by parallelizing

on the GPU are shown in Chapter 4.

The MC solution is used as the benchmark due to its accuracy and reliability. Given the expected value of the true probability of collision  $p_T$  and the percentage error  $\epsilon$  with confidence  $\gamma$ , the lower bound on the number of independent samples, point pair, required for a MC simulation is [43]

$$n_{MC} > \frac{4(e-2)((1-P_T)P_T)}{\epsilon^2 P_T^2} \log\left(\frac{2}{1-\gamma}\right) \quad (5.1)$$

A 1% error with 95% confidence corresponds to  $\epsilon = 0.01$  and  $\gamma = 0.95$ . Two options for the number of samples are:

- One-on-one: Take a point from each distribution, check if collision occurs, and then remove both points from the respective clouds.
- All-on-all: For each point from the cloud, check if a collision occurs with all the points in the other distribution.

For checking  $N$  cases, the first method requires  $N$  samples from each RSO distribution. The second method requires only  $\sqrt{N}$  samples. The first method is statistically more accurate because all the sample pairs are independent, however, the computation time saved by propagating fewer points makes the all-on-all method more desirable if MC is used for Phase I. If another uncertainty propagation method is used during Phase I, method 2 is more efficient only if the ephemeris of the samples during the search window is stored. Storing the ephemeris, however, increases the complexity of the code and has a large memory footprint.

The benchmark MC computations in this chapter use the one-on-one method with interpolation from Chapter 4 to find the RSO states at the closest approach.

### 5.2.2 Collision Probability for Gaussian Distributions in Cartesian Coordinates

Computationally inexpensive methods approximate the collision probability accurately under certain simplifying assumptions. The main assumptions in the approximate methods are that the uncertainty distributions of the RSOs are Gaussian in Cartesian coordinates and are mutually independent. The states of the primary and secondary RSO have the following 6-dimensional Gaussian distributions:

$$\mathbf{x}_P(t) \sim \mathcal{N}(\boldsymbol{\mu}_P(t), \mathbf{P}_P(t)) \quad (5.2a)$$

$$\mathbf{x}_S(t) \sim \mathcal{N}(\boldsymbol{\mu}_S(t), \mathbf{P}_S(t)) \quad (5.2b)$$

Coppola's method [40] computes the instantaneous probability rate and integrates over the time window in Phase II using numerical quadrature rules. The instantaneous probability of collision requires an integration over the surface of a sphere with the combined radius of the RSOs  $R$ , which is also computed using a quadrature rule. The sphere is located on the secondary body and the 6-dimensional relative state  $\mathbf{x}$ , relative position  $\mathbf{r}$ , relative velocity  $\mathbf{v}$ ,

and the combined covariance  $\mathbf{P}$  are:

$$\mathbf{x} = \mathbf{x}_P - \mathbf{x}_S \quad (5.3a)$$

$$\mathbf{r} = \mathbf{r}_P - \mathbf{r}_S \quad (5.3b)$$

$$\mathbf{v} = \mathbf{v}_P - \mathbf{v}_S \quad (5.3c)$$

$$\mathbf{P} = \mathbf{P}_P + \mathbf{P}_S \quad (5.3d)$$

Note that all the above variables are a function of time. The combined covariance matrix is expressed in terms of the  $3 \times 3$  position covariance  $\mathbf{P}_r$ , velocity covariance  $\mathbf{P}_v$ , and the correlation  $\mathbf{P}_{vr}$ .

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_r & \mathbf{P}_{vr}^T \\ \mathbf{P}_{vr} & \mathbf{P}_v \end{bmatrix} \quad (5.4)$$

The collision probability over a time window  $t_0 \rightarrow t_f$  is:

$$P_c = P_0 + P_I \quad (5.5)$$

$P_0$  accounts for all the set of initial conditions which occur in a collision at  $t_0$  and  $P_I$  handles the set of initial conditions that result in a collision during  $t_0 < t \leq t_f$ . Initial conditions resulting in a collision at various times are shown in Figure 5.2.

$P_0$  is expressed in spherical coordinates after integrating over the velocity as:

$$\begin{aligned} P_0 &= \int_0^R \frac{dP_0(\boldsymbol{\mu}_P(t_0), \mathbf{P}_P(t_0), \boldsymbol{\mu}_S(t_0), \mathbf{P}_S(t_0))}{dr'} dr' \\ &= \int_0^R \int_0^{2\pi} \int_{-\pi/2}^{\pi/2} p_g(\mathbf{r}'; \boldsymbol{\mu}_r(t_0), \mathbf{P}_r(t_0)) r' \cos \theta d\theta d\phi dr' \end{aligned} \quad (5.6)$$

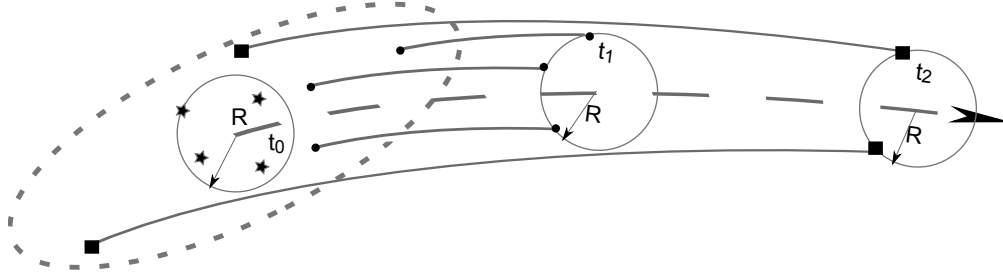


Figure 5.2: Initial conditions from the combined covariance matrix at  $t_0$  that result in collisions at  $t_0$  ( $\star$ ), and at times  $t_1$  ( $\bullet$ ), and  $t_2$  ( $\blacksquare$ ) during the conjunction

where  $p_g(\mathbf{x}'; \boldsymbol{\mu}, \mathbf{P})$  is the multivariate Gaussian probability distribution function with mean  $\boldsymbol{\mu}$  and covariance  $\mathbf{P}$  evaluated at  $\mathbf{x}'$ . The initial time of the conjunction computation is typically chosen such that  $P_0 = 0$  because the two RSOs are far apart. After integrating over the velocity,  $P_I$  is:

$$\begin{aligned}
 P_I &= \int_{t_0}^{t_f} \dot{P}_I(\boldsymbol{\mu}_P(t), \mathbf{P}_P(t), \boldsymbol{\mu}_S(t), \mathbf{P}_S(t)) dt \\
 &= \int_{t_0}^{t_f} \int_0^{2\pi} \int_{-\pi/2}^{\pi/2} p_g(\mathbf{r}'; \boldsymbol{\mu}_r(t), \mathbf{P}_r(t)) \nu(\hat{\mathbf{r}}', t) R^2 \cos\theta d\theta d\phi dt
 \end{aligned} \tag{5.7}$$

where

$$\nu(\hat{\mathbf{r}}', t) = \frac{\sigma}{\sqrt{2\pi}} \left( e^{-\tilde{\nu}^2} - \sqrt{\pi} (1 - \text{erf}(\tilde{\nu})) \right) \tag{5.8a}$$

$$\sigma^2 = \hat{\mathbf{r}}'^T (\mathbf{P}_v - \mathbf{P}_{vr} \mathbf{P}_r^{-1} \mathbf{P}_{vr}) \hat{\mathbf{r}}' \tag{5.8b}$$

$$\tilde{\nu} = \frac{\hat{\mathbf{r}}'^T (\boldsymbol{\mu}_v + \mathbf{P}_{vr} \mathbf{P}_r^{-1} (R\hat{\mathbf{r}}' - \boldsymbol{\mu}_r))}{\sqrt{2}\sigma} \tag{5.8c}$$

The surface integral over the sphere in Equations (5.6) and (5.7) is efficiently solved using Lebedev quadrature [87]. The outer integral over time is computed using any standard quadrature formulas such as Newton-Cotes [41],

Gauss-Konrod [33], and Clenshaw-Curtis [37]. In practice adaptive quadrature rule [109] or a composite rule is used where a quadrature rule is implemented over subintervals instead of using a high degree quadrature rule. A composite Newton-Cotes quadrature rule is used in this work.

If it is additionally assumed that the relative velocity of the two RSOs is above a threshold and most of the position uncertainty is in the along track direction, resulting in a brief encounter, a computationally fast linear collision probability method may be used. A comprehensive overview of the various linear methods is provided by Alfano [17]. The covariance matrices of the primary and the secondary RSOs are added to create a combined covariance matrix Eq. (5.3d) at the time of closest approach. The covariance ellipsoid of the position is located on either the primary or the secondary RSO. The velocities of the RSOs and their covariances are frozen at this instant and the combined covariance is located on the secondary RSO. The primary RSO is assumed to be a sphere with the combined radius of the two RSOs. As the sphere passes through the combined position covariance, it traces out a tube.

The covariance ellipsoid is a three-dimensional PDF and the collision probability is the integral of the PDF over the volume of the tube. The volume integral is assumed to be equivalent to the area integral of a two-dimensional PDF over a circle on the encounter plane [13] with the same radius as the tube. The encounter plane is by definition perpendicular to the plane formed by the vector from the primary to the secondary RSO, and the relative velocity. The encounter plane slices and takes a cross-section of the tube and the three-dimensional PDF at the time of closest approach to form the

circle and the two-dimensional PDF, respectively. Various methods are used to solve this two-dimensional integral. Patera's [105] method uses numerical quadratures and Alfano's [11] method uses error functions. Chan's [35] method makes further assumptions and converts the two-dimensional PDF into a one-dimensional Rician PDF, which is then approximately solved by using Bessel functions. Alfano's method is chosen because of its efficiency, robustness, and ease of implementation. The necessary steps and equations for computing the linear collision probability are now outlined for the sake of completeness and because an easy to implement summary is not available in literature.

The mean and the covariance of the two RSOs at TCA ( $t_{TCA}$ ) expressed in ECI Cartesian coordinates  $\mathcal{F}_{ECI}$ , is computed using a dense ephemeris or interpolation. The collision probability is then:

$$P_c = f_{Lin}(\boldsymbol{\mu}_P(t_{TCA}), \mathbf{P}_P(t_{TCA}), \boldsymbol{\mu}_S(t_{TCA}), \mathbf{P}_S(t_{TCA}), R) \quad (5.9)$$

where the computations inside  $f_{Lin}(\cdot)$  are found in Equations (5.10) - (5.20). A new encounter coordinate system  $\mathcal{F}_{ECS}$  is created where the  $y$ -axis is parallel to the relative velocity of the Primary RSO with respect to the Secondary RSO. The  $z$ -axis is perpendicular to both velocities, and the  $x$ -axis completes the frame. The perpendicular vector to both velocities  $\mathbf{n}$  is:

$$\mathbf{n} = \mathbf{v}_P(t_{TCA}) \times \mathbf{v}_S(t_{TCA}) \quad (5.10)$$

The unit axes of  $\mathcal{F}_{ECS}$  expressed in  $\mathcal{F}_{ECI}$  are:

$$\hat{\mathbf{Z}} = \hat{\mathbf{n}} \quad (5.11a)$$

$$\hat{\mathbf{Y}} = \hat{\mathbf{v}} \quad (5.11b)$$

$$\hat{\mathbf{X}} = \hat{\mathbf{Y}} \times \hat{\mathbf{Z}} \quad (5.11c)$$

Note that the unit axes computed above and the remaining computations till Eq. (5.20) are only computed at  $t_{TCA}$ . The rotation matrix from  $\mathcal{F}_{ECI}$  to  $\mathcal{F}_{ECS}$  is

$$R_{ECI}^{ECS} = [ \hat{\mathbf{X}} \quad \hat{\mathbf{Y}} \quad \hat{\mathbf{Z}} ]^T \quad (5.12)$$

