Applications of cubature methods to initial orbit determination in cislunar space

Kellan Meacham

Follow this and additional works at: https://louis.uah.edu/uah-theses

Recommended Citation
Meacham, Kellan, "Applications of cubature methods to initial orbit determination in cislunar space" (2023). Theses. 467.
https://louis.uah.edu/uah-theses/467

This Thesis is brought to you for free and open access by the UAH Electronic Theses and Dissertations at LOUIS. It has been accepted for inclusion in Theses by an authorized administrator of LOUIS.
APPLICATIONS OF CUBATURE METHODS TO INITIAL ORBIT DETERMINATION IN CISLUNAR SPACE

Kellan Meacham

A THESIS

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Aerospace Engineering in The Department of Mechanical and Aerospace Engineering to The Graduate School of The University of Alabama in Huntsville

May 2023

Approved by:
Dr. Naga Venkat Adurthi, Research Advisor/Committee Chair
Dr. Robert Frederick, Committee Member
Dr. Jason Cassibry, Committee Member
Dr. Keith Hollingsworth, Department Chair
Dr. Shankar Mahalingam, College Dean
Dr. Jon Hakkila, Graduate Dean
Abstract

APPLICATIONS OF CUBATURE METHODS TO INITIAL ORBIT DETERMINATION IN CISLUNAR SPACE

Kellan Meacham

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Aerospace Engineering

Mechanical and Aerospace Engineering
The University of Alabama in Huntsville
May 2023

This paper aims to examine the viability of using cubature methods, instead of the computationally expensive Monte Carlo method, to model cislunar orbits. The cubature methods examined are the Unscented Kalman Filter (UKF) and Conjugate Unscented Transform (CUT). This study considers only specific halo and cycler orbits with no perturbations or propulsion systems. Orbits were modeled by choosing a most likely initial condition, generating sigma points, propagating those sigma points forwards in time, and calculating the average final state. It found that cubature methods can model cislunar orbits far more efficiently than the Month Carlo method, at the cost of some accuracy. Furthermore, higher-order CUT methods had similar efficiency and higher accuracy than lower-order methods. Therefore, for the conditions examined, higher-order methods can be used instead of lower-order methods and could be a viable alternative to the Monte Carlo method in cases where computational efficiency is valued over accuracy.
Acknowledgements

I would like to thank the Mechanical and Aerospace Engineering Department for supporting my research. In particular, I would like to thank Dr. Naga Venkat Adurthi for his patience and continued support as my thesis advisor throughout the 2022 academic year. Dr. Adurthi taught me the statistical concepts, including cubature methods, that I needed to complete my research. I am also grateful for my parents’ support throughout my education. I would also like to thank my partner, Luke, for his constant support during my graduate degree, as well as his help in proofreading and collecting data.
Table of Contents

Abstract ................................................................. ii

Acknowledgements ...................................................... iv

Table of Contents ...................................................... vi

List of Figures ........................................................... vii

List of Tables ........................................................... ix

List of Symbols .......................................................... xi

Chapter 1. Introduction ............................................... 1

Chapter 2. Literature Review ......................................... 2

  2.1 Initial Orbit Determination ................................. 3

  2.2 Modeling Motion in Cislunar Space ...................... 4

  2.2.1 Circular Restricted 3-Body Problem ................. 6

  2.2.2 Bicircular Restricted 4-Body Problem ............. 7

  2.3 Accounting for Uncertainty in Initial Conditions .... 9

  2.3.1 Monte Carlo Method ................................. 10
2.3.2 Expectation Integral and Cubature Methods .......... 11
2.3.3 Unscented Kalman Filter .......................... 14
2.3.4 Conjugate Unscented Transforms ................. 18

Chapter 3. Methods .............................................. 49
3.1 Generating Sets of Initial Conditions ................. 50
3.2 Equations for Propagating Orbits ..................... 51
3.2.1 2-Body Problem .................................. 52
3.2.2 Circular Restricted 3-Body Problem ............... 53
3.2.3 Bicircular Restricted 4-Body Problem .......... 57
3.3 Determining Locations of the Sun and Moon .......... 62
3.4 Numerical Integration in MATLAB ................... 72

Chapter 4. Simulations ........................................... 73
4.1 Example 1: Spatial Cycler Orbit ....................... 74
4.2 Example 2: Spatial Halo Orbit ....................... 81
4.3 Comparing Cubature and Monte Carlo Methods .... 88

Chapter 5. Conclusions .......................................... 90

References ..................................................... 93
## List of Figures

2.1 Visualizing the BCR4BP .............................................. 8
2.2 Symmetric set of points and axes 2D and 3D space ............. 19
2.3 comparison of number of points .................................... 44

3.1 Locations of the Sun, Earth, and Moon as Assumed by the BCR4BP . 58
3.2 Position of Sun, Earth, and Moon at Time $t = 0$ ................. 63

4.1 Approximate Trajectory for Ex.1a (Cycler CR3BP) ............... 75
4.2 Approximate Trajectory for Ex.1b (Cycler BCR4BP) ............. 75
4.3 Moments of MC Points from Cycler CR3BP (Ex.1a) .......... 76
4.4 Moments of MC Points from Cycler BCR4BP (Ex.1b) .......... 77
4.5 Final Positions after 15 Days for Cycler CR3BP ................. 78
4.6 Final Positions after 15 Days for Cycler BCR4BP ............... 78
4.7 Absolute Position Error vs Time for Cycler Obit ............... 79
4.8 Modeled Trajectories for 15-day Cycler CR3BP ................. 80
4.9 Modeled Trajectories for 15-day Cycler BCR4BP ............... 81
4.10 Approximate Trajectory for Ex.2a (Halo CR3BP) ............... 82
4.11 Approximate Trajectory for Ex.2b (Halo BCR4BP) ............. 82
4.12 Moments of MC Points from Halo CR3BP (Ex.2a) ............ 83
4.13 Moments of MC Points from Halo BCR4BP (Ex.2b) ............ 83
4.14 Final Positions after 15 Days for Halo CR3BP (Ex.2a) ....... 84
4.15 Final Positions after 15 Days for Halo BCR4BP (Ex.2b) ..... 85
4.16 Absolute Position Error vs Time for Halo Orbit ............... 86
4.17 Modeled Trajectories for 15-Day Halo CR3BP .................. 87
4.18 Modeled Trajectories for 15-Day for Halo BCR4BP 87
4.19 Magnitude of Absolute Position Error vs Time 88
List of Tables

2.1 Fully Symmetric set of points ........................................ 22
2.2 Coefficients of the set of Fully Symmetric points .......... 24
2.3 Fully Symmetric set of points for CUT4 ....................... 26
2.4 Fully Symmetric set of points for CUT4 ....................... 28
2.5 Sigma Points for CUT4 .................................................. 30
2.6 CUT4: Optimized Solution for $n = 1$ and $n = 2$ ........ 31
2.7 Sigma Points for CUT6, $(n \leq 6)$ ................................. 34
2.8 Sigma Points for CUT6, $(7 \leq n \leq 9)$ ......................... 34
2.9 Fully Symmetric set of points for CUT6 ....................... 36
2.10 Solutions for $2 \leq n \leq 9$, $6^{th}$ moment constraint equations, CUT6 ... 37
2.11 Sigma Points for CUT8, $(2 \leq n \leq 6)$ ....................... 39
2.12 Fully Symmetric set of points for CUT8 ....................... 42
2.13 Solutions for $2 \leq n \leq 6$, $8^{th}$ moment constraint equations, CUT8 ... 42
2.14 Number of Points for $5^{th}$ order accurate cubature methods. ....... 46
2.15 Number of Points for $7^{th}$ order accurate cubature points. ......... 47
2.16 Number of Points for $9^{th}$ order accurate cubature points. .......... 48

3.1 Constants in CR3BP Model [7, 9] .................................... 57
3.2 Constants in BCR4BP Model [7, 10, 9] .............................. 62
3.3 Calculation of Lunar Eclipse Times [37, 38] ....................... 70
3.4 Constants in Model of Sun, Earth, and Moon Equations [7, 10, 9] ... 71

4.1 Examples ................................................................. 74
4.2 Absolute Position Errors of Cubature Methods for Cycler Orbit .... 79
4.3 Absolute Position Errors of Cubature Methods for Halo Orbit . . . . . 86
4.4 Computational Efficiency for Monte Carlo and Cubature Methods . . 89
## List of Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\vec{F}$</td>
<td>Total gravitational force on small object</td>
</tr>
<tr>
<td>$m$</td>
<td>Mass of small object</td>
</tr>
<tr>
<td>$\vec{r}$</td>
<td>Position of small object</td>
</tr>
<tr>
<td>$\dot{r}$</td>
<td>Velocity of small object</td>
</tr>
<tr>
<td>$\ddot{r}$</td>
<td>Acceleration of small object</td>
</tr>
<tr>
<td>$G$</td>
<td>Gravitational constant</td>
</tr>
<tr>
<td>$N$</td>
<td>In orbital mechanics, the total number of bodies in a gravitational system</td>
</tr>
<tr>
<td>$\vec{r}_n$</td>
<td>Position of the $n^{th}$ massive body</td>
</tr>
<tr>
<td>$m_n$</td>
<td>Mass of the $n^{th}$ massive body</td>
</tr>
<tr>
<td>$\mu_n$</td>
<td>Gravitational parameter of the $n^{th}$ massive body</td>
</tr>
<tr>
<td>$\vec{r}_{Sb}$</td>
<td>Position of the Sun in barycentric frame</td>
</tr>
<tr>
<td>$\vec{r}_{EMb}$</td>
<td>Position of the Earth-Moon barycenter in barycentric frame</td>
</tr>
<tr>
<td>$\vec{r}_{Eb}$</td>
<td>Position of the Earth in barycentric frame</td>
</tr>
<tr>
<td>$\ddot{r}_{Eb}$</td>
<td>Acceleration of the Earth in barycentric frame</td>
</tr>
<tr>
<td>$\vec{r}_{Mb}$</td>
<td>Position of the Moon in barycentric frame</td>
</tr>
</tbody>
</table>
\( \vec{r}_b \) Position of small body in barycentric frame
\( \vec{\ddot{r}}_b \) Acceleration of small body in barycentric frame
\( \vec{r}_{Sg} \) Position of the Sun in geocentric frame
\( \vec{r}_{EMg} \) Position of the Earth-Moon barycenter in geocentric frame
\( \vec{r}_{Eg} \) Position of the Earth in geocentric frame
\( \vec{\ddot{r}}_{Eg} \) Acceleration of the Earth in geocentric frame
\( \vec{r}_{Mg} \) Position of the Moon in geocentric frame
\( \vec{r}_g \) Position of small body in geocentric frame
\( \vec{\ddot{r}}_g \) Acceleration of small body in geocentric frame
\( m_S \) Sun’s mass
\( m_E \) Earth’s mass
\( m_M \) Moon’s mass
\( \mu_S \) Sun’s gravitational parameter
\( \mu_E \) Earth’s gravitational parameter
\( \mu_M \) Moon’s gravitational parameter
\( r_{SEM} \) Distance between Sun and Earth-Moon barycenter
\( r_{EM} \) Distance between Earth and Moon
\( x_1 \) Negative distance between the Sun and Sun-(Earth-Moon) barycenter
\( x_2 \) Positive distance between the Earth-Moon barycenter and Sun-(Earth-Moon) barycenter

\( x_3 \) Negative distance between the Earth and Earth-Moon barycenter

\( x_4 \) Positive distance between the Moon and Earth-Moon barycenter

\( \omega \) Angular velocity of the (Earth-Moon)-Sun system

\( \Omega \) Angular velocity of the Earth-Moon system

\( t \) Time since most recent total lunar eclipse

\( t_0 \) Time between initial measurement and most recent total lunar eclipse

\( \Delta t \) Time since initial measurement

\( UTC \) Coordinated universal time

\( UT \) Universal time

\( TD \) Terrestrial dynamical time

\( \Delta T \) Difference between \( UT \) and \( TD \)

\( JD \) Julian date

\( y \) In datetime, calendar year \(((1901 \leq y \leq 2099))\)

\( m \) In datetime, calendar month \((1 \leq m \leq 12)\)

\( d \) In datetime, calendar day \((1 \leq d \leq 31)\)

\( E \) Expectation integral
In statistics, order or degree of system

In statistics, average

Covariance matrix

Weight of \( i^{th} \) sigma point

Position of \( i^{th} \) sigma point

Tuning parameter

Principal axes

\( m^{th} \) conjugate axes

\( m^{th} \) scaled conjugate axes

Parameter of scaled conjugate axes

In statistics, distance variable of sigma points

Weight corresponding to \( r_j \)
Chapter 1. Introduction

Space Situational Awareness (SSA) is essential in maintaining current and future space assets. In order to maintain SSA, it is important to be able to predict where an object in orbit will be after an amount of time, given some initial measurements [25]. In this paper, initial measurements include a velocity and position vector. Therefore, it is of interest to be able to calculate the position of an object after a certain amount of time given only its initial conditions, a process known as Initial Orbit Determination (IOD) [7].

This paper aims to evaluate various methods for the uncertainty analysis of orbit propagation in cislunar space. These methods include both the widely used Unscented Kalman Filter (UKF), discussed further in Chapter 2.3.3, and the more recently derived Conjugate Unscented Transform (CUT), discussed further in Chapter 2.3.4 [1].

While some existing literature has considered the use of UKF and CUT in modelling orbits in a two-body system, few have used either to model orbits in cislunar space. Furthermore, many models of cislunar space only include planar orbits rather than spatial orbits as well.
Chapter 2. Literature Review

Various models can be used to predict motion in cislunar space. The term “massive body” indicates a body that exerts a significant gravitational force on a small object, such as a star, planet, or moon. The Circular Restricted 3-Body Problem (CR3BP) accounts for the gravitational influences of the Earth and Moon, and therefore considers the influence of two massive bodies on a small body [14]. The Bicircular Restricted 4-Body Problem (BCR4BP) accounts for the gravitational influences of the Sun, Earth, and Moon, and therefore considers the influence of three massive bodies on a small body [14]. Both methods have been used in past studies to model motion in cislunar space [4, 5, 8, 12, 13, 27, 34].

While existing studies have used various models to describe motion in cislunar space, few of them have analyzed the uncertainty associated with Initial Orbit Determination in cislunar space. If the initial measurements have an uncertainty, that uncertainty must be propagated through time to determine the uncertainty of the final state [50]. Past studies have used several methods of uncertainty analysis, such as the Monte Carlo method, the Unscented Kalman Filter, and the Conjugate Unscented Transform, but none so far have compared these methods in the context of cislunar space [1, 44, 50].
2.1 Initial Orbit Determination

When the number of initial observations describing an object’s motion is equal to the number of unknowns in the object’s final state, orbital determination is performed by solving a nonlinear system of equations. This process is known as Initial Orbit Determination (IOD) [2, 43, 16]. For the purposes of this paper, these initial observations are a velocity and position vector, for a total of six initial observations: the x, y, and z components of position and the x, y, and z components of velocity. IOD typically produces a single solution and measurement errors are usually not taken into account [2].

For a two-body problem, such as a small object under only the Earth’s gravitational influence, there exists an analytical solution describing the object’s motion over time [18]. Therefore, in a two-body problem, IOD from an initial position and velocity vector can be performed without numerical integration using methods such as the Lagrange coefficients [7].

However, when there are more than two massive bodies in the gravitational system, there exists no a closed-form analytical solution to describe the motion of the small body over time [33]. Instead, the body’s motion is modeled with a second-order differential equation, shown below in Equation 2.1, which must be solved numerically to determine the final position of the object [7]. This equation is derived from Newton’s Law of Universal Gravitation [26, 41]:

\[
F = m\ddot{r} = \sum_{n=2}^{N} \frac{Gm_m n}{|\vec{r}_n - \vec{r}|^3}(\vec{r}_n - \vec{r}) = \sum_{n=2}^{N} \frac{\mu_n (\vec{r}_n - \vec{r})}{|\vec{r}_n - \vec{r}|^3}
\]  

(2.1)
where

- \( F \) = total force on the small object
- \( m \) = mass of small object
- \( \ddot{\vec{r}} \) = acceleration of small object
- \( N \) = number of massive bodies
- \( G \) = gravitational constant, \( 6.67408 \times 10^{-11} Nm^2kg^{-2} \)
- \( m_n \) = mass of massive body
- \( \vec{r}_n \) = location of massive body
- \( \vec{r} \) = location of small object
- \( \mu_n \) = gravitational parameter, \( \mu_n = Gm_n \).

Like any other second order differential equation, integrating Equation 2.1 with any numerical integration technique will produce a solution, but some numerical integration techniques are higher fidelity than others [19]. Boone’s CR3BP and BCR4BP models, for instance, both use MATLAB’s ODE45, a six-stage fifth-order Runge Kutta method, to predict the locations of objects in cislunar space [4, 30].

2.2 Modeling Motion in Cislunar Space

In cislunar space, which is the space in which both the Earth’s gravity and the Moon’s gravity is relevant, it is generally necessary to consider the gravitational
influence of more than just the Earth [22]. The farther a small object is from the Earth, the greater the relative influence of other massive bodies, and the more important it is to consider the influence of those other bodies when modeling the object’s motion [13]. The Moon’s gravitational influence must be considered, and, in some models, the Sun’s gravitational influence is considered as well [4]. Since there are two or more massive bodies considered in cislunar space, no closed-form solution for a small object’s motion exists, and IOD must be performed through numerical integration [15, 31]. Although IOD for objects near the Earth is well-studied, few prior studies have investigated this type of IOD in cislunar space [4].

