Spring 2024

Application of the Fokker-Planck Equation for Quantifying Initial Condition Uncertainty of Reversible Dynamic Systems

Troy S. Newhart
Old Dominion University, tnewh001@odu.edu

APPLICATION OF THE FOKKER-PLANCK EQUATION FOR QUANTIFYING INITIAL CONDITION UNCERTAINTY OF REVERSIBLE DYNAMIC SYSTEMS

by

Troy S. Newhart
B.S. May 2015, The Pennsylvania State University
M.S. December 2019, Old Dominion University

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

AEROSPACE ENGINEERING

OLD DOMINION UNIVERSITY
May 2024

Approved by:

Gene Hou (Director)

Brett Newman (Member)

Drew Landman (Member)

Hong Yang (Member)

ABSTRACT

APPLICATION OF THE FOKKER-PLANCK EQUATION FOR QUANTIFYING INITIAL CONDITION UNCERTAINTY OF REVERSIBLE DYNAMIC SYSTEMS

Troy Newhart
Old Dominion University, 2023
Director: Dr. Gene Hou

Characterizing the behavior of dynamic systems requires the inclusion of initial conditions to propagate behavior forward in time. More realistic representations of system behavior quantify uncertainty about the initial conditions to assess sensitivity, reliability, and other stochastic response parameters. In many engineering applications, the uncertain initial conditions may be unknown given a desired response. This research applies the Fokker-Planck equation to reversible dynamic systems of select multi-dimensional nonlinear differential equations as a means for predicting the uncertainty about initial conditions. An alternating directions implicit numerical scheme is used to numerically solve the Fokker-Planck equation for both forward and reversed equations of motion. Initial conditions are predicted for a linear oscillating system, nonlinear trim solution, and atmospheric reentry equations. A use case is presented where the initial atmospheric entry conditions are predicted for a Mars reentry vehicle given a landing zone and parachute deployment response conditions. Monte Carlo simulations are also implemented to verify the outputs of the numerical scheme. Additional verification is conducted by comparing forward and reverse transient results of each problem set. It is shown that the initial conditions can be adequately predicted using the presented methodology. Computational resources quickly become a limitation as additional dimensions of variability are added to the problem.

This dissertation is dedicated to my friends and family. Their love and encouragement have motivated me to see this through. To my wife, who has endured much with me over this time but continues to inspire with her fortitude, work ethic, and compassion. To my “xbox” friends, who provided many much need laughs over the years.

ACKNOWLEDGMENTS

I would like to thank Dr. Gene Hou for his encouragement and enthusiasm throughout my time with Old Dominion University. I would also like to thank Dr. Brett Newman, Dr. Drew Landman, and Dr. Hong Yang for their dedication of time to review and provide feedback on this research. These gentlemen have shown tremendous commitment to their students through teaching and mentoring. They continue to inspire and influence our future engineers toward great things. Thank you!

NOMENCLATURE

A_i	Drift Vector
B_{ij}	Diffusion Matrix
B_t	Wiener Process
c	Initial Conditions
C_D	Drag Coefficient
d	Great Circle Distance
D	Drag Force
$E[\ ]$	Expectation
f	Function
g	Gravitational Constant
G	System Noise Mapping
h	Altitude
J	Jacobian
m	Mass
M	Number of dimensions
n	Number of time steps
N	Number of nodes
p	Probability Density
pr	Pressure
R	Sperical Radius
S_{ref}	Reference Area

t	Time
T	Temperature
x	State Variable
X_t	Stochastic Process
$\xi\mu$	Random Excitation
μ	Drift coefficient
μ_c	Trim coefficient
σ	Diffusion coefficient
τ	New Time
ρ	Density
ω	Angular Velocity
ϕ	Latitude
λ	Longitude
1,2,…,M	Specific state
0	Initial condition
i,j	Indexing points

TABLE OF CONTENTS

Page

Chapter Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

CHAPTER 1

INTRODUCTION

In this chapter, the reader is introduced to concepts of uncertainty and the relationship uncertainty has with dynamic processes. Of primary note for this research is a reentry problem of a space vehicle within the atmosphere of Mars. The primary objective of this research is then defined and expanded upon. The remaining scope if this dissertation is then discussed.

1.1 BACKGROUND

Uncertainty is an inherent aspect of all facets in engineering, ranging from epistemic (uncertainty in the knowledge of a system or phenomenon) to aleatoric (randomness encountered in nature) [1]. The propagation of uncertainty, or uncertainty quantification (UQ), has been a topic of interest to the scientific community for decades, gaining much of its traction with the advent of computers [2]. Derivation of methods combining classical deterministic engineering approaches with stochastic elements of uncertainty have been a natural means to advancing the field of UQ and provide not only the ability to assess the impact of uncertainty on system properties but also system reliability and parameter sensitivities. UQ can be further categorized into the forward and inverse problems [3] where the forward problem aims to propagate system uncertainty through a model. Forward problems are the most prevalent in literature and span a wide array of problem sets to determine parameter sensitivity, response statistics, failure likelihood, etc. Inverse UQ attempts to determine input randomness given response data. Model calibration using Bayesian approaches is a common example of inverse UQ. Many methods have been developed and modified throughout the years to solve a

multitude of forward and inverse UQ problems. As the digital age progresses, modeling and simulation (M&S) continues to advance and become an essential component of projects throughout many disciplines of study. Benefits of implementing M&S are well documented, demonstrating cost and time savings as well as a better understanding of modeled systems [4]. Although computational M&S has advanced significantly in the past decade, demand for faster M&S at higher fidelity introduces complexities that require innovative techniques [5, 6]. Of the many complexities that must be overcome to further M&S capabilities, uncertainty remains a central topic.

Traditional approaches in engineering problems simplify quantification by handling uncertainties as deterministic values [7]. This approach is common in most undergraduate curriculum as the reinforcement of fundamentals is of primary concern. This can be done using mean values from test data (Modulus of Elasticity, specific impulse of a batch of solid rocket fuel, ducting conditions off the coast of California at a specific time of the year, etc.) or by taking some worst-case measurement (highest flood condition, lowest shear strength to failure, etc.) and applying a factor of safety. In general, the deterministic process has proven insightful, efficient, and useful toward engineering design and analysis. The major downside occurs in the resulting outputs and the inability to assess how input variability propagates through a system or process. This downside continues to become more protuberant as system design complexity increases, especially amongst system-of-systems structures and simulations. Adequately quantifying and propagating this uncertainty through a model elevates the fidelity at which we may assess the system; revealing information that would not have been found otherwise. Of particular interest for this research is the UQ applied to dynamic systems.

Literature is abundant in presenting dynamic problems characterized by differential equations that describe the motion of its specific system given initial conditions, boundary values, model coefficients, etc. The ability to determine responses of systems with uncertain or random parameters is the natural extension of these deterministic approaches and yields a transient stochastic system, or stochastic process. A stochastic process, much like its deterministic counterpart, may be represented by a differential equation but requires new or modified solution methods due to the added stochastic aspects [8]. Examples demonstrate the novelty of assessing the transient nature of not only a stochastic process marching forward in time but also (given a desired response) marching backward in time, allowing for a determination of initial conditions and the uncertainty about them. The novelty here being a direct application of the Fokker-Planck equation to forward and reverse reentry dynamics. Although multiple examples are explored in this research, the primary example explored relates to uncertainty within the initial conditions of a space reentry vehicle.

Reentry vehicles entering an atmosphere follow a set of equations of motion that characterize the flight path to an impact point on a planet's surface. Forward UQ applied to this problem allows for the assessment of potential impact regions. The benefits of which are obvious when looking to avoid regions that may have drastic elevation changes, populated areas, environmental concerns, etc. Although literature is present on the topic of the forward problem applied to this scenario, it is sparse. Literature pertaining to the inverse problem in this context is even more uncommon. Enabling engineers to solve the inverse stochastics of a dynamic system provides the ability to assess initial conditions given system responses (or desired responses). Finding ideal input conditions and uncertainty bounds may influence design

or employment strategies. In the case of assessing the landing of a vehicle on the surface of Mars, eight potential landing points were identified by NASA scientists [9] depicted in Figure 1. If an area in the Holden crater is chosen, a joint normal distribution can be defined about the Martian latitude and longitude as an optimal landing location as seen in Figure 2 where the "hotter" regions depict a desired higher probability of that area being the landing zone.

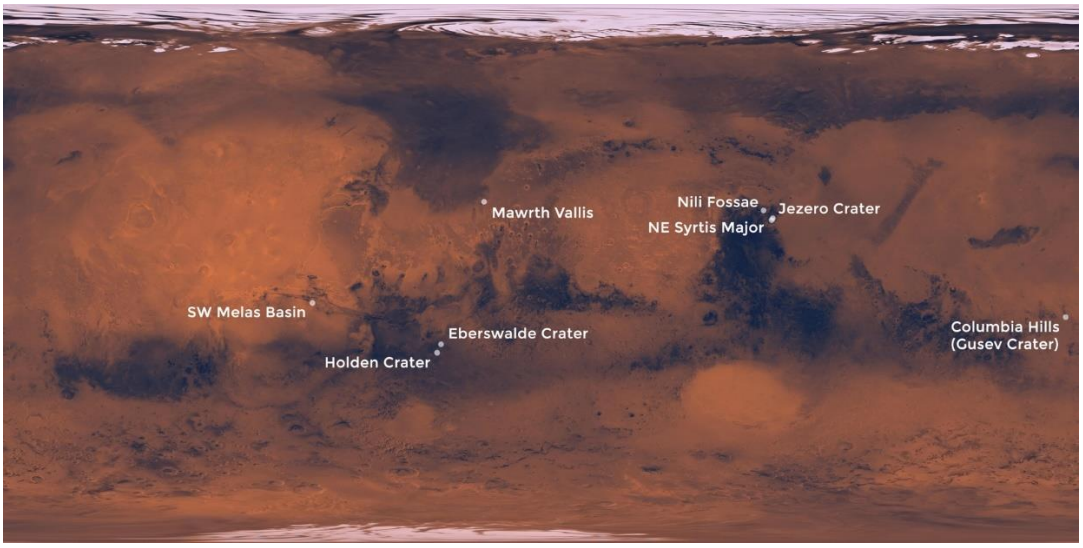

Figure 1. Potential Landing Sites, ref. [9]

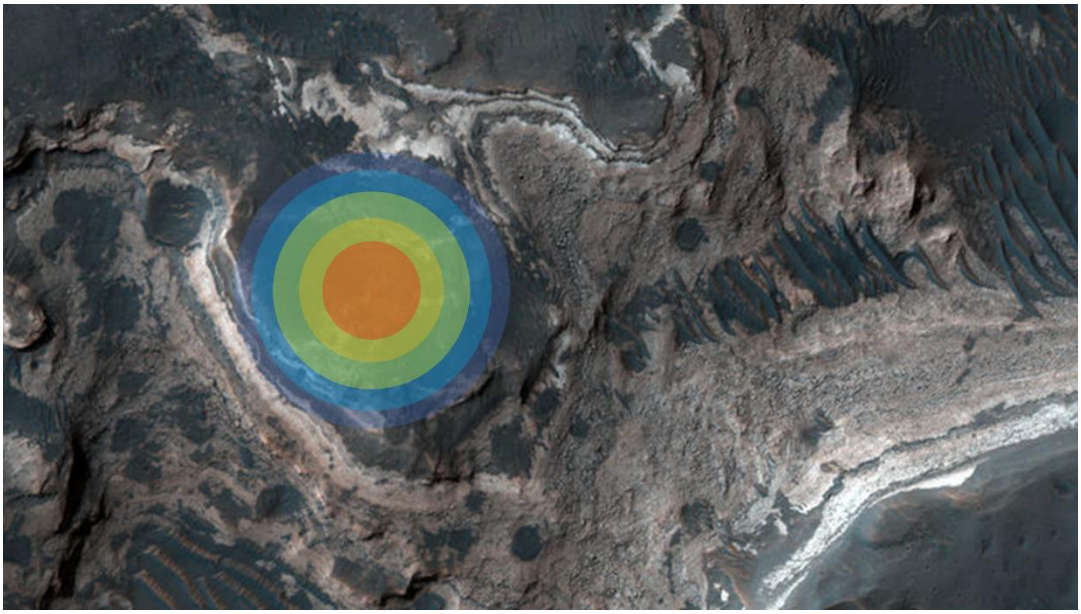

Figure 2. Holden Crater Preferred Landing Zone, ref. [9]

Although a joint distribution of latitude and longitude may characterize the desired landing zone within the Holden crater, additional dimension are also of interest to include velocity and flight path angle. Visualizing and quantifying higher dimensional joint distributions add another level of complexity to adequately assessing the initial conditions necessary to reach a satisfactory state that allows for a safe landing within the specified region.

1.2 OBJECTIVES

The objective of this dissertation is to derive and apply a scheme for solving the uncertainty about initial conditions of dynamic systems given a desired response. Examples are first assessed on single and 2-dimension cases for intuitive purposes. Consideration of higher dimensions is necessary when observing reentry equations of motion or other flight path computations. The primary applicable case is considered for determining bounds on uncertain initial conditions for a reentry body given impact criteria described in Section 1.1.

The employment of Kolmogorov's forward (Fokker-Planck equation) is observed assuming that no diffusion (no random excitation) occurs for both the forward and reverse dynamics of a system. Diffusion is investigated and shown to cause large instabilities within outputs. The alternating directions implicit numerical method is leveraged to solve Fokker-Planck formulations of the various examples. Monte Carlo Simulations are leveraged as verification cases for the methodology.

1.3 SCOPE

Chapter 2 provides a comprehensive literature survey of the field of uncertainty quantification as applied to engineering, specifically time-dependent differential equations. An overview of stochastic time-dependent differential equations, types of uncertainty inherent in engineering problems, stochastic characterization of uncertainty, forward and inverse methods, and numerical methods are all discussed at length. The literature survey concludes with a discussion of current methods for characterizing uncertainty in dynamics with emphasis on flight path uncertainty of reentry vehicles.

Chapter 3 depicts the derivation and methodology for using the presented forward and inverse uncertainty quantification schemes. The probability density is computed using a finite difference numerical method for solving a diffusion-less Fokker-Planck (FP) equation. Numerous accounts for solving the FP equation in this manner are available and discussed. The forward problem is presented primarily for completeness of the subject matter. The novel approach presented for solving the reverse kinematics is then derived. This approach requires modification to system equations of motion and implementation of a numerical scheme that computes the inputs of the stochastic process backwards through time.

Academic example problems demonstrating the forward and inverse computation of uncertain system response and initial conditions are depicted in Chapter 4. The example problems advance in difficulty starting with the observation of a linear oscillating system. The next problem then depicts a nonlinear set of equations. Both initial cases provide differential equations composed of 2 dimensions. Chapter 5 depicts the primary reentry cases for both Earth and Mars reentry. For this final case, equations of motion containing higher order of dimensions are observed as compared to the academic examples of Chapter 4. These chapters depict the method using ADI for solving these problem sets as compared to Monte Carlo Simulations for validation.

The report is concluded with Chapter 6. A discussion of the results is presented followed by final thoughts on the proposed methods. This includes the accuracy and computational efficiency of the various methods employed. Future work is discussed to close out the report and provide a comprehensive starting point for follow on work aimed to improve and expand this research. Appendices are provided listing out MATLAB® scripts used to generate results for this research.

CHAPTER 2

LITERATURE REVIEW

A comprehensive literature review is discussed in this chapter, expanding upon multiple fields that build upon the novel approach presented in this research. A stochastic differential equation and stochastic process are presented first as these are the primary building blocks for assessing uncertainty within system dynamics. Various methods for characterizing uncertainty within system inputs is then reviewed. After characterizing uncertainty, methods for quantifying it over a process are presented. An in depth look at the primary method used in this research (the Fokker-Planck equation) is then studied along with respective numerical approaches.

2.1 STOCHASTIC DIFFERENTIAL EQUATIONS

Differential equations (DE) are commonly formulated to describe the motion of a system. The derivation and application of DEs cover a wide range of topics discussed in numerous academic texts. Differential equations with random functions (i.e. uncertain initial conditions, random excitations, stochastic coefficients) formulate a stochastic process [8] that can be expressed in one of two ways [10]. The first is described as the stochastic differential equation (SDE) with no random excitation as depicted in Equation (1). This follows a more traditional DE with the addition of some randomness, whether it be in the initial condition (c), boundary constraints, or coefficients of the function f. Given the addition of some random excitation or noise (ξ), Equation (2) is leveraged for characterizing the SDE where G is a mapping of system parameters to the noise component. The three types of randomness that

may be contained in an SDE are: initial conditions, model coefficients, and excitations. These types may be observed in varying levels and configurations dependent on the problem space being explored. Both equations are indicative of a Markov process, defined as a stochastic process in which the future states are derived solely from knowledge of the present state (initial conditions) and do not depend on information of the system from past states [11].

$$\dot{x} = f(t, x),\ x_0 = c \tag{1}$$

$$\dot{x} = f(t, x) + G(t, x)\xi,\ x_0 = c \tag{2}$$

Equation (2) is commonly represented in the Langevin form as Equation (3). Where μ is considered the drift coefficient, σ is the diffusion coefficient, and B_t is a Wiener process representing the integral of a Gaussian white noise process.

$$\dot{\xi}_i = \mu_i(\boldsymbol{\xi}, t)dt + \sigma_{ij}(\boldsymbol{\xi}, t)dB_t \tag{3}$$

Statistical calculus and numerical methods provide approaches for computing the time dependent stochastic process X_t over a probability space. These methods are described further in Sections 2.3 and 2.4.

What is observed in later example problems are discrete-time Markov chains, a Markov processes on a finite state space. Stationary and nonstationary categories compose Markov processes describing whether the distribution remains the same over time (stationary) or changes (nonstationary) [12]. The stationary system can also be thought of as “time-invariant”, such as the addition of small perturbations of inputs due to background noise (Gaussian White

Noise) [13]. Response distributions in many cases can vary drastically from the initial distributions observed over a space. Figure 3 depicts this change in the distribution of a linear oscillator where the initial position is a Gaussian distribution centered on 5 units with a small standard deviation.