The relative position vector of the Primary w.r.t. the Secondary and the combined position covariance in  $\mathcal{F}_{ECS}$  are:

$$\mathbf{r}_{PS}^{ECS} = R_{ECI}^{ECS} \mathbf{r} = [ X \quad Y \quad Z ]^T \quad (5.13)$$

$$\mathbf{P}^{ECS} = R_{ECI}^{ECS} \mathbf{P} (R_{ECI}^{ECS})^T = \begin{bmatrix} \sigma_X^2 & \rho_{XY} \sigma_X \sigma_Y & \rho_{XZ} \sigma_X \sigma_Z \\ \rho_{YX} \sigma_Y \sigma_X & \sigma_Y^2 & \rho_{YZ} \sigma_Y \sigma_Z \\ \rho_{ZX} \sigma_Z \sigma_X & \rho_{ZY} \sigma_Y \sigma_X & \sigma_Z^2 \end{bmatrix} \quad (5.14)$$

The plane defined by  $\hat{\mathbf{X}} - \hat{\mathbf{Z}}$  is the encounter plane. The separation vector and the combined covariance on the 2-dimensional reference frame on the encounter plane,  $\mathcal{F}_{EP}$ , are:

$$\mathbf{r}^{EP} = [ X \quad Z ]^T \quad (5.15)$$

$$\mathbf{P}^{EP} = \begin{bmatrix} \sigma_X^2 & \rho_{XZ} \sigma_X \sigma_Z \\ \rho_{ZX} \sigma_Z \sigma_X & \sigma_Z^2 \end{bmatrix} \quad (5.16)$$

To compute the probability on the encounter plane, a new 2-dimensional reference frame  $\mathcal{F}_{ES}$  is defined where the combined covariance matrix is diagonal. The rotation matrix is formed from the eigenvectors of  $\mathbf{P}_C^{EP}$ . The eigenvalues



and associated eigenvectors pairs of  $\mathbf{P}_C^{EP}$  are  $(\lambda_1, \boldsymbol{\xi}_1)$  and  $(\lambda_2, \boldsymbol{\xi}_2)$  with  $\lambda_2 \geq \lambda_1$ . The rotation matrix from  $\mathcal{F}_{EP}$  to  $\mathcal{F}_{ES}$  is:

$$R_{EP}^{ES} = [ \boldsymbol{\xi}_1 \quad \boldsymbol{\xi}_2 ] \quad (5.17)$$

In  $\mathcal{F}_{ES}$ , the relative position and the combined position covariance are:

$$\mathbf{r}^{ES} = R_{EP}^{ES} \mathbf{r}^{EP} = [ x \quad y ]^T \quad (5.18)$$

$$\mathbf{P}^{ES} = R_{EP}^{ES} \mathbf{P}^{EP} (R_{EP}^{ES})^T = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \quad (5.19)$$

The following integral is evaluated to compute the collision probability:

$$P_c = \frac{1}{2\pi\sigma_x\sigma_y} \int_{-R}^R \int_{-\sqrt{R^2-x'^2}}^{\sqrt{R^2-x'^2}} \exp \left\{ -\frac{1}{2} \left[ \left( \frac{x' - x}{\sigma_x} \right)^2 + \left( \frac{y' - y}{\sigma_y} \right)^2 \right] \right\} dy' dx' \quad (5.20)$$

In Alfano's method [11], which is used in this work, the 2-dimensional integration in Eq. (5.20) is converted to a 1-dimensional integral using the error function. The 1-dimensional integration is converted to a series expression, which Alfano solves separately for the odd and even terms for better numerical accuracy. For the sake of brevity, the series solution is not presented here.

In a Fortran implementation of the two collision probability methods, the time taken to compute Eqs. (5.10) - (5.20) for the linear collision takes approximately  $7.6 \times 10^{-6}$  seconds. Computing the integrand  $\dot{P}_I(t)$  from Eq. (5.7) takes approximately  $9.4 \times 10^{-4}$  and  $2.8 \times 10^{-3}$  seconds for a Lebedev quadrature of 590 and 1730 nodes, respectively. It is emphasized that the main contribution of this chapter is to use MGMMs and therefore, any collision probability computation method works with Gaussian distributions can be used.

### 5.2.3 Collision Probability for MGMMs

Although state uncertainty expressed in orbit elements remains Gaussian for times of flight compared to Cartesian coordinates, minimum separation for close approaches need to be computed for the collision probability. Therefore, the RSO samples need to be in Cartesian coordinates during MC simulations. When MGMMs are used for uncertainty propagation in Phase I, millions of samples can be drawn from the MGMMs to use with MC simulation in Phase II. To reduce the computational burden, either the linear method (Eqs. (5.10) - (5.20)) or Coppola's method (Eqs. (5.4) - (5.8)) for collision probability computation are used. For either of the three options, the states have to be represented in Cartesian coordinates during Phase II. For the propagation in Phase I, however, any orbit element set may be chosen to represent the state uncertainty.

The computation methodology used in this work for computing the collision probability using MGMMs is now outlined. MGMMs in Cartesian coordinates are used in both Phase I and Phase II.

#### 5.2.3.1 Chose Splitting Directions

Find the splitting directions so that the initial Gaussian PDFs of both RSOs are converted to MGMMs in Cartesian coordinates:

$$p_P(\mathbf{x}(t)) = \sum_{i=1}^{N_P} \alpha_{P_i} p_g(\mathbf{x}(t); \boldsymbol{\mu}_{P_i}(t), \mathbf{P}_{P_i}(t)) \quad (5.21a)$$

$$p_S(\mathbf{x}(t)) = \sum_{j=1}^{N_S} \alpha_{S_j} p_g(\mathbf{x}(t); \boldsymbol{\mu}_{S_j}(t), \mathbf{P}_{S_j}(t)) \quad (5.21b)$$

The nonlinearity measure from Eq. (2.25) on page 51 is used to find the optimal splitting directions for both RSOs. The nonlinear function  $\mathbf{f}$  is the propagation of the orbital state in Cartesian coordinates from epoch to the time of closest approach of the nominal states. All computations are in normalized coordinates where the distance unit (DU) is the radius of the Earth and the time unit (TU) is computed such that the gravitational parameter of the Earth is equal to 1. The scalar sensitivity parameter is the norm of the 6-dimensional nonlinearity vector from Eq. (2.25),  $\|\phi\|_2$ . Note that the norm operator implies the importance of using normalized units, such that the nominal position and velocity magnitudes have the same order of magnitude.

For the initial uncertainty in Cartesian coordinates, 14 possible splitting directions that are investigated in this work are: inertial coordinates  $\mathbf{r}_x, \mathbf{r}_y, \mathbf{r}_z, \mathbf{v}_x, \mathbf{v}_y, \mathbf{v}_z$  (corresponding to the spectral directions of the diagonal covariance matrix),  $\mathbf{r}_r, \mathbf{r}_v, \mathbf{r}_h, \mathbf{r}_t, \mathbf{v}_r, \mathbf{v}_v, \mathbf{v}_h$ , and  $\mathbf{v}_t$ . The last 8 directions split the position and velocity along the unit radial ( $\mathbf{r}$ ), velocity ( $\mathbf{v}$ ), cross-track ( $\mathbf{h}$ ), and in-track ( $\mathbf{t} = \mathbf{h} \times \mathbf{r}$ ) directions. The directions with higher nonlinearity values benefit more from splitting. The number of splits is dependent on the available computational budget since increasing the number of elements better approximates the final non-Gaussian distribution. The standard deviation of the uncertainty along an arbitrary unit vector direction  $\hat{\mathbf{u}}$ ,  $\sigma_{\|\hat{\mathbf{u}}}$ , is:

$$\sigma_{\|\hat{\mathbf{u}} = \|\mathbf{S}^{-1}\hat{\mathbf{u}}\|_2^{-1} \quad (5.22)$$

where  $\mathbf{S}$  is any square-root of the covariance matrix, such as the Cholesky factor. Using Eq. (2.25) to chose the optimal splitting directions is shown in

Section 5.3.1 for a Molniya and a Sun-synchronous orbit.

### 5.2.3.2 Propagation

All the elements of both MGMMs are propagated using any desired Gaussian propagation method till the time window of interest (Phase II in Figure 5.1). The second order Divided difference method (DD2) is used in this work to propagate the mean and covariance of each MGMM element through the nonlinear orbit dynamics using 13 sigma points (orbit propagations) per MGMM element. Higher fidelity propagation of each element is possible by using more sigma points [9].

### 5.2.3.3 Compute Collision Probability

The collision probability during Phase II ( $t \in [t_0, t_f]$ ) is now computed using a weighted all-on-all approach between all the elements of both RSO MGMMs. When Coppola's quadrature is used to compute the collision probability, the all-on-all method translates to:

$$P_0 = \int_0^R \sum_{i=1}^{N_P} \sum_{j=1}^{N_S} \alpha_{P_i} \alpha_{S_j} \frac{dP_0(\boldsymbol{\mu}_{P_i}(t_0), \mathbf{P}_{P_i}(t_0), \boldsymbol{\mu}_{S_j}(t_0), \mathbf{P}_{S_j}(t_0))}{dr'} dr' \quad (5.23)$$

$$P_I = \sum_{k=1}^{N_q} w_k \left\{ \sum_{i=1}^{N_P} \sum_{j=1}^{N_S} \alpha_{P_i} \alpha_{S_j} \dot{P}_I(\boldsymbol{\mu}_{P_i}(t_k), \mathbf{P}_{P_i}(t_k), \boldsymbol{\mu}_{S_i}(t_k), \mathbf{P}_{S_i}(t_k)) \right\} \quad (5.24)$$

where  $w_k$  and  $t_k$  are the  $N_q$  weights and nodes, respectively for a quadrature rule from  $t_0$  to  $t_f$ . In Eq. (5.24),  $dP_0/dr'$  and  $\dot{P}_I$  are computed from Eqs. (5.6) and (5.7), respectively. At the beginning of Phase II, the two RSOs are usually far enough apart that  $P_0 = 0$ . All the MGMM elements of both RSOs

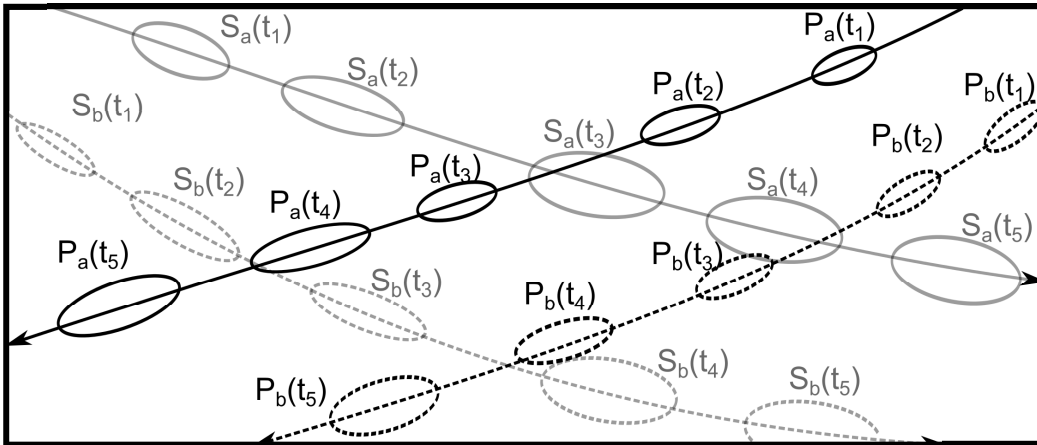
are propagated through time and the double sum inside the curly braces in Eq. (5.24) is evaluated at each  $t_k$ , which is found from the quadrature rule in time, as shown in Figure 5.3a.