The motion of an object in cislunar space is chaotic, meaning that small changes in initial conditions can produce large changes in final conditions, especially over long periods of time [50]. Furthermore, in addition to the gravitational influences of the Earth, Moon, and Sun, an object in cislunar space may also be influenced to some degree by gravity from other objects in the solar system, solar radiation pressure, oblateness effects, or atmospheric drag [4]. However, objects in cislunar space tend to be mostly influenced by the Earth and Moon, and secondarily influenced by the Sun. Therefore, there are two main models used in this paper to describe motion in cislunar space. The Circular Restricted 3-Body Problem considers the gravitational influences of the Earth and Moon, while the Bicircular Restricted 4-Body Problem considers the gravitational influences of the Earth, Moon, and Sun [4].
2.2.1 Circular Restricted 3-Body Problem

The Circular Restricted 3-Body Problem (CR3BP) can be used to model the motion of a small object under the influence of two massive bodies. It assumes that the two massive bodies move in circular orbits around their barycenter, or center of mass, with a constant angular velocity, and that the small object has a negligible mass compared to that of the two massive bodies, and therefore does not exert any significant gravitational force on them [8, 4]. The planar CR3BP considers a small body that is coplanar with both massive bodies, while the spatial CR3BP considers a small body that is not coplanar with the massive bodies. If the small body has coplanar initial conditions, it will remain coplanar; likewise, if it has spatial initial conditions, it will remain spatial [8]. Like any other problem that considers more than two bodies, there is no analytical solution to the CR3BP, so IOD with the CR3BP must be performed through numerical integration.

The CR3BP assumptions are generally considered valid for the Earth-Moon system, as the Earth and Moon are both massive compared to spacecrafts and satellites, and they move in an almost circular orbit around their barycenter [7, 6]. Although it ignores the Sun’s gravity and several perturbations, it is generally considered a useful tool for describing motion and providing insights into the dynamical properties of cislunar space [4].

Consequentially, many models in existing literature have used the CR3BP to describe motion in cislunar space. Dahlke [8] used the planar CR3BP to model optimal low-thrust trajectory designs in two dimensions. Davis et. al. [13] used the spatial
CR3BP to model Near Rectilinear Halo Orbits (NRHOs), which are three-dimensional precisely periodic orbits that go around both the Earth and Moon, passing close to the moon [5, 4]. Davis et. al. [13] also verified their CR3BP results using NAIF SPICE planetary ephemerides in conjunction with the GRAIL model for lunar gravity. The NAIF SPICE model and the GRAIL model were developed by NASA to account for other sources of perturbation not considered by the CR3BP [39, 40].

2.2.2 Bicircular Restricted 4-Body Problem

While the CR3BP is a useful model, it ignores the Sun’s gravitational influence. The Sun’s gravitational influence can have significant effects on spacecraft in cis-lunar orbits, especially over long time spans [11]. Near Earth, perturbing accelerations caused by the Sun and Moon are of about the same order of magnitude, and the Sun’s gravitational influence plays a non-trivial role in the change of an orbit over time [34]. Additionally, similar to the Moon’s influence, the relative gravitational influence of the Sun, as compared to the Earth, increases as the distance from Earth increases [13].

The Bicircular Restricted 4-Body Problem (BCR4BP) addresses these issues by incorporating the Sun’s gravitational influence into a model similar to the CR3BP. The BCR4BP model describes a small object that moves under the gravitational influences of the Earth, Moon, and Sun. Like the CR3BP, it assumes that the Earth and Moon both revolve in circular orbits around the Earth-Moon barycenter (EMB). It also assumes that the Sun and the EMB both revolve in circular orbits around the barycenter of the Sun and the Earth-Moon system, referred to as the Sun-Earth-Moon barycenter.
(SEMB). The orbits of all massive bodies are assumed to be coplanar [4, 49]. The BCR4BP is illustrated in Figure 2.1.

Figure 2.1: Visualizing the BCR4BP

Unlike the CR3BP, the motion of the massive bodies in the BCR4BP is not consistent with Newton’s Laws of Gravitation. In the BCR4BP, the Earth and Moon are assumed to revolve in perfectly circular orbits around their barycenter with a constant angular velocity, despite the Sun’s influence. According to Newton’s Laws of Gravitation, the Sun’s gravitational influence would make a perfectly circular Earth-Moon orbit impossible [27]. Despite this inconsistency, the standard BCR4BP is still incredibly useful for modeling motion in cislunar space [4].
The BCR4BP has been used in a number of models for cislunar space. Namazy-fard [34] used the BCR4BP to model cislunar orbits over decadal time spans, though they also considered the perturbing effects of Earth’s oblateness, and Koon [27] used the model for cislunar trajectory generation. Boone [4] used the BCR4BP to model small particles near the Earth-Moon Lagrange points. Davis et al [12] and Boudad [5] both used the BCR4BP to model NRHOs and the escape from such orbits, and both successfully verified their results using NAIF SPICE planetary ephemerides in conjunction with the GRAIL model for lunar gravity [40, 39].

2.3 Accounting for Uncertainty in Initial Conditions

IOD typically produces a single final condition for a single initial condition, and measurement errors are usually not taken into account [2]. However, there is some uncertainty associated with measurements, including the measured initial positions and velocities in IOD. Orbital motion in cislunar space is chaotic, meaning that small changes in the initial conditions can produce large changes in position and velocity over time [4, 50]. Therefore, it is important to account for the uncertainties associated with the initial measurements. This uncertainty can be expressed as a covariance matrix, vector of variances, or vector of standard deviations.

Generally, propagating uncertainty through nonlinear dynamical systems with uncertain initial conditions cannot be done analytically [20]. Instead, it involves computing multi-dimensional expected value integrals with respect to the appropriate probability density function (PDF). This process involves generating multiple sets of initial conditions, propagating each set of initial conditions forward in time, and observing the
distribution of final values resulting from the varying initial conditions \[50, 19\]. Common techniques to approximate the expectation integral with respect to a Gaussian PDF include the Monte Carlo method and unscented transform methods \[1, 50\].

The order of a system refers to the number of initial conditions, or random variables, it requires as an input. For instance, the IOD described in this paper is based on a 6th order system, since it has 3 initial conditions that describe the position vector and 3 initial conditions that describe the velocity vector. In order to calculate the expectation integral, each initial condition must be varied; more initial conditions results in more possible combinations of initial conditions, which the expectation integral must consider. Therefore, computational expense is greatly increased for higher order systems, since the relationship between the number of initial conditions and the number of points required to approximate the expectation integral is exponential \[1\]. For instance, in order to approximate the expectation integral of a 2nd order system with only 5 points in each direction, \(5^2 = 25\) points are required. However, to approximate the expectation integral of a 6th order system with 5 points in each direction, \(5^6 = 15,625\) points are required \[1\].

2.3.1 Monte Carlo Method

The Monte Carlo (MC) method involves generating a large number of random points using the Gaussian PDF that describes the initial conditions, and propagating these points forwards in time \[45\]. It is reliable, but it can be computationally expensive even for lower-order systems, since it requires many points to achieve high accuracy. Additionally, computational expense is greatly compounded for higher-order systems,
such as the 6th order system of equations required for IOD. This large number of required points can render the Monte Carlo method, which is already time consuming for lower-order systems, incredibly computationally expensive [1].

2.3.2 Expectation Integral and Cubature Methods

Let us consider the problem of computing the expected value of a real valued function $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ with respect to a multivariate Gaussian PDF. Without loss of generality, consider the mean and covariance of the Gaussian PDF to be zero and unity, respectively. The multidimensional expectation integral is generally approximated by a linear functional which can be interpreted as a weighted sum of function evaluations at specific pre-determined points $x_i \in \mathbb{R}^n$ and weights $w_i \in \mathbb{R}$:

$$E[f(x)] = \int \int \cdots \int f(x) \mathcal{N} x \mu \mathbf{P} dx_1 dx_2 \cdots dx_n \approx \sum_{i=1}^{N} w_i f(x_i)$$  (2.2)

where $\mathcal{N} x \mu \mathbf{P}$ represents a $n \mathbb{D}$ Gaussian PDF with mean $\mu$ and covariance $\mathbf{P}$. The expectation integral in Equation 2.2 for a zero mean and identity covariance Gaussian PDF is given by the equation:

$$E[f(x)] = \int \int \cdots \int f(x) \mathcal{N} x \mathbf{0} \mathbf{I} dx_1 dx_2 \cdots dx_n.$$  (2.3)

Assuming that $f(x)$ has a valid Taylor series expansion about the mean $\mathbf{0}$, Equation 2.3 can be re-written as:
\[
E[f(x)] \approx \sum_{N_1=0}^{\infty} \sum_{N_2=0}^{\infty} \cdots \sum_{N_n=0}^{\infty} \frac{E[x_1^{N_1} x_2^{N_2} \cdots x_n^{N_n}]}{N_1! N_2! \cdots N_n!} \frac{\partial^{N_1+N_2+\cdots+N_n} f}{\partial x_1^{N_1} \partial x_2^{N_2} \cdots \partial x_n^{N_n}}(0). \tag{2.4}
\]

Notice that the problem of evaluating the expected value of nonlinear function \(f(x)\) has reduced to computing higher order moments of the Gaussian PDF. Thus, one can obtain a more accurate value of the expectation integral in Equation 2.3 by increasing the number of terms in the Taylor series expansion. Furthermore, consider the discrete approximation of Equation 2.3 as a weighted average of \(f(x)\) evaluated at the cubature points \((x_1, x_2, \cdots, x_N)\) with corresponding weights \((w_1, w_2, \cdots, w_N)\), where each \(x_i\) is a \(n\)-D vector as \(x_i = [x_{(i,1)}, x_{(i,2)}, \cdots, x_{(i,n)}]^T\), i.e. \(x_{(i,j)}\) represents the \(j^{th}\) coordinate of the \(i^{th}\) point. Equation 2.3 can be approximated as:

\[
E[f(x)] \approx \sum_{i=1}^{N} w_i f(x_i). \tag{2.5}
\]

Now, substituting the Taylor series expansion of \(f(x_i)\) about the 0 in Equation 2.5 leads to:

\[
E[f(x)] \approx \sum_{N_1=0}^{\infty} \sum_{N_2=0}^{\infty} \cdots \sum_{N_n=0}^{\infty} \frac{(\sum_{i=1}^{N} w_i \{x_{(i,1)}^{N_1} x_{(i,2)}^{N_2} \cdots x_{(i,n)}^{N_n}\})}{N_1! N_2! \cdots N_n!} \frac{\partial^{N_1+N_2+\cdots+N_n} f}{\partial x_1^{N_1} \partial x_2^{N_2} \cdots \partial x_n^{N_n}}(0). \tag{2.6}
\]

Comparing the coefficients of \(f(0)\) and derivatives of \(f(x)\) evaluated at 0 from Equations 2.4 and 2.6 leads to the following set of equations known as the Moment Con-
straint Equations (MCE):

\[
\sum_{i=1}^{N} w_i \{ x_{(i,1)}^{N_1} x_{(i,2)}^{N_2} \cdots x_{(i,n)}^{N_n} \} = E[x_1^{N_1} x_2^{N_2} \cdots x_n^{N_n}] 
\]

(2.7)

where \( \{N_1 + N_2 \cdots + N_n = d\} \) represents the order/degree of the moment of the Gaussian PDF. Alternatively, the MCE in Equation 2.7 can be determined by stating that the sigma point set \((X, w)\) can integrate all monomials (hence polynomials) of degree up to and including \(d\). This is known as the moment matching method.

Notice that these constraint equations simply convey that the sigma point set \((X, w)\) should ideally capture all the infinite moments of the PDF. Often, this is not required, so one seeks to find a sigma point set \((X, w)\) that can reproduce the first \(d\) moments of the PDF exactly to guarantee the exact evaluation of the expectation integral of Equation 2.3 when \(f(x)\) is a polynomial function of total degree \(d\) or less. On the other hand, when integrating non-polynomial functions, one usually does not know beforehand the number of moments to be satisfied but one can expect that higher order cubature points would have lower error than the lower order cubature points. These moment constraint equations are high degree multivariate polynomial equations, the solution of which is not trivial and in some cases even intractable. The Gauss-Hermite (GH) Product rule in \(n\mathbb{D}\) indeed satisfy these moment constraint equations theoretically to any prescribed order \(d\) by first generating points in \(1\mathbb{D}\) that are exact up to order \(d\) and then taking the tensor product in \(n\mathbb{D}\). Hence, they suffer from the **curse of dimensionality** with the number of points increasing exponentially with dimension. Nevertheless, it is interesting to observe that Gauss-Hermite quadratures are symmetric
about the mean and thereby inherently satisfy all moments involving any odd exponent; i.e; \( N_i \) is odd.

As an alternative to the Gaussian quadrature scheme, the non-product cubature methods solve these moment constraint equations by judiciously selecting sigma point set in \( \mathbb{R}^n \) and offer similar numerical accuracy with fewer points. However, the development of a cubature rule of any given degree \( 'd' \) that is applicable to any dimension is still an open problem. The Smolyak quadrature scheme and Unscented Transform (UT) are one of the most widely used non-product quadrature schemes. The Smolyak quadrature scheme is a general cubature rule of any given degree \( 'd' \) and involves sparse tensor product of 1D Gaussian quadrature rule. However, it can result in quadrature points with negative weights. In [32] and [51], the authors specifically reiterate the necessity of positive weight cubature points for stable and accurate computations of expectation integrals. The UT is only degree three quadrature rule and involves the judicious selection of sigma set by constraining the points to lie on prescribed directions in \( \mathbb{R}^n \). The main focus of this work is to extend the UT rules to higher order quadrature rules by selecting additional axes or directions in \( \mathbb{R}^n \).

### 2.3.3 Unscented Kalman Filter

The Unscented Kalman Filter (UKF) is a type of UT cubature method that provides a potential alternative to the Monte Carlo method and has excellent applications to nonlinear fitting. Instead of randomly generating points like the Monte Carlo Method, the UKF generates specific points for initial conditions, assigns them weights,
and then propagates them forwards in time. Each set of initial conditions, referred to as a sigma point, results in a final state, which is still associated with a weight \([1, 48]\).

According to Adurthi (2017), the UKF can be used to describe an \(n^{th}\) order system with only \(2n + 1\) sigma point sets. For an arbitrary covariance matrix \(P\) and mean vector \(\mu\), the suggested sigma point set \((X, \bar{w})\) is generated as shown in Equations 2.8-2.11:

\[
\begin{align*}
\bar{x}_0 &= \mu \\
\bar{w}_0 &= \frac{\kappa}{n + \kappa} \\
\bar{x}_i &= \mu + (\sqrt{(n + \kappa)P})_i \\
\bar{w}_i &= \frac{1}{2(n + \kappa)} \\
\bar{x}_{i+n} &= \mu + (\sqrt{(n + \kappa)P})_i \\
\bar{w}_{i+n} &= \frac{1}{2(n + \kappa)} \\
\kappa &= \begin{cases} 
3 - n & n \leq 3 \\
1 & n > 3 
\end{cases}
\end{align*}
\]

where

- \(\bar{x}_0, \bar{x}_i, \bar{x}_{i+n}\) = sigma points
- \(\bar{w}_0, \bar{w}_i, \bar{w}_{i+n}\) = weights associated with each sigma point.

Once calculated, each row of the matrix \(X\) contains a sigma point, \(\bar{x}\), which describes a set of initial conditions. Each sigma point is propagated forward in time by numerically integrating the IOD equations, resulting in a final state. These final
states, when considered with their corresponding weights, can be used to calculate the weighted average and covariance matrix of an orbital object’s final position and velocity. Because it requires far fewer points than the Monte Carlo method, the UKF could present a far more computationally efficient way of calculating the expectation integral [1].

2.3.3.1 Unscented Transforms and Higher Order Moment Constraint Equations

Let us consider the problem of selecting sigma points such that the first four moment equations (MCEs) are satisfied. Assuming Gaussian random variables $\mathbf{x} = [x_1, x_2, \cdots, x_n]^T$ with zero mean and identity covariance, the higher order moments up to fourth order are given as:

\[
E[x_i^2] = 1, \quad E[x_i x_j] = 0, \quad E[x_i^4] = 3, \\
E[x_i^3 x_j] = 0, \quad E[x_i^2 x_j^2] = 1, \quad E[x_i^2 x_j x_k] = 0, \\
E[x_i x_j x_k x_l] = 0, \quad \qquad \quad (2.12)
\]

where $\{i, j, k, l\} \subset \{1, 2, 3, \ldots, n\}$. As in the conventional UT scheme, a fully symmetric set of sigma points are chosen such that they lie on the orthogonal cartesian axes at a distance $r_1$ from the origin and each have equal weight of $w_1$. The central point on the origin (mean) has weight $w_0$. When enumerated the set of points are $(r_1, 0, \cdots, 0)$, $(0, r_1, \cdots, 0)$, $(0, 0, \cdots, r_1)$, $(-r_1, 0, \cdots, 0)$, $(0, -r_1, \cdots, 0)$, $(0, 0, \cdots, -r_1)$, $(0, 0, \cdots, 0)$.
and the corresponding MCE are:

\[ E[x_i^0] \equiv w_0 + 2nw_1 = 1 \quad (2.13) \]
\[ E[x_i^2] \equiv 2r_1^2w_1 = 1 \quad (2.14) \]
\[ E[x_i^4] \equiv 2r_1^4w_1 = 3 \quad (2.15) \]
\[ E[x_i^2x_j^2] \equiv 0 \neq 1. \quad (2.16) \]

It can be seen that the choice of \( r_1 = \sqrt{n + \kappa} \) in Equation 2.14 leads to the same weights in Equations 2.9 and 2.10. Similarly, the calculation of the central weight in Equation 2.13 eventually leads to Equation 2.8. It should be noticed that moments of Equation 2.12 containing odd exponents are already satisfied due to the symmetry of \( 2n + 1 \) sigma points. Furthermore, the \( 4^{th} \) order moment in Equation 2.15 leads to \( n + \kappa = 3 \). The cross moment \( E[x_i^2x_j^2] \) in Equation 2.16 cannot be satisfied by this particular selection of symmetric sigma points that are constrained to lie only on the principal axes.