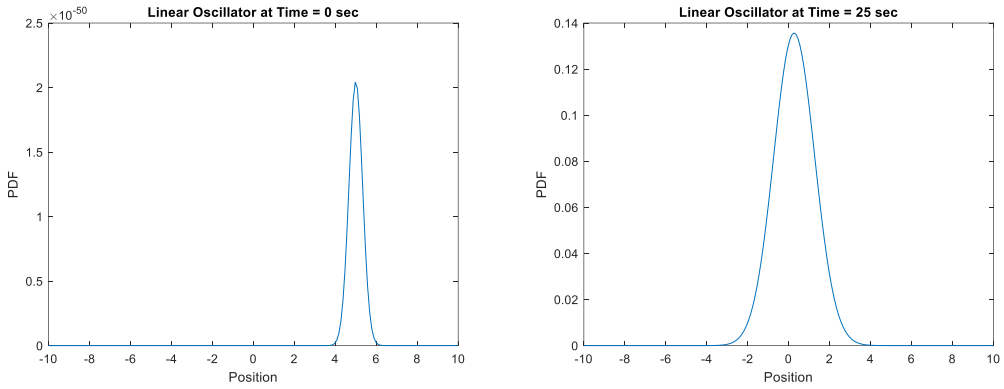

Figure 3. Nonstationarity of Position for a Linear Oscillator

The linear oscillator example is a benign example of a nonstationary system, where the response is still found to be Gaussian with a different mean and standard deviation. In many cases, the response may evolve into a completely different distribution. New distributions may or may not be characterized by a function, adding to the difficulty in quantifying transient response PDFs.

2.2 CHARACTERIZATION OF UNCERTAINTY

As previously discussed, stochastic differential equations may contain uncertainty in initial conditions, model coefficients, and/or excitations. The uncertainty, or randomness, inherent in these parameters has already been depicted for the linear oscillator example. In

general, PDFs are powerful functions for characterizing the variability of some parameter or set of parameters in a random sampling. Quantifying statistical properties of a parameter follows a systemic process inclusive of data collection, statistical analysis, and probability distribution assessment. Random variables may be of a discrete (random variable describes some integer) or continuous (random variable consists of an infinite range of values on a number line) nature [14]. Through empirical data or known probabilities of system inputs, probability mass functions (discrete) and probability density functions (continuous) are used to characterize the randomness of an input variable. When multiple random variables are assessed, the ability to observe their combined probabilities is called a joint PDF. Marginal PDFs then define the observation of a single variable that has all other variable effects removed from its computation. Conditional probability distribution is then the probability of one variable conditional on the values of another variable.

In many research fields, the use of a Gaussian distribution is of primary interest due to its span, symmetry, and statistical characteristics. Although Gaussian distributions are commonly used, they are not necessarily the most representative of engineering systems. Additional PDFs of interest to the engineering community include continuous functions such as lognormal, uniform, triangular, Rayleigh, and beta distributions and discrete functions including binomial, Poisson, Bernoulli, and exponential distributions. The information that is stored within a PDF allows for computation of stochastic moments, probabilities, and cumulative distribution function of the system subject to a normalization condition depicted in Equation (4), where $\Pr$ is the probability, $p(x)$ is the probability distribution of some continuous

distribution, and x is the random variable [15]. These properties make PDFs powerful tools for understanding the statistics of a system.

$$\Pr(-\infty < x < \infty) = \int_{-\infty}^{\infty} p(x)\, dx = 1 \tag{4}$$

Table 1 depicts a list of common PDFs found in engineering applications and may be used as a reference for subsequent example problems, where μ_X is the mean and σ_X is the standard deviation [14, 16]. Γ is a factorial function that may be evaluated using the process defined in Appendix 2 of [18].

Table 1. Common Probability Distribution Functions in Engineering

Continuous Functions		
Distribution	**PDF**	**Use Cases**
Gaussian	$f_X(x) = \frac{1}{\sigma_X\sqrt{2\pi}} exp\left[-\frac{1}{2}\left(\frac{x-\mu_X}{\sigma_X}\right)^2\right]$, $-\infty < x < \infty$	Random vibration [14], generalizations of parameters with no a priori knowledge
Lognormal	$f_X(x) = \frac{1}{\zeta_X x\sqrt{2\pi}} exp\left[-\frac{1}{2}\left(\frac{\ln(x)-\lambda_X}{\zeta_X}\right)^2\right]$, $0 < x < \infty$	Variables physically cannot be negative, defined by logarithmic parameters ζ_X (Eq. 5) and λ_X (Eq. 6)
Uniform	$f_X(x) = \frac{1}{b-a},\ a < x < b$	Equally probable outcomes over specified bounds a and b
Triangular	$f_X(x) = \frac{(b-c)-\lvert c-x\rvert}{(b-c)^2},\ a < x < b$	Centrally probable outcomes over c and specified bounds a and b
Rayleigh	$f_X(x) = \frac{x}{\sigma^2} exp\left[-\frac{x^2}{2\sigma^2}\right],\ 0 < x < \infty$	Chi squared distribution with 2 degrees of freedom, assesses peak values
Beta	$f_X(x) = \frac{1}{B(q,r)}\frac{(x-a)^{q-1}(b-x)^{r-1}}{(b-a)^{q+r-1}}$, $a \le x \le b, 0\ otherwise$	A flexible distribution over specified bounds a and b where q and r are Beta function (Eq 7) parameters estimated through their relationship with mean and variance (Eq. 8 and 9)
Discrete Functions		
Distribution	**PDF**	**Use Cases**
Binomial	$P(X = x, n\lvert p) = \binom{n}{x} p^x (1-p)^{n-x}$	Characterizes the randomness of a pass/fail event or systems with only 2 discrete outcomes given n number of observations
Poisson	$P(x\ occurences\ in\ time) = \frac{(\upsilon t)^x}{x!} e^{-\upsilon t}$	Probability of several events occurring over time (t) and event occurrence per time (υ).
Bernoulli	$P(k) = \begin{cases} p, & k = 1 \\ 1-p, & k = 0 \end{cases}$	Probability that a system is at a specific state (k).
Exponential	$P(T > t) = e^{-\upsilon t}$	Probability that no events occur over time.

$$\zeta_X^2 = \ln\left[1 + \left(\frac{\sigma_X}{\mu_X}\right)^2\right] \tag{5}$$

$$\lambda_X = \ln(\mu_X) - \frac{1}{2}\zeta_X^2 \tag{6}$$

$$B(q,r) = \frac{\Gamma(\mathrm{q})\Gamma(\mathrm{r})}{\Gamma(\mathrm{q}+\mathrm{r})} \tag{7}$$

$$E(X) = a + \frac{q}{q+r}(b-a) \tag{8}$$

$$Var(X) = \frac{qr}{(q+r)^2(q+r+1)}(b-a)^2 \tag{9}$$

All three of the random cases described above are intrinsic to engineering problems and experiments. Attempts to create more realistic models require uncertainty of all three to some capacity. This allows for sensitivity analysis of system responses to the corresponding variability introduced by uncertain inputs [17] and reliability of engineering design [18]. Probabilities and statistical moments of response variables provide insight as to the risk that is taken in specific design or implementation of a system. For the example described in Chapter 1, after assessing the uncertain inputs, a probability is computed for where the reentry vehicle will land. If some significant portion of the probability overlays an area that could result in mission failure, this area is the risk given specific design and implementation parameters.

The first source of randomness discussed is introduced via initial conditions. Generally, initial conditions are provided to solve differential equations (initial value problems). Applying variability to these inputs can characterize events such as but not limited to weather prediction [19, 20], structural geometry [21], spread of disease [22], and (of particular interest to this

research) initial kinematics of dynamic systems [23-26]. Additionally, randomness can be attributed to model coefficients as well. This uncertainty is observed within the equations that have been derived for characterizing a system. Some examples of random coefficients include ballistic coefficients [27], modulus of elasticity [21], or basically any coefficient within a mathematical model. The final random type is that of the excitation applied to a model. This random excitation may encompass events such as sea state, wind load, vehicle vibration over an interstate, etc. [28].

The accurate characterization of uncertainty within a differential equation is an important step toward representing realistic responses of systems through modeling and simulation. Numerous experiments are conducted to adequately quantize the variability surrounding random inputs so that the community may better represent these systems through modeling and simulation. Unfortunately, these stochastic parameters or PDFs cannot be directly applied to deterministic equations and models.

2.3 FORWARD/INVERSE METHODS FOR UNCERTAINTY QUANTIFICATION

Various methods have been derived for propagating the different types of uncertainty discussed above, and new methods continue to be a topic of research to minimize computational cost while maximizing accuracy of results. As defined in Section 1.1, UQ can be categorized as forward and inverse problems where the propagation of uncertainty through either category is done using unique derivations of system equations or processes. In the case of forward UQ, the most popular method is the non-intrusive [29] Monte Carlo Simulation (MCS). The MCS method may be applied to systems with uncertain initial conditions, model coefficients, and excitations. This is unique as other methods seldom handle uncertainty of all

three categories. Another unique attribute of MCS is the ability to compute transient response PDFs. It was discussed in section 2.2 that these response PDFs through time provide critical information as to the stochastic properties of a system. Other methods such as Taylor series expansion [30] and polynomial chaos [31] are examples of forward methods that can handle various degrees of uncertainty for a wide array of problems. Of primary concern in this research are specific methods used to determine the uncertainty associated with initial conditions.

A great deal of literature is available on MCS and its diverse applications. Devised during World War II by John von Neumann and Stanislaw Ulam to improve decision making given stochastic inputs [32], the Monte Carlo method quickly spread to countless disciplines where its use made for better decision-making strategies when faced with uncertainty. No limitations exist for number of random variables, or the distributions used to characterize the random variables. Use of this method is simple, as its process is easily applied to any existing model. Uncertain values with known probability distribution are set as inputs for a simulation. A random number is generated to select samples and discretize the problem. The simulation is then run, generating desired outputs. This process is repeated a finite number of times to achieve a desired accuracy of results [33]. Limitations of this method are dependent on computational resources. A variety of “smart sampling” techniques have been explored to minimize computational resources while adequately characterizing the uncertain space. Due to the simplicity, applicability, and history of MCS, it is commonly used to validate new processes. MCS is employed in the example problems of Chapters 4 and 5 as ground truth for comparison of the presented method.

For completeness, it is important to mention the existence of closed form solutions for computing responses of systems with uncertain inputs. Various closed form solutions have been developed to characterize uncertainty within systems. The major limitation with these methods is their applicability and robustness. Closed form solutions are generally derived from first principles of the specific system being assessed and are dependent only on factors identified as significant at the time of derivation. An example of a closed form solution of propagating uncertainty in mechanical design is presented in [34]. The number of cycles to failure is computed for mechanical designs under varying loads using Manson's method. This is done by establishing a maximum and minimum stress, along with many table lookups to characterize defects in the material. For each cycle, mean and alternating stresses are computed from the defined minimum and maximum stress. Each cycle is assessed for damage and summed until a threshold is met indicating mechanical failure. It is obvious here that no true randomness is applied to the problem. This closed form solution, much like many others, leverages a mean value given knowledge of the min and max characteristics of a specific system. Not only is this method limited to computing cyclic loading of mechanical systems, but it is also assuming a uniform distribution of the loads.

The first inverse UQ of initial conditions that is assessed is the Mean Square Method presented in [10]. Given the initial state and a random initial condition depicted in Equations (10) and (11), assume an explicit solution can be found as depicted in Equation (12).

$$\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t), t) \tag{10}$$

$$\boldsymbol{x}(t_0) = \boldsymbol{x}_0 \tag{11}$$

$$\boldsymbol{x}(t) = h(\boldsymbol{x}_0, t) \tag{12}$$

The inverse is then assessed to determine the initial conditions.

$$\boldsymbol{x}_0(t) = h^{-1}(\boldsymbol{x}, t) \tag{13}$$

The determinate of the Jacobian of Equation (13) is then found using Equation (14) and used to determine the joint density function of the initial conditions as depicted in Equation (15).

$$|J| = \left|\frac{\partial \boldsymbol{x}_0^T}{\partial \boldsymbol{x}}\right| \tag{14}$$

$$f(\boldsymbol{x}, t) = f_0[\boldsymbol{x}_0 = h^{-1}(\boldsymbol{x}, t)]|J| \tag{15}$$

This method is useful in providing an analytic probability density function that characterizes the initial conditions based on response variability. Unfortunately, this method requires defined distribution functions for response variables as well as an analytical solution for the systems differential equation. Having this information is rare in realistic engineering problems. Other methods like the Mean Square Method compute initial conditions using processes such as multivariate intervals [35], stochastic adaptive interpolation [25], and alpha path equation [36].

Although numerous methods exist for determining uncertainty in initial conditions of a physical system, they are limited in applicability and scope. Requirements of formulating an analytical solution, dimensionality, and defining response PDFs can be severely limiting in engineering problems. This is less true with propagating uncertain initial conditions forward in

time using MCS, but computational resources are always a concern when employing this method. Efficiently solving physical system ODEs with uncertain initial conditions that do not meet the requirements of the previous methods, [37] derives a Partial Differential Equation (PDE) that is capable of computing the transient PDF of an ODE as depicted in Equation (16). Upon further investigation, Equation (16) is the Fokker-Planck equation of a stochastic differential equation with zero diffusion (no random excitation) also known as the Liouville equation. Here t is time, $p(\boldsymbol{x}, t|\boldsymbol{x}_0, t_0)$ is the probability density as a function of the variable vector given initial conditions, d is the number of variables, and $A_i(\boldsymbol{x}, t)$ is the drift vector.

$$\frac{\partial}{\partial t}p(\boldsymbol{x}, t|\boldsymbol{x}_0, t_0) = -\sum_{i=1}^{d}\frac{\partial}{\partial x_i}\left(A_i(\boldsymbol{x}, t)p(\boldsymbol{x}, t|\boldsymbol{x}_0, t_0)\right) \tag{16}$$

2.4 FOKKER-PLANCK EQUATION

Originally derived to describe Brownian motion of particles in 1914, the Fokker-Planck (FP) equation is a second order parabolic PDE also known as the forward Kolmogorov equation [38]. The FP equation provides an "equation of motion for the distribution function of fluctuating macroscopic variables," or, in simpler terms, characterizes the variability surrounding some motion (described by an ODE) throughout time given uncertainty in initial conditions or random excitation. Also known as Kolmogorov's Equations or Smoluchowski Equation; the FP equations origins are rooted in various scientific derivations that lead to varying forms of what is now primarily known as the FP equation [39]. The most rigorous derivation stems from the differential Chapman-Kolmogorov equation, composed of drift, diffusion, and jump characteristics. Derivations of this form and others are available in [24, 37,

39]. Following one of the more popular derivations [38] a set of stochastic variables is defined as $\boldsymbol{\xi} = \xi_1, \xi_2, \ldots, \xi_N$, where N is the total number of random variables. Equation (17) demonstrates the relationship between a probability density ($P(x, t+\tau)$) and its transition ($P_{tr}(x, t+\tau|x_0, t)$) at some new time (τ) and initial conditions (x_0) for a single variable case.

$$p(x, t+\tau) = \int p_{tr}(x, t+\tau|x_0, t)p(x_0, t)dx \tag{17}$$

Assuming all moments of the transition probability are defined by Equation (18) for $n \geq 1$, the transition probability may be expanded using a variety of techniques.

$$\begin{aligned} M_n(x_0, t, \tau) &= \langle[\xi(t+\tau) - \xi(t)]^n\rangle|_{\xi(t)=x_0} \\ &= \int (x - x_0)^n p_{tr}(x, t+\tau|x_0, t)dx \end{aligned} \tag{18}$$

Defining $\Delta = x - x_0$, the integrand in Equation (17) may be expanded in a Taylor series as:

$$\begin{aligned} &p_{tr}(x, t+\tau|x_0, t)p(x_0, t) \\ &\quad = p_{tr}(x - \Delta + \Delta, t+\tau|x - \Delta, t)p(x - \Delta, t) \\ &\quad = \sum_{n=0}^{\infty} \frac{(-1)^n}{n!} \Delta^n \left(\frac{\partial}{\partial x}\right)^n p_{tr}(x + \Delta, t+\tau|x, t)p(x, t) \end{aligned} \tag{19}$$

Applying this to Equation (17) and integrating over Δ leads to the formulation in Equation (20).

$$p(x, t+\tau) - p(x, t) = \sum_{n=1}^{\infty} \left(-\frac{\partial}{\partial x}\right)^n \left[\frac{M_n(x, t, \tau)}{n!}\right] p(x, t) \tag{20}$$

Expanding the moments using Taylor series about τ results in the following equation:

$$\frac{M_n(x, t, \tau)}{n!} = D^{(n)}(x, t) + O(\tau^2) \tag{21}$$

Where the Kramers-Moyal operator is defined as:

$$L(x,t) = \sum_{n=1}^{\infty} \left(-\frac{\partial}{\partial x}\right)^n D^{(n)}(x,t) \tag{22}$$

And yields:

$$p(x,t+\tau) - p(x,t) = \sum_{n=1}^{\infty} \left(-\frac{\partial}{\partial x}\right)^n D^{(n)}(x,t)\, p(x,t) \tag{23}$$

After omitting the case where $\tau = 0$ (because no change in time yields no change in the probability density from its initial conditions), an infinite sequence of coefficients is observed for defining the transitional scale of the probability density. The coefficients $D^{(n)}(x,t)$ are attainable from the functions defining the stochastic differential equation of interest. From the Langevin form defined in Equation (3), the first two coefficients of the series are derived [38]. These coefficients are the drift ($A_i(\boldsymbol{x},t)$) and diffusion ($B_{ij}(\boldsymbol{x},t)$) coefficients, where all other coefficients ($n > 2$) are equal to zero.

$$A_i(\boldsymbol{x},t) = \lim_{\tau \to 0} \frac{1}{\tau} \langle \xi_i(t+\tau) - x_i \rangle|_{\xi_k(t)=x_k}, \qquad k = 1,2,\dots,N \tag{24}$$

$$B_{ij}(\boldsymbol{x},t) = \frac{1}{2} \lim_{\tau \to 0} \frac{1}{\tau} \langle [\xi_i(t+\tau) - x_i][\xi_j(t+\tau) - x_j] \rangle|_{\xi_k(t)=x_k}, \quad k = 1,2,\dots,N \tag{25}$$

Knowing the deterministic differential equations of a system provides the necessary information for obtaining the drift coefficients. For Brownian motion that is applicable to the diffusion coefficient, problem heuristics may define the matrix scaling coefficients [38]. Obtaining the coefficients simply requires a decomposition of the system into a set of first-

order equations that define its motion. Inserting Equations (24) and (25) into Equation (23) yields the traditional Fokker-Planck (forward Kolmogorov) equation.

$$\frac{\partial}{\partial t}p(\boldsymbol{x},t|\boldsymbol{x}_0,t_0) = -\sum_i \frac{\partial}{\partial x_i}[A_i(\boldsymbol{x},t)p(\boldsymbol{x},t|\boldsymbol{x}_0,t_0)] + \sum_{i,j}\frac{1}{2}\frac{\partial^2}{\partial x_i \partial x_j}\left[B_{ij}(\boldsymbol{x},t)p(\boldsymbol{x},t|\boldsymbol{x}_0,t_0)\right] \quad (26)$$

Mathematically, the FP equation is known as a diffusion process. Many diffusion processes found in literature are driven by a standard Wiener process (Gaussian White Noise excitation) [40], although other formulations have been explored to provide non-Gaussian excitation to a system [41]. As mentioned above, many cases may exist where the Brownian motion is determined heuristically. In the case where no diffusion exists, Equation (26) can be further decomposed resulting in a special case of the Liouville's Equation which was depicted earlier in Equation (16).