For the linear collision method, the probability is computed in an all-on-all way as:

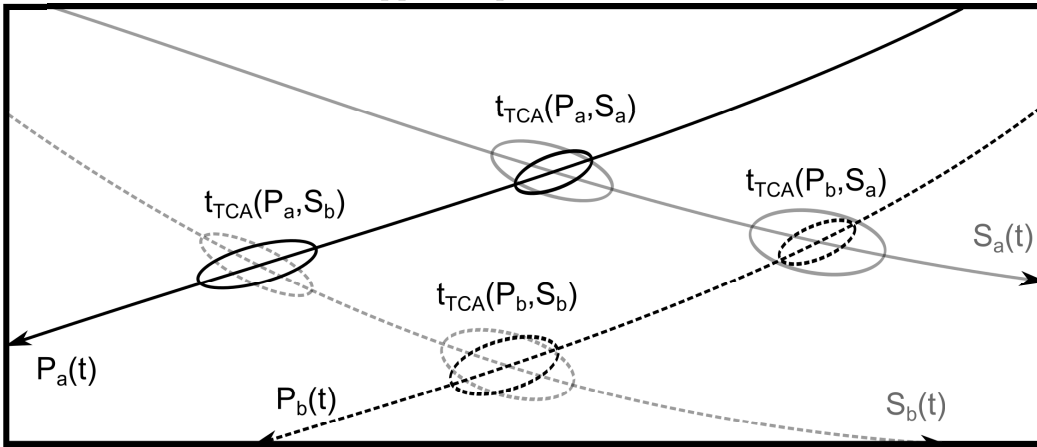
$$P_c = \sum_{i=1}^{N_P} \sum_{j=1}^{N_S} \alpha_{P_i} \alpha_{S_j} f_{Lin} \left( \boldsymbol{\mu}_{P_i}(TCA(i, j)), \mathbf{P}_{P_i}(TCA(i, j)), \boldsymbol{\mu}_{S_j}(TCA(i, j)), \mathbf{P}_{S_j}(TCA(i, j)), R \right), \quad (5.25)$$

where the computations inside  $f_{Lin}(\cdot)$  are found in Equations (5.10) - (5.20).  $TCA(i, j)$  is the time of closest approach  $t_{TCA}$  between element  $i$  of the Primary RSO's MGMM and element  $j$  of the Secondary RSO's MGMM as seen in Figure 5.3 b.

It is emphasized that using MGMMs only extends the two fast collision probability computation methods (Coppola's quadrature and Linear collision) to non-Gaussian distributions. All the other assumptions built into the theories remain i.e. independence of the two RSOs' distributions in Cartesian coordinates. The additional constraint of a fast encounter for the linear collision method still remains. Splitting the initial distribution of an RSO along the velocity direction results in a small spread of the  $t_{TCA}$  with slightly different encounter geometries over Phase II in case the encounter is relatively long. However, spreading out the  $t_{TCA}$  of the MGMM elements is not guaranteed to work if the true distributions are additionally non-Gaussian. Even if the encounter of the nominal trajectories of the RSOs is fast, the full non-Gaussian PDFs of the RSOs might not have a fast encounter.



a. Coppola's quadrature method



b. Linear collision method

Figure 5.3: The all-on-all approach for collision probability computation using 2 element MGMMs for the primary and the secondary RSOs

## 5.3 Results

Using GMMs to propagate the uncertainty of RSOs for computing the probability of collision is demonstrated in this section. The utility of picking the correct splitting directions using the sensitivity measure is first shown for propagating the uncertainty in Cartesian coordinates during Phase I. Finally, GMMs and MGMMs are used to compute the collision probability for some test cases.

### 5.3.1 Perturbed Uncertainty Propagation

The importance of the splitting direction is shown by propagating the state and uncertainty of a Molniya and a Sun-synchronous orbit for 5 days with the initial conditions from Table 5.1. The perturbations included are: a  $33 \times 33$  degree and order spherical harmonics gravity potential field, drag from an exponential atmosphere model, SRP, and luni-solar attraction. Both orbits have an initial uncertainty with a standard deviation of 10 m in the position coordinates and 10 m/s in the velocity coordinates [75]. An MC simulation using 100,000 points is used as the benchmark solution. The accuracy of a GMM with 19 elements is plotted in Figure 5.4 against the sensitivity measure of the splitting directions for both orbits.

The accuracy of the propagated GMM with respect to the benchmark MC run is expressed as the log-likelihood of the  $M$  propagated MC points being generated by the  $N$ -element GMM [126].

$$LL = \sum_{i=1}^M \log \left( \sum_{j=1}^N \alpha_j p_g(\mathbf{x}_i; \boldsymbol{\mu}_j, \mathbf{P}_j) \right) \quad (5.26)$$

The log-likelihood and sensitivity  $\|\phi\|_2$  of the various splitting directions from Eq. (2.25) for the Molniya and Sun-synchronous orbits are found in Figure 5.4. The function in Eq. (2.25),  $\mathbf{f}$ , is the final 6-dimensional state in Cartesian coordinates of the RSO after propagation and the plotted sensitivity value is the norm of this vector. The most effective splitting directions (highest LL) have the largest sensitivity measure direction ( $x$ -axis). In orbit propagation scenarios, most of the non-linearity lies along a few directions only. For the two orbits presented here, splitting along the velocity direction  $\mathbf{v}_v$  results in the most accurate GMM in Cartesian coordinates. Splitting along any other direction results in a much lower LL, and the effect of splitting is not noticeable. For both orbits, the benefit of splitting along any of the seven directions in the position space is negligible and the sensitivities amongst the various directions are indistinguishable. The directions in velocity space that are almost parallel for the Molniya orbit are  $\mathbf{v}_y$  with  $\mathbf{v}_v$ , and  $\mathbf{v}_z$  with  $\mathbf{v}_h$ . The velocity vector is almost entirely along the  $y$ -axis and therefore, splitting along  $\mathbf{v}_y$  results in only a slightly worse result compared to  $\mathbf{v}_v$ . The directions in position space that are almost parallel for the Sun-synchronous orbit are  $\mathbf{v}_v$  with  $\mathbf{v}_t$ . The consideration and selection of split directions (amongst the infinite options available) is an important contribution of this paper. Existing methods simply choose the direction with the highest eigenvalue or the largest norm of the column of the square-root matrix.



Table 5.1: Initial conditions for the uncertainty propagation runs with a standard deviation of 10 m and 1 m/s in each of the position and velocity coordinates, respectively

Orbital Parameter	Molniya	Sun-synchronous
Semi-major axis ( $a$ )	26562.00 km	7172.57 km
Eccentricity ( $e$ )	0.741	$1.12 \times 10^{-3}$
Inclination ( $i$ )	$63.40^\circ$	$98.62^\circ$
Longitude of ascending node ( $\Omega$ )	$85^\circ$	$253.73^\circ$
Argument of periapsis ( $\omega$ )	$-90^\circ$	$77.60^\circ$
True anomaly at epoch ( $\nu$ )	$0^\circ$	$59.21^\circ$

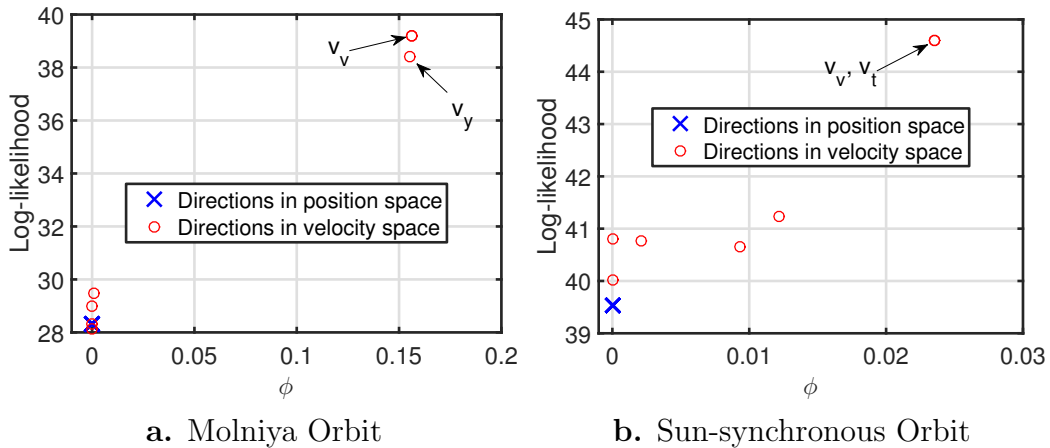


Figure 5.4: Sensitivity and accuracy of the splitting directions using a 19 element split

### 5.3.2 GMM Test Case

One of the twelve cases in Alfano's [17] paper, Case 7, is used to show the utility of using GMM's to compute the collision probability. Case 7 has two RSOs in LEO and does not work with the GMM linear collision equations because the PDF at TCA is non-Gaussian and the relative velocity is below the threshold specified by Alfano [17]. An accurate probability is also not computed when the quadrature method is used without GMMs since the PDFs are additionally non-Gaussian during Phase II. Therefore, the quadrature method is used with GMMs to accurately compute the probability of collision.

The nonlinearity measure for the full state is tested along all the important directions to find the preferred splitting direction. For both the primary and secondary RSOs, the highest nonlinearity corresponds to splitting along  $\mathbf{r}_r$ . Converting the initial distribution of both the primary and the secondary RSOs into GMMs by splitting along the  $\mathbf{r}_r$  direction provides an accurate estimate of the collision probability. Figure 5.5 shows that increasing the total number of elements increases accuracy of the computation. The number of propagations is the sum of elements of the primary and the secondary GMMs multiplied by 13 because each GMM element requires 13 sigma points when DD2 is used for propagation of a 6-dimensional uncertainty distribution.

Alfano's test cases 1, 2, 4, and 10 also do not work with the linear collision method. They were tested using GMMs and since the distribution remains Gaussian during the entire conjunction, they require the quadrature method and not GMMs.

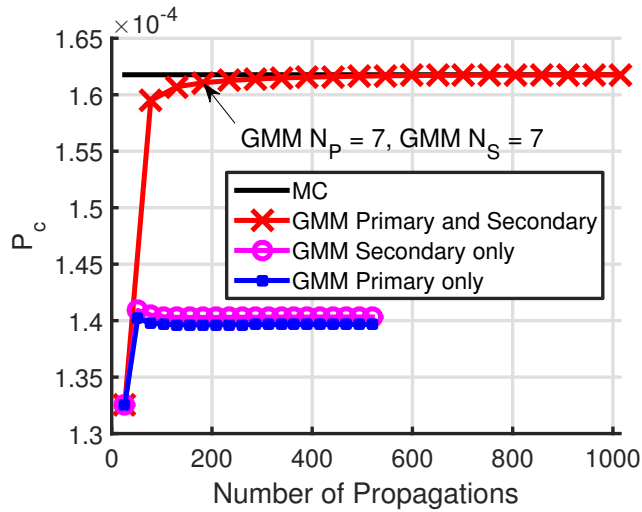


Figure 5.5: Collision probability performance using GMMs for Alfano Case 7 and the quadrature method

### 5.3.3 MGMMs

The previous test case requires splitting the initial Gaussian PDF of the RSOs along only one direction. Two cases using the two-body dynamics are presented in this section, which require an MGMM instead of a GMM to compute an accurate collision probability. The two cases are:

**LEO** Case 7 from Alfano with two RSOs in LEO and added uncertainty along the in-track velocity direction

**GEO** One RSOs in GEO and one in GTO with TCA of approximately 2.5 days after epoch

An MC simulation with  $10^8$  independent cases is used to generate the benchmark. Equation 5.1 estimates the minimum number of samples needed to

reach a factor of  $\epsilon$  error and a confidence factor of  $\gamma$ . The equation is rearranged to solve for an approximation of the error percentage  $\epsilon$  with a 95% ( $\gamma = 0.95$ ) confidence when the number of MC samples is  $10^8$ .

$$\epsilon^2 = \frac{4(e-2)((1-P_T)P_T)}{n_{MC}P_T^2} \log\left(\frac{2}{1-\gamma}\right) \approx \frac{4(e-2)((1-P_C)P_C)}{10^8P_C^2} \log(40) \quad (5.27)$$

The value of  $\epsilon$  depends on the MC collision value, and is shown as the  $\pm 95\%$  bounds in Figures 5.6, 5.7, and 5.8.