In fact, no cross moment of any order can be satisfied by the selection of symmetric points constrained to lie only on the orthogonal cartesian axes. Hence, by merely adding more points or scaling points, they can only minimize the error in the higher order moments but cannot capture even-order cross moments. For a Gaussian PDF, moments such as \( E[x_1^{N_1}x_2^{N_2} \cdots x_n^{N_n}] \) are non-zero when \( N_1, N_2, \ldots, N_n \) are all even numbers. However, points on the orthogonal cartesian axes, which are of the form \((\pm r, 0, 0, \ldots, 0)\), always produce a zero when substituted into the left hand side of the MCE for all cross moments.
Also notice that the UKF is a special case of UT when $\kappa = 0$. Hence the UT and UKF have discrepancy in the 4th order moments. In summary, the set of non-linear moment constraint equations are tedious to solve for a general $nD$ system. Either one needs to break the symmetry of the sigma points or one needs to look for alternative axes to define new sigma points. Breaking the symmetry of sigma points is not desirable since this will require one to include moment equations containing odd exponents in the formulation. In this paper, the second option is pursued by including additional axes which are labeled as conjugate axes such that the resultant points are more ‘spread out’ symmetrically in space.

### 2.3.4 Conjugate Unscented Transforms

In this section, the Conjugate Unscented Transform (CUT) is introduced as an extension of the conventional UT to define new sigma points which can capture higher order moments. As discussed in the previous section, the UT provides the insight that “the points can be constrained to some carefully selected axes” and one only needs to solve for the distances of these points from the origin and their corresponding weights. The set of moment constraint equations along with these additional constraints can make the system tractable for higher orders and dimensions. However, choosing appropriate axes is still not a trivial problem. Hence, as a first step towards outlining a generic approach, the following important axes are defined to select higher order sigma points.
Figure 2.2: Symmetric set of points and axes 2D and 3D space
In \( n \)-D Cartesian space, the \textit{Principal axes} are defined as the \( n \) orthogonal axes centered at the origin. The points on the principal axes are enumerated as

\[
\sigma_i \in \{ \pm (I)_k | k \in \mathcal{D} \}, \quad i = 1, 2, 3, \ldots, 2n
\]  

(2.17)

where \( \mathcal{D} = \{1, 2, 3, \ldots, n\} \) and \((I)_k\) represents the \(k^{th}\) row or column of the \(n \times n\) dimensional identity matrix \(I\). Each point on the principle axes is at a unit distance from the origin.

In \( n \)-D Cartesian space, the \textit{mth-conjugate axes} with \( m \leq n \) is defined as the directions that are constructed from all the combinations, including the sign permutations, of the set of principal axes taken \( m \) at a time. The set of \textit{mth} conjugate axes labeled as \( c^m \), where the points are listed as \( c^m_i \) are

\[
c^m_i \in \{ \pm (\sigma_{N_1} \pm \sigma_{N_2} \ldots \pm \sigma_{N_m}) | \{N_1, N_2, \ldots, N_m\} \subset \mathcal{D} \} \quad i = 1, 2, \ldots, 2^m \binom{n}{m}
\]

In \( n \)-D Cartesian space, the \textit{mth-Scaled conjugate axes} is defined as the set of directions constructed from all combinations, including sign permutations, of the set of principal axes such that in every combination exactly one principal axis is scaled by a parameter \( h \in \mathbb{R} \). The set of \textit{mth}-Scaled conjugate axes are labeled as \( s^m \), and the points are listed as \( s^m_i \).

\[
s^m_i \in \{ \pm h(\sigma_{N_1} \pm \sigma_{N_2} \ldots \pm \sigma_{N_m}) | \{N_1, N_2, \ldots, N_m\} \subset \mathcal{D} \} \quad i = 1, 2, \ldots, n2^m \binom{n}{m}
\]
The various axes for the 2D case are shown in Fig. 2.2(a) while Figs. 2.2(b) and 2.2(c) show two different perspective views of the first octant for a 3D case. It should be mentioned that all the eight octants in the 3D case are symmetrical.

The next step involves constraining the sigma points to lie on these special axes so that they satisfy higher order MCE. The main steps of generic procedure are outlined as follows:

1. As the Gaussian PDF is symmetric, choosing points on the symmetric axes inherently satisfy all the moment constraint equations involving odd exponent. If the set $X = \{x_1, x_2, \ldots, x_n\}$ is a fully symmetric set, then it is closed under the operations of coordinate position and sign permutations. For example, in 2D, if $x_i = [a, b]^T \in X$, then also $\{[b, a]^T, [-a, b]^T, [a, -b]^T, [-a, -b]^T, [-b, a]^T, [b, -a]^T, [-b, -a]^T\} \in X$. The various structures of symmetric sigma point sets used to develop the CUT method are summarized in Table 2.1. Only one sample point is shown for each type of axes. Given a single point, the fully symmetric set for each type can be easily completed by taking all possible permutations of coordinate position and sign.

Equation 2.18 provides an example of a 3 dimensional system in which the symmetric points from Table 2.1 are explicitly illustrated.
Table 2.1: Fully Symmetric set of points

<table>
<thead>
<tr>
<th>Type</th>
<th>Sample Point</th>
<th>No. of points</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma$</td>
<td>$(1, 0, 0, \cdots, 0)$</td>
<td>$2n$</td>
</tr>
<tr>
<td>$c^m$</td>
<td>$(1, 1, \cdots, 1, 0, 0, \cdots, 0)$</td>
<td>$2^m \binom{n}{m}$</td>
</tr>
<tr>
<td>$s^n(h)$</td>
<td>$(h, 1, 1, \cdots, 1)$</td>
<td>$n2^n$</td>
</tr>
</tbody>
</table>

$c_i^3 \in \{ (1,1,0), (0,1,1), (-1,1,0), (-1,0,1), (0,-1,1) \}$, $c_i^3 \in \{ (1,1,1), (1,0,-1), (-1,1,0), (-1,0,-1), (0,-1,1) \}$, $\sigma_i \in \{ (1,1,1), (1,0,0), (0,1,0), (0,0,1) \}$.
Points on the same set of symmetric axes are equidistant from the mean and have equal weights. For the $i^{th}$ set of axes, the distance scaling variables are labeled as $r_i$ and weight variables are labeled as $w_i$.

2. The moment constraint equations up to the desired order are derived in terms of the unknown variables $r_i$ and $w_i$ by making use of the moments of a Gaussian PDF. Each sigma point set in Table 2.1 can be scaled by a variable $r_i$, assigned a weight $w_i$ and substituted into the left hand side of Equation 2.7 resulting in a polynomial equation. Hence, each moment constraint equation would contain monomials with variables $r_i$ and $w_i$ corresponding to each sigma point set. The coefficients of these monomials are tabulated in Table 2.2 and can be used directly in generating the moment constraint equations. The triplet $[a, b, c]$ represents the monomial $ar_i^bw_i^c$ for the $i^{th}$ set of axes, where $r_i$ and $w_i$ are the unknown variables. Here, $n$ is the dimension of the integral and $\binom{n}{m}$ is the binomial coefficient.
In the following sections, these new sigma point sets are explicitly derived by making use of these **conjugate axes** to satisfy the moment constraint equations up to order eight. Without loss of generality, the Gaussian density function is assumed to have zero mean and identity covariance matrix. If \( \mathbf{x}_i \) represent the sigma points corresponding to \( \mathcal{N} \mathbf{x} \mathbf{0} \mathbf{I} \) and \( \mathbf{y}_i \) are the sigma points corresponding to the generic Gaussian PDF \( \mathcal{N} \mathbf{y} \mu \mathbf{P} \),

<table>
<thead>
<tr>
<th>Moment</th>
<th>( \sigma )</th>
<th>( c^m )</th>
<th>( s^m(h) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( E[x_i^p] ) ( p=1,2,3,4 )</td>
<td>( [2, p, 1] )</td>
<td>( [2^m \binom{n-1}{m-1}, p, 1] )</td>
<td>( [(n-1)2^m + h^2 2^m, p, 1] )</td>
</tr>
<tr>
<td>( E[x_i^2 x_j^2] )</td>
<td>( [0, 4, 1] )</td>
<td>( [2^m \binom{n-2}{m-2}, 4, 1] )</td>
<td>( [2^{n+1}h^2 + (n-2)2^n, 4, 1] )</td>
</tr>
<tr>
<td>( E[x_i^4 x_j^2] )</td>
<td>( [0, 6, 1] )</td>
<td>( [2^m \binom{n-2}{m-2}, 6, 1] )</td>
<td>( [2^m h^2 + 2^n h^4 + (n-2)2^n, 6, 1] )</td>
</tr>
<tr>
<td>( E[x_i^6 x_j^2] )</td>
<td>( [0, 8, 1] )</td>
<td>( [2^m \binom{n-2}{m-2}, 8, 1] )</td>
<td>( [2^{n+1}h^4 + (n-2)2^n, 8, 1] )</td>
</tr>
<tr>
<td>( E[x_i^8 x_j^2] )</td>
<td>( [0, 8, 1] )</td>
<td>( [2^m \binom{n-3}{m-3}, 8, 1] )</td>
<td>( [2^n h^6 + 2^m h^2 (n-2)2^n, 8, 1] )</td>
</tr>
<tr>
<td>( E[x_i^4 x_j x_k^2] )</td>
<td>( [0, 6, 1] )</td>
<td>( [2^m \binom{n-3}{m-3}, 6, 1] )</td>
<td>( [3(2^n)h^2 + (n-3)2^n, 6, 1] )</td>
</tr>
<tr>
<td>( E[x_i^4 x_j x_k^2 x_l^2] )</td>
<td>( [0, 8, 1] )</td>
<td>( [2^m \binom{n-3}{m-3}, 8, 1] )</td>
<td>( [2^n h^4 + 2^{n+1}h^2 + (n-3)2^n, 8, 1] )</td>
</tr>
<tr>
<td>( E[x_i^2 x_j^2 x_k^2 x_l^2] )</td>
<td>( [0, 8, 1] )</td>
<td>( [2^m \binom{n-4}{m-4}, 8, 1] )</td>
<td>( [2^{n+2}h^2 + (n-4)2^n, 8, 1] )</td>
</tr>
</tbody>
</table>

3. This non-linear set of equations are solved for \( r_i \) and \( w_i \), yielding the required sigma point set. Usually for lower order moment constraint equations, the polynomial equations can be solved analytically or by the help of symbolic computation software. For higher order moment constraint equations, efficient polynomial solvers such as “Bertini” [3] can be used.
the points $x_i$ can be transformed to the points $y_i$ by an affine transformation:

$$y_i = \Sigma x_i + \mu, \Sigma = \sqrt{P}.$$  \hspace{1cm} (2.19)

### 2.3.4.1 4th Order Conjugate Unscented Transformation (CUT4)

CUT4 stands for the conjugate unscented transformation sigma points that are 4th moment equivalent or the sigma points that can completely satisfy the moment constraint equations up to the 4th order and can integrate all polynomials with total degree 5 or less. The moment constraint equations that are not automatically satisfied by the selection of the symmetric set of sigma points are:

$$E[x_i^2] = 1, \quad E[x_i^4] = 3, \quad E[x_i^2x_j^2] = 1.$$  \hspace{1cm} (2.20)

These equations can be solved by introducing a minimum of five variables $r_1, r_2, w_0, w_1$ and $w_2$. Note that the sum of the weights should be equal to 1. The resulting problem is an underdetermined problem which can be solved by either posing an optimization problem or by arbitrarily setting $w_0$ to 0 and solving for the remaining parameters. Since the conjugate axes provide a non-unique selection of points to solve the MCEs shown in Equation 2.20, potential benefits of selection of specific axes must be analyzed.

Table 2.3 illustrates the growth in the number of points as a function of dimension for various conjugate axes. The sigma and $c^n$ columns and cells refer to the principal and second conjugate axis which were used by Julier and Uhlmann [24]. Note
Table 2.3: Fully Symmetric set of points for CUT4

<table>
<thead>
<tr>
<th>Dimension</th>
<th>$\sigma$</th>
<th>$c^2$</th>
<th>$c^3$</th>
<th>$c^4$</th>
<th>$c^5$</th>
<th>$c^6$</th>
<th>$c^7$</th>
<th>$c^8$</th>
<th>$s^n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>4</td>
<td>4</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>12</td>
<td>8</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>24</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>24</td>
<td>32</td>
<td>16</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>64</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>40</td>
<td>80</td>
<td>80</td>
<td>32</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>160</td>
</tr>
<tr>
<td>6</td>
<td>12</td>
<td>60</td>
<td>160</td>
<td>240</td>
<td>192</td>
<td>64</td>
<td>-</td>
<td>-</td>
<td>384</td>
</tr>
<tr>
<td>7</td>
<td>14</td>
<td>84</td>
<td>280</td>
<td>560</td>
<td>672</td>
<td>448</td>
<td>128</td>
<td>-</td>
<td>896</td>
</tr>
<tr>
<td>8</td>
<td>16</td>
<td>112</td>
<td>448</td>
<td>1120</td>
<td>1792</td>
<td>1792</td>
<td>1024</td>
<td>256</td>
<td>2048</td>
</tr>
<tr>
<td>9</td>
<td>18</td>
<td>144</td>
<td>672</td>
<td>2016</td>
<td>4032</td>
<td>5376</td>
<td>4608</td>
<td>2304</td>
<td>512</td>
</tr>
</tbody>
</table>

that for $n = 3$, selection of the principal axis in conjunction with the 3rd conjugate axis results in the smallest number of points necessary to satisfy the required constraints.

Julier and Uhlmann have described an approach in the Appendix of Ref. [24] to calculate the sigma points that can capture all the fourth order moments. However, this method suffers from the presence of a negative/zero weight for dimensions higher than or equal to 4. It is to be noted that this cubature rule with a similar drawback can be seen in [46]. To gain more insight, these equations are re-derived and solved.

Let us consider that one sigma point of weight $w_0$ lies on the origin, $2n$ points of weight $w_1$ lie symmetrically on each principal axes at a distance of $r_1$ and $2n(n - 1)$ points of weight $w_2$ lie symmetrically on the 2nd-conjugate axes ($c^2$) at a distance scaled by $r_2$. As the points are symmetric about the origin, the moments involving odd
The moment constraint equations up to order 4 for this particular selection of points are given as:

\begin{align*}
2r_1^2w_1 + 4(n - 1)r_2^2w_2 &= 1 \quad (2.21) \\
2r_1^4w_1 + 4(n - 1)r_2^4w_2 &= 3 \quad (2.22) \\
4r_2^4w_2 &= 1 \quad (2.23) \\
1 - 2nw_1 - 2n(n - 1)w_2 &= w_0. \quad (2.24)
\end{align*}

The central weight merely helps to sum the weights to 1. The variables \( r_1, w_1, r_2, w_2 \) can be solved from Equations 2.21-2.23 and \( w_0 \) can be found from Equation 2.24. Hence, there are only 4 main variables \( r_1, w_1, r_2, w_2 \) and 3 equations leading to an underdetermined system of equations. In [24], the authors choose to minimize the error in one of the 6\(^{th}\) order moment. Before the optimization is carried out, one can simplify the constraint equations as follows:

\begin{align*}
w_2 &= \frac{1}{4r_2^4}, \quad w_1 = \frac{4 - n}{2r_1^4} \quad (2.25) \\
r_1^2r_2^2 &= r_1^2(n - 1) + r_2^2(4 - n). \quad (2.26)
\end{align*}

As a consequence of this simplification, there is only one constraint equation (2.26) to solve in terms of the variables \( r_1 \) and \( r_2 \). Even though it can be solved by minimizing the 6\(^{th}\) moment constraint violation, the weight \( w_1 \) in Equation 2.25 would always result in a negative weight for dimension greater than 4. One can address this problem of negative weight by choosing a different set of conjugate axes. Table 2.4
Table 2.4: Fully Symmetric set of points for CUT4

<table>
<thead>
<tr>
<th>Dimension</th>
<th>$\sigma$</th>
<th>$c^2$</th>
<th>$c^3$</th>
<th>$c^4$</th>
<th>$c^5$</th>
<th>$c^6$</th>
<th>$c^7$</th>
<th>$c^8$</th>
<th>$c^9$</th>
<th>$s^9$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>4</td>
<td>4</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>12</td>
<td>8</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>24</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>24</td>
<td>32</td>
<td>16</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>64</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>40</td>
<td>80</td>
<td>80</td>
<td>32</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>160</td>
</tr>
<tr>
<td>6</td>
<td>12</td>
<td>60</td>
<td>160</td>
<td>240</td>
<td>192</td>
<td>64</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>384</td>
</tr>
<tr>
<td>7</td>
<td>14</td>
<td>84</td>
<td>280</td>
<td>560</td>
<td>672</td>
<td>448</td>
<td>128</td>
<td>-</td>
<td>-</td>
<td>896</td>
</tr>
<tr>
<td>8</td>
<td>16</td>
<td>112</td>
<td>448</td>
<td>1120</td>
<td>1792</td>
<td>1792</td>
<td>1024</td>
<td>256</td>
<td>-</td>
<td>2048</td>
</tr>
<tr>
<td>9</td>
<td>18</td>
<td>144</td>
<td>672</td>
<td>2016</td>
<td>4032</td>
<td>5376</td>
<td>4608</td>
<td>2304</td>
<td>512</td>
<td>4608</td>
</tr>
</tbody>
</table>

highlights the symmetric sets of points which lie on the principal and the $n^{th}$-conjugate axes which will be used to derive the moment constraint equations.