2.4.1 FINDING THE DRIFT VECTOR

FP formulations result in nonlinear PDEs that describe the transient nature of a systems probability distribution. The states are specific to ODEs with uncertain initial conditions and random excitation. The drift vector is created directly from the set of first-order ODEs used to specify a system, whereas the diffusion matrix is subject to not only a vector of magnitudes but also the equations defining its randomness. Applying diffusion in the reverse case incorporates large instabilities within the solution sets as explained in later sections. For this reason, the diffusion-less process is of primary concern for this research and derivation of the drift vector is

prioritized. For example, consider the differential equation characterizing a linear oscillator depicted in Equation (27), where x is position, $\dot{x}$ is velocity, $\ddot{x}$ is acceleration, k is spring stiffness, and c is the damping factor. There is no random excitation in this problem statement, so the diffusion matrix can be neglected for now.

$$m\ddot{x} = -kx - c\dot{x} \tag{27}$$

Let $x_1 = x$ and $x_2 = \dot{x}$, such that Equation (27) can be described by a set of ordinary differential equations depicted in Equation (28).

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -\frac{k}{m}x_1 - \frac{c}{m}x_2 \end{bmatrix} \tag{28}$$

The drift vector is then taken as this set of ordinary differential equations and substituted as the $A_i(\boldsymbol{x}, t)$ vector in Equation 26. After differentiating, and observing no random excitation ($B_{ij}(\boldsymbol{x}, t) = 0$) the FP equation for the linear oscillating system is:

$$\begin{aligned} &\frac{\partial}{\partial t} p(\boldsymbol{x}, t|\boldsymbol{x}_0, t_0) \\ &\qquad = -x_2 \frac{\partial}{\partial x_1} [p(\boldsymbol{x}, t|\boldsymbol{x}_0, t_0)] \\ &\qquad + \frac{\partial}{\partial x_2} \left[\left(\frac{k}{m} x_1 + \frac{c}{m} x_2 \right) p(\boldsymbol{x}, t|\boldsymbol{x}_0, t_0) \right] \end{aligned} \tag{29}$$

Equation (29) is the product of a relatively simple dynamic system. Provided with simple joint probability distributions characterizing the behavior of each state, an analytical solution is

possible. As the joint distribution increases in complexity, numerical methods are necessary to solve for each state. Add to this a case where diffusion is present and the need for numerical methods becomes more prevalent.

For completeness a brief discussion on the diffusion matrix is presented although this matrix is omitted for much of the research due to limitations explained in Section 3.2. Generally, the diffusion matrix is composed of scaling factors that are applied to some level of randomness usually in the form of Gaussian white noise [48]. Scaling factors are obtained from experimental data, trial and error, or educational assumptions. Other levels of randomness may be applied but further complicate the ability to numerically solve the FP equation.

2.4.2 ALTERNATING DIRECTION IMPLICIT NUMERICAL METHOD

Given the nonlinear nature of transient PDF equations, there is seldom an analytical solution available (primarily only cases containing linear systems driven by Gaussian excitations [42]). Numerical methods such as finite elements and finite differences are well documented [43-44] with [45] providing a comprehensive comparison of the two as applied to the FP equation. From [45], the Alternating Directions Implicit (ADI) finite difference method was adopted for use in this research due to its accuracy, computational efficiency, and ability to handle multi-dimensional problem spaces. ADI solves finite difference steps in each direction separately, implicitly for one dimension and explicitly for all other dimensions resulting in stable formulations. For a two-dimensional problem over variables x_1 and x_2, the initial steps of ADI for the FP equation are depicted in Equations (30) and (31) where the A and B matrices are derived as tridiagonal coefficients. The A matrix is composed of coefficients defined over each node given the value of the central difference of the specified state equation of motion, where

N is the total number of nodes defined for all directions and M is the total number of dimensions.

$$A_{x_M} \equiv \begin{bmatrix} r_1 & q_1 & 0 & 0 & 0 \\ s_1 & r_2 & q_2 & 0 & 0 \\ 0 & s_2 & \ddots & \ddots & 0 \\ 0 & 0 & \ddots & \ddots & q_{N-1} \\ 0 & 0 & 0 & s_{N-1} & r_N \end{bmatrix} \tag{30}$$

where:

$$\begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{r} \\ \boldsymbol{s} \end{bmatrix} = \begin{bmatrix} \dot{\boldsymbol{x}}_M|_{1:N-1} \\ \mathbf{0}|_{1:N} \\ -\dot{\boldsymbol{x}}_M|_{1:N-1} \end{bmatrix} \tag{31}$$

A series of tridiagonal matrices are formulated for each dimension 1 through *M*. The diffusion matrix (if existing) is formulated similarly using scaling factors defined from heuristics applied to a tridiagonal matrix of ones. The $\boldsymbol{r}$ vector in the case of no diffusion ends up being zeros. The transient probability distribution may then be solved using Equations (32)-(33) for a two-dimensional case, where $\boldsymbol{p}$ is a vector of the probability density at each node of the finite difference grid for the specified time step.

$$\left(I - \frac{\Delta t}{2}\left[\frac{A_{x_1}}{2\Delta x_1} - \frac{B_{x_1}}{{\Delta x_1}^2}\right]\right)\boldsymbol{p}^{n+\frac{1}{2}} = \left(I + \frac{\Delta t}{2}\left[\frac{A_{x_2}}{2\Delta x_2} - \frac{B_{x_2}}{{\Delta x_2}^2}\right]\right)\boldsymbol{p}^{n} \tag{32}$$

$$\left(I - \frac{\Delta t}{2}\left[\frac{A_{x_2}}{2\Delta x_2} - \frac{B_{x_2}}{{\Delta x_2}^2}\right]\right)\boldsymbol{p}^{n+1} = \left(I + \frac{\Delta t}{2}\left[\frac{A_{x_1}}{2\Delta x_1} - \frac{B_{x_1}}{{\Delta x_1}^2}\right]\right)\boldsymbol{p}^{n+\frac{1}{2}} \tag{33}$$

For additional dimensions, the number of time steps are increased. The number of steps then correspond to the number of states (each state must be solved implicitly) where $n + \frac{1}{d}$, changes such that *d* is the number of states or dimensions being implicitly solved. This results in a set of equations equal to *d* that are solved over all dimensions. The resulting matrices, on both the left and right side of the equations, are tridiagonal. Tridiagonal matrices consist of non-zero diagonal elements along the upper and lower diagonal components with the remainder of the matrix being zero. This is depicted in Equation (34) leveraging the x_2-direction from previous equations, where a, b, and c are computed coefficients of the system.

$$\left(I - \frac{\Delta t}{2}\left[\frac{A_{x_2}}{2\Delta x_2} - \frac{B_{x_2}}{{\Delta x_2}^2}\right]\right) \equiv \begin{bmatrix} b_1 & a_1 & 0 & 0 & 0 \\ c_1 & b_2 & a_2 & 0 & 0 \\ 0 & c_2 & \ddots & \ddots & 0 \\ 0 & 0 & \ddots & \ddots & a_{N-1} \\ 0 & 0 & 0 & c_{N-1} & b_N \end{bmatrix} \tag{34}$$

Tridiagonal matrices provide numerous computational advantages that are a primary driver for the popularity of the ADI method. Sparse matrices can be leveraged to reduce the memory required to compute multidimensional PDFs along fine step sizes due to the large number of zero cells in each matrix. Inversion of a tridiagonal matrix is another computationally efficient property when applying the Thomas algorithm [46]. The Thomas algorithm drastically reduces the number of computations necessary to find an inverse of a matrix when compared to Gaussian elimination methods. It is observed in Equations (32) and (33) that inverting matrices

is necessary to solve for $\boldsymbol{p}$ at each step of the evaluation. Equation (33) is reformulated as Equation (35).

$$\boldsymbol{p}^{n+1} = \begin{bmatrix} b_1 & a_1 & 0 & 0 & 0 \\ c_1 & b_2 & a_2 & 0 & 0 \\ 0 & c_2 & \ddots & \ddots & 0 \\ 0 & 0 & \ddots & \ddots & a_{N-1} \\ 0 & 0 & 0 & c_{N-1} & b_N \end{bmatrix}^{-1} \left(I + \frac{\Delta t}{2}\left[\frac{A_{x_1}}{2\Delta x_1} - \frac{B_{x_1}}{{\Delta x_1}^2}\right]\right)\boldsymbol{p}^{n+\frac{1}{2}}. \tag{35}$$

Let $p_1^{n+1} = \frac{d_1}{b_1}$ and the initial $\tilde{b} = b_1$. Forward substitution is done from p_2^{n+1} to p_N^{n+1} (or N = 2 to N) via Equations (36)-(38).

$$c'_N = \frac{c_{N-1}}{\tilde{b}} \tag{36}$$

$$\tilde{b} = b_N - a_N * c'_N \tag{37}$$

$$p_N^{n+1} = \frac{d_N - a_N * p_{N-1}^{n+1}}{\tilde{b}} \tag{38}$$

Backward substitution is then performed to find all unknowns for $N = N - 1$ to 1 using Equation 39.

$$p_N^{n+1} = p_N^{n+1} - c'_{N+1} * p_{N+1}^{n+1} \tag{39}$$

An example of the above algorithm in MATLAB® is depicted in Appendix A for the first example problem of Section 4. The ADI method proves to be useful for numerically solving parabolic and elliptic partial differential equations. Eventually, the step size and number of dimensions add a significant number of computations to this process. Comparison of ADI to Monte Carlo convergence rates are assessed for example problems in Chapter 4 and 5. A 5-point-stencil of 2-dimensional ADI over a single step is depicted in Figure 4.

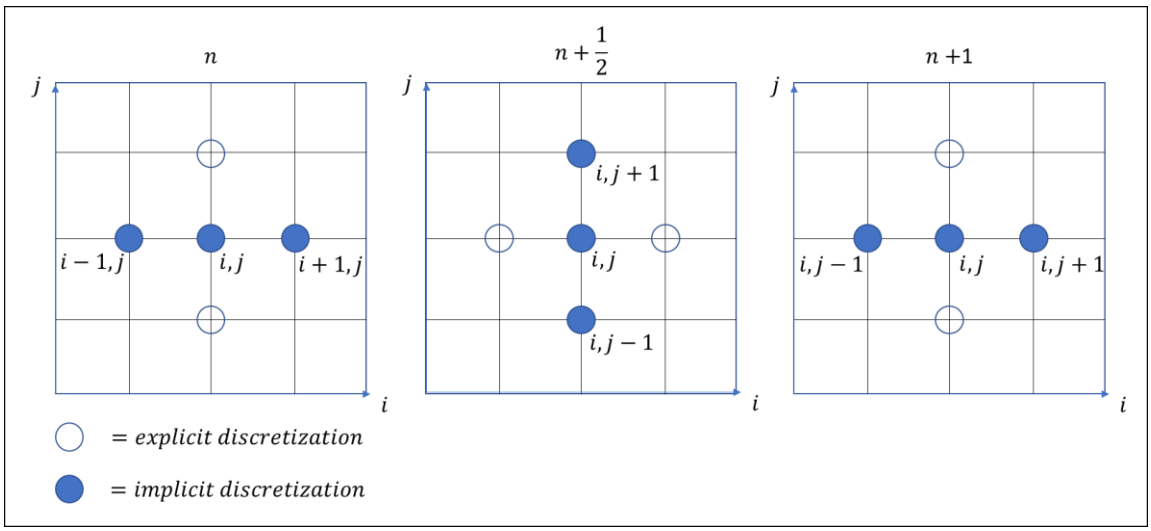

Figure 4. ADI Graphical Depiction

Application of the FP equation spans multiple disciplines within the scientific and engineering communities. Regarding dynamics, literature is found in some capacity on solving various forward problems. Orbital dynamic input uncertainty is assessed using the FP equation in [21, 44, 47]. Within these examples, initial condition uncertainty of orbital state vectors are imposed and PDFs of response variables are computed at varying times using finite element methods. Applications to control systems are discussed to determine system stability given

variance in the initial conditions and varying amplitudes of white noise excitation. Multiple forward problems are observed in [45] to include two- and three-dimensional linear oscillators, as well as a two- and three-dimensional Duffing oscillators. Here, the initial position and velocity of the mass is varied and randomly excited to determine equilibrium conditions. Ship rolling motion is presented in [48] where the roll and roll rate transient PDFs are computed for cases varying wave amplitude and white noise excitation. Copious amounts of literature are available on the FP equation in theoretical mathematical and physics applications. Although well documented and leveraged amongst many different disciplines, a direct application to reverse reentry dynamics is absent.

CHAPTER 3

FORWARD AND REVERSE UNCERTAINTY PROPAGATION

Chapter 3 provides an overview of the methodology adopted for assessing uncertainty about a dynamic system and ultimately determining uncertainty about initial conditions given a desired response. Forward and reverse deterministic cases are reviewed, ensuring an understanding of the basic dynamic process before the addition of uncertainty. The concept of time reversible systems is then discussed and how uncertain time reversible systems are solved using a Fokker-Planck equations. This chapter concludes with remarks on additional material that will assist the reader in understanding assumptions and interpreting results.

3.1 FORWARD AND REVERSE DYNAMICS

Initially, inverting the FP equation was assessed to compute PDFs of initial conditions from response information. As discussed in the previous section, solving the forward FP equation is not a trivial endeavor and computing its inverse only provides a solution for the terms that formulate the drift and diffusion coefficients. Given the problem set of interest for this research, the ability to determine a drift coefficient is dependent on the existence of a state space form of the differential equation that is being assessed. For completeness about inverse FP equations, [49] provides a comprehensive overview of methods available for solving the inverse FP equation using neural networks.

Within a forward dynamic problem, the state space form is substituted within the FP equation as the drift vector. Using the numerical method discussed above, a transient PDF of the system responses is computed. Since this response is attainable from the forward dynamics

given input conditions, it was hypothesized that the input conditions could be attainable from the reverse dynamics given a response (so long as the system is time-symmetric). To find the reverse dynamics of a system, the equations of motion are computed backwards in time. This is done by applying a negative sign to the differential equation and ensuring decimal points are computed out to the maximum extent to minimize error. This time-reversal is only possible for systems that are time-reversal symmetric. Fortunately, this encompasses any mechanical system of the form $m\ddot{x} = F(x)$ [50]. These invariant systems are defined as reversible under $t \to -t$. Applying this to a differential equation from [51] presented as the original forward formulation in Equation (40), the forward and reverse computations are plotted against time for position (orange line) and velocity (blue line) in Figures 5 and 6.

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \end{bmatrix} = \begin{bmatrix} x_2 \\ -5\mu_c(1 - x_1^2)x_2 - 9\sin(x_1) \end{bmatrix} \tag{40}$$

Where the reversible state space representation is depicted in Equation (41).

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \end{bmatrix} = \begin{bmatrix} -x_2 \\ 5\mu_c(1 - x_1^2)x_2 + 9\sin(x_1) \end{bmatrix} \tag{41}$$

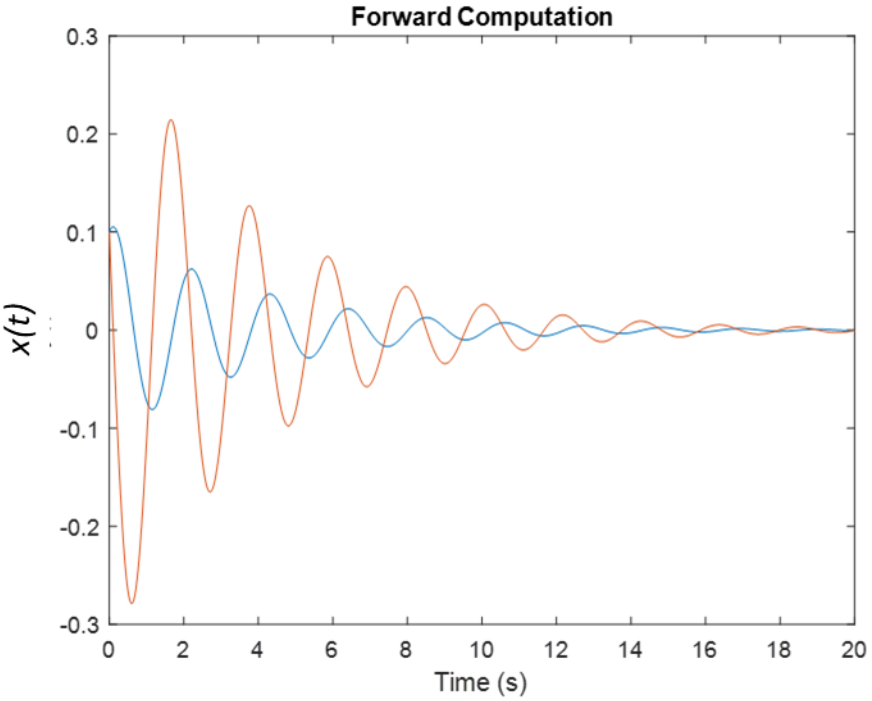

Figure 5. Forward Dynamic Propagation

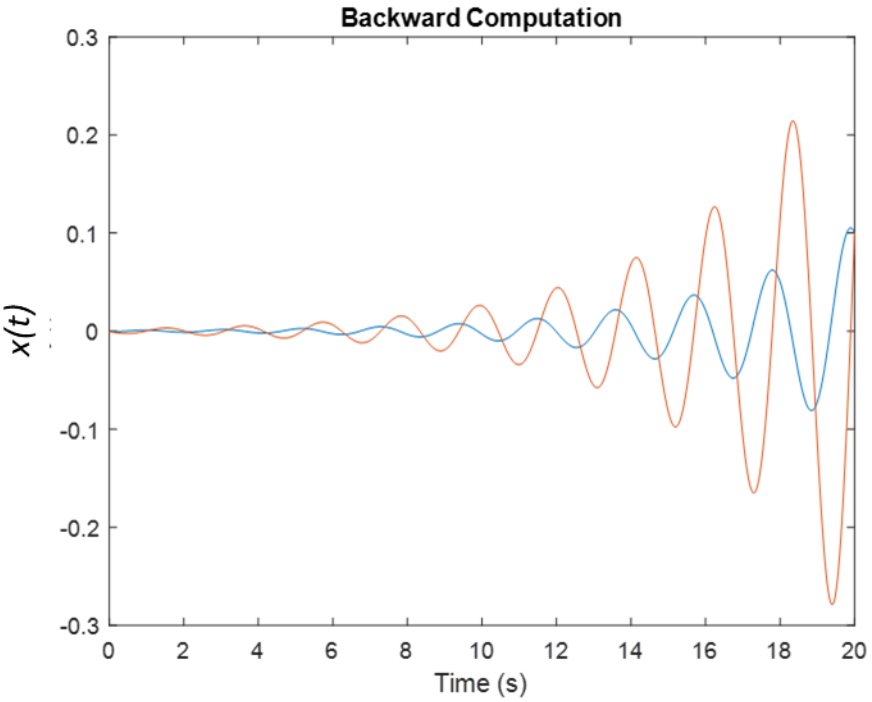

Figure 6. Reverse Dynamic Propagation

Minimal errors between the forward and reverse dynamics are observed in Figure 7. This provides confidence in the observed results obtained from reversing the dynamics of the system. These reversed conditions may then be applied within the FP equation to determine the backwards transient PDF of a system, eventually leading to initial conditions of interest.