For the LEO case, the standard deviation of the in-track velocity of the primary RSO is increased by a factor of 30 compared to Alfano's test case 7. Since the encounter geometry is not changed, the quadrature method is still the appropriate choice. The initial conditions of the secondary RSO are not altered, so the initial Gaussian uncertainty is once again split along only the  $\mathbf{r}_r$  direction. However, an MGMM is required for the primary RSO to fully capture the non-Gaussian behavior. The largest value in the sensitivity test on the primary RSO is the  $\mathbf{v}_v$  direction. The second largest value approximately halfway between the smallest value of the sensitivity measure and the  $\mathbf{v}_v$  direction is the  $\mathbf{r}_r$  direction. Therefore, four splitting strategies are investigated:

**Strategy A** Secondary:  $N \parallel \mathbf{r}_r$  , Primary:  $(N \parallel \mathbf{v}_v) \times (\text{odd}(N/2) \parallel \mathbf{r}_r)$

**Strategy B** Secondary:  $N \parallel \mathbf{r}_r$  , Primary:  $(N \parallel \mathbf{v}_v) \times (N \parallel \mathbf{r}_r)$

**Strategy C** Secondary:  $N \parallel \mathbf{r}_r$  , Primary:  $(N \parallel \mathbf{v}_v)$

**Strategy D** Secondary:  $N \parallel \mathbf{r}_r$  , Primary:  $(N \parallel \mathbf{r}_r)$

where  $odd(N)$  computes the closest odd number to  $N$  because the univariate library used consists of an odd number of elements only.

$$odd(N) = 2 \times \text{ceiling}(N/2) - 1 \quad (5.28)$$

The probability of collision as a function of the total number of propagations in the GMMs of both the RSOs for the four strategies is seen in Figures 5.6 and 5.7 for  $R$  of 10 m and 20 m, respectively. The number of propagations is the sum of elements of the primary and the secondary GMMs multiplied by 13 because each GMM element requires 13 sigma points when DD2 is used for propagation. Only the MGMM strategies A and B are able to converge and pierce the 95% confidence bounds of the MC simulation. It is sufficient to use half as many elements in the  $\mathbf{r}_r$  direction of the primary RSO, as predicted by the nonlinearity measure.

The initial states and the uncertainties for the GEO and GTO case are found in Tables 5.2 and 5.3, respectively. The orbit of the primary GEO RSO is assumed to be derived from optical observations, right ascension (RA) and declination (DEC), which in general create larger errors in the range direction for GEO objects. The encounter is brief and therefore, the linear collision probability method is used with GMMs. Due to the highly nonlinear and elliptic motion of the secondary RSO, an MGMM is required to describe the uncertainty of the secondary RSO only.

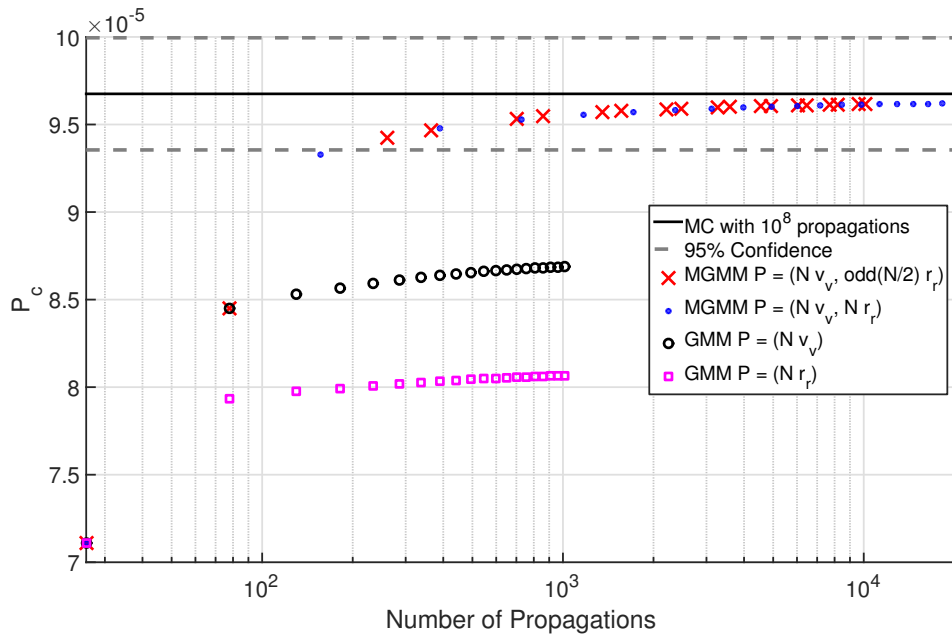


Figure 5.6: Collision probability performance using MGMMs for the LEO case with  $R = 10$  m and the quadrature method, compared to a MC simulation with  $10^8$  cases ( $2 \times 10^8$  propagations)

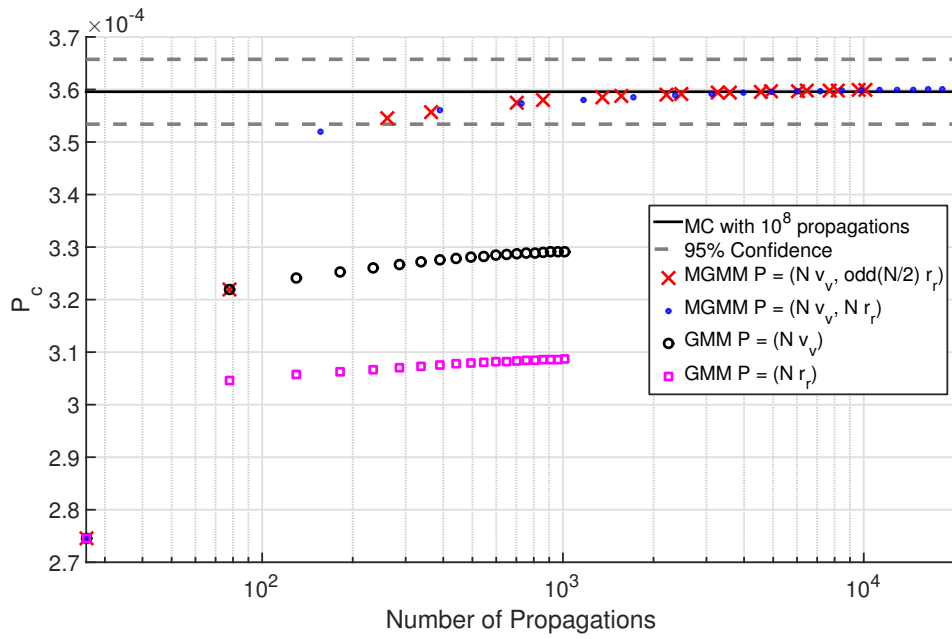


Figure 5.7: Collision probability performance using MGMMs for the LEO case with  $R = 20$  m and the quadrature method, compared to a MC simulation with  $10^8$  cases ( $2 \times 10^8$  propagations)

Table 5.2: State conditions in the ECI frame at epoch ( $t_{CA}$ -2.5 orbital periods of the primary RSO) for the for the GEO (Primary) and GTO (Secondary) RSOs

	Primary	Secondary
$x$ [km]	42057.9	-24374.96499496852
$y$ [km]	0	-16016.93987982133
$z$ [km]	0	0
$v_x$ [km/s]	0	3.25222774265
$v_y$ [km/s]	3.0800809759824	-0.62421144888
$v_z$ [km/s]	0	0

Table 5.3: State uncertainty in the RIC frame at epoch ( $t_{CA}$ -3 days) for the GEO (Primary) and GTO (Secondary) RSOs

	Primary	Secondary
$\sigma_{\mathbf{r}}$	5 km	1 m
$\sigma_{\mathbf{t}}$	0.1 km	1 m
$\sigma_{\mathbf{h}}$	0.1 km	5 m
$\sigma_{\mathbf{vr}}$	0.12 m/s	4 m/s
$\sigma_{\mathbf{vt}}$	0.6 m/s	4 m/s
$\sigma_{\mathbf{vh}}$	0.025 m/s	0.1 m/s



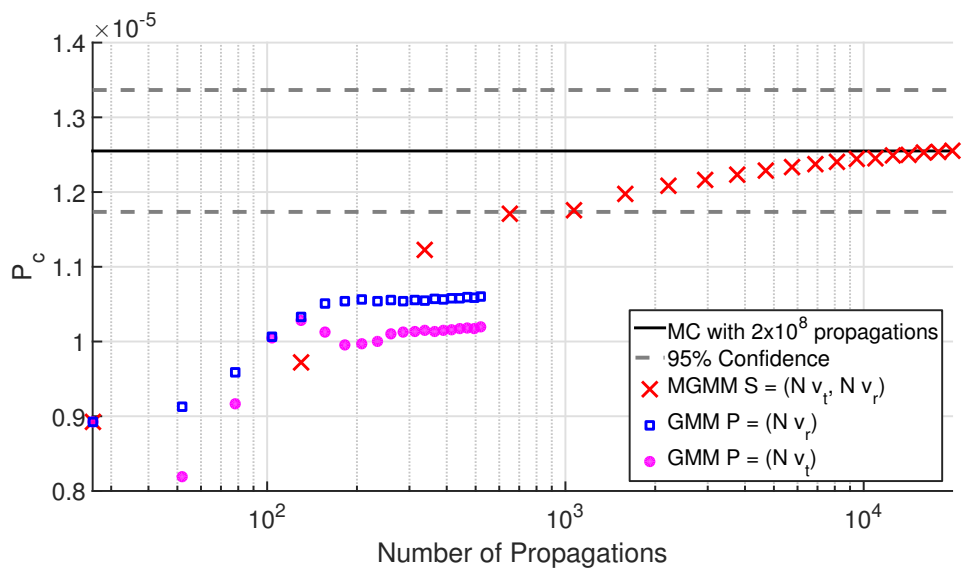


Figure 5.8: Collision probability performance using MGMMs for the GEO case with  $R = 40$  m and the linear method, compared to an MC simulation with  $2 \times 10^8$  cases ( $4 \times 10^8$  propagations)

Due to the high nonlinearity in the  $\mathbf{v}_t$  and  $\mathbf{v}_r$  directions of the secondary RSO, three splitting strategies are investigated:

**Strategy A** Secondary:  $(N \parallel \mathbf{v}_t) \times (N \parallel \mathbf{v}_r)$

**Strategy B** Secondary:  $(N \parallel \mathbf{v}_r)$

**Strategy C** Secondary:  $(N \parallel \mathbf{v}_t)$

The probability of collision as a function of the total number of elements in the GMMs of both the RSOs for the four strategies is shown in Figure 5.8 for  $R$  of 40 m. This is analogous to a case where the RSO with the small uncertainty is an active space asset and its location is desired to be kept secret from other objects. Only the MGMM Strategy A is able to converge by piercing the 95% confidence bounds of the MC simulation using a total of 1,000 propagations.