The new sigma points are selected such that $2n$ points of weight $w_1$ lie symmetrically on each principal axes at a distance of $r_1$ and $2^n$ points of weight $w_2$ lie on the $n^{th}$-conjugate axes at a distance of $r_2\sqrt{n}$. Substitution of this particular selection of sigma points as summarized in Table 2.5 leads to the following $4^{th}$ order moment constraint equations:

\[
2r_1^2w_1 + 2^nr_2^2w_2 = 1 \quad (2.27)
\]

\[
2r_1^4w_1 + 2^nr_2^4w_2 = 3 \quad (2.28)
\]

\[
2^nr_2^4w_2 = 1 \quad (2.29)
\]
1 − 2nw_1 − 2^nw_2 = w_0. \hspace{1cm} (2.30)

The weight \( w_0 \) corresponds to the central point at the mean and does not contribute to any of the moment constraint equations of order \( d \geq 1 \). There are 5 variables and only 4 equations. One can eliminate the central point by choosing \( w_0 = 0 \). This selection of \( w_0 \) results in an analytical solution for the points, which is shown in Equations 2.31-2.32.

\[
\begin{align*}
  r_1 &= \sqrt{\frac{n+2}{n-2}}, & r_2 &= \sqrt{\frac{n+2}{n-2}} \\
  w_1 &= \frac{1}{2n^2} = \frac{4}{(n+2)^2}, & w_2 &= \frac{1}{2^nr_2^4} = \frac{(n-2)^2}{2^n(n+2)^2}
\end{align*}
\hspace{1cm} (2.31)

(2.32)

Notice that \( r_2 \) is undefined for \( n \leq 2 \) and thereby this solution is valid for \( n \geq 3 \). Alternatively, one can also find the central weight \( w_0 \) by minimizing the error in one of the \( 6^{th} \) moment constraint equation namely \((2r_1^6w_1 + 2^nr_2^6w_2 - 15)^2\). The solution for dimension \( n = 2 \) is shown in Table 2.6 where the \( 6^{th} \) moment constraint equation error has been minimized. In addition, for \( n = 1 \), there is no cross order moment and Equation 2.29 can be replaced by the \( 6^{th} \) moment constraint equation leading to the standard Gauss-Hermite quadrature with 4 points. The take away of this analysis is the user should select the \( n^{th} \)- conjugate axes in conjunction with the principal axes for CUT4.

Note that, according to the CUT4, one would need 14 and 1044 sigma points to exactly evaluate the expectation of a polynomial function of degree 4 in 3- and 10-dimensional space, whereas the GH Product rule would require a minimum of 27 and
Table 2.5: Sigma Points for CUT4

<table>
<thead>
<tr>
<th>Position</th>
<th>Weights</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1 \leq i \leq n)</td>
<td>(x_i = r_1 \sigma_i) (w_i = w_1)</td>
</tr>
<tr>
<td>(1 \leq i \leq 2^n)</td>
<td>(x_{i+2N} = r_2 c_i^N) (w_{i+2N} = w_2)</td>
</tr>
<tr>
<td>Central weight</td>
<td>(x_0 = 0) (w_0 = w_0)</td>
</tr>
</tbody>
</table>

\[ N = 2n + 2^n \quad (+1 \text{ when } w_0 \neq 0) \]

59,049 points, respectively. This striking difference is the motivation to develop a 6\(^{th}\) moment equivalent method, ‘CUT6’, in the next section.
Table 2.6: CUT4: Optimized Solution for \( n = 1 \) and \( n = 2 \)

<table>
<thead>
<tr>
<th>Variable</th>
<th>( n = 1 )</th>
<th>( n = 2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( r_1  )</td>
<td>0.7419637843027252</td>
<td>2.606099476935847</td>
</tr>
<tr>
<td>( r_2  )</td>
<td>2.3344142183389796</td>
<td>1.190556300661233</td>
</tr>
<tr>
<td>( w_0  )</td>
<td>0</td>
<td>0.41553535186548973</td>
</tr>
<tr>
<td>( w_1  )</td>
<td>0.45412414523193145</td>
<td>0.021681819434216532</td>
</tr>
<tr>
<td>( w_2  )</td>
<td>0.04587585476806854</td>
<td>0.12443434259941118</td>
</tr>
<tr>
<td>No. of points</td>
<td>4</td>
<td>9</td>
</tr>
</tbody>
</table>

2.3.4.2 6\(^{th}\) Order Conjugate Unscented Transform: CUT6

This section describes a similar procedure to solve all the moment constraint equations up to 6\(^{th}\) order. The even order moments up to order 6 that must be satisfied are:

\[
E[x_i^2] = 1, \quad E[x_i^4] = 3, \quad E[x_i^2 x_j^2] = 1, \\
E[x_i^6] = 15, \quad E[x_i^4 x_j^2] = 3, \quad E[x_i^2 x_j^2 x_k^2] = 1. \quad (2.33)
\]

These equations can be solved by introducing at least 6 variables \( r_1, r_2, r_3, w_1, w_2, w_3 \) to make the system consistent. Specifically, the sigma point on the principal axes have been assigned a weight of \( w_1 \) and are constrained to lie symmetrically at a distance \( r_1 \) from the origin. Points on the \( n^{th}\)-conjugate axes have been chosen with a weight \( w_2 \) and are constrained to lie symmetrically at a distance scaled by \( r_2 \). Finally, the third
set of points are selected with weight \( w_3 \) and are constrained to lie symmetrically at a distance scaled by \( r_3 \) along the 2\textsuperscript{nd}-conjugate axes. These points are enumerated in Table 2.7. The set of moment constraint equations using points in Table 2.7 are given as:

\[
\begin{align*}
2r_1^2w_1 + 2^n r_2^3w_2 + 4(n - 1)r_3^2w_3 &= 1 \quad (2.34) \\
2r_1^4w_1 + 2^n r_2^4w_2 + 4(n - 1)r_3^4w_3 &= 3 \quad (2.35) \\
2^n r_2^4w_2 + 4r_3^4w_3 &= 1 \quad (2.36) \\
2r_1^6w_1 + 2^n r_2^6w_2 + 4(n - 1)r_3^6w_3 &= 15 \quad (2.37) \\
2^n r_2^6w_2 + 4r_3^6w_3 &= 3 \quad (2.38) \\
2^n r_2^6w_2 &= 1 \quad (2.39) \\
1 - 2nw_1 - 2^n w_2 - 2n(n - 1)w_3 &= w_0. \quad (2.40)
\end{align*}
\]

Solving Equations 2.37-2.39 for unknown weights leads to the analytical expressions in Equation 2.41.

\[
w_1 = \frac{8 - n}{r_1^6} \quad w_2 = \frac{1}{2n r_2^6} \quad w_3 = \frac{1}{2r_3^6} \quad (2.41)
\]

The substitution of the analytical expressions in Equation 2.41 for the weight variables in Equations 2.34-2.36 leads to 3 polynomial equations in terms of \( r_1, r_2 \) and \( r_3 \).

\[
\begin{align*}
-2r_1^4r_2^4 + 2nr_1^4r_2^4 + r_1^4r_3^4 + 16r_2^4r_3^4 - 2nr_2^4r_3^4 - r_1^4r_2^4r_3^4 &= 0 \\
-2r_1^2r_2^2 + 2nr_1^2r_2^2 + r_1^2r_3^2 + 16r_2^2r_3^2 - 2nr_2^2r_3^2 - 3r_1^2r_2^2r_3^2 &= 0 \\
\end{align*}
\]
Furthermore, by defining \( r_1 = \frac{1}{\sqrt{a_1}} \), \( r_2 = \frac{1}{\sqrt{a_2}} \) and \( r_3 = \frac{1}{\sqrt{a_3}} \), the system of 3 equations in Equation 2.42 can be rewritten as:

\[
2r_2^2 + r_3^2 - r_2^2 r_3^2 = 0 \quad (2.42)
\]

\[
2(8 - n)a_1^2 + a_2^2 + 2a_3^2(n - 1) = 1
\]

\[
2(8 - n)a_1 + a_2 + 2a_3(n - 1) = 3
\]

\[
a_2 + 2a_3 = 1. \quad (2.43)
\]

This reduced system of equations in (2.43) is simpler to solve than the original system of equations given by (2.34)-(2.40). For dimensions greater than 6, the central weight becomes negative and above dimension 8 weight \( w_1 \) becomes negative. Therefore, this procedure is valid for \( n \leq 6 \). A similar method with the same drawback is presented in [47], for which the author has derived the 7th order equivalent set of sigma points valid only for specific dimensions \( n = 3, 4, 6, 7 \).

Furthermore, the problem of negative weight can be avoided by choosing 3rd-conjugate axes instead of 2nd-conjugate axes, which makes the proposed approach valid for \( 7 \leq n \leq 9 \). The set of 6th order moment constraint equations in terms of these new sigma points, as enumerated in Table 2.8, are given as:

\[
2r_1^2 w_1 + 2^n r_2^2 w_2 + 4(n - 1)(n - 2)r_3^2 w_3 = 1 \quad (2.44)
\]

\[
2r_1^4 w_1 + 2^n r_2^4 w_2 + 4(n - 1)(n - 2)r_3^4 w_3 = 3 \quad (2.45)
\]

\[
2^n r_2^4 w_2 + 8(n - 2)r_3^4 w_3 = 1 \quad (2.46)
\]
**Table 2.7:** Sigma Points for CUT6, \((n \leq 6)\)

<table>
<thead>
<tr>
<th>Position</th>
<th>Weights</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 \leq i \leq 2n</td>
<td>(x_i = r_1 \sigma_i)</td>
</tr>
<tr>
<td>1 \leq i \leq 2n</td>
<td>(x_{i+2n} = r_2 c_i^2)</td>
</tr>
<tr>
<td>1 \leq i \leq 2n(n-1)</td>
<td>(x_{i+2n+2n} = r_3 c_i^2)</td>
</tr>
<tr>
<td>Central weight</td>
<td>(x_0 = 0)</td>
</tr>
</tbody>
</table>

\[N = 2n^2 + 2^n + 1\]

**Table 2.8:** Sigma Points for CUT6, \((7 \leq n \leq 9)\)

<table>
<thead>
<tr>
<th>Position</th>
<th>Weights</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 \leq i \leq 2n</td>
<td>(x_i = r_1 \sigma_i)</td>
</tr>
<tr>
<td>1 \leq i \leq 2n</td>
<td>(x_{i+2n} = r_2 c_i^2)</td>
</tr>
<tr>
<td>1 \leq i \leq 4n(n-1)(n-2)/3</td>
<td>(x_{i+2n+2n} = r_3 c_i^2)</td>
</tr>
<tr>
<td>Central weight</td>
<td>(x_0 = 0)</td>
</tr>
</tbody>
</table>

\[N = 2n + 2^n + 4n(n-1)(n-2)/3 + 1\]
\[
2r_1^6w_1 + 2n r_2^6w_2 + 4(n-1)(n-2)r_3^6w_3 = 15 \tag{2.47}
\]
\[
2^n r_2^6w_2 + 8(n-2)r_3^6w_3 = 3 \tag{2.48}
\]
\[
2^n r_2^6w_2 + 8r_3^6w_3 = 1 \tag{2.49}
\]
\[
1 - 2nw_1 - 2^n w_2 - (4n(n-1)(n-2)/3)w_3 = w_0. \tag{2.50}
\]

Once again exploiting the linearity in the weights, one can first solve for weights analytically from Equations 2.47-2.49:

\[
\begin{align*}
w_1 &= -\frac{24r_2^2 + 3(11 - n)r_3^2 + (n-3)r_2^2r_3^2}{4r_1^2(r_2^2 - r_3^2)}, \\
w_2 &= \frac{-2^{-n} (r_3^2 - 3)}{r_2^4 (r_2^2 - r_3^2)}, \\
w_3 &= -\frac{3 - r_2^2}{8(n-2)r_3^4 (r_2^2 - r_3^2)}.
\end{align*}
\]

The solution for \(r_1\), \(r_2\) and \(r_3\) can be computed from the remaining three equations 2.44-2.46. Notice that this aforementioned set of equations 2.44-2.49 are preferred till dimension 9, at which point the central weight becomes negative for \(n > 9\). However, if one allows the presence of negative weights, the equations can be solved up to dimension \(n = 13\), after which real roots cannot be guaranteed. The results of evaluating the expectation integral by the CUT6 method show a contrasting difference in the number of points required. For example, the CUT6 would need only 49 and 1203 function evaluation, while the GH product rule would require a minimum of 256 and 262,144 function evaluations for the exact computation of the expectation integral of a polynomial function of degree 6 in 4- and 9-dimensional space, respectively. The solutions to the MCEs given in Equations 2.34-2.39 and 2.44-2.49 are provided in Table 2.10. By using Table 2.10 along with Tables 2.7 and 2.8, one can directly generate the sigma point set of order 6.
Table 2.9: Fully Symmetric set of points for CUT6

<table>
<thead>
<tr>
<th>Dimension</th>
<th>$\sigma$</th>
<th>$c^2$</th>
<th>$c^3$</th>
<th>$c^4$</th>
<th>$c^5$</th>
<th>$c^6$</th>
<th>$c^7$</th>
<th>$c^8$</th>
<th>$c^9$</th>
<th>$s^n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>4</td>
<td>4</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>12</td>
<td>8</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>24</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>24</td>
<td>32</td>
<td>16</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>64</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>40</td>
<td>80</td>
<td>80</td>
<td>32</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>160</td>
</tr>
<tr>
<td>6</td>
<td>12</td>
<td>60</td>
<td>160</td>
<td>240</td>
<td>192</td>
<td>64</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>384</td>
</tr>
<tr>
<td>7</td>
<td>14</td>
<td>84</td>
<td>280</td>
<td>560</td>
<td>672</td>
<td>448</td>
<td>128</td>
<td>-</td>
<td>-</td>
<td>896</td>
</tr>
<tr>
<td>8</td>
<td>16</td>
<td>112</td>
<td>448</td>
<td>1120</td>
<td>1792</td>
<td>1792</td>
<td>1024</td>
<td>256</td>
<td>-</td>
<td>2048</td>
</tr>
<tr>
<td>9</td>
<td>18</td>
<td>144</td>
<td>672</td>
<td>2016</td>
<td>4032</td>
<td>5376</td>
<td>4608</td>
<td>2304</td>
<td>512</td>
<td>4608</td>
</tr>
</tbody>
</table>


Table 2.10: Solutions for $2 \leq n \leq 9$, $6^{th}$ moment constraint equations, CUT6

<table>
<thead>
<tr>
<th>Variable</th>
<th>2D</th>
<th>3D</th>
<th>4D</th>
<th>5D</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_1$</td>
<td>2.4494897427</td>
<td>2.3587090379</td>
<td>2.2520650012</td>
<td>2.1213203430</td>
</tr>
<tr>
<td>$r_2$</td>
<td>1.1147379454</td>
<td>1.1198362859</td>
<td>1.1260325006</td>
<td>1.1338934189</td>
</tr>
<tr>
<td>$r_3$</td>
<td>3.2004125801</td>
<td>3.1421303838</td>
<td>3.0763780026</td>
<td>3</td>
</tr>
<tr>
<td>$w_1$</td>
<td>0.0277777777</td>
<td>0.0290351301</td>
<td>0.0306601632</td>
<td>0.0329218107</td>
</tr>
<tr>
<td>$w_2$</td>
<td>0.1302876649</td>
<td>0.0633844605</td>
<td>0.0306601632</td>
<td>0.0147033607</td>
</tr>
<tr>
<td>$w_3$</td>
<td>0.0004653012</td>
<td>0.0005195469</td>
<td>0.0005898367</td>
<td>0.0006858710</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variable</th>
<th>6D</th>
<th>7D</th>
<th>8D</th>
<th>9D</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_1$</td>
<td>1.9488352799</td>
<td>2.5512003554</td>
<td>2.4494897427</td>
<td>2.3439073215</td>
</tr>
<tr>
<td>$r_2$</td>
<td>1.1445968942</td>
<td>0.9642630979</td>
<td>1</td>
<td>1.0232622230</td>
</tr>
<tr>
<td>$r_3$</td>
<td>2.9068006056</td>
<td>2.3255766977</td>
<td>2.449489742</td>
<td>2.5342864499</td>
</tr>
<tr>
<td>$w_1$</td>
<td>0.0365072564</td>
<td>0.0126940628</td>
<td>0.0138888888</td>
<td>0.0150763910</td>
</tr>
<tr>
<td>$w_2$</td>
<td>0.0069487173</td>
<td>0.0048594459</td>
<td>0.00234375</td>
<td>0.0011342717</td>
</tr>
<tr>
<td>$w_3$</td>
<td>0.0008288549</td>
<td>0.0003950899</td>
<td>0.0002314814</td>
<td>0.0001572731</td>
</tr>
</tbody>
</table>
2.3.4.3 $8^{th}$ Order Conjugate Unscented Transform: CUT8

In this section sigma points are selected such that they satisfy all the moment constraint equations up to order 8. The 11 non-zero moments up to the $8^{th}$ order for a Gaussian PDF with zero mean and identity covariance are:

\[
\begin{align*}
E[x_i^2] &= 1, & E[x_i^4] &= 3, & E[x_i^2 x_j^2] &= 1, \\
E[x_i^6] &= 15, & E[x_i^4 x_j^2] &= 3, & E[x_i^8] &= 105, \\
E[x_i^4 x_j^2] &= 15, & E[x_i^4 x_j^4] &= 9, & E[x_i^4 x_j^2 x_k^2] &= 3, \\
E[x_i^2 x_j^2 x_k^2] &= 1, & E[x_i^2 x_j^2 x_k^2 x_l^2] &= 1. 
\end{align*}
\] (2.51)

To solve the system of 11 moment constraint equations one would require at least 11 variables. As the distance and weight variables appear in pairs, 6 sets of points including the scaled conjugate axes parameter $h$ would lead to 13 variables. The specific structure of points chosen to capture these 11 MCE are shown in Table 2.11.