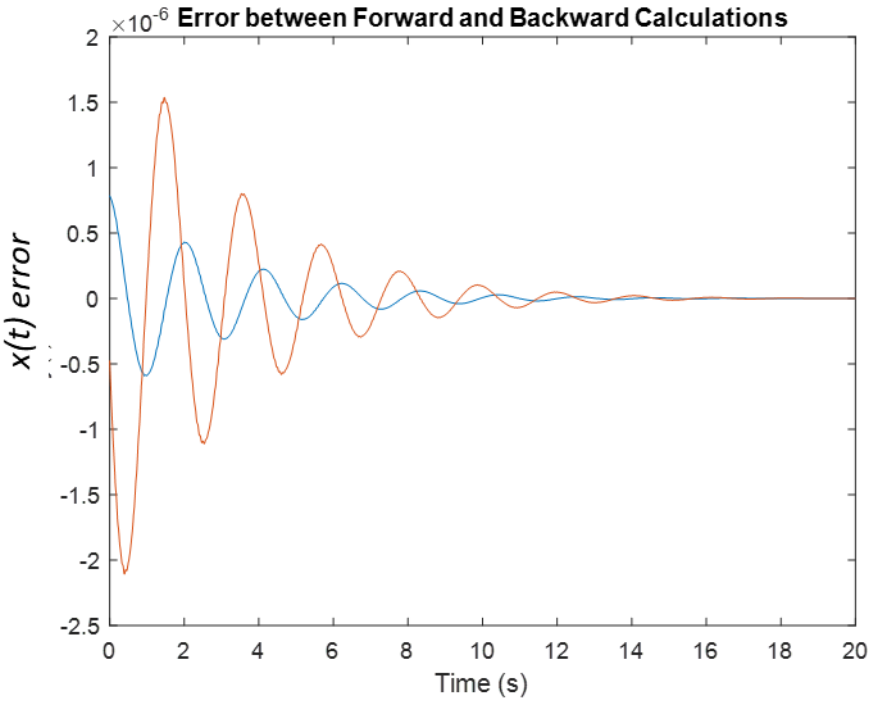

Figure 7. Error Between Forward and Reverse Dynamics

Time reversible systems are apparent in many academic applications of classical mechanics that are time symmetric. Time reversibility is restricted to closed systems where inputs such as guidance commands or entropy may skew any time symmetries of a system.

3.2 TIME REVERSIBILITY AND THE FP EQUATION

Applying the forward dynamic case to the FP equation is demonstrated in numerous papers where an equation of motion is deconstructed into a set of ODEs and applied to the first

unknown within the FP equation (A_i). This drift vector is essentially the translation of the uncertainty about specific states. For example, take the case with no diffusion and a system composed of a single state characterized by the EOM $\dot{x}_1 = 0.1$. The resulting FP equation for this EOM is defined in Eq. (42). If the initial conditions are defined at time zero as a normal distribution with mean zero and standard deviation 0.2618, this uncertainty is easily propagated through time. Since this is a simple system, the distribution does not change but translates to different mean values. This is observed in Figure 8, where the transient probably density (PD) of the single state is depicted at two times. Plot colors are based on the magnitude of the probability density. For the two-dimensional plots, the colors are easily relatable to the magnitudes depicted on the vertical axes. The color schemes become more important with three-dimensional plots depicted in later sections. For three-dimensional plots, provided color bars define magnitudes. Additional information on the probability density is presented in the next section.

$$\frac{\partial}{\partial t} p(\boldsymbol{x}, t|\boldsymbol{x}_0, t_0) = -0.1 \frac{\partial}{\partial x_1} [p(\boldsymbol{x}, t|\boldsymbol{x}_0, t_0)] \tag{42}$$

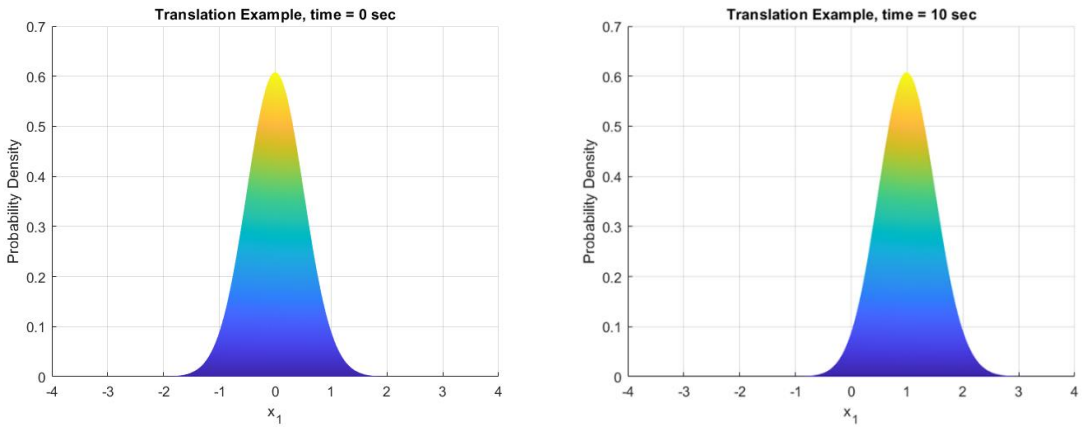

Figure 8. Translation Example, Forward In Time

Since this simple system is time reversible, the PD for x can be propagated backwards in time as well. The reverse time PD is depicted in Figure 9.

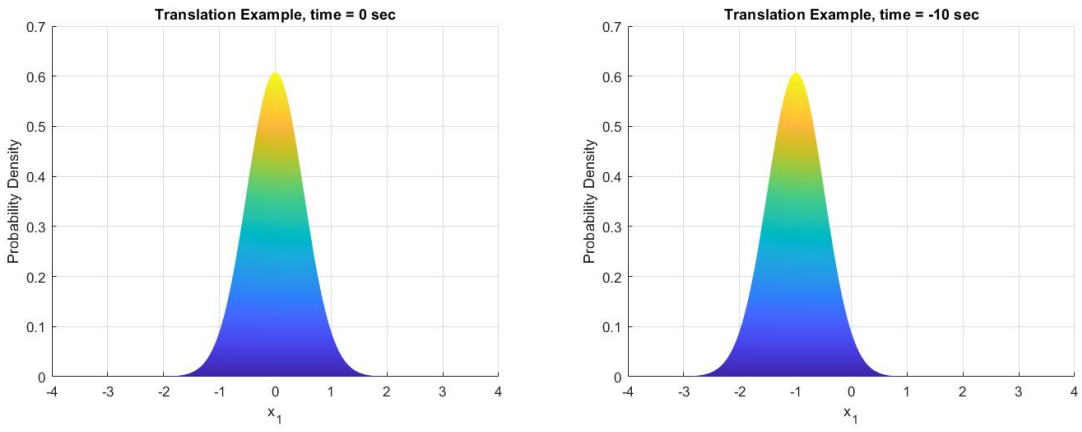

Figure 9. Translation Example, Backward In Time

As previously stated, Figures 8-9 lack diffusion within the system. The translation example is used to depict how the state space ODEs of a system directly influence the output PDF of the FP equation. Diffusion is generally added to a system to account for randomness or noise in system excitations. This is applied within the FP equation as matrix B_{ij}. The diffusion matrix scales Gaussian white noise (GWN) for the traditional FP equation [10, 48]. GWN is a random process applicable to many engineering systems that accounts for small perturbations of system inputs due to background noise. GWN is specified from a Gaussian distribution with mean of zero, and the property that any two values at different times are statistically independent [13]. It is composed of all frequency content, leading to the name “white” derived from white light which is composed of all wavelengths of the light spectrum. The use of GWN is generally scaled by a spectral density value, allowing for assessment of varying levels of white

noise on system output. For the simple example demonstrated in this section, GWN is applied as spectral densities of 0.01, 0.1, and 1.0. Figures 10, 11, and 12 depict the impact of diffusion on the simple system at times 0 and 10. The FP equation with GWN scaled at 0.01 is depicted in Equation (43).

$$\frac{\partial}{\partial t}p(\boldsymbol{x}, t|\boldsymbol{x}_0, t_0) = -0.1\frac{\partial}{\partial x_1}[p(\boldsymbol{x}, t|\boldsymbol{x}_0, t_0)] + 0.01\frac{1}{2}\frac{\partial}{\partial x_1}[p(\boldsymbol{x}, t|\boldsymbol{x}_0, t_0)] \tag{43}$$

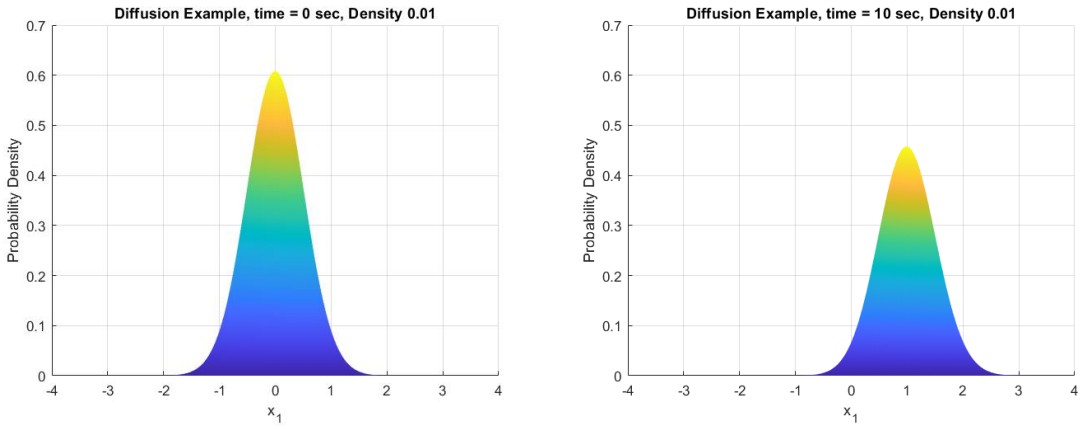

Figure 10. Diffusion Example, Density = 0.01

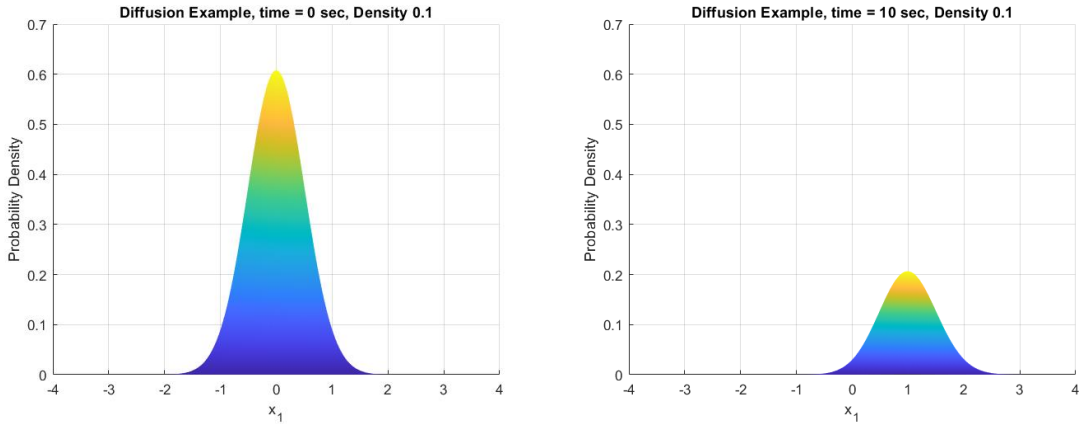

Figure 11. Diffusion Example, Density = 0.1

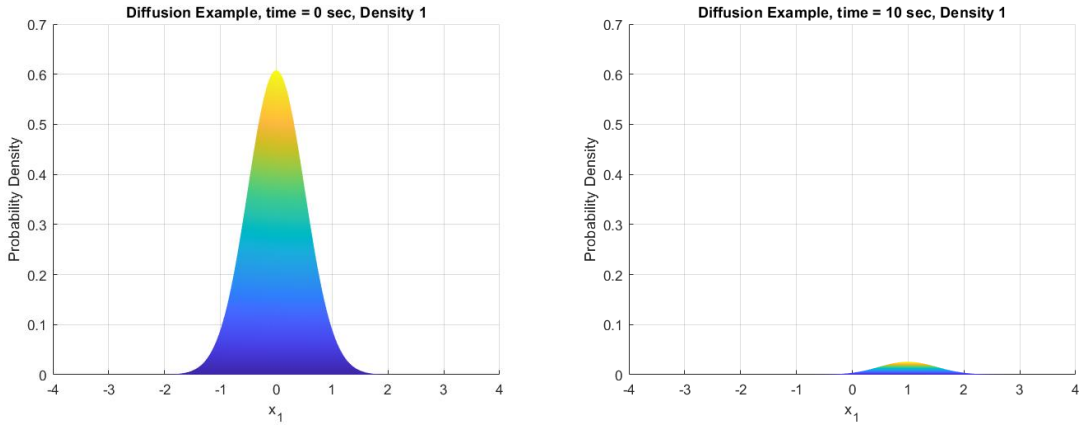

Figure 12. Diffusion Example, Density = 1

As the GWN is amplified, the probability density begins to approach a uniform distribution about all possible points within the standard deviation bounds. This approach to uniformity occurs as the magnitude of noise begins to outweigh all other sources of uncertainty within the state space. Unfortunately, the application of diffusion via the FP equation is not time symmetric. If the same density is applied to the reverse time system, the noise acts similarly to the forward case and causes the probability density to approach a uniform

distribution. As the system marches backward in time, the noise should be implementing the opposite effect. This is easily observed via Figure 13 and can be compared to the forward results in Figure 11.

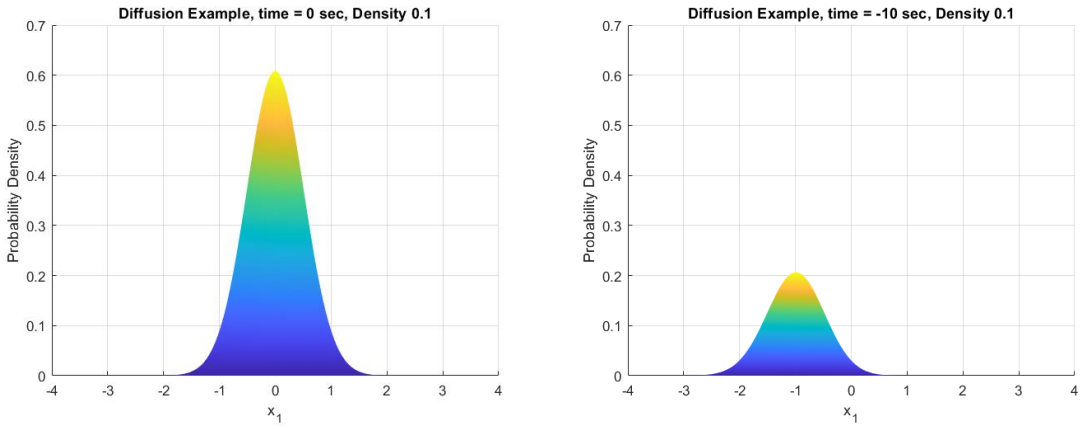

Figure 13. Reverse Diffusion, Density = 0.1

Subtracting the noise component immediately causes probability density instabilities as seen in Figure 14.

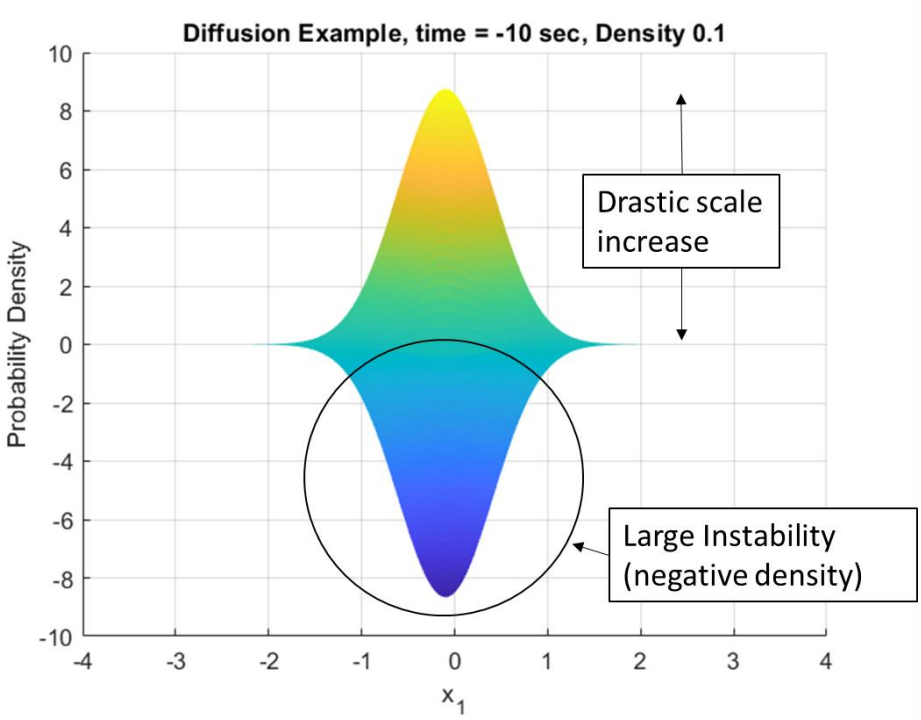

Figure 14. Reverse Diffusion Instability

Instabilities with the addition of diffusion are due to non-time symmetry of the diffusive process. Because the diffusion cannot be applied as a time reversible system, it is omitted from the future examples used to predict initial condition uncertainty.