## 5.4 Conclusion

MGMMs and GMMs provide a simple and computationally efficient way of capturing the evolution of uncertainties. Expressing the state uncertainty in orbit elements keeps the state uncertainty closer to a Gaussian distribution for longer times of flights. However, collision probability computations require the state uncertainty of both space objects in Cartesian coordinates. The computationally efficient collision probability computation methods, i.e., linear conjunction and Coppola's quadrature method additionally require the state uncertainty to be Gaussian in Cartesian coordinates. With MGMMs,

these efficient conjunction techniques are extended to non-Gaussian distributions. The highly nonlinear conjunctions used as examples in this chapter demonstrate that MGMMs show a savings of four orders of magnitude compared to MC, i.e., the same accuracy is attained using  $10^3$  to  $10^4$  propagations for MGMMs instead of  $10^8$  for MC.

A multivariate Gaussian distribution is converted to a GMM by applying the univariate splitting library from Chapter 2 along a desired direction. The uncertainty, i.e., variance along the splitting direction is reduced and the means lie on the vector of the splitting direction passing through the mean of the original Gaussian distribution. Due to the nonlinearity of orbit dynamics, splitting the initial Gaussian uncertainty along only one direction may not sufficiently capture the final non-Gaussian distribution of the uncertainty after propagation. Splitting along multiple directions using a tensor product of the univariate splitting libraries results in an MGMM, which reduce the uncertainty along multiple directions of the original Gaussian distribution . It is important to split the initial Gaussian state uncertainty of an RSO along the directions that most affect the extent of the non-Gaussian behavior of the propagated state uncertainty. This effect is quantified using the second-order divided difference of the state in normalized coordinates after propagation. Selected initial states are propagated through the non-linear dynamics to find the directions that have the most influence on the extent of the non-Gaussian character of the final PDF. A higher nonlinearity value corresponds to a larger benefit due to splitting along that direction.

Long and short duration conjunction scenarios using two-body dynam-

ics demonstrate that splitting the initial multivariate Gaussian distributions of the RSOs along one or two of the most influential directions accurately computes the collision probability using orders of magnitude fewer function propagations compared to an MC simulation. The method is simple to implement without converting to orbit elements, adapting the weights, or adaptively splitting the MGMM elements mid-propagation. Therefore, the method works using black box propagation functions, an important feature when considering high fidelity and broad use applications.

## Chapter 6

### Conclusions

The aim of this dissertation is uncertainty propagation pertaining to Resident Space Objects (RSOs) and the computation of the collision probability, which are both essential for Space Situational Awareness. Accurate propagation techniques developed here have advanced the state of the art and are applicable in many fields where uncertainty propagation or quantification is required for nonlinear problems. The second half of the dissertation uses the uncertainty representation of the RSOs to accurately and efficiently compute the collision probability. An outline of the work and the major contributions are presented in Section 1.6 and the resulting publications are shown in Appendix A. The major conclusions for the uncertainty propagation in Phase I (Figure 1.7) and the collision probability computation in Phase II are summarized here and some avenues of future work are considered.

#### 6.1 Uncertainty Propagation

The initial Gaussian uncertainty of the RSOs is first approximated by a Gaussian Mixture Model (GMM). Multidirectional Gaussian Mixture Models (MGMMs) are constructed, and their implementation is detailed in Chapter 2. Univariate splitting libraries are used to reduce the uncertainty of the initial

distribution along a specified direction. Univariate splitting libraries of up to 39 elements that are carefully fit in advance are generated by minimizing the  $L_2$  distance with respect to a standard normal distribution. Instead of applying the univariate library along a single direction, the univariate library is applied recursively along multiple directions forming a regular grid over multiple dimensions. The MGMMs approximate the propagated multivariate non-Gaussian distribution more accurately than if the split is made along only one direction. A second-order divided difference is used to measure nonlinearity along directions of interest. Thus, the directions that benefit most from splitting are identified and can be exploited. The nonlinearity merit provides a ratio to relate the number of splits along each direction. Therefore, a practitioner can implement the MGMM technique with the selection of only one fidelity tuning parameter. A user enforces a number of splits along a single direction, then the number of splits in the other directions can be chosen according to their relative nonlinearity. A threshold for the weight further reduces the computational load by ensuring that the MGMM only contains elements above a certain weight.

In Chapter 3, PCE is used to propagate each element of the GMM. PCE is a higher order interpolant compared to the Gaussian approximation and therefore, each element better approximates the true non-Gaussian distribution after the propagation. Splitting the initial Gaussian distribution into a GMM with the means of the weighted Gaussian distributions along the most influential direction (initial velocity vector for the Cartesian Coordinates cases and semi-major axis for the EE case presented here) reduces the uncertainty

along that direction. Therefore, a more accurate description of the propagated uncertainty is found at a lower computation cost, compared to simply increasing the order of the PCE, which is  $p$ -refinement. However, the GMM-PC technique is recommended for scenarios where a PCE does not converge, as is the case for orbit problems with large propagation times, large initial uncertainties, and highly eccentric orbits. The weighted least squares method for computing the coefficients of the PCE is capable of reusing the function evaluations so that accuracy of the GMM-PC method can be increased without increasing the computational load.

A distinction is made between local methods of uncertainty propagation, such as State Transition Tensors (STTs), and the super-local methods developed in this dissertation. STTs are local approaches because high order derivatives of the function are only computed at the mean of the initial uncertainty distribution. Therefore, information about the nonlinear function is only available at one location and all the constraints for Taylor series expansions apply. An implicit assumption is made that the function is smooth and continuous over the domain of the initial uncertainty. The super-local uncertainty propagation methods slightly relax the assumptions. Even the standard sigma-point methods such as the Unscented Transform (UT) and the Divided Difference Transform (DD) are super-local because the function is uniformly sampled in the  $2\sigma$  range. The sigma-point methods and the STTs have the same order of accuracy if only the mean and covariance are propagated. The first-order DD (DD1) has a similar accuracy to using a first-order STT for mean and covariance propagation. The UT and the second order DD

(DD2) have an accuracy comparable to using STTs of up to order 2 and 3, depending on the specific problem. The MGMMs and the GMM-PC methods developed in this dissertation use a denser sampling strategy to give a more accurate representation of the entire post-propagation PDF at the cost of a higher computational load.

The advantage of using a local STT method is that they do not depend on the initial uncertainty distribution, but only on the location of the mean. The size and shape of the initial distribution can be changed, i.e. the covariance matrix of a Gaussian distribution, and the uncertainty can be propagated without recomputing the STTs. The sample points of the super-local methods also depend on the initial uncertainty distribution and therefore, new samples have to be selected and propagated through the function if the initial uncertainty distribution is changed. The main advantages of the super-local methods developed here are that they are easier to use, especially with black box functions, and information about the nonlinear function is used at multiple locations within the initial uncertainty domain. Recently, Fujimoto and Scheeres [53] have combined GMMs with STTs but the STTs are used as a surrogate model to replace the underlying nonlinear function by its Taylor series expansion.

## 6.2 Collision Probability Computation

A Monte Carlo simulation to compute the probability of collision is very general and provides accurate results for any conjunction scenario. Samples from any uncertainty propagation methods, including MGMMs and GMM-



PC, can readily be used with Monte Carlo. Although millions of samples are required to compute the collision probability, the use of a Graphics Processing Unit provides one to two orders of magnitude speedups when compared to a serial CPU implementation for all the scenarios and this technique is feasible on a single workstation without using a top of the line GPU.

The PDFs of the RSOs can even be combined to a joint PDF of a 12 dimensional state in order to handle collision probability between RSOs that are not independent. The samples are integrated forward and backwards in time so that the closest separation over the entire encounter window per sample pair, i.e. one sample from the primary RSO and one from the secondary RSO, is accurately computed. Fixed step RK4 integration ensures that the computational load is uniform for all the samples in the MC simulation and reduces the memory lead required per run in order to optimize the computation on the GPU. A one sided Hermite interpolation is used to find the closest separation between the realization of the two RSOs during each step. A maximum time step size of 10 s was shown to accurately compute the minimum separation to an accuracy of less than 1 m.

Collision probability computations are always carried out in Cartesian coordinates. The computationally efficient collision probability computation methods, i.e., linear conjunction and Coppola's quadrature method additionally require the state uncertainty to be Gaussian in Cartesian coordinates. With MGMMs, these efficient conjunction techniques are extended to non-Gaussian distributions. The highly nonlinear conjunctions used as examples in Chapter 5 demonstrate that MGMMs show a savings of four orders of mag-

nitude compare to MC, i.e., the same accuracy is attained using  $10^3$  to  $10^4$  propagations for MGMMs instead of  $10^8$  for MC.

The initial Gaussian state uncertainty of an RSO is only split along the directions that most affect the extent of the non-Gaussian behavior of the propagated state uncertainty. This effect is quantified using the second-order divided difference from Chapter 2 of the state in normalized coordinates after propagation. Selected initial states are propagated through the nonlinear orbit dynamics to find the directions that have the most influence on the extent of the non-Gaussian character of the final PDF.

### **6.3 Future Work**

In the current study GMMs and MGMMs have been used to propagate uncertainty. The next logical step is to use the GMMs in a Bayesian Filtering framework. Therefore, process noise and measurement updates using measurements and measurement noise should be added. GMMs have been previously used for filtering applications as a Gaussian Sum Filter (GSF). However, the implementation has been as an extension of a Kalman filter. The future goal is to approach the filtering procedure as a particle filter where each particle is a GMM element. First MGMMs should be constructed for measurements that are known to have non-Gaussian noise. Efficient resampling techniques should be investigated where the particles are Gaussian distributions, and finally merging algorithms should be developed that combine similar elements into one larger element. The goal is to construct a filter with a sliding dial where higher accuracy is achieved by increasing the computation load. The

extreme with the lowest computational load is a one element GMM which is the standard EKF or UKF. The other extreme has a computational load approaching that of a Particle Filter but with thousands to millions of Gaussian elements.

Only univariate splitting libraries have been used so far to create GMMs. The extension to an MGMM was a tensor product. Future work should involve computing multivariate libraries that directly approximate the multivariate IID Gaussian distribution. The goal is to reduce the required number of elements for an uncertainty propagation problem by removing the extra terms that result as part of a tensor product.

The Gaussian elements that still exhibit nonlinear behavior and could be split further in the MGMM simulations, should be identified. The DD2 propagation provides an easy check because it is capable of producing both the first and the second order estimate of the mean and the covariance. Although the second order estimate should be used, the difference can be compared with the first order estimate. The difference could either be the  $L_2$  norm or the entropy, which both have closed form solutions for Gaussian distributions. In case the difference is above a certain minimum, the pre-propagation element should be split further. The nonlinearity measure can be computed from the sigma points and therefore, the direction of splitting is also easily determined because the nonlinearity measure is readily available.

The GMM weights post-propagation can be updated slightly to better match the mean and the covariance of the true non Gaussian distribution.

Although higher order sigma point filters do not capture the overall non Gaussian post-propagation PDF as well GMMs because of the Gaussian constraint, they excel at estimating the true mean and covariance. A Conjugate Unscented Transform (CUT) or a high order cubature propagation will provide a more accurate estimate of the mean and the covariance. The weights of the GMM should be updated using an optimization process such that the mean and covariance match the values from the higher order sigma-point method.

The accurate surrogate model provided by the PCE and the GMM-PC are used to efficiently generate samples from the post-propagation PDF. However, there is no closed form function which is an approximation of the PDF itself. An MGMM can be used for propagation using the surrogate model instead of the original function. Since the surrogate model generated by the PCE or GMM-PC is a close representation of the original function, the propagation of the MGMM elements will be fast, accurate, and provide a closed-form approximation of the PDF.

Apart from the function evaluations, solving the linear system of equations for the coefficients and generating samples from the PCE is also computationally intensive when the polynomial order, or the number of samples is large. These computations can be parallelized on a Graphics Processing Unit (GPU) to investigate if there are any possible speedups.

The multivariate orthogonal polynomials used in this dissertation have the same order in all the univariate directions. With the GMM-PC, the size of the uncertainty in the splitting direction is reduced when the univariate

splitting library is applied. Therefore, multivariate orthogonal polynomials could be constructed that have a lower univariate polynomial order along the splitting direction.