The general form of the CUT8 equations is shown in Equation 2.52. Given a value of $n$ for $n\mathbb{D}$ space, these equations can be solved for $r_1$, $r_2$, $r_3$, $r_4$, $r_5$, $r_6$, $w_1$, $w_2$, $w_3$, $w_4$, $w_5$, $w_6$ and $w_0$. To solve these equations, it is necessary to assume an appropriate value for $h$ and one distance variable $r_i$ that renders the weights positive.

\[
\begin{align*}
E[x_i^2] &= 2r_1^2 w_1 + 2^n r_2^2 w_2 + 4(n-1) r_3^2 w_3 + 2^n r_4^2 w_4 + 4(n-1)(n-2) r_5^2 w_5 + [2^n(n-1) + 2^n h^2] r_6^2 w_6 = 1 \\
E[x_i^4] &= 2r_1^4 w_1 + 2^n r_2^4 w_2 + 4(n-1) r_3^4 w_3 + 2^n r_4^4 w_4 + 4(n-1)(n-2) r_5^4 w_5 + [2^n(n-1) + 2^n h^4] r_6^4 w_6 = 3 \\
E[x_i^6] &= 2r_1^6 w_1 + 2^n r_2^6 w_2 + 4(n-1) r_3^6 w_3 + 2^n r_4^6 w_4 + 4(n-1)(n-2) r_5^6 w_5 + [2^n(n-1) + 2^n h^6] r_6^6 w_6 = 15
\end{align*}
\]
Table 2.11: Sigma Points for CUT8, \((2 \leq n \leq 6)\)

<table>
<thead>
<tr>
<th>Position</th>
<th>Weights</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1 \leq i \leq 2n)</td>
<td>(x_i = r_i \sigma_i)</td>
</tr>
<tr>
<td>(1 \leq i \leq 2^n)</td>
<td>(x_i + 2n = r_2 s_1^n)</td>
</tr>
<tr>
<td>(1 \leq i \leq 2n(n-1))</td>
<td>(x_{i+2n} = r_s c_i^n)</td>
</tr>
<tr>
<td>(1 \leq i \leq 2^n)</td>
<td>(x_{i+2n+2} + 2(n-1) = r_4 c_i^n)</td>
</tr>
<tr>
<td>(1 \leq i \leq n1)</td>
<td>(x_i + 2n + 2(n-1) = r_5 c_i^n)</td>
</tr>
<tr>
<td>(1 \leq i \leq n2^n)</td>
<td>(x_i + 2n + 2(n-1) + 2^n + 2n+1 = r_6 c_i^n)</td>
</tr>
<tr>
<td>Central weight</td>
<td>(x_0 = 0)</td>
</tr>
</tbody>
</table>

\(N = 2n + 2^n + 2(n-1) + n1 + 2^n + n2^n + 1 : \{n1 = 4(n-1)(n-3)/3\}\)

\[
\begin{align*}
E[x_i^8] & = 2^n r_1^3 w_1 + 2^n r_2^3 w_2 + 4(n-1)r_2^3 w_3 + 2^n r_3^8 w_4 + 4(n-1)(n-2)r_3^8 w_5 + [2^n(n-1) + 2^n h^2]r_8^8 w_6 = 105 \\
E[x^2_i x^2_j] & = 2^n r_2^3 w_2 + 4^n r_2^3 w_3 + 2^n r_3^4 w_4 + 8(n-2)r_3^4 w_5 + [2^n(n-2) + 2^n h^2]r_8^8 w_6 = 1 \\
E[x^3_i x^3_j] & = 2^n r_2^3 w_2 + 4^n r_2^3 w_3 + 2^n r_4^8 w_4 + 8(n-2)r_4^8 w_5 + [2^n(n-2) + 2^n h^2 + 2^n R^4]r_8^8 w_6 = 3 \\
E[x^4_i x^4_j] & = 2^n r_2^3 w_2 + 4^n r_2^3 w_3 + 2^n r_3^8 w_4 + 8(n-2)r_3^8 w_5 + [2^n(n-2) + 2^n h^2 + 2^n h^2]r_8^8 w_6 = 9 \\
E[x^5_i x^5_j] & = 2^n r_2^3 w_2 + 4^n r_2^3 w_3 + 2^n r_4^8 w_4 + 8(n-2)r_4^8 w_5 + [2^n(n-2) + 2^n h^2 + 2^n h^6]r_8^8 w_6 = 15 \\
E[x^6_i x^6_j] & = 2^n r_2^3 w_2 + 2^n r_4^8 w_4 + 8^n r_5^8 w_5 + [2^n(n-3) + 3 x 2^n h^2]r_8^8 w_6 = 1 \\
E[x^7_i x^7_j x^7_k] & = 2^n r_2^3 w_2 + 2^n r_4^8 w_4 + 8^n r_5^8 w_5 + [2^n(n-3) + 2^n h^2 + 2^n h^4]r_8^8 w_6 = 3 \\
E[x^8_i x^8_j x^8_k x^8_l] & = 2^n r_2^3 w_2 + 2^n r_4^8 w_4 + [2^n(n-4) + 2^n h^2 + 2^n h^4]r_8^8 w_6 = 1 \\
2nw_1 + 2w_2 + n(n-1)w_3 + 2n w_4 + \frac{4(n-1)(n-2)}{3}w_5 + n^2 w_6 + w_0 & = 1
\end{align*}
\]

(2.52)
For instance, in $6\mathrm{D}$ space this particular selection of points leads to the following 11 moment constraint equations:

\[
E[x_1^2] = 2r_1^2 w_1 + 64r_2^2 w_2 + 20r_3^2 w_3 + 64r_4^2 w_4 + 80r_5^2 w_5 + [320 + 64h^2]r_6^2 w_6 = 1
\]
\[
E[x_4^2] = 2r_1^4 w_1 + 64r_2^4 w_2 + 20r_3^4 w_3 + 64r_4^4 w_4 + 80r_5^4 w_5 + [320 + 64h^4]r_6^4 w_6 = 3
\]
\[
E[x_i^2 x_j^2] = 64r_2^4 w_2 + 4r_3^4 w_3 + 64r_4^4 w_4 + 32r_5^4 w_5 + [256 + 128h^2]r_6^4 w_6 = 1
\]
\[
E[x_1^6] = 2r_1^6 w_1 + 64r_2^6 w_2 + 20r_3^6 w_3 + 64r_4^6 w_4 + 80r_5^6 w_5 + [320 + 64h^6]r_6^6 w_6 = 15
\]
\[
E[x_1^4 x_j^2] = 64r_2^6 w_2 + 4r_3^6 w_3 + 64r_4^6 w_4 + 32r_5^6 w_5 + [256 + 64h^2 + 64h^4]r_6^6 w_6 = 3
\]
\[
E[x_i^2 x_j^2] = 64r_2^6 w_2 + 64r_4^6 w_4 + 8r_5^6 w_5 + [192 + 192h^2]r_6^6 w_6 = 1
\]
\[
E[x_1^8] = 2r_1^8 w_1 + 64r_2^8 w_2 + 20r_3^8 w_3 + 64r_4^8 w_4 + 80r_5^8 w_5 + [320 + 64h^8]r_6^8 w_6 = 105
\]
\[
E[x_i^6 x_j^2] = 64r_2^8 w_2 + 4r_3^8 w_3 + 64r_4^6 w_4 + 32r_5^8 w_5 + [256 + 64h^2 + 64h^6]r_6^8 w_6 = 15
\]
\[
E[x_1^4 x_j^4] = 64r_2^8 w_2 + 4r_3^8 w_3 + 64r_4^8 w_4 + 32r_5^8 w_5 + [256 + 128h^4]r_6^8 w_6 = 9
\]
\[
E[x_i^4 x_j^2] = 64r_2^8 w_2 + 64r_4^8 w_4 + 8r_5^8 w_5 + [192 + 128h^2 + 64h^4]r_6^8 w_6 = 3
\]
\[
E[x_i^2 x_j^2 x_k^2] = 64r_2^8 w_2 + 64r_4^8 w_4 + [128 + 256h^2]r_6^8 w_6 = 1
\]
\[
12w_1 + 64w_2 + 60w_3 + 64w_4 + 160w_5 + 384w_6 + w_0 = 1.
\]

These 11 moment constraint equations can be solved in a similar manner as mentioned in the previous sections. One can analytically solve for six unknown weights from the last six equations of (2.53). The substitution of these expressions of weight variables in the first 5 equations leads to a system of 5 equations with 7 variables $r_1, r_2, r_3, r_4, r_5, r_6$ and $h$. One can minimize the error in the $10^{th}$ moment constraint
equation and find a solution for all 7 variables. A numerical optimization procedure with high degree polynomial equations as constraints is often tedious and in some cases even intractable. A work around for this issue is to assume appropriate values for $h$ and any one of the distance variables $r_i$ that render the weights positive. Such assumptions make the system of polynomials equations consistent and hence solvable by highly efficient polynomial solvers. The set of sigma points as described in Table 2.11, when used to solve the 11 moment constraint equations, are found to work very well up to $n \leq 6$, after which some weights become negative.

It should be noticed that there are only 8 moment constraint equations in 2D and hence the sigma points corresponding to $r_5$ and $w_5$ are unnecessary; in effect, there are only 8 variables: $r_1, r_2, r_3, r_4$ and $w_1, w_2, w_3, w_4$. Similarly, there are 10 moment constraint equations for 3D system and hence the sigma points corresponding to $r_5$ and $w_5$ are dropped. Using Table 2.13, the CUT8 method would need only 355 and 745 function evaluations to compute the expectation integral for a polynomial function of degree 8 in the 5D and 6D space, respectively, whereas the GH product rule would need 3,125 and 15,625 function evaluations for the same 5D and 6D space respectively. It is clear that the number of function evaluations required by the CUT methods are significantly less than that required by the GH quadrature rule while having the same order of accuracy when integrating polynomials. Using Table 2.11 and the solutions of the MCE till 8th order in Table 2.13, one can generate the required sigma point set for dimensions $2 \leq n \leq 6$. Table 2.12 illustrates with highlighted cells the axes which are sampled for the CUT points and the number of CUT points as a function of the dimension of the random variables.
Table 2.12: Fully Symmetric set of points for CUT8

<table>
<thead>
<tr>
<th>Dimension</th>
<th>(\sigma)</th>
<th>(c^2)</th>
<th>(c^3)</th>
<th>(c^4)</th>
<th>(c^5)</th>
<th>(c^6)</th>
<th>(c^7)</th>
<th>(c^8)</th>
<th>(s^n)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>4</td>
<td>4</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>12</td>
<td>8</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>24</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>24</td>
<td>32</td>
<td>16</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>64</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>40</td>
<td>80</td>
<td>80</td>
<td>32</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>160</td>
</tr>
<tr>
<td>6</td>
<td>12</td>
<td>60</td>
<td>160</td>
<td>240</td>
<td>192</td>
<td>64</td>
<td>-</td>
<td>-</td>
<td>384</td>
</tr>
</tbody>
</table>

Table 2.13: Solutions for \(2 \leq n \leq 6\), 8\(^{th}\) moment constraint equations, CUT8

<table>
<thead>
<tr>
<th>Variable</th>
<th>2D</th>
<th>3D</th>
<th>4D</th>
<th>5D</th>
<th>6D</th>
</tr>
</thead>
<tbody>
<tr>
<td>(r_1)</td>
<td>2.0681360611</td>
<td>2.2551372655</td>
<td>2.2017090714</td>
<td>2.3143708172</td>
<td>2.4498497427</td>
</tr>
<tr>
<td>(r_2)</td>
<td>0.8491938499</td>
<td>0.7174531274</td>
<td>0.7941993714</td>
<td>0.8390942773</td>
<td>0.893824694122111</td>
</tr>
<tr>
<td>(r_3)</td>
<td>1.1386549808</td>
<td>1.8430194370</td>
<td>1.8725743605</td>
<td>1.8307521253</td>
<td>1.7320508075</td>
</tr>
<tr>
<td>(r_4)</td>
<td>1.8616199350</td>
<td>1.5584810327</td>
<td>1.3291164300</td>
<td>1.3970397430</td>
<td>1.531963037906212</td>
</tr>
<tr>
<td>(r_5)</td>
<td>-</td>
<td>-</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>(r_6)</td>
<td>-</td>
<td>-</td>
<td>1.3055615004</td>
<td>1.1258655812</td>
<td>1.1134786327</td>
</tr>
<tr>
<td>(w_1)</td>
<td>0.0438226426</td>
<td>0.0246319934</td>
<td>0.0181100873</td>
<td>0.0105290342</td>
<td>0.0061728395</td>
</tr>
<tr>
<td>(w_2)</td>
<td>0.1405096621</td>
<td>0.081510094</td>
<td>0.0320632733</td>
<td>0.0151440196</td>
<td>0.0069134430</td>
</tr>
<tr>
<td>(w_3)</td>
<td>0.0009215768</td>
<td>0.009767232555</td>
<td>0.006614353</td>
<td>0.0052829996</td>
<td>0.0041152263</td>
</tr>
<tr>
<td>(w_4)</td>
<td>0.0124095396</td>
<td>0.0057724893</td>
<td>0.0034899065</td>
<td>0.0010671298</td>
<td>0.0002183265</td>
</tr>
<tr>
<td>(w_5)</td>
<td>-</td>
<td>-</td>
<td>0.0006510416</td>
<td>0.0006510416</td>
<td>0.00065104166</td>
</tr>
<tr>
<td>(w_6)</td>
<td>-</td>
<td>-</td>
<td>0.0002794729</td>
<td>0.0002521833</td>
<td>0.00013776017</td>
</tr>
<tr>
<td>(h)</td>
<td>3</td>
<td>2.74</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>
2.3.4.4 Comparison to Other Cubature Methods

The number of points required by each of the methods Gauss-Hermite Product rule (GH), Smolyak Sparse grid Gauss Hermite quadrature (Sparse-GH) and the Conjugate Unscented Transform (CUT) are compared in Fig. 2.3. Fig. 2.3(a) compares the number of points required to exactly evaluate the expectation integrals of polynomials of total degree 5 or less. Similarly, Fig. 2.3(b) and Fig. 2.3(c) compare the number of points required to exactly evaluate the expectation integrals of polynomials of total degree equal or less than 7 and 9 respectively. Compared to the GH rule, the Smolyak scheme has considerably lower number of points. However, these type of sparse grid quadratures suffer from the presence of large negative weights [17]. It is known that cubature rules with $\frac{\sum |w_i|}{\sum w_i} >> 1$ have large round-off errors that could lead to instability [46]. Hence, one is always interested in cubature rules that have all positive weights. Unlike the Smolyak sparse grid Gauss Hermite Quadrature rules, the CUT points have all positive weights and can achieve higher accuracy with fewer points.

For the dimensions and order described in this paper, the CUT is compared to equivalent Gauss-Hermite Quadrature, Gauss Hermite Smolyak quadrature (GHS) and Kronrod Patterson (see [42]) Smolyak quadrature (KPS). Kronrod Patterson univariate quadrature is a nested quadrature rule that can be used in the Smolyak algorithm to generate multidimensional sparse grids. The number of points for the sparse grid quadrature generated by Gauss-Hermite and Kronrod Patterson is shown in [21]. The recently developed High order cubature Kalman filter (see [23]) for 5th (denoted as HCKF-5) and 7th (denoted as HCKF-7) degree are also shown.
Figure 2.3: comparison of number of points
Tables 2.14, 2.15 and 2.16 show the number of points $N$ used by each of the methods considered for specific dimensions. In addition, the stability factor for each method is also shown. The stability factor, as described in [46] and [51], is defined as 

$$\sum_{i=0}^{N} |w_i|,$$ 

i.e. the sum of absolute values of the weights. If some weights are negative, then the stability factor would be greater than 1, leading to a cubature rule or formula that is inefficient and which introduces large roundoff errors.

GHS has all negative weights for the dimensions and orders considered. KPS method has negative weights from the 5th dimension for 5th order accurate rules and negative weights from 3rd dimension for 7th order accurate rules. In case of the 5th order accurate rules of Table 2.14, CUT4 has fewer points than GHS. Up to dimension 5, CUT4 has fewer points than KPS and HCKF-5. Only GH and CUT have positive weights.

For the 7th order accurate rules of Table 2.15, CUT6 has fewer points than KPS up to the 6th dimension. CUT6 also has fewer points than HCKF-7, GHS and GH for the dimensions $2 \leq n \leq 9$.