3.3 PROBABILITY DENSITY ANALYSIS, DIFFERENCE GRID, AND MONTE CARLO SIMULATIONS

The power and applicability of a PDF is discussed in Section 2.2. A PDF provides engineers and scientists with a means for characterizing the uncertainty about a parameter. High order statistical moments can be derived from PDFs as well as discrete probabilities from select samples. In the simple example described in Section 3.2, a Gaussian distribution is observed and is characterized by the Gaussian function in Table 1. It is convenient to specify the probability density as a function, allowing for analytic solutions of statistical moments and probability metrics along values of interest. In real world application, these functions are generally best fits to actual data, and in the case of a joint transient system there may not be a

function that describes the output PD. Fortunately, the function is not necessary to derive the higher order statistical moments if the PD data is available for a system. Referring to the simple case presented in the previous section, assuming the mean is zero and standard deviation is 0.2628 the Gaussian distribution for time zero is defined by $f_X(x) = \frac{1}{0.2628\sqrt{2\pi}} exp\left[-\frac{1}{2}\left(\frac{x}{0.2628}\right)^2\right]$.

Derivation of central statistical moments from a function may be leveraged to define the shape and location of a PDF [52]. This is generally done using the first four moments known as mean, variance, skewness, and kurtosis; all of which may be derived from a PDF using Equation (44) where k is the moment number and E is the expectation.

$$E[X^k] = \int_{-\infty}^{\infty} x^k \ f_X(x)dx \tag{44}$$

Although useful in providing PDF metrics, more applicable to this research is the ability to find probability of variable occurrence given a single instance or span of instances. The probability of occurrence is simply the area under the probability density curve, or the integral over the variables span of interest. When a function is provided an analytical solution is available to determine probability of occurrence using Equation (45).

$$P(x_1 < X < x_2) = \int_{x_1}^{x_2} f_X(x)dx \tag{45}$$

In the case where no function is defined, such as a response PD (p_i) computed by the FP equation, a numerical approach is taken to calculate the area. This numerical approach is a simple sampling method over the defined grid multiplying the grid span by the density to obtain a probability over defined points of interest. Equation (46) defines this sampling technique.

$$P(x_1 < X < x_2) = \sum_{i=x_1}^{x_2} p_i dx_i \tag{46}$$

To prove out the sampling technique the Gaussian distribution from above is used. Analytically, this is solved by taking the integral over a span that is defined from $x_1 = 0$ to $x_2 = 1$, where it is known that the probability is approximately fifty percent. An analytical solution for a Gaussian distribution is no trivial task [53]. Fortunately, functions exist for solving this integral and were leveraged to compare the sampling technique in this 1-dimensional example. Errors between the analytical solution and the sampling technique are shown to be on the order of 10^{-8} where dx_i is set to 0.001. Node spacing between each one of the sampled points is critical for solution accuracy. Not only does the grid spacing affect the sampling technique for determining probability of occurrence but is also essential for stability and convergence of the numerical solution for the FP equation using the ADI method.

The approximate solution set provided by the ADI method obtains an order of accuracy relative to the finite difference size along each direction and the time step. Ideally, small time steps and fine differences are leveraged to minimize computational errors. As system complexity is increased, the sheer number of computations may well outweigh capability of memory storage or run time feasibility. On the other hand, coarse differences and time steps contribute to instabilities within the ADI method. Instability and large computational errors then contribute to unrealistic results that violate PD assumptions. For the 1D case used in Section 3.2, coarse grid sizes and time steps are depicted in Figure 15 to demonstrate instabilities observed in the PD. Decreasing grid size and time step eventually results in a stable and accurate depiction of the PD at the cost of computations and run time [54]. As system

complexity increases, the need for finer step sizes is required to generate stable densities. For systems with additional dimensions or nonlinearities, these instabilities can become significantly worse as depicted in Figure 16.

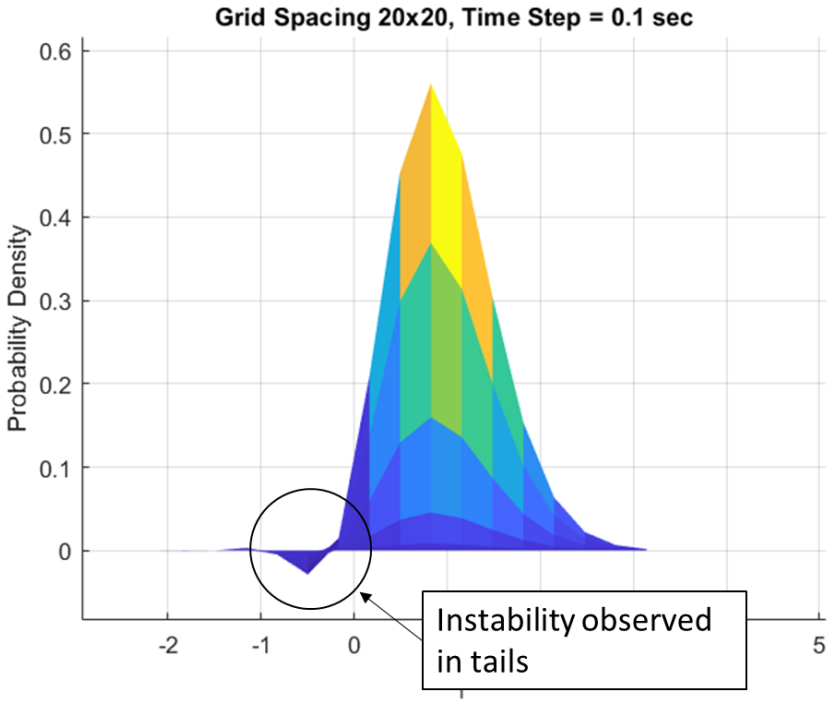

Figure 15. Grid Spacing Instability

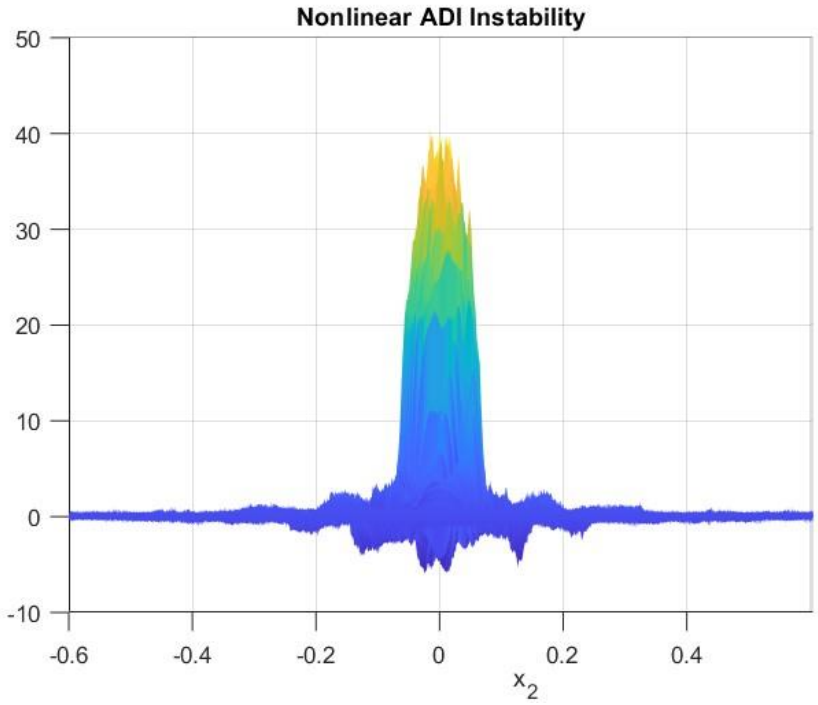

Figure 16. Nonlinear Grid Spacing Instability

3.4 SOFTWARE AND FUNCTIONS

Matrix Laboratory (MATLAB®) is a computer coding platform used academically and industrially by engineers and scientists as a highly robust data analysis tool [5]. MATLAB® is the primary tool used in this research for computing results and plotting data depicted in this report. Example MATLAB® scripts created in support of this research are illustrated in Appendix A.

Computations were run on an AMD Ryzen 7 4800H 2.90 GHz processor with Radeon Graphics and 8.0 GB of RAM. The MATLAB® version utilized is R2022a. General functions used from the MATLAB® toolkit include: ODE45, reshape, meshgrid, mvnpdf, and sparse.

CHAPTER 4

ACADEMIC EXAMPLE PROBLEMS

Chapter 4 depicts the use of a FP equation on both forward and reverse academic dynamic problems including a linear oscillator and a nonlinear trim solution. The FP equation is solved numerically for each example using an ADI method. Monte Carlo Simulations are employed as a secondary method for solving these problem sets, providing a benchmark for result comparison. MCS stop criteria is presented, showing the number of iterations necessary to obtain convergence of output results.

4.1 LINEAR OSCILLATOR

A 2-dimensional linear oscillator may be characterized mathematically by Equation (47) [45]. This differential equation can be represented by a set of ordinary differential equations depicted in Equation (48). Assessing this problem deterministically with input conditions equal to 5 for both states, Figure 17 depicts the system motion for 10 seconds.

$$\ddot{x} = -2\xi\omega\dot{x} - \omega^2 x \tag{47}$$

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}x_2\\ -2\xi\omega x_2 - \omega^2 x_1\end{bmatrix} \tag{48}$$

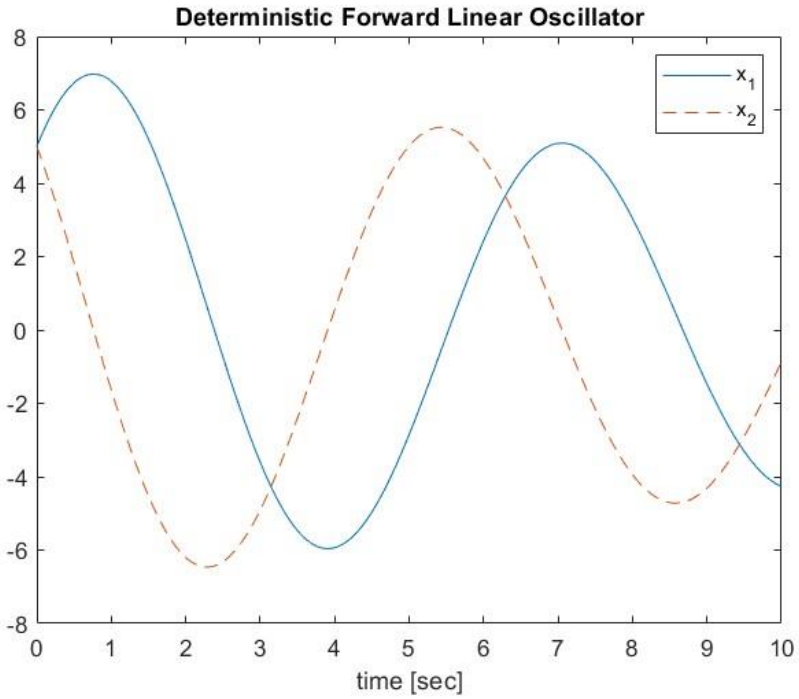

Figure 17. Linear Oscillator Deterministic Forward Motion

Applying the results as initial conditions and reversing the dynamic equations of each state results in reverse time motion depicted in Figure 18. Since the conditions return to their original state after 10 seconds, the system is confirmed to be time symmetric.

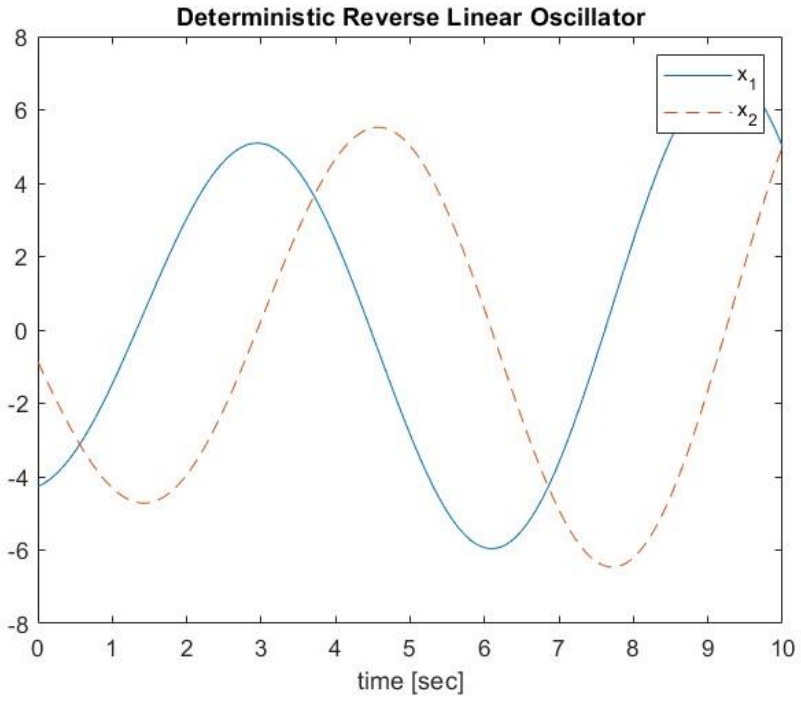

Figure 18.Linear Oscillator Deterministic Reverse Motion

Uncertainty is then applied to the system. As described previously, the FP equation drift matrix is replaced by the state vector of Equation (48), resulting in a PDE capable of computing the transient probability density at each state. Similarly to Equation (29), the FP equation yielded for this example becomes:

$$\frac{\partial}{\partial t}p(\boldsymbol{x},t|\boldsymbol{x}_0,t_0) = -x_2\frac{\partial}{\partial x_1}[p(\boldsymbol{x},t|\boldsymbol{x}_0,t_0)] + \frac{\partial}{\partial x_2}[(2\xi\omega x_2+\omega^2 x_1)p(\boldsymbol{x},t|\boldsymbol{x}_0,t_0)] \quad (49)$$

For this example, it is assumed that the initial conditions for both states follow a Gaussian distribution with a mean of 5 and standard deviation of 1. Grid sizes that ensure stability are 500 for each direction. Time step is set to 0.005 seconds. Solving this problem forward in time results in the probability densities depicted in Figures 19-21.