When the GMM-PC solution converges, the accuracy is limited by the initial approximation error caused by the splitting of the initial multivariate Gaussian distribution. Reducing the error of the univariate library with respect to the standard normal distribution will reduce the initial splitting error. A first step will be to take the present homoscedastic libraries and then let the individual variances vary to reduce the approximation error.

New orbit elements should be constructed that also absorb perturbations akin to the  $J_2$  Equinoctial Elements [23]. Vinti's potential theory analytically describes motion for RSOs due to the added perturbation of  $J_2$ ,  $J_3$ , and a part of  $J_4$  [129]. The uncertainty expressed in mean elements constructed using Vinti's theory might reduce the extent of the non-Gaussian behavior for a longer time of flight. The nonlinearity in the propagation, which is important for both GMM-PC and MGMMs, is reduced for perturbed orbits.

The MC method could be extended to handle other parameter uncertainties such as the gravity parameter, ballistic coefficient, and the SRP characteristics. Adding these parameter will increase the dimension of the states and result in extra memory load for the GPU. Therefore, the speedups might decrease and the effect of the added uncertainties on the collision probability is only noticeable on long duration encounters. The present speedup analyses are from an NVIDIA GPU with compute capability 2.0 and the computation

for the low fidelity dynamics is limited by the availability of fast access memory, i.e. registers. Therefore, a newer GPU with a higher compute capability that has more registers per thread should be used for larger speedups. Using mascons is better than using Spherical Harmonics on the GPU, however, accessing the mascon data from the global memory for each dynamics evaluation requires many relatively slow read and writes. Using an interpolation based gravity model such as Fetch will decrease the number of memory transfers and increase speedups on the GPU.

For long encounters, the encounter window can be split into smaller encounter windows that are adjacent in time. The PDF propagation method, uses the full accurate model to propagate the uncertainty and generate samples for each window. MC simulation with a lower fidelity model is used to compute the closest approach within the time window. Regenerating the samples from the PDF ensures that error due to the lower fidelity dynamics is negligible because of the smaller times of flights for the MC phase. It is important to note that his method only works when a particular sample can identified across the PDFs at different times.

The MGMM PDF of the RSOs can be expressed in orbit elements at the beginning of the encounter window. Using orbit elements will reduce the number of elements required for an accurate PDF during the uncertainty propagation phase (Phase I). The MGMMs can still be propagated in orbit elements during the encounter time, however, they will be split into more elements converted to Cartesian coordinate at each quadrature node in time where the probability of new collisions is computed, i.e. Equation (5.24) on

page 155. It should be noted that the MGMM should be in Cartesian coordinates for the entire encounter window if the linear collision probabilities are used.

## Appendices



# Appendix A

## Publications

The publications that include the author's contributions are stated here.

### Submitted to Refereed Journals

#### Chapter 2

- **V. Vittaldev** and R. P. Russell, Multidirectional Gaussian Mixture Models for Nonlinear Uncertainty Propagation, Submitted July 2015.

#### Chapter 3

- **V. Vittaldev**, R. Linares, and R. P. Russell, Spacecraft Uncertainty Propagation using Gaussian Mixture Models and Polynomial Chaos Expansion, Submitted August 2015.

#### Chapter 5

- **V. Vittaldev** and R. P. Russell, Space Object Collision Probability Using Multidirectional Gaussian Mixture Models, Submitted August 2015.

## Other Referred Journal Publications

- N. Arora, **V. Vittaldev**, and R. P. Russell, Parallel Computation of Trajectories using Graphics Processing Units and Interpolated Gravity Models, *Journal of Guidance, Navigation, and Control*, Vol. 38, No. 8, pp. 1345-1355, 2015, DOI: 10.2514/1.G000571.
- E. Pellegrini, R. P. Russell, and **V. Vittaldev**, F and G Taylor Series Solutions to the Stark and Kepler Problems with Sundman Transformations, *Celestial Mechanics and Dynamical Astronomy*, Vol. 118, No. 5, pp. 355-378, 2014, DOI: 10.1007/s10569-014-9538-7
- **V. Vittaldev**, E. Mooij, and M. C. Naeije, Unified State Model theory and application in Astrodynamics, *Celestial Mechanics and Dynamical Astronomy*, Vol. 112, No. 3, pp. 253-282, 2012, DOI: 10.1007/s10569-011-9396-5

## Conference and Presentations

### Chapter 2

- **V. Vittaldev** and R. P. Russell, Multidirectional Gaussian Mixture Models for Nonlinear Uncertainty Propagation, ICCES2015 Mini-Symposium of Computational Methods in Celestial Mechanics, Reno, NV, 2015.
- **V. Vittaldev** and R. P. Russell, Uncertainty Propagation using Gaussian Mixture Models, SIAM Conference on Uncertainty Quantification, Savannah, GA, 2014.

### Chapter 3

- **V. Vittaldev**, R. Linares, and R. P. Russell, Uncertainty Propagation using the Gaussian Mixture Model Polynomial Chaos Approach, Paper AAS 15-448, AAS/AIAA Space Flight Mechanics Meeting, Williamsburg, VA, 2015.

### Chapter 5

- **V. Vittaldev** and R. P. Russell, Collision Probability using Multidirectional Gaussian Mixture Models, Paper AAS 15-394, AAS/AIAA Space Flight Mechanics Meeting, Williamsburg, VA, 2015.
- **V. Vittaldev** and R. P. Russell, Collision Probability for Resident Space Objects using Gaussian Mixture Models, Paper AAS 13-351, AAS/AIAA Space Flight Mechanics Meeting, Kauai, HI, 2013.

### Other Conference Papers and Presentations

- **V. Vittaldev**, R. Linares, H. C. Godinez, J. Koller, and R. P. Russell, Improved Uncertainty Quantification for Physics-Based Atmospheric Models via Generalized Polynomial Chaos, Abstract NG21A-1476, Fall Meeting, AGU, San Francisco, CA, 2013.
- N. Arora, R. P. Russell, and **V. Vittaldev**, Parallel Computation of Multiple Space Trajectories using GPUs and Interpolated Gravity Models, Paper AAS 13-901, AAS/AIAA Astrodynamics Specialist Conference, Hilton Head, SC, 2013.

- E. Pellegrini, R. P. Russell, and **V. Vittaldev**, F and G Taylor Series Solutions to the Stark Problem, Paper AAS 13-725, AAS/AIAA Astrodynamics Specialist Conference, Hilton Head, SC, 2013.
- H. Shen, **V. Vittaldev**, C. Karlgaard, R. P. Russell, and E. Pellegrini, Parallelized Sigma Point and Particle Filters for Navigation Problems, Paper AAS 13-034, AAS G&C Conference, Breckenridge, CO, 2013.
- R. P. Russell, N. Arora, **V. Vittaldev**, D. Gaylor, J. Anderson, Ballistic Coefficient Prediction for Resident Space Objects, Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS), Maui, HI, 2012.
- H. Shen, **V. Vittaldev**, C. Karlgaard, R. P. Russell, and E. Pellegrini, Parallelized Sigma Point and Particle Filters for Orbit Determination, Krestovaya Pad, Listvyanka, Russia, 2012.
- **V. Vittaldev**, R. P. Russell, N. Arora, and D. Gaylor, Second Order Kalman Filter using Multicomplex Step Derivatives, Paper AAS 12-204, AAS/AIAA Space Flight Mechanics Meeting, Charleston, SC, 2012.
- J. R. Quinlan, C. A. Jones, **V. Vittaldev**, and A. Wilhite, On the Design of an Active Debris Removal Architecture for Low Earth Orbit Space Debris Remediation, Paper AIAA 2011-7250, AIAA SPACE Conference & Exposition, Long Beach, CA, 2011.
- **V. Vittaldev**, E. Mooij, and M. C. Naeije, Performance Aspects of Orbit Propagation using the Unified State Model, Paper AIAA 2010-

7658, AIAA/AAS Astrodynamics Specialist Conference, Toronto, Ontario, Canada, 2010.

## Bibliography

- [1] Caesar. <https://debris-spatiaux.cnes.fr/en/node/179>. Accessed: 20 November 2015.
- [2] Cara: Conjunction assessment risk analysis. <http://satellitesafety.gsfc.nasa.gov/cara.html>. Accessed: 20 November 2015.
- [3] *Orbital Debris: A Technical Assessment*. The National Academies Press.
- [4] Space debris and human spacecraft. [http://www.nasa.gov/mission\\_pages/station/news/orbital\\_debris.html](http://www.nasa.gov/mission_pages/station/news/orbital_debris.html). Accessed: 06 October 2015.
- [5] Usstratcom space control and space surveillance network. <http://www.stratcom.mil/f11.html>. Accessed: 18 November 2015.
- [6] *Orbital Debris Quarterly News*, 19(1):9, January 2015.
- [7] R. I. Abbot and T. P. Wallace. Decision support in space situational awareness. *Lincoln Laboratory Journal*, 16(2):297–335, 2007.
- [8] N. Adurthi and P. Singla. Conjugate unscented transformation-based approach for accurate conjunction analysis. *Journal of Guidance, Control, and Dynamics*, 38(9):1642–1658, September 2015.

- [9] N. Adurthi, P. Singla, and T. Singh. The conjugate unscented transform - an approach to evaluate multi-dimensional expectation integrals. In *American Control Conference (ACC), Montreal, QC, Canada*, pages 5556–5561, 2012.
- [10] M. R. Akella and K. T. Alfriend. Probability of collision between space objects. *Journal of Guidance, Control, and Dynamics*, 23(5):769–772, 2000.
- [11] S. Alfano. A numerical implementation of spherical object collision probability. *Journal of the Astronautical Sciences*, 53(1):103–109, 2005.
- [12] S. Alfano. Addressing nonlinear relative motion for spacecraft collision probability. In *15th AAS/AIAA Astrodynamics Specialist Conference, Keystone, CO*, 2006.
- [13] S. Alfano. Review of conjunction probability methods for short-term encounters. In *AAS/AIAA Space Flight Mechanics Meeting, Sedona, Arizona, AAS 07-148*, 2007.
- [14] S. Alfano. Eliminating assumptions regarding satellite conjunction analysis. *The Journal of Astronautical Sciences*, 59(4):676–705, 2012.
- [15] S. Alfano and D. Finkleman. Operating characteristic approach to effective satellite conjunction filtering. In *23rd AAS/AIAA Space Flight Mechanics Meeting, Kauai, Hawaii*, 2013.
- [16] Salvatore Alfano. Determining satellite close approaches, part ii. *The Journal of Astronautical Sciences*, 42(2):143–152, April-June 1994.

- [17] Salvatore Alfano. Satellite conjunction monte carlo analysis. In *AIAA Space Flight Mechanics Meeting, AAS 09-233*, 2009.
- [18] K. T. Alfriend, M. R. Akella, J. Frisbee, J. L. Foster, D. J. Lee, and M. Wilkins. Probability of collision error analysis. *Space Debris*, 1(1), 1999.
- [19] Daniel L. Alspach and Harold W. Sorenson. Nonlinear bayesian estimation using gaussian sum approximations. *IEEE Transactions on Automatic Control*, 17(4):439–448, 1972.
- [20] T. W. Anderson. On the distribution of the two sample cramer-von mises criterion. *The Annals of Mathematical Statistics*, 33(3):1148–1159, 1962.
- [21] N. Andrei. An unconstrained optimization test functions collection. *Advanced Modeling and Optimization*, 10(1):147–161, 2008.
- [22] I. Arasaratnam and S. Haykin. Cubature kalman filters. *IEEE Transactions on Automatic Control*, 54(6):1254 – 1269, 2009.
- [23] J. M. Aristoff, J. T. Horwood, and A. B. Poore. Implicit-runge-kutta-based methods for fast, precise, and scalable uncertainty propagation. *Celestial Mechanics and Dynamical Astronomy*, 122(2):169–182, 2015.
- [24] J. M. Aristoff, J. T. Horwood, T. Singh, and A. B. Poore. Nonlinear uncertainty propagation in orbital elements and transformation to cartesian space without loss of realism. In *AIAA/AAS Astrodynamics Specialist Conference, San Diego, CA, Aug 4 - Aug 7*, 2014.