For 9th order accurate rules of table 2.16, CUT8 has fewer number of points than GH, GHS and KPS. The stability factor $\sum |w|$ is always 1 for CUT6, since all the weights are positive. GHS has the largest stability factor.
**Table 2.14:** Number of Points for 5th order accurate cubature methods.

| Dim | GH $\sum |w_i|$ N | GHS $\sum |w_i|$ N | KPS $\sum |w_i|$ N | CUT4 $\sum |w_i|$ N | HCKF-5 $\sum |w_i|$ N |
|-----|----------------|----------------|-------------|----------------|----------------|
| 2   | 1 9            | 5 13           | 1 9         | 1 9            | 1 9            |
| 3   | 1 27           | 13 25          | 1 19        | 1 14           | 1 19           |
| 4   | 1 81           | 25 41          | 1 33        | 1 24           | 1 33           |
| 5   | 1 243          | 41 61          | 2.11 51     | 1 42           | 1.2041 51      |
| 6   | 1 729          | 61 85          | 3.67 73     | 1 76           | 1.37 73        |
| 7   | 1 2187         | 85 113         | 5.67 99     | 1 142          | 1.52 99        |
| 8   | 1 6561         | 113 145        | 8.11 129    | 1 272          | 1.64 129       |
| 9   | 1 19683        | 145 181        | 11 163      | 1 530          | 1.74 163       |
Table 2.15: Number of Points for $7^{th}$ order accurate cubature points.

<table>
<thead>
<tr>
<th>Dim</th>
<th>GH</th>
<th>GHS</th>
<th>KPS</th>
<th>CUT6</th>
<th>HCKF-7</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\sum</td>
<td>w_i</td>
<td>N$</td>
<td>$\sum</td>
<td>w_i</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>16</td>
<td>7</td>
<td>17</td>
<td>13</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>64</td>
<td>25</td>
<td>69</td>
<td>39</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>256</td>
<td>63</td>
<td>137</td>
<td>81</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1024</td>
<td>129</td>
<td>241</td>
<td>151</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>4096</td>
<td>231</td>
<td>389</td>
<td>257</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>16384</td>
<td>377</td>
<td>589</td>
<td>407</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>65536</td>
<td>575</td>
<td>849</td>
<td>609</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>262144</td>
<td>833</td>
<td>1177</td>
<td>871</td>
</tr>
</tbody>
</table>
**Table 2.16:** Number of Points for 9\textsuperscript{th} order accurate cubature points.

| Dim | GH $\Sigma |w_i|$ | N | GHS $\Sigma |w_i|$ | N | KPS $\Sigma |w_i|$ | N | CUT8 $\Sigma |w_i|$ | N |
|-----|-------------|---|-------------|---|-------------|---|-------------|---|
| 2   | 1           | 25 | 9           | 53 | 1.485       | 37 | 1           | 21 |
| 3   | 1           | 125 | 41         | 165 | 2.6360     | 93 | 1           | 59 |
| 4   | 1           | 625 | 129        | 385 | 3.825      | 201 | 1           | 161 |
| 5   | 1           | 3125 | 321       | 781 | 4.95       | 401 | 1           | 355 |
| 6   | 1           | 15625 | 681       | 1433 | 6.07       | 749 | 1           | 745 |
Chapter 3. Methods

The purpose of this chapter is to detail the analysis methods used in this paper.

A computer program first accepts user inputs, which govern the initial conditions, the type of simulation, and the duration of time which the simulation studies. The program uses these inputs to generate outputs in the form of possible final positions and velocities, from which the average or “most likely” final state can be calculated. The required user inputs are:

- time and date of initial measurements,
- time interval over which to model the orbit,
- measured initial position and initial velocity vectors,
- position and velocity standard deviations, variances, or covariances, and
- whether or not to consider the sun’s gravitational influence.

The general process of the computer program is described as follows.

1. The program uses the time and date of initial measurements to determine the initial position of the Moon, and, if applicable, the Sun.
2. The program generates sets of initial conditions according to the applicable statistical method. This statistical method is the Monte Carlo method, the Unscented Kalman Filter (UKF), or one of the Conjugate Unscented Transformations (CUT).

3. For each set of initial conditions, the program numerically integrates the applicable governing equations of motion to find the final position. At each time step in the integration process, the program calculates the position of the Moon, and, if applicable, the Sun. Additionally, since it is necessary for second-order integration, the velocity and acceleration are calculated at each time step.

4. Once all generated initial conditions have been integrated, and all final states have been found, the program calculates the average using the applicable statistical method.

This process is discussed in further detail in Sections 3.1-3.4.

3.1 Generating Sets of Initial Conditions

The statistical method used for a problem determines how the initial conditions are generated and how the final states are used to predict the most likely final position. In this paper, the Monte Carlo method, UKF, CUT4, CUT6, and CUT8 methods are considered as statistical methods.

In the Monte Carlo method, initial conditions are randomly generated from a Gaussian probability density function (PDF) using the given initial position and velocity and their standard deviations. The number of sets of initial conditions required
to sufficiently represent the PDF is determined by testing different numbers of initial conditions and seeing how many sets of initial conditions it takes for the fourth order moment, or kurtosis, to converge or become stable. Once all final states for a representative number of initial conditions has been calculated, the average final position is determined by taking a simple non-weighted average of all final states.

For the UKF and CUT methods, each set of initial conditions, or sigma point is generated non-randomly. The UKF’s sigma points are generated using Equations 2.8-2.11. The CUT4 sigma points are generated using Equations 2.31 and 2.32 with Table 2.5. The CUT6 sigma points are generated using Equations 2.41 and 2.42 with Table 2.8. The CUT8 sigma points are generated using Equation 2.53 and Table 2.13. All of these equations are shown in Section 2.3. In all methods, after each sigma point is propagated forwards in time to find its final state, the average final position is determined by taking a weighted average of the final states.

### 3.2 Equations for Propagating Orbits

Each set of initial conditions must be propagated forwards in time via the numerical integration of a second-order differential equation, as discussed in Chapter 2. The equations used to propagate the orbit in cislunar space depend on whether the CR3BP or BCR4BP model is being used. The derivations of the governing equations of motion used in this paper for the CR3BP and the BCR4BP are shown below.

Consider an orbital system consisting of \( N \) objects, one of which has negligible mass compared to the others. The object of negligible mass has mass \( m \) and position \( \vec{r} \). Each massive object exerts a gravitational force on the small object. The force \( \vec{F}_n \)
that a single massive object, with mass $m_n$ and position $\vec{r}_n$, exerts on the small object is represented below in Equation 3.1. This equation is Newton’s Law of Universal Gravitation [41].

$$\vec{F}_n = \frac{Gm_n m}{|\vec{r}_n - \vec{r}|^3} \left( \vec{r}_n - \vec{r} \right) = \frac{\mu_n m}{|\vec{r}_n - \vec{r}|^3} \left( \vec{r}_n - \vec{r} \right)$$  \hspace{1cm} (3.1)

The total force on the smaller object is the sum of the forces caused by each massive object, as shown in Equation 2.1, which is restated below. This equation can be rearranged to determine the acceleration of the small object, as shown in Equation 3.2.

$$\vec{F} = m \ddot{\vec{r}} = \sum_{n=2}^{N} \frac{\mu_n m}{|\vec{r}_n - \vec{r}|^3} \left( \vec{r}_n - \vec{r} \right)$$

$$\ddot{\vec{r}} = \sum_{n=2}^{N} \frac{\mu_n \left( \vec{r}_n - \vec{r} \right)}{|\vec{r}_n - \vec{r}|^3} \hspace{1cm} (3.2)$$

The formula shown in Equation 3.2 can be generalized to a system with any number of bodies, as shown in Sections 3.2.1-3.2.3. These sections discuss orbital systems that consist of 2, 3, and 4 bodies.

### 3.2.1 2-Body Problem

The 2-Body Problem (2BP) considers only the effects of the earth on the small body. This problem considers a fixed, non-rotating frame of reference centered on the earth, which is referred to in this paper as the geocentric frame. The differential equa-
tion of motion describing the 2BP, based on Equation 3.2, is shown below in Equation 3.3:

\[ \ddot{\vec{r}}_g = \sum_{n=2}^{2} \frac{\mu_n (\vec{r}_n - \vec{r})}{|\vec{r}_n - \vec{r}|^3} = \frac{\mu_E (r_{Eg} - \vec{r}_g)}{|r_{Eg} - \vec{r}_g|^3} \]  

(3.3)

where

- \( \vec{r}_g \) = position of small body in geocentric frame
- \( r_{Eg} \) = position of Earth in geocentric frame
- \( \mu_E \) = gravitational parameter of Earth.

In the geocentric frame, \( r_{Eg} = 0 \). Therefore, Equation 3.3 can be rewritten as Equation 3.4, which summarizes the motion of a small body in the 2BP.

\[ \ddot{\vec{r}}_g = -\frac{\mu_E \vec{r}_g}{|\vec{r}_g|^3} \]  

(3.4)

### 3.2.2 Circular Restricted 3-Body Problem

The Circular Restricted 3-Body Problem (CR3BP) considers the earth and moon, each of which is in a circular orbit around the center of mass of the earth-moon system. This problem first considers a fixed, non-rotating frame of reference centered on the center of mass of the earth-moon system, which is referred to in this paper as the barycentric frame for the CR3BP.

The CR3BP barycentric differential equation of motion is simply the sum of the accelerations caused by the earth and moon, as shown below in Equation 3.5:
\[
\vec{r}_b = \sum_{n=2}^{3} \frac{\mu_n (\vec{r}_n - \vec{r})}{|\vec{r}_n - \vec{r}|^3} = \frac{\mu_E (\vec{r}_{Eb} - \vec{r}_b)}{|\vec{r}_{Eb} - \vec{r}_b|^3} + \frac{\mu_M (\vec{r}_{Mb} - \vec{r}_b)}{|\vec{r}_{Mb} - \vec{r}_b|^3} 
\] (3.5)

where

- \( \vec{r}_b \) = position of small body in barycentric frame
- \( \vec{r}_{Eb} \) = position of Earth in barycentric frame
- \( \vec{r}_{Mb} \) = position of Moon in barycentric frame
- \( \mu_M \) = gravitational parameter of Moon.

The equations representing the locations of the Earth and Moon in the CR3BP barycentric frame of reference are shown below in Equations 3.6 and 3.7:

\[
\vec{r}_{Eb} = x_3 \cos \Omega t \hat{i} + x_3 \sin \Omega t \hat{j} 
\] (3.6)

\[
\vec{r}_{Mb} = x_4 \cos \Omega t \hat{i} + x_4 \sin \Omega t \hat{j} 
\] (3.7)

where

- \( x_3 \) = negative distance between Earth and EMB
- \( x_4 \) = positive distance between Moon and EMB
- \( \Omega \) = angular velocity of the Earth-Moon system
- \( t \) = time since most recent total lunar eclipse.
It is useful to represent the CR3BP equations of motion in a geocentric frame, which is centered on the earth, rather than a barycentric frame, since the Earth can provide a simpler frame of reference regarding measurements taken from its surface. The following generic transformation equations, Equations 3.8-3.10, can be used in this process.

\[
\begin{align*}
\vec{r}_g &= \vec{r}_b - \vec{r}_{Eb} \quad (3.8) \\
\dot{\vec{r}}_g &= \dot{\vec{r}}_b - \dot{\vec{r}}_{Eb} \quad (3.9) \\
\ddot{\vec{r}}_g &= \ddot{\vec{r}}_b - \ddot{\vec{r}}_{Eb} \quad (3.10)
\end{align*}
\]

Below are the derivations of the processes of transforming the coordinates of the massive bodies of the solar system from the barycentric to geocentric frame. The following derivations are based on Equation 3.8:

\[
\begin{align*}
\vec{r}_{Eg} &= \vec{r}_{Eb} - \vec{r}_{Eb} = 0 \quad (3.11) \\
r_{Mg} &= r_{Mb} - r_{Eb} = [x_4 \cos \Omega t \hat{i} + x_4 \sin \Omega t \hat{j}] - [x_3 \cos \Omega t \hat{i} + x_3 \sin \Omega t \hat{j}] \\
&= (x_4 - x_3) \cos \Omega t \hat{i} + (x_4 + x_3) \sin \Omega t \hat{j} \quad (3.12)
\end{align*}
\]

where

- \( \vec{r}_{Eg} \) = position of Earth in geocentric frame
- \( r_{Mg} \) = position of Moon in geocentric frame.

The expressions for the vectors from the massive bodies to the small body, in barycentric frame and geocentric frame, are shown below.
Due to the $\vec{r}_{Eb}$ term in Equation 3.10, it is also necessary to calculate the acceleration of the earth in the barycentric frame by taking derivatives of Equation 3.6. This calculation is shown below.

$$\vec{r}_{Eb} = x_3 \cos \Omega t \hat{i} + x_3 \sin \Omega t \hat{j}$$

$$\dot{\vec{r}}_{Eb} = -x_3 \Omega \sin \Omega t \hat{i} + x_3 \Omega \cos \Omega t \hat{j}$$

$$\ddot{\vec{r}}_{Eb} = -x_3 \Omega^2 \cos \Omega t \hat{i} - x_3 \Omega^2 \sin \Omega t \hat{j}$$ (3.15)

The general equation for the acceleration of a small body in the geocentric frame can therefore be derived by combining Equations 3.5 and 3.10, and is shown in Equation 3.16. Furthermore, the motion of the small body in a geocentric frame for the CR3BP can be summarized by Equations 3.16, 3.17, and 3.18, shown below. The constants used in these equations are shown in Table 3.1.

$$\ddot{\vec{r}}_g = -\frac{\mu_E \vec{r}_g}{|\vec{r}_g|^3} + \frac{\mu_M (\vec{r}_{Mg} - \vec{r}_g)}{|\vec{r}_{Mg} - \vec{r}_g|^3} - \ddot{\vec{r}}_{Eb}$$ (3.16)

$$\ddot{\vec{r}}_{Eb} = -x_3 \Omega^2 \cos \Omega t \hat{i} - x_3 \Omega^2 \sin \Omega t \hat{j}$$ (3.17)

$$\vec{r}_{Mg} = (x_4 - x_3) \cos \Omega t \hat{i} + (x_4 - x_3) \sin \Omega t \hat{j}$$ (3.18)
Table 3.1: Constants in CR3BP Model [7, 9]

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Omega$</td>
<td>Angular velocity of Earth-Moon system (s$^{-1}$)</td>
<td>$2.66170 \times 10^{-6}$</td>
</tr>
<tr>
<td>$x_3$</td>
<td>Distance between Earth and EMB (km)</td>
<td>$-4670.63$</td>
</tr>
<tr>
<td>$x_4$</td>
<td>Distance between Moon and EMB (km)</td>
<td>$379,729$</td>
</tr>
<tr>
<td>$\mu_E$</td>
<td>Earth gravitational parameter ($m^3s^{-1}$)</td>
<td>$3.98616 \times 10^5$</td>
</tr>
<tr>
<td>$\mu_M$</td>
<td>Moon gravitational parameter ($m^3s^{-1}$)</td>
<td>$4.90294 \times 10^3$</td>
</tr>
</tbody>
</table>

3.2.3 Bicircular Restricted 4-Body Problem

The BCR4BP considers the influence of the Earth, Moon, and Sun on a small object. It assumes that the center of mass of the Earth-Moon system is in a circular orbit around the Sun, and the Moon is in a circular orbit around the Earth. This model first considers a fixed, non-rotating frame of reference whose origin is on the center of mass of the sun-earth-moon system, which is referred to as the barycentric reference frame for the BCR4BP. The positions of the Sun, Earth, and Moon, as assumed in the BCR4BP, are illustrated in Figure 3.1.

The acceleration of the small body in the barycentric frame is the sum of the accelerations due to the Sun, Earth, and Moon, and is described below in Equation 3.19:
\[ \ddot{\vec{r}}_b = \sum_{n=2}^{4} \frac{\mu_n (\vec{r}_n - \vec{r})}{|\vec{r}_n - \vec{r}|^3} - \frac{\mu_E (\vec{r}_{E_b} - \vec{r}_b)}{|\vec{r}_{E_b} - \vec{r}_b|^3} - \frac{\mu_M (\vec{r}_M - \vec{r}_b)}{|\vec{r}_M - \vec{r}_b|^3} - \frac{\mu_S (\vec{r}_{S_b} - \vec{r}_b)}{|\vec{r}_{S_b} - \vec{r}_b|^3} \]  

(3.19)

where

- \( \mu_S = \) gravitational parameter of Sun

- \( \vec{r}_{S_b} = \) position of Sun in barycentric frame.

\(58\)
The equations representing the locations of the Sun, Earth, and Moon in the BCR4BP barycentric frame of reference are shown in Equations 3.20 - 3.23:

\[
\begin{align*}
\vec{r}_{SB} &= x_1 \cos \omega t \hat{i} + x_1 \sin \omega t \hat{j} \\
\vec{r}_{EMb} &= x_2 \cos \omega t \hat{i} + x_2 \sin \omega t \hat{j} \\
\vec{r}_{Eb} &= \vec{r}_{EM} + x_3 \cos \Omega t \hat{i} + x_3 \sin \Omega t \hat{j} \\
&= (x_2 \cos \omega t + x_3 \cos \Omega t) \hat{i} + (x_2 \sin \omega t + x_3 \sin \Omega t) \hat{j} \\
\vec{r}_{Mb} &= \vec{r}_{EM} + x_4 \cos \Omega t \hat{i} + x_4 \sin \Omega t \hat{j} \\
&= (x_2 \cos \omega t + x_4 \cos \Omega t) \hat{i} + (x_2 \sin \omega t + x_4 \sin \Omega t) \hat{j}
\end{align*}
\] (3.20 - 3.23)

where

- \( \vec{r}_{SB} \) = position of sun in BCR4BP barycentric frame
- \( \vec{r}_{EMb} \) = position of Earth-Moon barycenter (EMB) in BCR4BP barycentric frame
- \( \vec{r}_{Eb} \) = position of Earth in BCR4BP barycentric frame
- \( \vec{r}_{Mb} \) = position of Moon in BCR4BP barycentric frame
- \( x_1 \) = negative distance between the Sun and Sun-(Earth-Moon) barycenter (SEMB)
- \( x_2 \) = positive distance between EMB and SEMB
- \( x_3 \) = negative distance between Earth and EMB
- \( x_4 \) = positive distance between Moon and EMB
• \( \omega \) = angular velocity of Sun-(Earth-Moon) system
• \( \Omega \) = angular velocity of Earth-Moon system
• \( t \) = time since most recent total lunar eclipse.