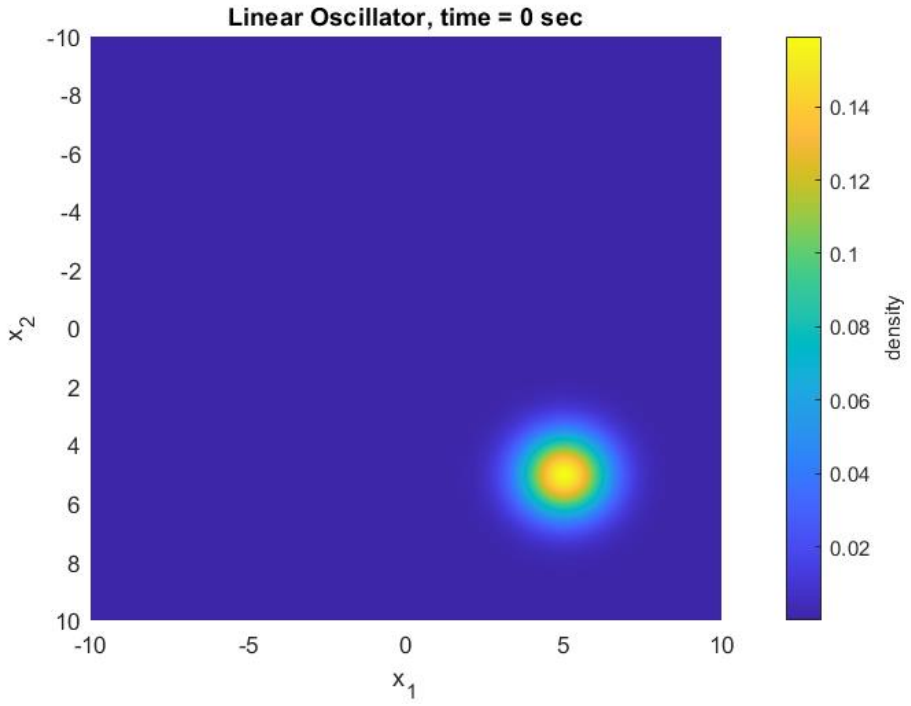

Figure 19. Linear Oscillator Forward PD at Time 0 sec

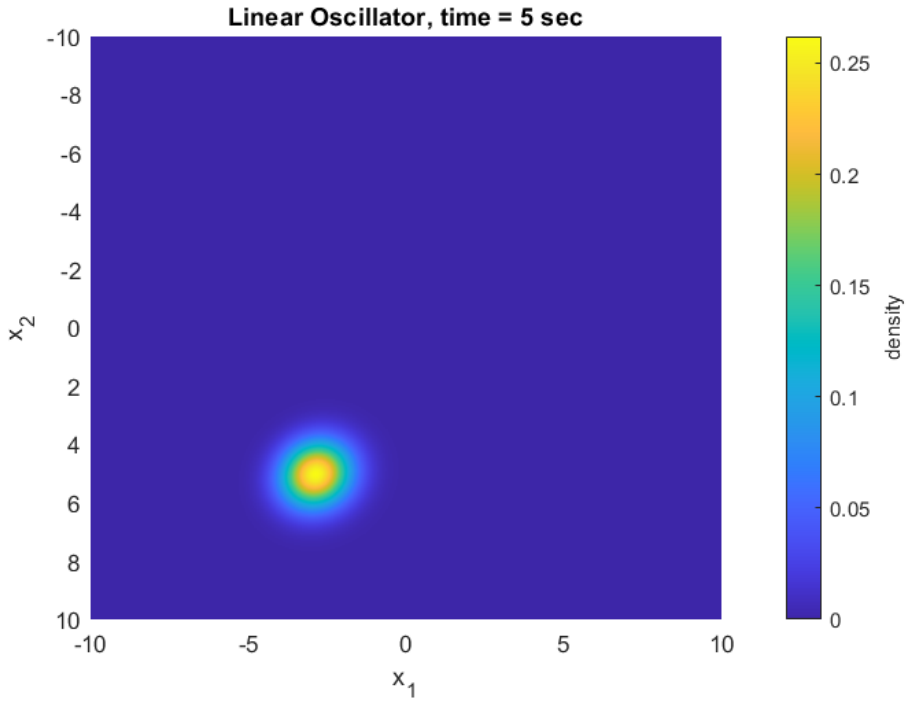

Figure 20. Linear Oscillator Forward PD at Time 5 sec

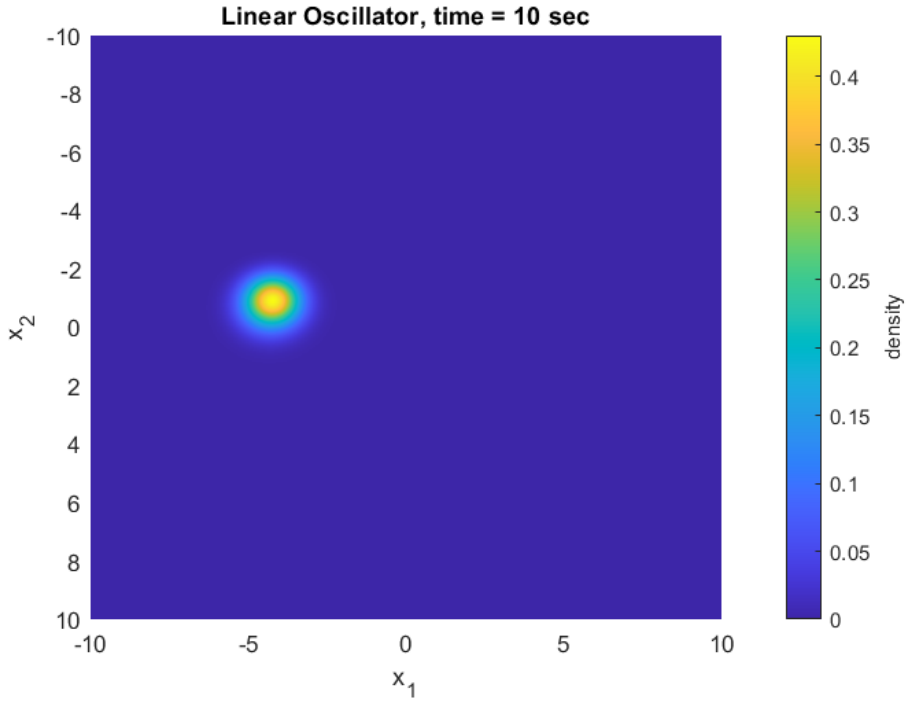

Figure 21. Linear Oscillator Forward PD at Time 10 sec

Where the density magnitudes at 10 seconds for each state are illustrated in Figures 22-23. MCS results confirm the validity of FP equation results for this system in Figures 24-26. MCS convergence criteria is depicted in Figure 27.

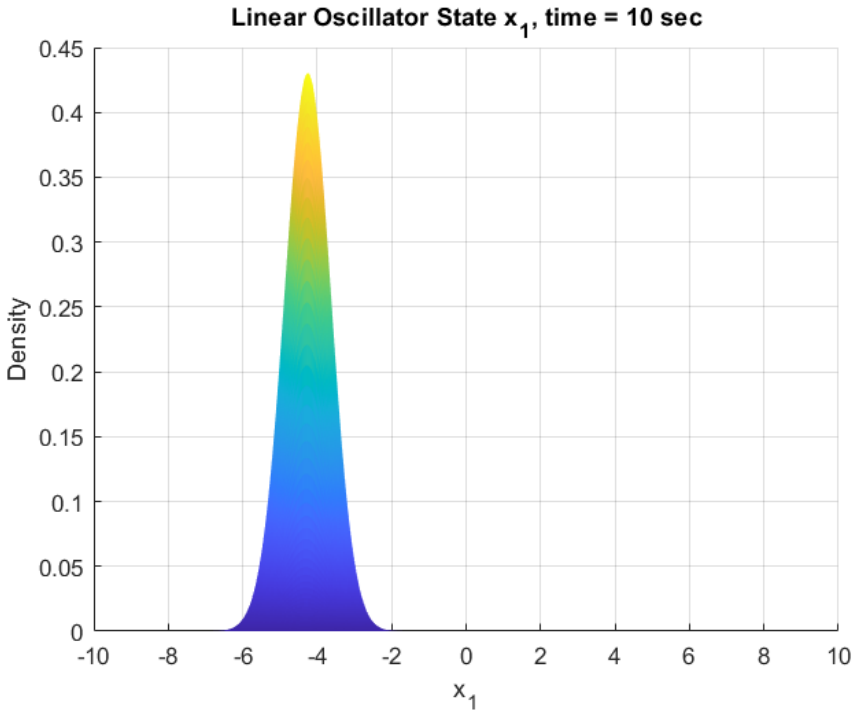

Figure 22. Linear Oscillator First State Density

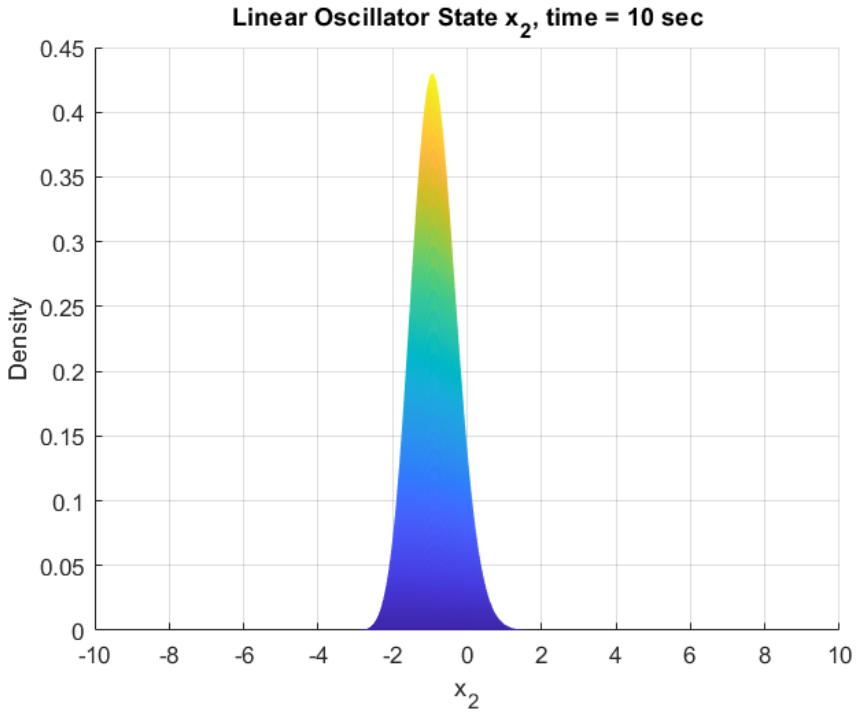

Figure 23. Linear Oscillator Second State Density

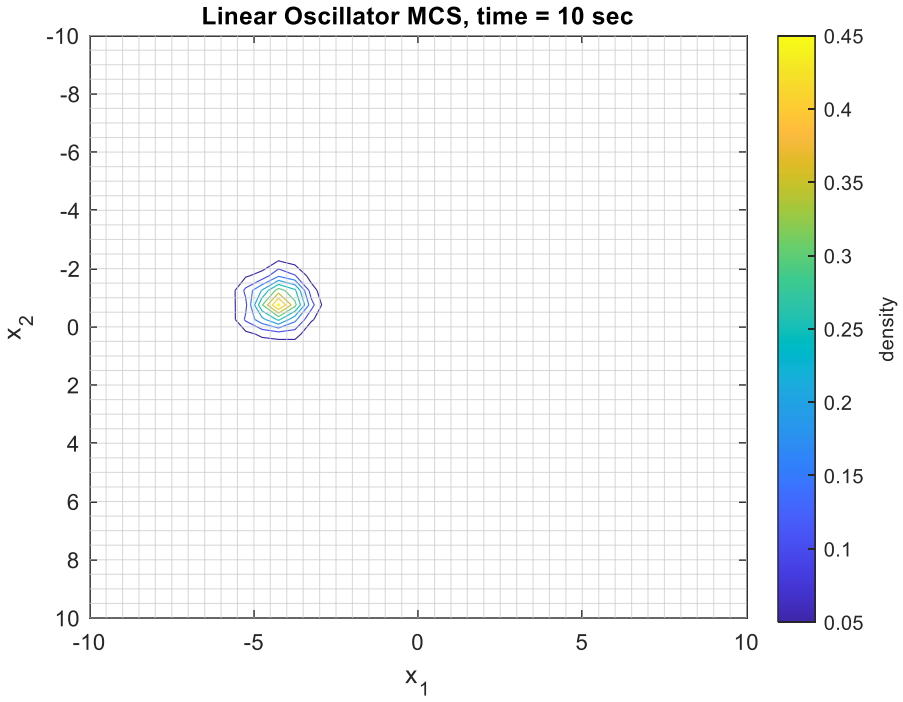

Figure 24. Linear Oscillator MCS PD

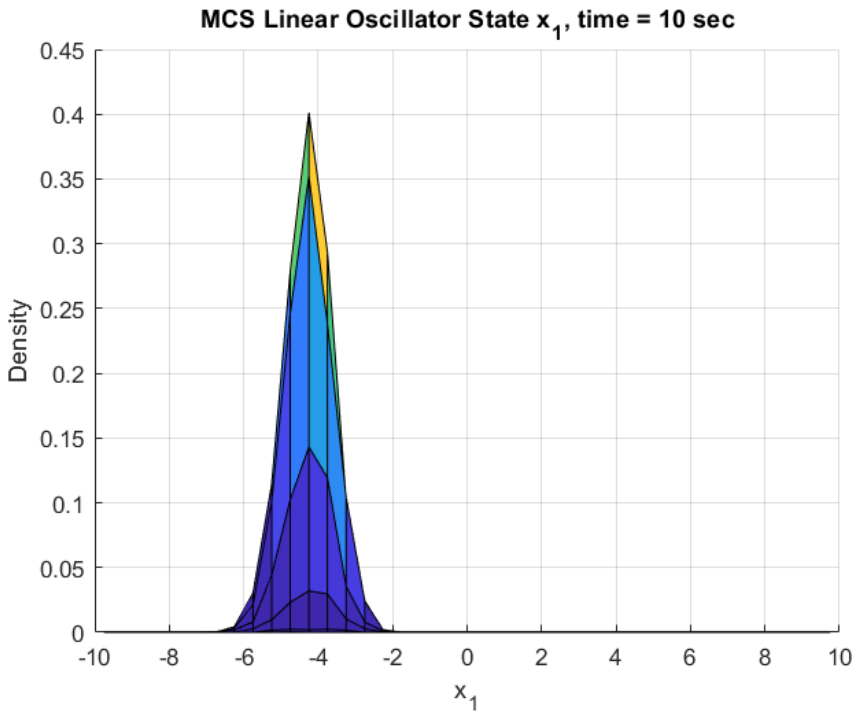

Figure 25. Linear Oscillator MCS First State Density

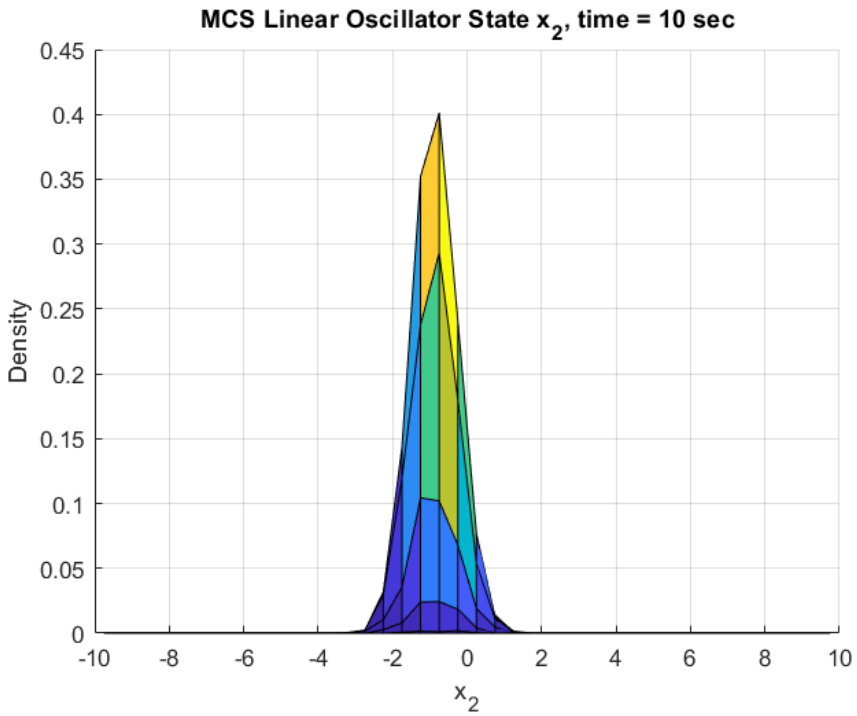

Figure 26. Linear Oscillator MCS Second State Density

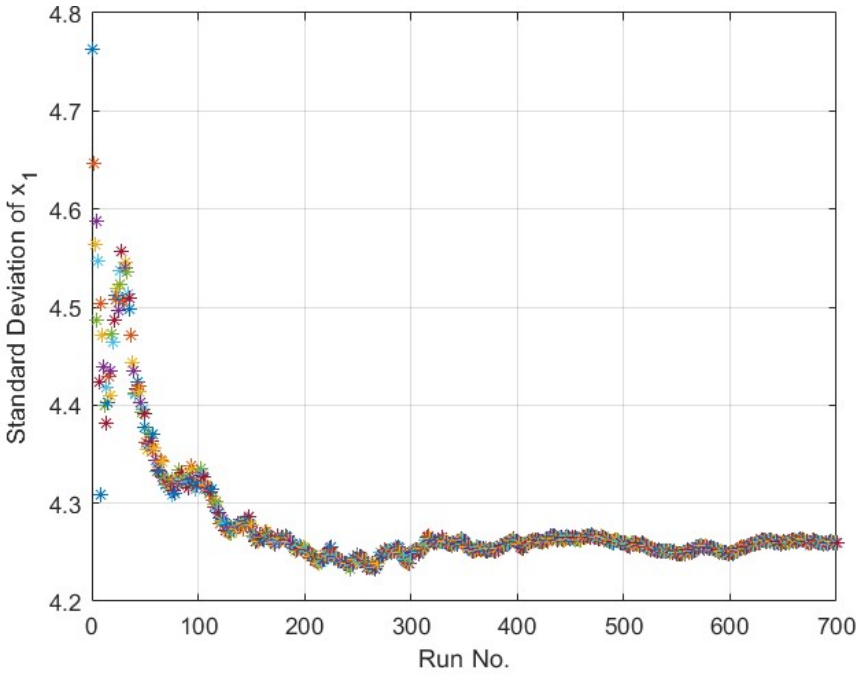

Figure 27. Linear Oscillator MCS Convergence

Leveraging the time symmetry of the system, a negative sign is applied at the drift vector to propagate the system backwards in time. The response observed at 10 seconds in Figure 21 is used as the initial conditions for the backwards FP equation. The backward results are depicted in Figures 28-29. The errors observed between the true initial conditions and the predicted initial conditions are depicted in Figure 30.