- [25] N. Arora, V. Vittaldev, and R. P. Russell. Parallel computation of trajectories using graphics processing units and interpolated gravity models. *Journal of Guidance, Control, and Dynamics*, 38(8):1345–1355, 2015.
- [26] M. Athans, R. P. Wishner, and A. Bertolini. Suboptimal state estimation for continuous-time nonlinear systems from discrete noisy measurements. *IEEE Transactions on Automatic Control*, AC-13(5):504–514, 1968.
- [27] I. Babuska and B. Q. Guo. The h, p and h-p version of the finite element method: basis theory and applications. *Advances in Engineering Software*, 15(3-4):159–174, 1992.
- [28] K. Baek and H. Bang. Adaptive sparse grid quadrature filter for spacecraft relative navigation. *Acta Astronautica*, 87:96–106, 2013.
- [29] U. Baspinar, H. S. Varol, and V. Y. Senyurek. Performance comparison of artificial neural network and gaussian mixture model in classifying hand motions by using semg signals. *Biocybernetics and Biomedical Engineering*, 33(1):33–45, 2013.
- [30] S. Bhaumik. Cubature quadrature kalman filter. *IET Signal Processing*, 7(7):533–541, September 2013.
- [31] A. Brown, J. Tichy, M. Demoret, and D. Rand. Gpu accelerated conjunction assessment with applications to formation flight and space debris, paper aas 13-902. In *AAS/AIAA Astrodynamics Specialist Conference, August 11-15, Hilton Head, SC*, 2013.

- [32] R. H. Byrd, R. B. Schnabel, and G. A. Schultz. A trust region algorithm for nonlinearly constrained optimization. *SIAM Journal on Numerical Analysis*, 24(5):1152 – 1170, 1987.
- [33] D. Calvetti, G. H. Golub, W. B. Gragg, and L. Reichel. Computation of gauss-konrod quadrature rules. *Mathematics of Computation*, 69(231):1035–1052, Feb 2000.
- [34] S. Challa, Y. Bar-Shalom, and V. Krishnamurthy. Nonlinear filtering via generalized edgeworth series and gauss-hermite quadrature. *IEEE Transactions on Signal Processing*, 48(6):1812–1820, June 2000.
- [35] F. Kenneth Chan. *Spacecraft Collision Probability*. The Aerospace Press, American Institute of Aeronautics and Astronautics, Inc., 2008.
- [36] K. Chan. Non-gaussian collision probability, paper aas 13-908. In *AAS/AIAA Astrodynamics Specialist Conference, Aug. 11 - 15, Hilton Head, South Carolina, 2013*.
- [37] C. W. Clenshaw and A. R. Curtis. A method for numerical integration on an automatic computer. *Numerische Mathematik*, 2(1):197–205, 1960.
- [38] S. L. Coffey, H. L. Neal, and M. M. Berry. Uncorrelated observations processing at naval space command. *Journal of Guidance, Control, and Dynamics*, 25(4):676–684, 2002.

- [39] A. R. Conn, N. I. M. Gould, and P. L. Toint. *Trust region methods*. Number 1 in MPS-SIAM Series on Optimization. Society for Industrial and Applied Mathematics, January 1987.
- [40] V. Coppola. Including velocity uncertainty in the probability of collision between space objects. In *AAS/AIAA Spaceflight Mechanics Meeting, Charleston, SC, AAS 12-247*, 2012.
- [41] D. Corbit. Numerical integration: From trapezoids to rms: Object-oriented numerical integration. *Dr. Dobb's Journal*, (252):117–120, Oct 1996.
- [42] M. K. Cowles and B. P. Carlin. Markov chain monte carlo convergence diagnostics: A comparative review. *Journal of the American Statistical Association*, 91(434):883–904, 1996.
- [43] P. Dagum, R. Karp, M. Luby, and S. Ross. An optimal algorithm for monte carlo estimation. *SIAM Journal on Computing*, 29(5):1484–1496, March 2000.
- [44] 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*, 36(4):1047–1057, 2013.
- [45] K. J. DeMars and M. K. Jah. Collision probability with gaussian mixture orbit uncertainty. *Journal of Guidance, Control, and Dynamics*, 37(3):979–985, 2014.

- [46] L. Demkowicz, W. Rachowicz, and Ph. Devloo. A fully automatic hp-adaptivity. *Journal of Scientific Computing*, 17(1-4):117–142, 2002.
- [47] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B*, 39(1):1–38, 1977.
- [48] M. Dodson and G. T. Parks. Robust aerodynamic design optimization using polynomial chaos. *Journal of Aircraft*, 46(2):635–646, 2009.
- [49] O. E. Drummond, Ogle T. L., and S. Waugh. Metrics for evaluating track covariance consistency. In *SPIE 6969, Signal and Data Processing of Small Targets*, 2008.
- [50] D. M. Endres and J. E. Schindelin. A new metric for probability distributions. *IEEE Transactions on Information Theory*, 49(7):1858 – 1860, 2003.
- [51] W. Feller. The fundamental limit theorems in probability. *Bulletin of the American Mathematical Society*, 51:800–832, 1945.
- [52] C. E. Froberg. *Introduction to Numerical Analysis*. Addison Wesley Publishing Company, Boston, Massachusetts, 1969.
- [53] K. Fujimoto and D. J. Scheeres. Tractable expressions for nonlinear propagated uncertainties. *Journal of Guidance, Control, and Dynamics*, 38(6):1146–1151, 2015.

- [54] A. T. Fuller. Analysis of nonlinear stochastic systems by means of the fokker-planck equation. *International Journal of Control*, 9(6):603–655, 1969.
- [55] A. Genz and B. D. Keister. Fully symmetric interpolatory rules for multiple integrals over finite regions with gaussian weight. *Journal of Computational and Applied Mathematics*, 71(2):299–309, 1996.
- [56] J. D. Giorgini, L. A. M. Benner, S. J. Ostro, M. C. Nolan, and M. W. Busch. Predicting the earth encounters of (99942) apophis. *Icarus*, 193:1–19, 2008.
- [57] Michael Grant and Stephen Boyd. Graph implementations for nonsmooth convex programs. In V. Blondel, S. Boyd, and H. Kimura, editors, *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pages 95–110. Springer-Verlag Limited, 2008. [http://stanford.edu/~boyd/graph\\_dcp.html](http://stanford.edu/~boyd/graph_dcp.html).
- [58] Michael Grant and Stephen Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. <http://cvxr.com/cvx>, March 2014.
- [59] H. Haario, M. Laine, A. Mira, and E. Saksman. Dram: Efficient adaptive mcmc. *Statistics and Computing*, 16(4):339–354, 2006.
- [60] L. M. Healy. Close conjunction detection on parallel computer. *Journal of Guidance, Control, and Dynamics*, 18(4):824–829, 1995.
- [61] J. R. Hershey and P. A. Olsen. Approximating the kullback leibler divergence between gaussian mixture models. In *IEEE International*

- Conference on Acoustics, Speech and Signal Processing, Honolulu, HI, 2007.*
- [62] K. Hill, C. Sabol, and K. T. Alfriend. Comparison of covariance based track association approaches using simulated radar data. *The Journal of the Astronautical Sciences*, 59(1-2):281–300, 2012.
- [63] T. Hobson, N. Gordon, I. Clarkson, M. Rutten, and T. Bessell. Dynamic steering for improved sensor autonomy and catalogue maintenance. In *Advanced Maui Optical and Space Surveillance Technologies Conference, Wailea, HI*, September 2014.
- [64] M. J. Holzinger, K. K. Luu, C. Sabol, and K. Hill. Probabilistic tracklet characterization and prioritization using admissible regions. In *Advanced Maui Optical and Space Surveillance Technologies Conference, Wailea, HI*, September 2014.
- [65] F. R. Hoots, L. L. Crawford, and R. L. Roehrich. An analytical method to determine future close approaches between satellites. *Celestial Mechanics*, 33, 1984.
- [66] J. T. Horwood and A. B. Poore. Gauss von mises distribution for improved uncertainty realism in space situational awareness. *SIAM/ASA Journal on Uncertainty Quantification*, 2(1):276–304, 2014.
- [67] J. T. Horwood and A. B. Proore. Orbital state uncertainty realism. In *Advanced Maui Optical and Space Surveillance Technologies Conference*, 2012.

- [68] Joshua T. Horwood, Nathan D. Aragon, and Aubrey B. Poore. Gaussian sum filters for space surveillance: Theory and simulations. *Journal of Guidance, Control, and Dynamics*, 34(6):1839–1851, 2011.
- [69] S. Hosder, R. W. Walters, and M. Balch. Efficient sampling for non-intrusive polynomial chaos applications with multiple uncertain input variables. *Proceedings of the 48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, number AIAA-2007-1939, Honolulu, HI*, 125, 2007.
- [70] Marco F. Huber, Tim Bailey, Hugh Durrant-Whyte, and Uwe D. Hanebeck. On entropy approximation for gaussian mixture random vectors. *Multi-sensor Fusion and Integration for Intelligent Systems, 2008. MFI 2008. IEEE International Conference on*, pages 181–188, 2008.
- [71] B. Jia, M. Xin, and Y. Cheng. Sparse gaussian-hermite quadrature filter with application to spacecraft attitude estimation. *Journal of Guidance, Control, and Dynamics*, 34(2):367–379, 2011.
- [72] B. Jia, M. Xin, and Y. Cheng. High-degree cubature kalman filter. *Automatica*, 49(2):510–518, February 2013.
- [73] N. L. Johnson, E. Stansbery, D. O. Whitlock, K. J. Abercromby, and D. Shoots. History of on-orbit satellite fragmentations. Technical Memorandum NASA/TM-2008-214779, NASA, June 2008.
- [74] B. A. Jones and A. Doostan. Satellite collision probability estimation using polynomial chaos. *Advances in Space Research*, 52(11):1860–1875,

2013.

- [75] 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*, 36(2):430–444, 2013.
- [76] B A. Jones, N. Parrish, and A. Doostan. Post-maneuver collision probability estimation using sparse polynomial chaos expansions. *Journal of Guidance, Control, and Dynamics*, Accepted for publication, 2014.
- [77] S.J. Julier and J. K. Uhlmann. Unscented filtering and nonlinear estimation. In *Proceedings of the IEEE*, volume 92, pages 401–422, 2004.
- [78] J. L. Junkins, M. R. Akella, and K. T. Alfriend. Non gaussian error propagation in orbital mechanics. *Journal of the Astronomical Sciences*, 44(4):541–563, 1996.
- [79] C. Kalender and A. Schottl. Sparse grid-based nonlinear filtering. *IEEE Transactions on Aerospace and Electronic Systems*, 49(4):2386 – 2396, October 2013.
- [80] T. S. Kelso. Norad two-line element sets current data. [https://www.space-track.org/documents/CSM\\_Guide.pdf](https://www.space-track.org/documents/CSM_Guide.pdf). Accessed: 19 November 2015.
- [81] T. S. Kelso. Norad two-line element sets current data. <http://www.celestrak.com/NORAD/elements/>. Accessed: 21 July 2015.