Similar to the CR3BP case, it is useful to represent the BCR4BP equations of motion in a geocentric frame, which is centered on the Earth, rather than a barycentric frame, whose origin would be somewhere inside the Sun. The generic transformation equations, shown above in Equations 3.8-3.10, can be used in this process.

Below, in Equations 3.24, 3.25, and 3.26, are the derivations for the transformations of the massive bodies of the solar system from barycentric to geocentric frames. Note that the geocentric-frame locations of the Earth and Moon for the BCR4BP are the same as the geocentric-frame locations of the Earth and Moon for the CR3BP, which are shown above in Equations 3.11 and 3.12, respectively.

\[
\vec{r}_g = \vec{r}_b - \vec{r}_{Eb}
\]

\[
\vec{r}_{Sg} = \vec{r}_{Sb} - \vec{r}_{Eb} = [x_1 \cos \omega t \hat{i} + x_1 \sin \omega t \hat{j}]
- \left[(x_2 \cos \omega t + x_3 \cos \Omega t) \hat{i} + (x_2 \sin \omega t + x_3 \sin \Omega t) \hat{j}\right] = [(x_1 - x_2) \cos \omega t - x_3 \cos \Omega t] \hat{i} + [(x_1 - x_2) \sin \omega t - x_3 \sin \Omega t] \hat{j}
\]

\[
r_{Eg} = r_{Eb} - r_{Eb} = 0
\]

\[
r_{Mg} = r_{Mb} - r_{Eb} = [r_{EM} + x_4 \cos \Omega t \hat{i} + x_4 \sin \Omega t \hat{j}]
- \left[r_{EM} + x_3 \cos \Omega t \hat{i} + x_3 \sin \Omega t \hat{j}\right] = (x_4 - x_3) \cos \Omega t \hat{i} + (x_4 - x_3) \sin \Omega t \hat{j}
\]
The expressions for the vectors from the massive bodies to the small body, in barycentric frame and geocentric frame, are shown below.

\[
\vec{r}_{Sb} - \vec{r}_b = (\vec{r}_{Sg} + \vec{r}_{Eb}) - (\vec{r}_g + \vec{r}_{Eb}) = \vec{r}_{Sg} - \vec{r}_g
\]  
(3.27)

\[
\vec{r}_{Eg} - \vec{r}_b = (\vec{r}_{Eg} + \vec{r}_{Eb}) - (\vec{r}_g + \vec{r}_{Eb}) = \vec{r}_{Eg} - \vec{r}_g = -\vec{r}_g
\]  
(3.28)

\[
\vec{r}_{Mb} - \vec{r}_b = (\vec{r}_{Mg} + \vec{r}_{Eb}) - (\vec{r}_g + \vec{r}_{Eb}) = \vec{r}_{Mg} - \vec{r}_g
\]  
(3.29)

Similar to the CR3BP, due to the \(\vec{r}_{Eb}\) term in Equation 3.10, it is necessary to calculate the acceleration of the earth in the barycentric BCR4BP frame. This is done by taking the second derivative of Equation 3.22, as shown below.

\[
\vec{r}_{Eg} = (x_2 \cos \omega t + x_3 \cos \Omega t) \hat{i} + (x_2 \sin \omega t + x_3 \sin \Omega t) \hat{j}
\]

\[
\vec{r}_{Eg} = (-x_2 \omega \sin \omega t - x_3 \Omega \sin \Omega t) \hat{i} + (x_2 \omega \cos \omega t + x_3 \Omega \cos \Omega t) \hat{j}
\]

\[
\vec{r}_{Eg} = (-x_2 \omega^2 \cos \omega t - x_3 \Omega^2 \cos \Omega t) \hat{i} + (-x_2 \omega^2 \sin \omega t - x_3 \Omega^2 \sin \Omega t) \hat{j}
\]  
(3.30)

The general equation for the acceleration of the small body in the geocentric frame can therefore be derived as follows in Equation 3.31 by combining Equations 3.19 and 3.10. Furthermore, the motion of the small body in a geocentric frame for the BCR4BP can be summarized by Equations 3.31, 3.32, 3.33, and 3.34, shown below. The constants in these equations are shown in Table 3.2.

\[
\ddot{r}_g = \mu_S \frac{(\vec{r}_{Sg} - \vec{r}_g)}{|\vec{r}_{Sg} - \vec{r}_g|^3} + \mu_E \frac{(\vec{r}_g)}{|\vec{r}_g|^3} + \mu_M \frac{(\vec{r}_{Mg} - \vec{r}_g)}{|\vec{r}_{Mg} - \vec{r}_g|^3} - r_{Eg}^\omega
\]  
(3.31)
\[\vec{r}_{Eb} = (-x_2 \omega^2 \cos \omega t - x_3 \Omega^2 \cos \Omega t) \hat{i} + (-x_2 \omega^2 \sin \omega t - x_3 \Omega^2 \sin \Omega t) \hat{j} \quad (3.32)\]

\[\vec{r}_{Sg} = [(x_1 - x_2) \cos \omega t - x_3 \cos \Omega t] \hat{i} + [(x_1 - x_2) \sin \omega t - x_3 \sin \Omega t] \hat{j} \quad (3.33)\]

\[\vec{r}_{Mg} = (x_4 - x_3) \cos \Omega t \hat{i} + (x_4 - x_3) \sin \Omega t \hat{j} \quad (3.34)\]

**Table 3.2: Constants in BCR4BP Model [7, 10, 9]**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\omega)</td>
<td>Angular velocity of Sun-EMB system (s(^{-1}))</td>
<td>1.99099 \times 10^{-7}</td>
</tr>
<tr>
<td>(\Omega)</td>
<td>Angular velocity of Earth-Moon system (s(^{-1}))</td>
<td>2.66170 \times 10^{-6}</td>
</tr>
<tr>
<td>(x_1)</td>
<td>Distance between Sun and SEMB (km)</td>
<td>-453.906</td>
</tr>
<tr>
<td>(x_2)</td>
<td>Distance between EMB and SEMB (km)</td>
<td>1.50004 \times 10^{8}</td>
</tr>
<tr>
<td>(x_3)</td>
<td>Distance between Earth and EMB (km)</td>
<td>-4670.63</td>
</tr>
<tr>
<td>(x_4)</td>
<td>Distance between Moon and EMB (km)</td>
<td>379,729</td>
</tr>
<tr>
<td>(\mu_S)</td>
<td>Sun gravitational parameter (m(^3)s(^{-1}))</td>
<td>1.33353 \times 10^{11}</td>
</tr>
<tr>
<td>(\mu_E)</td>
<td>Earth gravitational parameter (m(^3)s(^{-1}))</td>
<td>3.98616 \times 10^{5}</td>
</tr>
<tr>
<td>(\mu_M)</td>
<td>Moon gravitational parameter (m(^3)s(^{-1}))</td>
<td>4.90294 \times 10^{3}</td>
</tr>
</tbody>
</table>

### 3.3 Determining Locations of the Sun and Moon

To calculate the position of the Sun and Moon at each time step, it is necessary to define a system to represent the bodies’ locations relative to one another. Boone (2021) defined a system in which time \(t=0\) whenever the Sun, Earth, and Moon are
in alignment and the Earth is between the Sun and the Moon, as occurs during a total lunar eclipse. This alignment is shown in Figure 3.2. This system is used in this paper. Therefore, the time $t$ in any equation describing the position of the Earth, Moon, or Sun indicates the time since the most recent total lunar eclipse. For simplicity, this system is utilized in both the CR3BP and the BCR4BP models, though it is only strictly necessary in the BCR4BP model.

![Figure 3.2: Position of Sun, Earth, and Moon at Time $t = 0$]

Julian date (JD) represents dates and times as a decimal system and can be calculated from coordinated universal time (UTC) [7]. In this paper, the initial time $t$ in days is calculated by subtracting the JD of the most recent eclipse from the JD of the initial measurement. This difference is then converted into days as needed.

The times and dates of lunar eclipses are generally reported in Terrestrial Dynamical (TD) time [37, 38]. Therefore, in order to calculate the Julian date of a lunar eclipse, the TD time must be converted to UT, which must then be converted to JD. In this paper, UT is assumed to be approximately equal to UTC, since the two times are al-
ways within 0.9 seconds of one another, which is an acceptable error when calculating
the positions of planetary bodies [36].

The equation to determine UT from TD is shown below in Equation 3.35 [36].

\[ UTC \approx UT = TD - \Delta T \quad (3.35) \]

\( \Delta T \), which is defined only as the difference between UT and TD, is based on
observed eclipse times between 500 BCE and 2005 CE [35]. For \( \Delta T \) calculations
outside of this time interval, Equation 3.36 is used to extrapolate \( \Delta T \) [35].

\[ \Delta T = -20 + 32 \left( \frac{\text{year} - 1820}{100} \right)^2 \text{ seconds} \quad (3.36) \]

Equations 3.35 and 3.36 can be combined to determine the UT corresponding
to any TD, as shown below in Equation 3.37.

\[ UTC \approx UT = TD - \left[ -20 + 32 \left( \frac{\text{year} - 1820}{100} \right)^2 \text{ seconds} \right] \quad (3.37) \]

The equation to calculate the Julian date from the UT is shown in Equation 3.38
[7]:

\[ JD = 367y - \text{INT} \left( \frac{7 \left[ y + \text{INT} \left( \frac{m+9}{12} \right) \right]}{4} \right) + \text{INT} \left( \frac{275m}{9} \right) + d + 1721013.5 + \frac{UT}{24} \quad (3.38) \]
where

- \( y \) = calendar year \((1901 \leq y \leq 2099)\)
- \( m \) = calendar month \((1 \leq m \leq 12)\)
- \( d \) = calendar day \((1 \leq d \leq 31)\)
- \( \text{INT}(a) \) rounds \( a \) towards 0.

MATLAB has a built-in function, \text{juliandate(t)}, that converts UTC to JD [28]. For simplicity, this function was implemented in the model for converting UTC to JD, rather than manually calculating JD using Equation 3.38.

The lunar eclipse schedule used in the model was obtained from NASA’s lunar eclipse schedule [37, 38], and is shown below in Table 3.3. Table 3.3 also includes the UT and JD of the lunar eclipses, calculated as described in Equations 3.37 and 3.38.

The model used in this paper determines the JD of the most recent lunar eclipse and considers it to be the time at which \( t = 0 \). It then determines the time in seconds between the time of measurement and the most recent lunar eclipse, and considers that time to be \( t_0 \). Therefore, the time variable \( t \) in all equations for this model can be determined as shown in Equations 3.39 and 3.40:

\[
\begin{align*}
    t &= t_0 + \Delta t \\
    t_0 &= \left[ 24 \frac{\text{hours}}{\text{day}} \times 3600 \frac{\text{seconds}}{\text{hour}} \times (JD_M - JD_{LE}) \right] \text{seconds}
\end{align*}
\]

where
• $t_0 =$ time in seconds between initial measurement and most recent lunar eclipse

• $\Delta t =$ time in seconds since initial measurement

• $JD_M =$ JD of initial measurement

• $JD_{LE} =$ JD of most recent lunar eclipse.

Additionally, certain orbital information is necessary to determine the orbits of the Sun and Moon. Some of these calculations are shown below, including the calculations for $x_1$, $x_2$, $x_3$, and $x_4$.

The calculations for the distances between the Earth and EMB and the Moon and EMB, represented by $x_3$ and $x_4$ respectively, are shown in Equations 3.41 and 3.42:

$$x_3 = - \frac{m_M r_{EM}}{m_E + m_M}$$  \hspace{1cm} (3.41)

$$x_4 = r_{EM} \left( 1 - \frac{m_M}{m_E + m_M} \right)$$  \hspace{1cm} (3.42)

where

• $x_3 =$ negative distance between Earth and EMB

• $x_4 =$ positive distance between Moon and EMB

• $m_E =$ mass of Earth

• $m_M =$ mass of Moon

• $r_{EM} =$ distance between Earth and Moon.
To calculate $x_1$ and $x_2$, it is useful to define the mass of the Earth-Moon system and the distance between the Sun and the EMB. These values are defined in Equations 3.43 and 3.44, respectively:

$$m_{EM} = m_E + m_M$$  \hspace{1cm} (3.43)

$$r_{SEM} = r_{SE} - x_3 = r_{SE} + \frac{m_M r_{EM}}{m_E + m_M}$$  \hspace{1cm} (3.44)

where

- $m_{EM}$ = total mass of Earth-Moon system
- $r_{SEM}$ = distance between Sun and EMB
- $r_{SE}$ = distance between Sun and Earth.

The calculations for the distance between the Sun and SEMB ($x_1$) and the EMB and SEMB ($x_2$), are derived in Equations 3.45 and 3.46:

$$x_1 = -\frac{m_E r_{SEM}}{m_S + m_{EM}} = -\frac{m_E}{m_S + m_E + m_M} \left( r_{SE} + \frac{m_M r_{EM}}{m_E + m_M} \right)$$  \hspace{1cm} (3.45)

$$x_2 = r_{SEM} \left( 1 - \frac{m_E}{m_S + m_{EM}} \right)$$

$$= \left( r_{SE} + \frac{m_M r_{EM}}{m_E + m_M} \right) \left( 1 - \frac{m_E}{m_S + m_E + m_M} \right)$$  \hspace{1cm} (3.46)

where

- $x_1$ = negative distance between Sun and SEMB
- $x_2$ = positive distance between EMB and SEMB
• $m_S =$ mass of Sun.

Therefore, the governing equations for the locations of the Sun and Moon in a geocentric frame can be summarized in Equations 3.33, 3.34, 3.45, 3.46, 3.41, 3.42, and 3.39. These equations are used in combination with the constants in Tables 3.2 and 3.4 to calculate the positions of the Sun and Moon in the geocentric frame. For reference, these equations are restated below:

\[
\vec{r}_{Sg} = \left[ (x_1 - x_2) \cos \omega t - x_3 \cos \Omega t \right] \hat{i} + \left[ (x_1 - x_2) \sin \omega t - x_3 \sin \Omega t \right] \hat{j} \quad (Eq. 3.33)
\]

\[
\vec{r}_{Mg} = (x_4 - x_3) \cos \Omega t \hat{i} + (x_4 - x_3) \sin \Omega t \hat{j} \quad (Eq. 3.34)
\]

\[
x_1 = -\frac{m_E}{m_S + m_E + m_M} \left( r_{SE} + \frac{m_{MRE}}{m_E + m_M} \right) \quad (Eq. 3.45)
\]

\[
(x_2 = \left( r_{SE} + \frac{m_{MRE}}{m_E + m_M} \right) \left( 1 - \frac{m_E}{m_S + m_E + m_M} \right) \quad (Eq. 3.46)
\]

\[
x_3 = -\frac{m_{MRE}}{m_E + m_M} \quad (Eq. 3.41)
\]

\[
x_4 = r_{EM} \left( 1 - \frac{m_M}{m_E + m_M} \right) \quad (Eq. 3.42)
\]

\[
t = \left[ 24 \times 3600 \times (JD_M - JD_{LE}) \right] + \Delta t \quad (Eq. 3.39)
\]

where

• $\vec{r}_{Sg} =$ Location of Sun in geocentric frame

• $\vec{r}_{Mg} =$ Location of Moon in geocentric frame
• \( x_1 \) = negative distance between the sun and the center of mass of (Earth-Moon)-Sun system

• \( x_2 \) = positive distance between the center of mass of the earth-moon system and the center of mass of the (Earth-Moon)-Sun system

• \( x_3 \) = negative distance between earth and center of mass of earth-moon system

• \( x_4 \) = positive distance between moon and center of mass of the Earth-Moon system

• \( \Omega \) = angular velocity of the Earth-Moon system

• \( \omega \) = angular velocity of the Sun-Earth system.
<table>
<thead>
<tr>
<th>TD Date</th>
<th>∆T</th>
<th>UT Date</th>
<th>Julian Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>2021-05-26</td>
<td>96.7392</td>
<td>6/15/2011 20:12</td>
<td>2455728.342</td>
</tr>
<tr>
<td>2022-05-16</td>
<td>96.7392</td>
<td>12/10/2011 14:31</td>
<td>2455906.105</td>
</tr>
<tr>
<td>2022-11-08</td>
<td>100.4352</td>
<td>4/15/2014 7:45</td>
<td>2456762.823</td>
</tr>
<tr>
<td>2025-03-14</td>
<td>100.4352</td>
<td>10/8/2014 10:54</td>
<td>2456938.954</td>
</tr>
<tr>
<td>2025-09-07</td>
<td>101.68</td>
<td>4/4/2015 11:59</td>
<td>2457117</td>
</tr>
<tr>
<td>2026-03-03</td>
<td>101.68</td>
<td>9/28/2015 2:46</td>
<td>2457293.616</td>
</tr>
<tr>
<td>2028-12-31</td>
<td>105.4528</td>
<td>1/31/2018 13:29</td>
<td>2458150.062</td>
</tr>
<tr>
<td>2029-12-20</td>
<td>106.7232</td>
<td>1/21/2019 5:11</td>
<td>2458504.716</td>
</tr>
<tr>
<td>2032-04-25</td>
<td>109.2832</td>
<td>5/26/2021 11:18</td>
<td>2459360.971</td>
</tr>
<tr>
<td>2032-10-18</td>
<td>110.5728</td>
<td>5/16/2022 4:10</td>
<td>2459715.674</td>
</tr>
<tr>
<td>2033-04-14</td>
<td>110.5728</td>
<td>11/8/2022 10:58</td>
<td>2459891.957</td>
</tr>
<tr>
<td>2033-10-08</td>
<td>114.48</td>
<td>3/14/2025 6:58</td>
<td>2460748.79</td>
</tr>
<tr>
<td>2036-02-11</td>
<td>114.48</td>
<td>9/7/2025 18:11</td>
<td>2460926.258</td>
</tr>
<tr>
<td>2036-08-07</td>
<td>115.7952</td>
<td>3/3/2026 11:32</td>
<td>2461102.981</td>
</tr>
<tr>
<td>2037-01-31</td>
<td>118.4448</td>
<td>12/31/2028 16:51</td>
<td>2462137.202</td>
</tr>
<tr>
<td>2040-05-26</td>
<td>119.7792</td>
<td>6/26/2029 3:21</td>
<td>2462313.64</td>
</tr>
</tbody>
</table>
Table 3.4: Constants in Model of Sun, Earth, and Moon Equations [7, 10, 9]