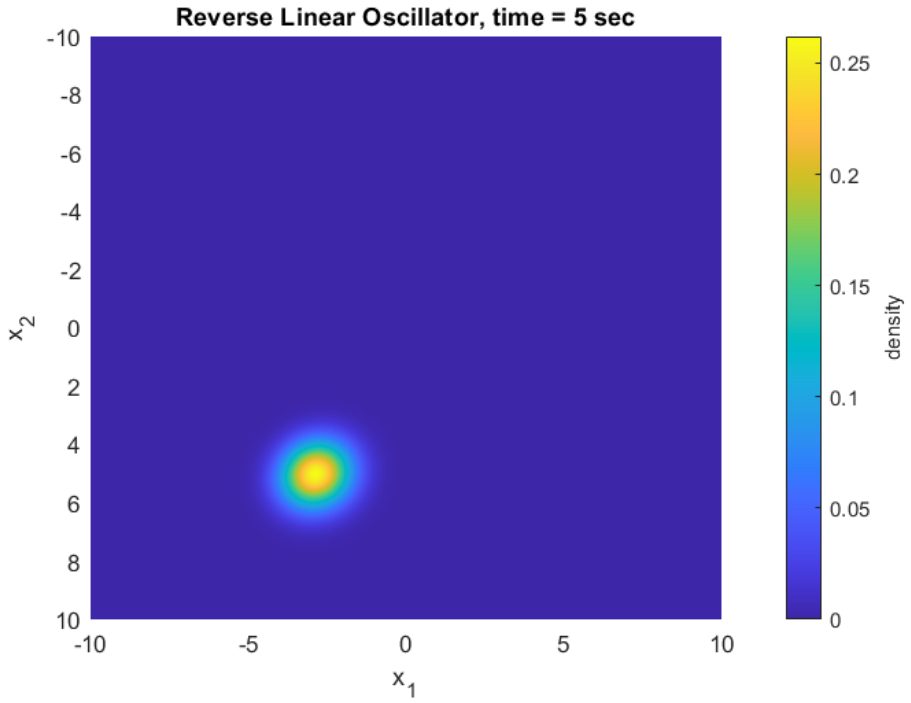

Figure 28. Linear Oscillator Reverse PD at Time 5 sec

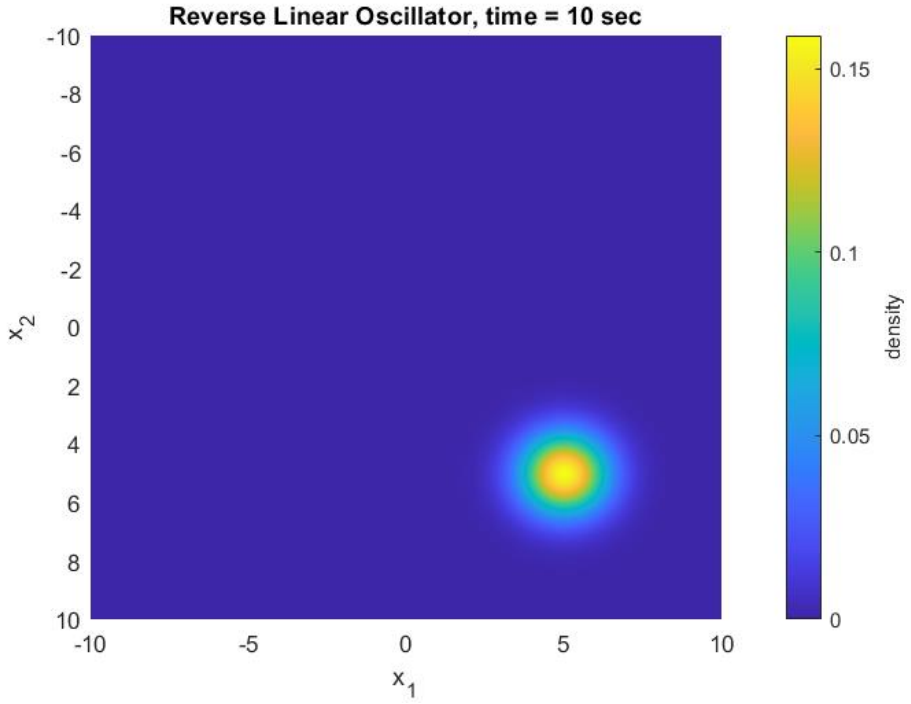

Figure 29. Linear Oscillator Reverse PD at Time 10 sec

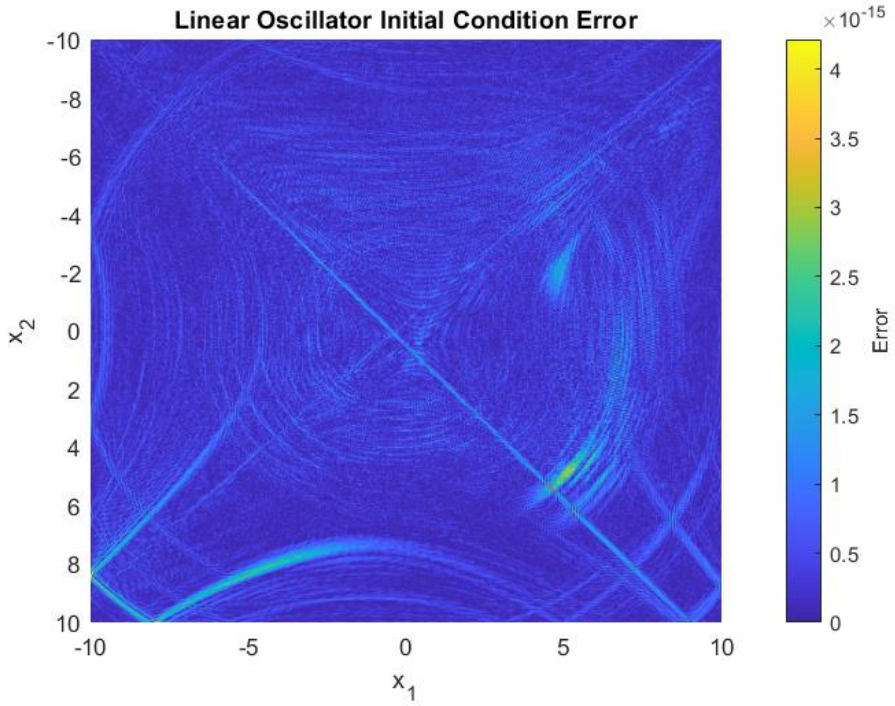

Figure 30. Linear Oscillator Error Between True and Reverse Predicted Initial Conditions

4.2 NONLINEAR TRIM

A 2-dimensional nonlinear oscillating system presented in [51] was demonstrated as a reversible system in Section 3.1 via Equation (41) with deterministic motion illustrated in Figures 5-6. Leveraging the set of ODEs as the drift vector the FP equation is formulated as:

$$\begin{aligned}\frac{\partial}{\partial t}p(\boldsymbol{x},t|\boldsymbol{x}_0,t_0)&\\ &= -x_2\frac{\partial}{\partial x_1}[p(\boldsymbol{x},t|\boldsymbol{x}_0,t_0)]\\ &+\frac{\partial}{\partial x_2}[(5\mu_c(1-x_1^2)x_2\\ &+9\sin(x_1))p(\boldsymbol{x},t|\boldsymbol{x}_0,t_0)]\end{aligned} \tag{50}$$

The forward problem is numerically solved using the ADI method. The additional complexity introduced by the nonlinear system requires finer differences and time steps. The time step for this problem is set to 0.005 seconds with grids containing 990 nodes along each of the two state directions. Initial conditions for both states are Gaussian distributions with a standard deviation of 0.001 and means of 0 at x_1 and 0.4 at x_2.

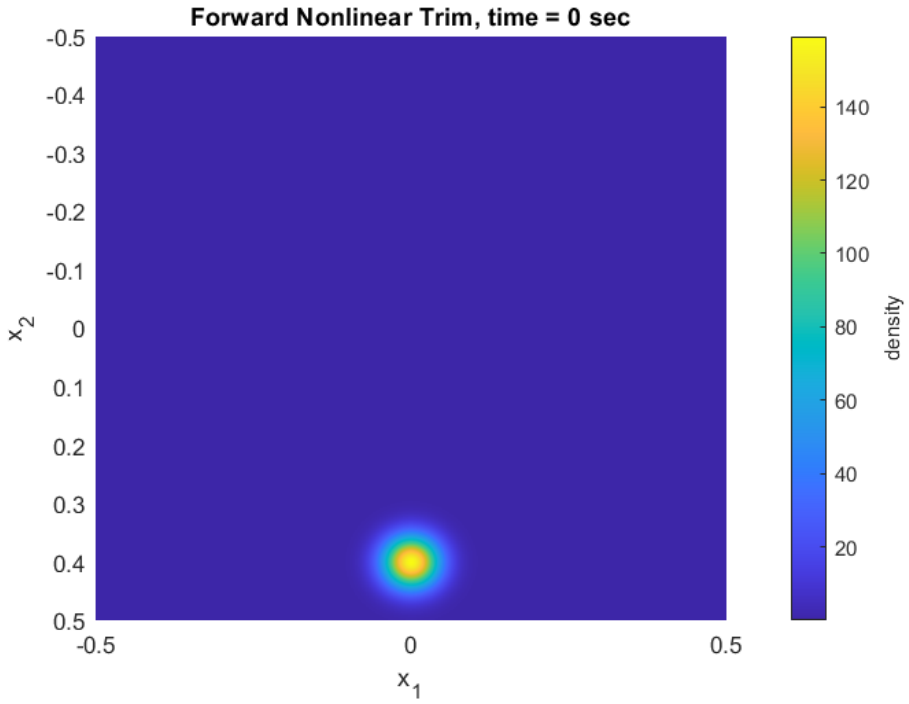

Figure 31. Nonlinear Trim Forward PD at Time 0 sec

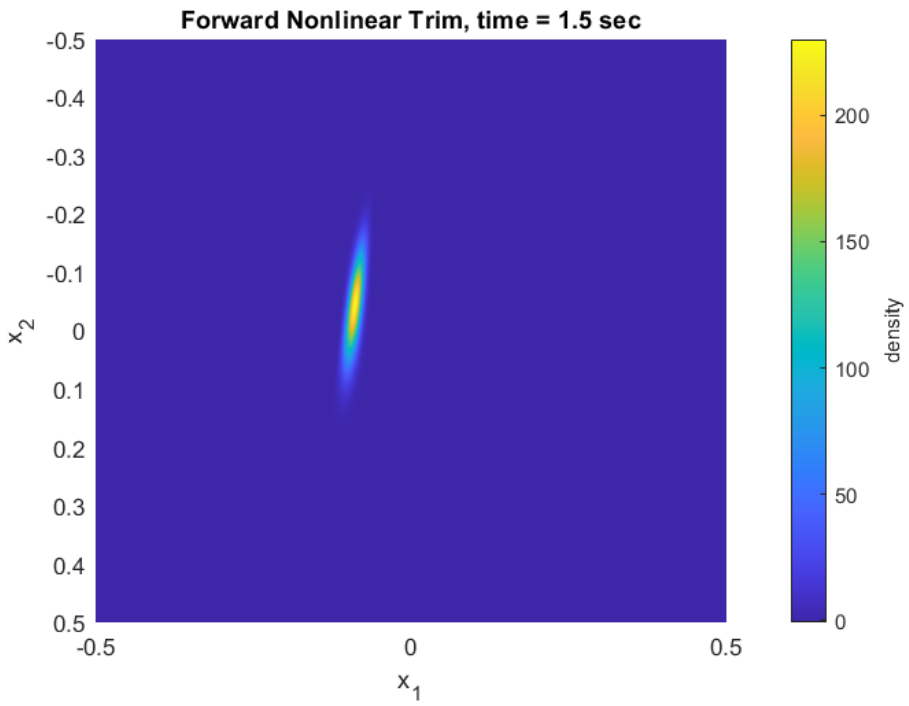

Figure 32. Nonlinear Trim Forward PD at Time 1.5 sec

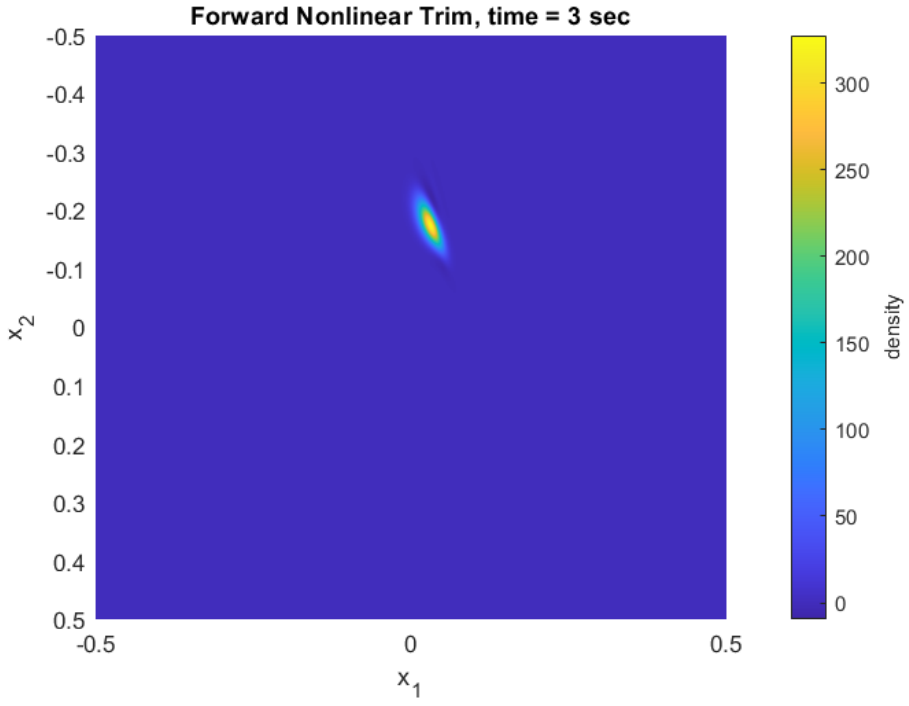

Figure 33. Nonlinear Trim Forward PD at Time 3 sec

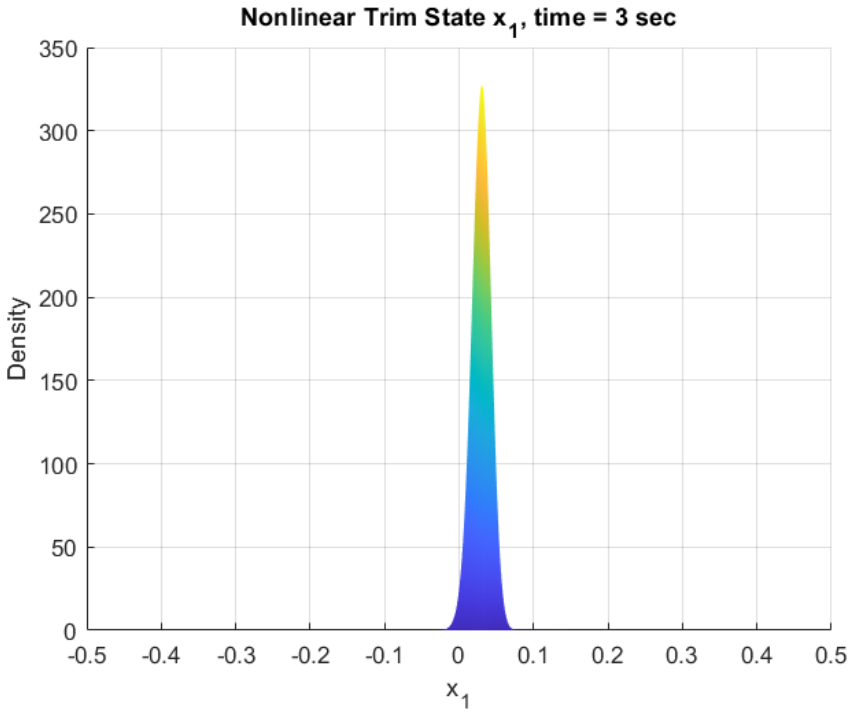

Figure 34. Nonlinear Trim First State Density

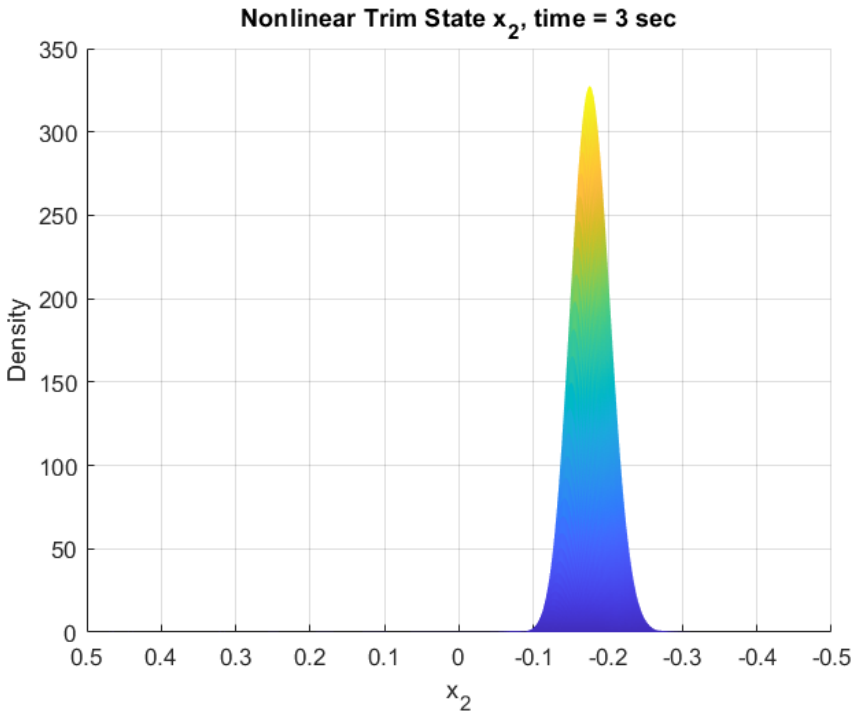

Figure 35. Nonlinear Trim Second State Density

MCS is leveraged to validate the computed FP response. Figures 36-38 depict comparable results to the probability densities computed via the ADI method. Convergence criteria for MCS is illustrated in Figure 39. The reverse nonlinear trim distributions are depicted in Figures 40 and 41 using the response from Figure 33 as the initial conditions for the reverse time propagation. Figure 42 concludes this section with a surface chart of the small errors observed between the true initial conditions and the predicted initial conditions using reverse dynamics within a FP equation.