- [82] D. J. Kessler, N. L. Johnson, J. C. Liou, and M. Matney. The kessler syndrome: Implications to future space operations. In *33rd Annual AAS Guidance and Control Conference*, 2010.
- [83] S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79 – 86, March 1951.
- [84] M. Kumar and S. Chakravorty. Nonlinear filter based on the fokker-planck equation. *Journal of Guidance, Control, and Dynamics*, 35(1):68–79, 2012.
- [85] M. Kumar, S. Chakravorty, P. Singla, and J. L. Junkins. The partition of unity finite element approach with hp-refinement for the stationary fokker-planck equation. *Journal of Sound and Vibration*, 327(1-2):144–162, 2009.
- [86] G. Lantoine, R. P. Russell, and T. Dargent. Using multicomplex variables for automatic computation of high-order derivatives. *ACM Transactions on Mathematical Software*, 38(3), April 2012.
- [87] V. I. Lebedev and D. N. Laikov. A quadrature formula for the sphere of the 131st algebraic order of accuracy. *Doklady Mathematics*, 59(3):477–481, 1999.
- [88] W. Li, S. Prasad, and J. E. Fowler. Hyperspectral image classification using gaussian mixture models and markov random fields. *Geoscience and Remote Sensing Letters, IEEE*, 11(1):153–157, Jan 2014.

- [89] X. Li, P. B. Nair, Z. Zhang, L. Gao, and C. Gao. Aircraft robust trajectory optimization using nonintrusive polynomial chaos. *Journal of Aircraft*, 51(5):1592–1603, 2014.
- [90] R. Linares and J. L. Crassidis. Generalized gaussian cubature for nonlinear filtering, aas paper 2015-423. In *25th AAS/AIAA Space Flight Mechanics Meeting, Williamsburg, VA*, 2015.
- [91] D. M. Luchtenburga, S. L. Bruntonc, and C. W. Rowleyb. Long-time uncertainty propagation using generalized polynomial chaos and flow map composition. *Journal of Computational Physics*, 274(1):783–802, October 2014.
- [92] L. Mathelin, M. Y. Hussaini, and T. A. Zang. Stochastic approaches to uncertainty quantification in cfd simulations. *Numerical Algorithms*, 38(1-3):209–236, 2005.
- [93] H. M. T. Menegaz, J. Y. Ishihara, G. A. Borges, and A. N. Vargas. A systematization of the unscented kalamn filter theory. *IEEE Transactions on Automatic Control*, 60(10):2583–2598, October 2015.
- [94] R. Mikulevicius and B. Rozovskii. Fourier-hermite expansions for nonlinear filtering. *Theory Probab. Appl.*, 44(3):606–612, 2000.
- [95] J. G. Miller. A new sensor allocation algorithm for the space surveillance network. *Military Operations Research*, 12(1):57–70, 2007.
- [96] J. J. Moré and D. C. Sorensen. Computing a trust region step. *SIAM Journal on Scientific and Statistical Computing*, 4(3):553–572, 1983.

- [97] A. Morselli, R. Armellin, P. D. Lizia, and F. B. Zazzera. A high order method for orbital conjunctions analysis: Monte carlo collision probability computation. *Advances in Space Research*, 55(1):311–333, 2015.
- [98] N. Nakhjiri and B. F. Villac. An algorithm for trajectory propagation and uncertainty mapping on gpu, paper aas 13-376. In *23rd AAS/AIAA Space Flight Mechanics Meeting, Kauai, HI*, 2013.
- [99] M. Nayak. Impact of national space policy on orbital debris mitigation and us air force end of life satellite operations. In *SpaceOps Conference, Stockholm, Sweden*, 2012.
- [100] M. Norgaard, N. K. Poulsen, and O. Ravn. New developmeents in state estimation for nonlinear systems. *Automatica*, 36(11):1627–1638, 2000.
- [101] S. Oladyshkin and W. Nowak. Data-driven uncertainty quantification using the arbitrary polynomial chaos expansion. *Reliability Engineering & System Safety*, 106:179–190, October 2012.
- [102] S. S. Oliver, A. J. Perticia, and J. R. Henderson. Intelligent sensor tasking for space collision mitigation. Technical Report LLNL-TR-427454, Lawrence Livermore National Laboratory, April 2010.
- [103] 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*, 29(6):1367–1375, 2006.
- [104] R. P. Patera. Satellite collision probability for nonlinear relative motion. *Journal of Guidance, Control, and Dynamics*, 26(5):728–733, 2003.

- [105] R. P. Patera. Calculating collision probability for arbitrary space-vehicle shapes via numerical quadrature. *Journal of Guidance, Control, and Dynamics*, 28(6):1326–1328, 2005.
- [106] T. Payne and R. Morris. The space surveillance network (ssn) and orbital debris. In *33rd Annual AAS Guidance and Control Conference, Breckenridge, Colorado*, 2010.
- [107] A. B. Poore. Propagation of uncertainty in support of ssa missions. In *25th AAS/AIAA Space Flight Mechanics Meeting, Williamsburg, VA*, 2015.
- [108] M. L. Psiaki, J. R. Schoenberg, and I. T. Miller. Gaussian sum reapproximation for use in a nonlinear filter. *Journal of Guidance, Control, and Dynamics*, 38(2):292–303, 2015.
- [109] J. R. Rice. A metalgorithm for adaptive quadrature. *Journal of the ACM*, 22(1):61–82, 1975.
- [110] D. Richmond. Space situational awareness (ssa) research findings. In *Advanced Maui Optical and Space Surveillance Technologies Conference*, volume 1, page 3, 2008.
- [111] J. R. A. Rodriguez, Fadrique F. M. M., and H. Klinkrad. Collision risk assessment with a smart sieve method. In *Joint ESA-NASA Space-Flight Safety Conference, ESTEC, Noordwijk, the Netherlands*, 2002.

- [112] R. P. Russell and N. Arora. Global point mascon models for simple, accurate and parallel geopotential computation. *Journal of Guidance, Control, and Dynamics*, 35(5):1568–1581, 2012.
- [113] C. Sabol, C. Binz, A. Segerman, K. Roe, and P. W. Schumacher. Probability of collision with special perturbations dynamics using the monte carlo method. In *AAS/AIAA Astrodynamics Specialist Conference, Girdwood, Alaska, AAS 11-435*, 2011.
- [114] C. Sabol, T. Sukut, K. Hill, K. T. Alfriend, B. Wright, Y. Li, and P. Schumacher. Linearized orbit covariance generation and propagation analysis via simple monte carlo simulations, paper aas 10-134. In *20th AAS/AIAA Space Flight Mechanics Meeting, Feb 14-17, San Diego, CA*, 2010.
- [115] J. Sarmavuori and S. Sarkka. Fourier-hermite kalman filter. *IEEE Transactions on Automatic Control*, 57(6):1511–1515, June 2012.
- [116] D. J. Scheeres, F.-Y. Hsiao, R. S. Park, B. F. Villac, and J. M. Maruskin. Fundamental limits on spacecraft orbit uncertainty and distribution propagation. *The Journal of the Astronautical Sciences*, 54(3-4):505–523, 2006.
- [117] H. Shen, V. Vittaldev, C. D. Karlgaard, R. P. Russell, and E. Pellegrini. Parallelized sigma point and particle filters for navigation problems, paper aas 13-034. In *36th Annual AAS Guidance and Control Conference, Feb 1- 6, Breckenridge, CO*, 2013.

- [118] D. Siegmund. Importance sampling in the monte carlo study of sequential tests. *The Annals of Statistics*, 4(4):673–684, 1976.
- [119] J. B. Skelton. Data handling and protection of need-to-know data in a need-to-share netcentric enterprise. In *Advanced Maui Optical and Space Surveillance Technologies Conference*, page 93, 2012.
- [120] S. A. Smolyak. Quadrature and interpolation formulas for tensor products of certain classes of functions. *Doklady Akademii nauk SSSR*, 1(4):240–243, 1963.
- [121] W. Squire and G. Trapp. Using complex variables to estimate derivatives of real functions. *Siam Review*, 40(1):110 – 112, 1998.
- [122] C. L. Stokely, J. L. Foster, E. G. Stansbery, J. R. Benbrook, and Q. Juarez. Haystack and hax radar orbital debris environment; 2003. NASA Technical Report JSC-62815, NASA Johnson Space Center Orbital Debris Program Office, November 2006.
- [123] G. Szegő. *Orthogonal Polynomials*. Number v. 23 in American Mathematical Society. American Mathematical Society, 1967.
- [124] B. D. Tapley, B. E. Schutz, and G. H. Born. *Statistical Orbit Determination*. Elsevier Academic Press, 1 edition, June 2004.
- [125] The Analytic Sciences Corporation Technical Staff. *Applied Optimal Estimation*. The MIT Press, Cambridge, Massachusetts, 1 edition, 1974.

- [126] 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*, 31(6):1623 – 1633, 2008.
- [127] Sain-Zee Ueng, Melvin Lathara, Sara S. Bagsorkhi, and Wen-Mei W. Hwu. Languages and compilers for parallel computing. chapter CUDA-Lite: Reducing GPU Programming Complexity, pages 1–15. Springer-Verlag, Berlin, Heidelberg, 2008.
- [128] David A Vallado. *Fundamental of Astrodynamics and Applications*. Microcosm Press/Springer, 3 edition, April 2007.
- [129] J. P. Vinti. Inclusion of the third zonal harmonic in an accurate reference orbit of an artificial satellite. *Journal of Research of the National Bureau of Standards*, 70B(1):17–46, 1966.
- [130] K. Vishwajeet, P. Singla, and M. Jah. Nonlinear uncertainty propagation for perturbed two-body orbits. *Journal of Guidance, Control, and Dynamics*, 37(5):1415–1425, 2014.
- [131] K. Vishwajeet, P. Singla, and J. Moriba. Nonlinear uncertainty propagation for perturbed two-body orbits. *Journal of Guidance, Control, and Dynamics*, Available Online, 2014.
- [132] V. Vittaldev and R. Russell. Collision probability for space objects using gaussian mixture models. In *Proceedings of the 23rd AAS/AIAA Space Flight Mechanics Meeting*, volume 148, 2013.

- [133] V. Vittaldev and R. P. Russell. Collision probability for resident space objects using gaussian mixture models, aas 13-351. In *23rd AAS/AIAA Spaceflight Mechanics Meeting, Kauai, Hawaii*, 2013.
- [134] V. Vittaldev and R. P. Russell. Collision probability using multidirectional gaussian mixture models, aas 15-394. In *AAS/AIAA Space Flight Mechanics Meeting, Williamsburg, VA, January 11-15*, 2015.
- [135] X. Wan and G. E. Karniadakis. An adaptive multi-element generalized polynomial chaos method for stochastic differential equations. *Journal of Computational Physics*, 209, 2005.
- [136] N. Wiener. The homogeneous chaos. *Amer. J. Math.*, 60(4):897–936, 1938.
- [137] D. Xiu and G. E. Karniadakis. The wiener-asky polynomial chaos for stochastic differential equations. *SIAM J. Sci. Comput.*, 24:619–644, 2002.



## Vita

Vivek Vittaldev was born in Visakhapatnam, India on 6 January 1987. He spent his first 10 years in Rasayani, India before moving to the Hague, the Netherlands. Vivek graduated with a B.Sc. in Aerospace Engineering from Delft University of Technology (Delft, The Netherlands) in 2007. He received a M.Sc. in Aerospace Engineering in 2010 also from Delft University of Technology. He began his PhD at Georgia Institute of Technology in 2010, but transferred to The University of Texas at Austin in 2012 to continue working with Dr. Ryan Russell. Additionally, he has gained invaluable experience from internships at NGC Aerospace Ltd. (Sherbrooke, Canada), NASA Langley Research Center, National Institute of Aerospace, and Los Alamos National Laboratory.

Permanent address: vivek.vittaldev@gmail.com

This dissertation was typeset with  $\text{\LaTeX}^\dagger$  by the author.

---

<sup>†</sup> $\text{\LaTeX}$  is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's  $\text{\TeX}$  Program.