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m_S$</td>
<td>Sun’s mass (kg)</td>
<td>$1.998 \times 10^{30}$</td>
</tr>
<tr>
<td>$m_E$</td>
<td>Earth’s mass ($m^3s^{-1}$)</td>
<td>$5.9724 \times 10^{24}$</td>
</tr>
<tr>
<td>$m_M$</td>
<td>Moon’s mass ($m^3s^{-1}$)</td>
<td>$0.07346 \times 10^{24}$</td>
</tr>
<tr>
<td>$r_{SE}$</td>
<td>Distance between Sun and Earth (km)</td>
<td>$150 \times 10^6$</td>
</tr>
<tr>
<td>$r_{EM}$</td>
<td>Distance between Sun and Moon (km)</td>
<td>384400</td>
</tr>
<tr>
<td>$1/\omega$</td>
<td>Orbital Period of Earth around Sun (days)</td>
<td>365.256363</td>
</tr>
<tr>
<td>$1/\Omega$</td>
<td>Orbital Period of Moon around Earth (days)</td>
<td>27.3217</td>
</tr>
</tbody>
</table>
3.4 Numerical Integration in MATLAB

Matrix Laboratory (MATLAB) has several built-in solvers for differential equations that could be used with varying accuracy to integrate the governing equations of motion. Boone (2021) used MATLAB’s ODE45, a medium-accuracy nonstiff solver. This paper, however, utilizes the ODE89 solver, a nonstiff solver which was released in MATLAB R2021b and is currently MATLAB’s highest fidelity solver [29]. This solver was chosen after comparing 2BP results generated with ODE45, ODE23, ODE78, and ODE89 to analytically determined values.
Chapter 4. Simulations

The purpose of this chapter is to discuss the results of each model being applied to example initial conditions. It begins with a discussion of the chosen initial conditions and their corresponding approximate trajectories. It then discusses the validity of the Monte Carlo method to model these trajectories. It discusses the non-Gaussian nature of the final location probability distribution, and compares the tested methods of UKF, CUT4, CUT6, and CUT8 to the baseline MC values.

In this paper, initial conditions were selected to produce different types of trajectories. The main types of orbits of concern are halo orbits, which orbit the moon near a Lagrange point, and cycler orbits, which envelop both the earth and the moon. A spatial orbit was selected for both types of orbits, resulting in two sets of initial conditions. For each set of initial conditions, there are two computed trajectories. One of these trajectories represents the CR3BP, while the other represents the BCR4BP.

The details for each example are described in Table 4.1. These details include the initial geocentric position ($\vec{x}_G^0$), initial geocentric velocity ($\vec{v}_G^0$), and the standard deviations for both values ($\sigma$).
4.1 Example 1: Spatial Cycler Orbit

The trajectories computed in Example 1 are both spatial cycler orbits, though neither are exactly periodic. The approximate trajectories are shown below in Figures 4.1 and 4.2. Figure 4.1 shows the Ex.1a trajectory, which was generated by the CR3BP, while Figure 4.1 shows the Ex.1b trajectory, which was generated by the BCR4BP. Both trajectories were generated by propagating the average initial conditions only, and therefore do not account for uncertainty in initial conditions. In these figures, there does not appear to be much difference between the CR3BP and the BCR4BP trajectories, as differences are mostly visible on longer time scales.
Monte Carlo (MC) results calculated with a set size of 2,000,000 are used evaluate the errors of the cubature methods for both the CR3BP and BCR4BP models of this cycler orbit. To confirm that 2,000,000 MC runs provide sufficient accuracy for this purpose, the validity of the Monte Carlo results is assessed by calculating the mean, covariance, and kurtosis of the final positions for varying numbers of generated points. These moments are then compared to determine if sufficient numbers of points have been generated for model validity. A chart showing the mean, covariance, and kurtosis
of the MC runs for the cycler CR3BP trajectory is shown in Figure 4.3. A similar chart for the cycler BCR4BP trajectory is shown in Figure 4.4.

As evident in Figures 4.3 and 4.4, the moments for both example cycler orbits generally converge to within 1% of their final values with 2,000,000 points. Therefore, 2,000,000 Monte Carlo points should be enough to obtain accurate results for the examined cycler CR3BP and the cycler BCR4BP trajectories.

Because the orbit models are nonlinear, propagating a normally distributed set of initial conditions forwards in time does not necessarily result in a normally distributed final state. This concept is illustrated in the final position distributions from Example 1, shown in Figures 4.5 and 4.6. These figures show the MC final position distributions, overlaid by the final positions calculated from the sigma points of UKF,
CUT4, CUT6, and CUT8. For the final states calculated from sigma points, the size of a point on the plot corresponds to that point’s weight.

Figures 4.5 and 4.6 illustrate how cubature methods, especially higher-order cubature methods, can effectively represent the final probability distributions at a fraction of the computational expense. The final positions generated by the cubature methods hold closely to those generated by the MC method, with the heaviest-weighted cubature points typically being near the densest part of the final MC positions. Higher order cubature methods, such as CUT6 and CUT8, capture more details of the final probability distribution, such as the length of the curve of possible final positions in both models.

**Figure 4.4:** Moments of MC Points from Cycler BCR4BP (Ex.1b)
Figure 4.5: Final Positions after 15 Days for Cycler CR3BP

Figure 4.6: Final Positions after 15 Days for Cycler BCR4BP
Table 4.2: Absolute Position Errors of Cubature Methods for Cycler Orbit

<table>
<thead>
<tr>
<th>Cubature Method</th>
<th>CR3BP Errors (km)</th>
<th>BCR4BP Errors (km)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Day 5</td>
<td>Day 10</td>
</tr>
<tr>
<td>UKF</td>
<td>625.4</td>
<td>65.49</td>
</tr>
<tr>
<td>CUT4</td>
<td>428.1</td>
<td>73.87</td>
</tr>
<tr>
<td>CUT6</td>
<td>298.0</td>
<td>123.5</td>
</tr>
<tr>
<td>CUT8</td>
<td>229.7</td>
<td>27.64</td>
</tr>
</tbody>
</table>

The errors in position at 15 days are listed in Table 4.2. These errors are calculated by comparing the most likely final positions according to each cubature method with the average Monte Carlo final position. Note that the error at any particular point in this particular situation does not seem to correspond much with the order of the cubature method. However, the errors over time, shown in Figure 4.7, show that, over the entire examined time interval, the higher-order cubature method CUT8 tends to be more accurate than the other cubature methods examined.

Figure 4.7: Absolute Position Error vs Time for Cycler Orbit
Often, the error of cubature methods does not increase continuously with propagation time. Instead, the position error often increases near a massive body, since the object is moving at a greater speed. This concept is shown in Figures 4.8 and 4.9, which show the MC and estimated trajectories for the CR3BP and BCR4BP, respectively. As seen in Figure 4.8, the errors associated with the CR3BP trajectory spiked as the body neared a massive body, since this is when the body is moving the fastest, and near the end of the simulation, where the cubature methods begin to diverge.
4.2 Example 2: Spatial Halo Orbit

The expected trajectories resulting from the Ex.2 spatial halo orbit are shown below in Figures 4.10 and 4.11. Figure 4.10 shows the Ex.2a trajectory, which was generated by the CR3BP, while Figure 4.11 shows the Ex.2b trajectory, which was generated by the BCR4BP. Like the cycler orbit, both trajectories were generated by propagating the average initial conditions only, and therefore do not account for uncertainty in initial conditions. Once again, the CR3BP and BCR4BP trajectories seem similar, since large differences between the two models are most observable on longer time scales.

Monte Carlo (MC) results calculated with a set size of 4,000,000 are used evaluate the errors of the cubature methods for the halo orbit CR3BP trajectory. For the halo orbit BCR4BP trajectory, a set size of 2,500,000 was used. To confirm that these set sizes provide sufficient accuracy, the mean, covariance, and kurtosis of the final positions is calculated for varying numbers of generated points. These moments are
then compared to determine if sufficient numbers of points have been generated for model validity. A chart showing the mean, covariance, and kurtosis of the MC runs for the halo CR3BP trajectory is shown in Figure 4.12. A similar chart for the BCR4BP trajectory is shown in Figure 4.13.

As evident in Figure 4.12, the moments for the halo CR3BP trajectory generally converge to within 2.5% of their final value with 4,000,000 generated MC points. Similarly, for the halo BCR4BP trajectory, the moments converge to within 1% of their final values with a set size of approximately 2,500,000 generated points. Therefore,
Figure 4.12: Moments of MC Points from Halo CR3BP (Ex.2a)

Figure 4.13: Moments of MC Points from Halo BCR4BP (Ex.2b)
cubature method results for the halo CR3BP and BCR4BP trajectories are compared to MC simulation results of sizes 4,000,000 and 2,500,000, respectively.

The possible final positions for the spatial halo orbit are also not normally distributed, as shown in Figures, 4.14, and 4.15. These figures show the MC final position distributions, overlaid by the final positions calculated from the sigma points of UKF, CUT4, CUT6, and CUT8. For the final states calculated from sigma points, the size of a point on the plot corresponds to that point’s weight.

![Figure 4.14: Final Positions after 15 Days for Halo CR3BP (Ex.2a)](image_url)

As expected for the final states of a nonlinear system, the distributions shown in Figures 4.14 and 4.15 are fairly complex and do not resemble a normal distribution. Like the final positions for the cycler orbit, shown in Figures 4.5 and 4.6, the cubature methods decently represent the general shape of the final distribution, with higher-order
cubature methods representing finer details. In both models, the CUT methods were able to indicate some details not described by the UKF. Note that, in Figure 4.5, higher-order cubature methods better represent a wider arc of potential positions, rather than just the ones in the center. Also note that, in Figure 4.6, the curve of potential positions for the halo orbit fits more closely to the MC results for CUT8 than it does for any other method.

The errors in position at selected times throughout the simulation are listed in Table 4.3. Like Example 1, these errors are calculated by comparing the most likely final positions according to each cubature method with the average Monte Carlo final position. Note that, in this example, the errors of higher-order cubature methods tend to be lower than the errors of lower-order cubature methods. This relationship is also
### Table 4.3: Absolute Position Errors of Cubature Methods for Halo Orbit

<table>
<thead>
<tr>
<th>Cubature Method</th>
<th>CR3BP Errors (km)</th>
<th>BCR4BP Errors (km)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Day 5</td>
<td>Day 10</td>
</tr>
<tr>
<td>UKF</td>
<td>39.71</td>
<td>42.25</td>
</tr>
<tr>
<td>CUT4</td>
<td>32.07</td>
<td>27.50</td>
</tr>
<tr>
<td>CUT6</td>
<td>6.246</td>
<td>19.49</td>
</tr>
<tr>
<td>CUT8</td>
<td>20.56</td>
<td>13.88</td>
</tr>
</tbody>
</table>

![Figure 4.16: Absolute Position Error vs Time for Halo Orbit](image)

visible in Figure 4.16, which shows the errors of each cubature method throughout the duration of the simulation.

As with the the cycler orbit case, error does not necessarily increase constantly over time. Figures 4.17 and 4.18 compare the MC-generated and the cubature-generated trajectories for the halo orbits calculated with CR3BP and BCR4BP, respectively. Note that, for all cubature methods, the spikes in error correspond to the times at which the small body was closest to the moon. These spikes in error are likely because
the body is moving faster at these points, since its gravitational potential energy is being exchanged for kinetic energy. This increase in velocity would make its position more difficult to model with numerical integration methods.

Figure 4.17: Modeled Trajectories for 15-Day Halo CR3BP

Figure 4.18: Modeled Trajectories for 15-Day for Halo BCR4BP
4.3 Comparing Cubature and Monte Carlo Methods

As evident in Figures 4.17 and 4.18, the cubature methods all produce fairly accurate results relative to the large distances they are covering. However, there are some differences in accuracy between the cubature methods, as shown in Figure 4.19.

The cubature methods are compared to one another by calculating the magnitude of the error of the positions generated by the cubature methods throughout the simulated period of time. Because the Monte Carlo method is confirmed to be the most accurate, the error shown in Figure 4.19 is determined by calculating the distance from the position generated by the cubature method to the position generated by the Monte Carlo method.

Figure 4.19: Magnitude of Absolute Position Error vs Time
As is evident in Figure 4.19, higher-order CUT methods like CUT8 generally produce more accurate results for the conditions examined. The results considered to be the most accurate are the Monte Carlo results, which are considered to be the baseline for this study. However, computer programs using the Monte Carlo method take far longer to run. The computational efficiencies of these methods are compared in Table 4.4, which describes example run times for the conditions examined.

<table>
<thead>
<tr>
<th>Cubature Method</th>
<th>Cycler CR3BP</th>
<th>Cycler BCR4BP</th>
<th>Halo CR3BP</th>
<th>Halo BCR4BP</th>
</tr>
</thead>
<tbody>
<tr>
<td>10^6 MC Points Run Time (s)</td>
<td>4200</td>
<td>5300</td>
<td>6300</td>
<td>7500</td>
</tr>
<tr>
<td>UKF Run Time (s)</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>CUT4 Run Time (s)</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>CUT6 Run Time (s)</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>CUT8 Run Time (s)</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>19</td>
</tr>
</tbody>
</table>

As shown in Table 4.4, the Monte Carlo method, while somewhat more accurate than higher-order cubature methods, is far more computationally expensive and, for some use cases, may have prohibitively long run times. The MC methods all took at least an hour to run, while the longest cubature methods took less than 20 seconds. CUT8, the cubature method with the longest run time, is at least 300 times faster than the Monte Carlo method in the situations examined.
Chapter 5. Conclusions

This paper aims to examine the applications of cubature methods to initial orbit determination in cislunar space, or the space in which both the Earth’s and the Moon’s gravity is relevant. Initial orbit determination can allow one to predict where an object will be in a given amount of time with given initial conditions. However, since motion is chaotic in cislunar space, small changes in initial conditions can result in large changes in final states. Therefore, it is necessary to account for possible uncertainties in initial measurements, since small uncertainties in initial measurements can cause large uncertainties in the final states.

The two orbital models used in this paper to model cislunar trajectories are the Circular Restricted 3-Body Problem (CR3BP), which accounts for the gravitational influences of the Earth and Moon, and the Bicircular Restricted 4-Body Problem (BCR4BP), which accounts for the gravitational influences of the Earth, Moon, and Sun. Both are nonlinear dynamical systems with no analytical solution.

Since orbit determination in cislunar space is based on a non-linear dynamical system, there is no established analytical method to propagate these uncertainties forward in time. Therefore, numerical methods like the Monte Carlo method and cubature methods are used to determine uncertainty in final states. This paper examines
the Monte Carlo Method, as well as the Unscented Kalman Filter and Conjugate Unscen
ted Transforms of the 4th, 6th, and 8th orders.

This paper found that, for the conditions examined, cubature methods can be used to account for uncertainties in initial conditions when propagating orbits in cis-
lunar space. Higher-order cubature methods that generate more points, such as the 8th order Conjugate Unscented Transform, performed particularly well, with relatively small errors. For instance, when modeling a halo orbit with the BCR4BP for 15 days, the UKF results had an error of 76 km, while the CUT8 method had an error of 24 km. This trend was present in all examined conditions. Therefore, for the conditions examined, higher-order CUT methods could be used instead of lower-order cubature methods to model cis-lunar orbits with higher accuracy and similar computational expense.

All cubature methods examined had some errors when compared to the Monte Carlo results; however, using the Monte Carlo method in this case can be prohibitively computationally expensive. For the cases examined, run times for the Monte Carlo method were at least 300 times longer than those for cubature methods. Therefore, for some use cases of cis-lunar orbit propagation, higher-order cubature methods could provide an efficient, viable alternative to the computationally expensive Monte Carlo method.

Although this paper establishes that higher-order cubature methods are effective in modeling certain cis-lunar orbits under certain conditions, future work would still be required. More simulations representing a wider variety of initial conditions should be performed to verify that these cubature methods are able to model a wider variety
of orbits and trajectories. Furthermore, this paper only accounts for uncertainties in initial conditions; future models could also account for small perturbations and uncertainties that can occur mid-orbit, such as solar winds, solar pressure, the oblateness of most massive bodies, and small gravitational influences from other massive bodies in the solar system. For applications to spacecraft, it would also be necessary to incorporate a propulsion system into the model, as well as the uncertainties inherent in any propulsion system. While this paper shows that cubature methods can be successfully applied in some cases, further studies to verify and append this model could increase its accuracy and broaden its applicability.
References