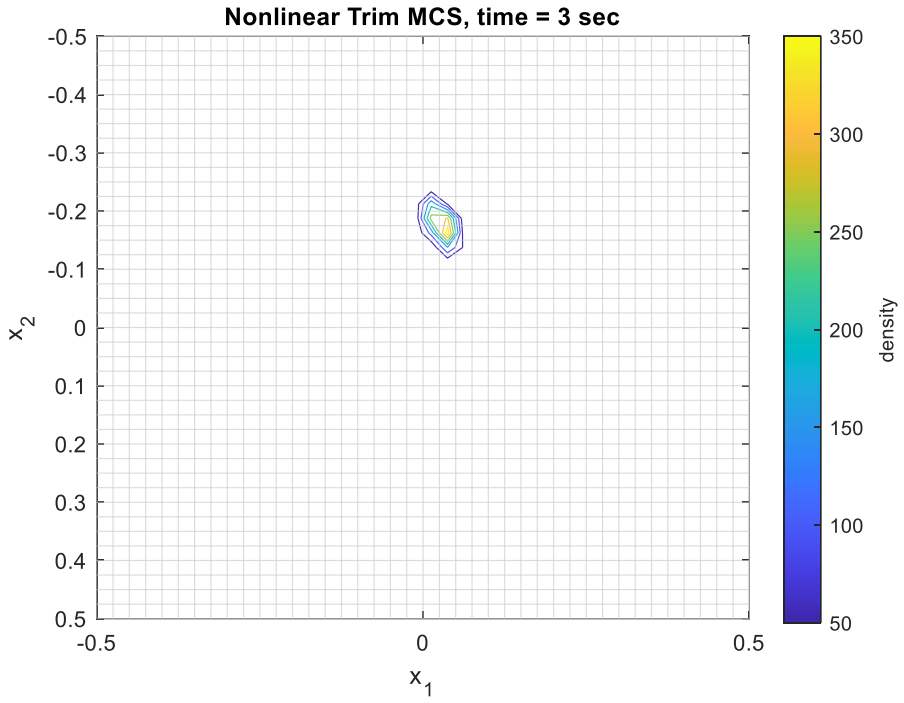

Figure 36. Nonlinear Trim MCS PD

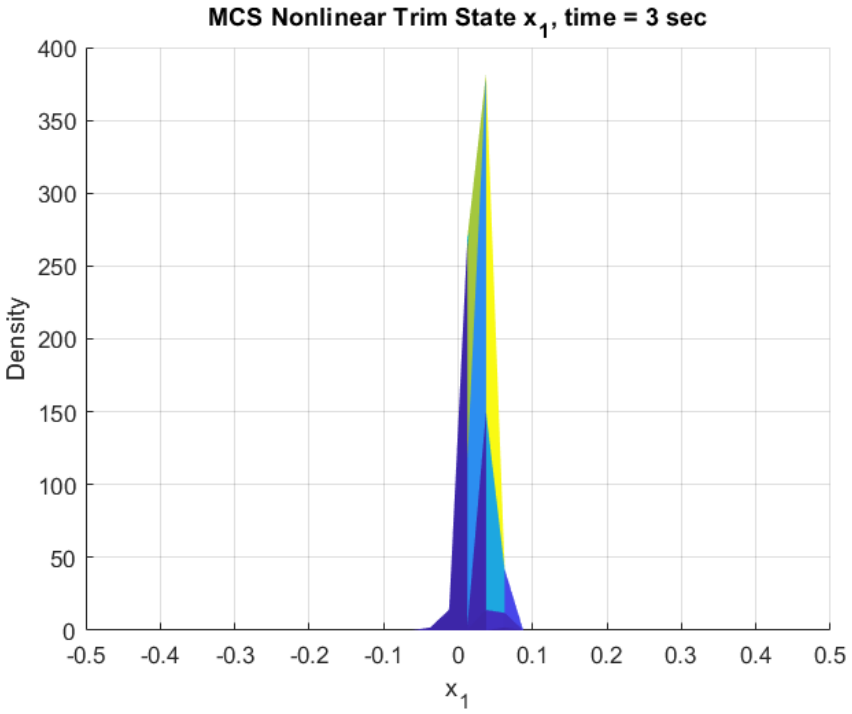

Figure 37. Nonlinear Trim MCS First State Density

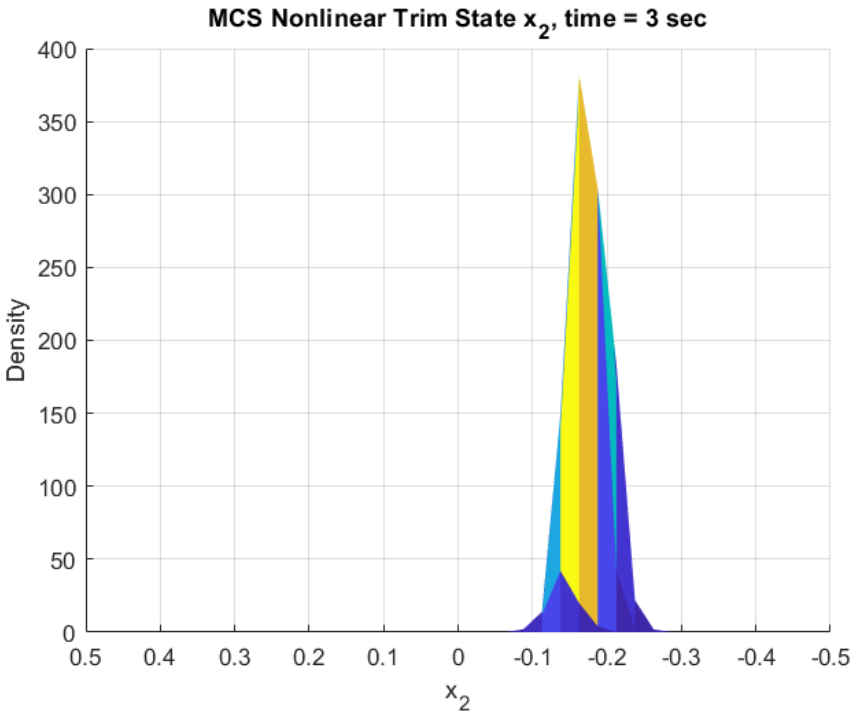

Figure 38. Nonlinear Trim MCS Second State Density

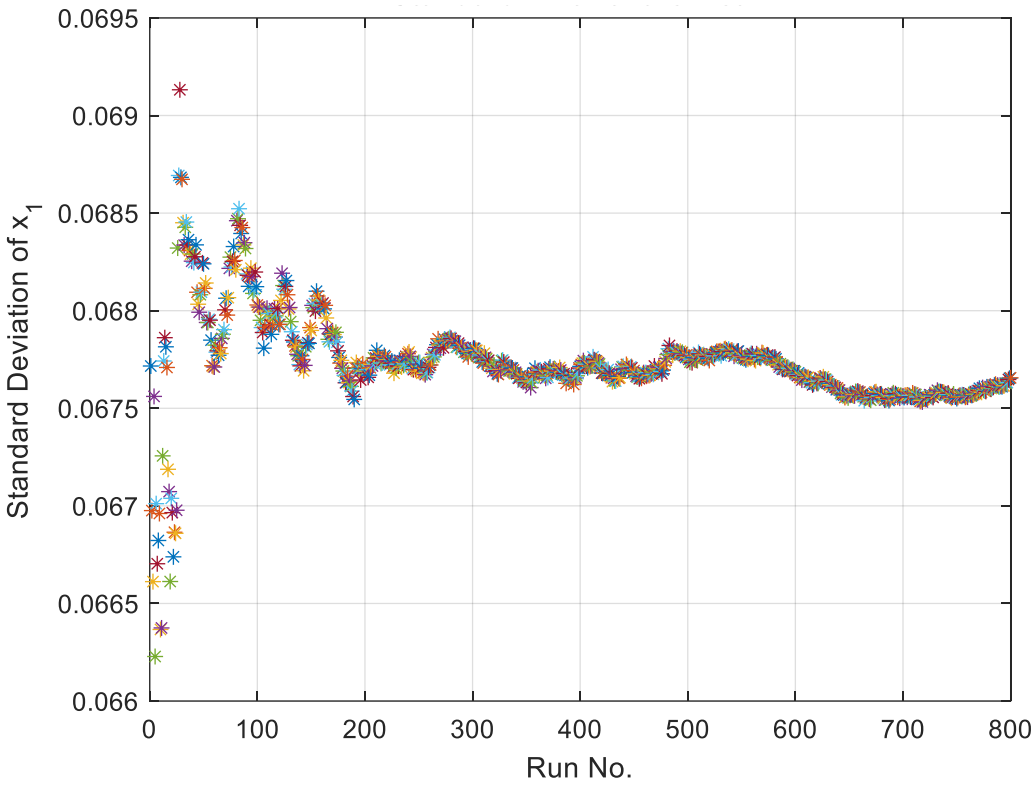

Figure 39. Nonlinear Trim MCS Convergence

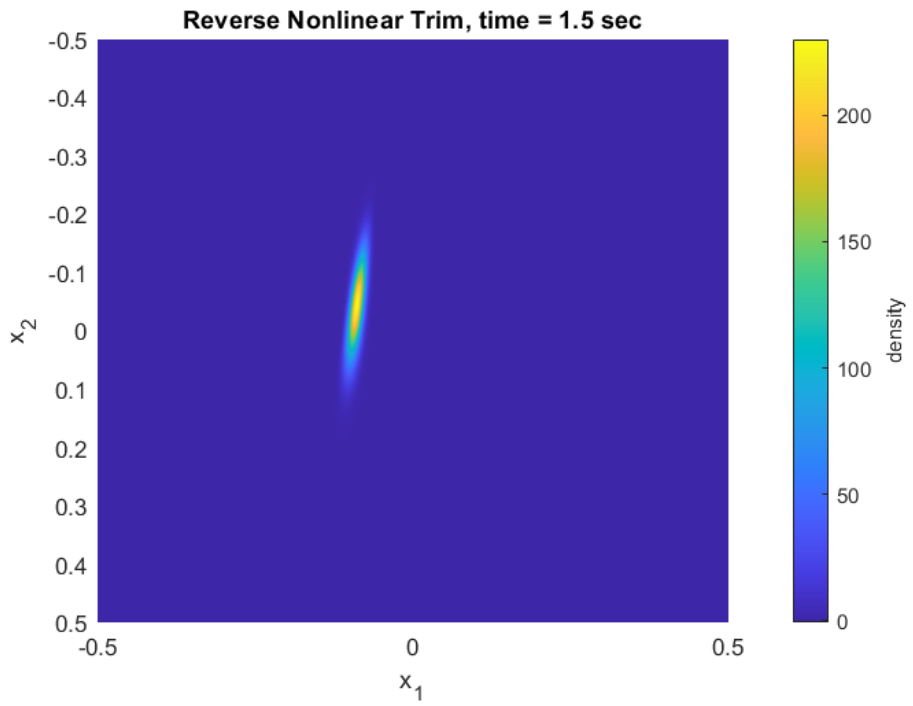

Figure 40. Nonlinear Trim Reverse PD at Time 1.5 sec

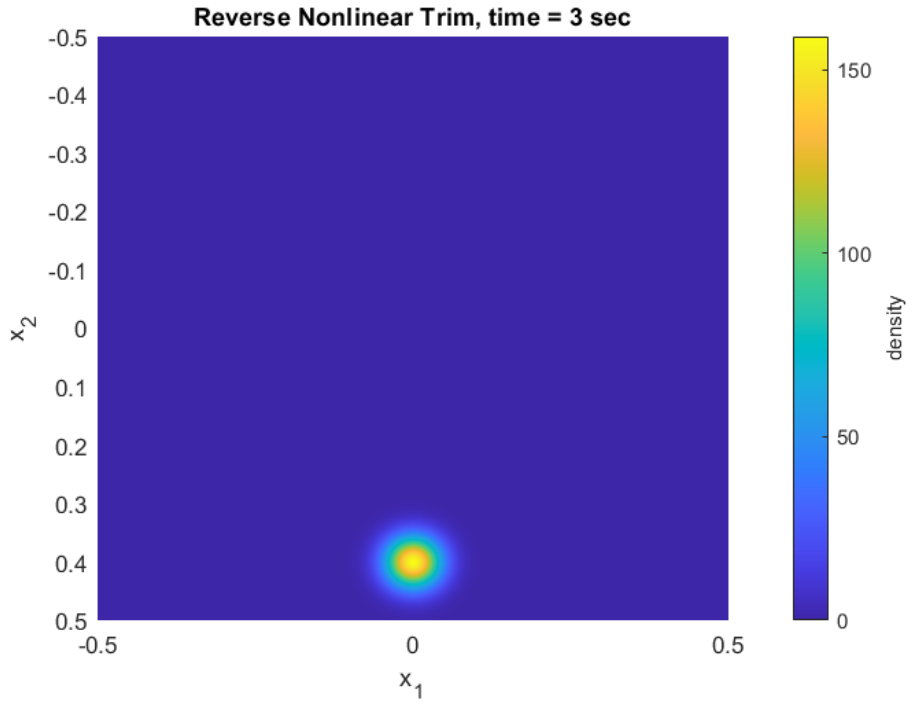

Figure 41. Nonlinear Trim Reverse PD at Time 3 sec

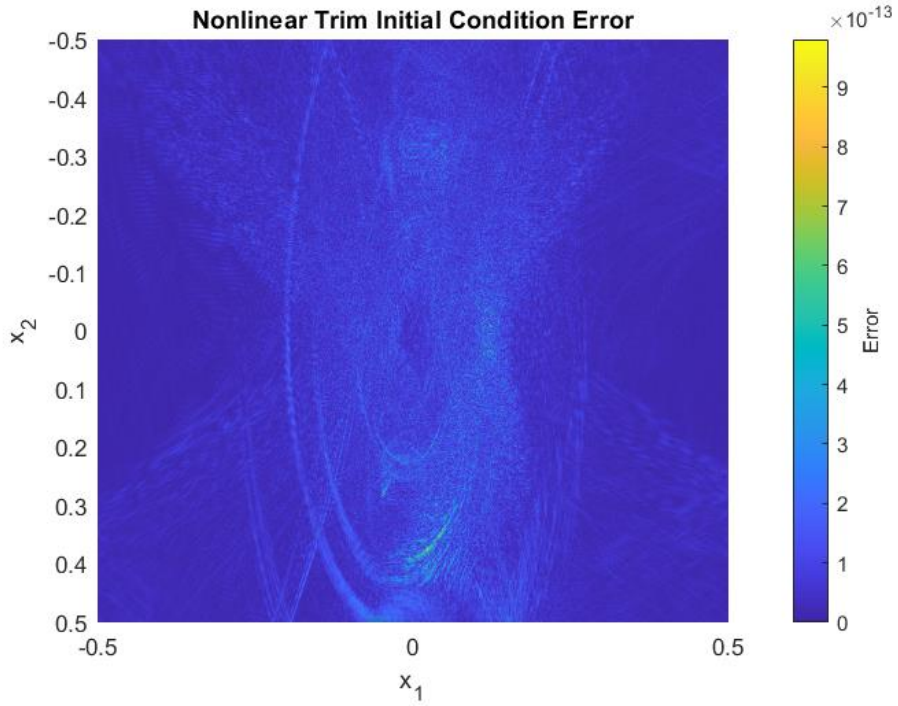

Figure 42. Nonlinear Trim Error Between True and Reverse Predicted Initial Conditions

CHAPTER 5

REENTRY VEHICLE

Chapter 5 illustrates the novel approach of leveraging the FP equation to propagate uncertainty in reverse reentry dynamics, ultimately predicting initial conditions of a reentry vehicle given a desired response. The chapter begins with an overview of the reentry equations of motion that can be leveraged by for any planet and ballistic vehicle. Example problems for an Earth and Mars reentry are then presented, culminating in a case for a Mars reentry where uncertainty bounds must be met for a parachute deployment. Initial conditions are computed for mission planning and design to ensure reentry results in the desired response.

5.1 BALLISTIC REENTRY EQUATIONS OF MOTION

Simplified reentry dynamics are defined in Chapter 12 of [55] treating the body as a point mass resulting in a three degree-of-freedom system. The reduced order characterization provides a means for using the FP equation to assess reentry dynamics throughout flight to some terminating condition. Time derivatives of the vehicles position vector are provided in Equations (51)-(53) establishing the kinematic equations of motion relative to a rotating body, where x_1 is longitude in radians, x_2 is latitude in radians, and x_3 is the radius in meters from the center of the rotating body to the reentry vehicle.

$$\dot{x}_1 = \frac{x_4 * \cos(x_5) \sin(x_6)}{x_3 \cos(x_2)} \tag{51}$$

$$\dot{x}_2 = \frac{x_4 * \cos(x_5) \cos(x_6)}{x_3} \tag{52}$$

$$\dot{x}_3 = x_4 * \sin(x_5) \tag{53}$$

For a rotating spherical body with angular velocity ω and gravitational constant g, with vehicle parameters of drag D and total mass m, the time rate of change for the relative velocity vector is defined by Equations (54)-(56), where x_4 is velocity in meters per second, x_5 is flight path angle in radians, and x_6 is heading in radians.

$$\begin{aligned}\dot{x}_4 = -gsin(x_5) + g\cos(x_6)\cos(x_5) + \frac{D}{m} \\ + \omega^2 x_3 \cos(x_2)[\sin(x_5)\cos(x_2) \\ - \cos(x_5)\cos(x_6)\sin(x_2)]\end{aligned} \tag{54}$$

$$\begin{aligned}&\dot{x}_5 \\ &= \left(\frac{x_4}{x_3} - \frac{g}{x_4}\right)\cos(x_5) - \frac{g\cos(x_6)\sin(x_5)}{x_4} + 2\omega\sin(x_6)\cos(x_2) \\ &+ \omega^2 x_3 \cos(x_2)\frac{[\cos(x_5)\cos(x_2) + \sin(x_5)\cos(x_6)\sin(x_2)]}{x_4}\end{aligned} \tag{55}$$

$$\begin{aligned}\dot{x}_6 = \frac{x_4\sin(x_6)\tan(x_2)\cos(x_5)}{x_3} - \frac{g\sin(x_6)}{x_4} \\ - 2\omega[\tan(x_5)\cos(x_6)\cos(x_2) - \sin(x_2)] \\ + \frac{\omega^2 x_3 \sin(x_6)\sin(x_2)\cos(x_2)}{x_4\cos(x_5)}\end{aligned} \tag{56}$$

5.2 EARTH REENTY

Deterministic inputs for the above equations of motion are first assessed given Earth conditions. Table 2 depicts the Earth input parameters for the model [55]. An Earth atmosphere

model based on 1976 U.S. Standard Atmosphere is leveraged for pressure and density parameters at varying altitudes. Figure 43 depicts the density change in log scale from sea-level to an Altitude of 300 km. Atmospheric measurements are used in tandem with body parameters specified in Table 3 to compute drag forces on the body via the drag equation $D = -\frac{1}{2}\rho x_4^2 S_{ref} C_D$, where ρ is density in kilograms per cubic meter, C_D is a Mach dependent drag coefficient for a generic reentry body specified in section 12.4.1 of [55], and S_{ref} is the body reference area.

Table 2. Earth Parameters

Standard Gravitational Parameter (μ)	3.986e14	m^3/s^2
Mean Radius (R)	6378140	m
Angular Velocity (ω)	7.292116e-5	rad/s
Acceleration Due to Gravity (g)	9.81	m/s^2

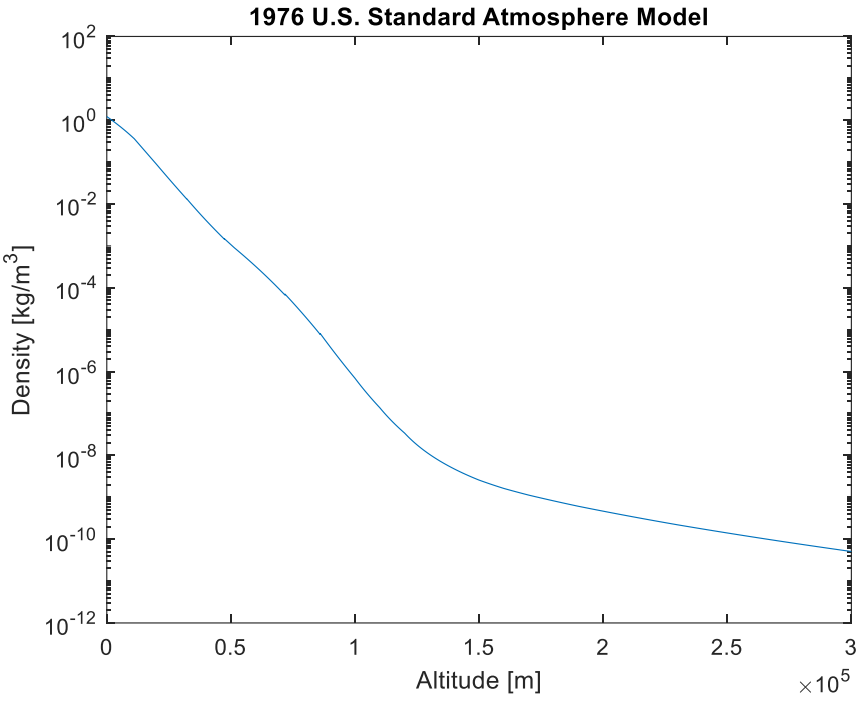

Figure 43. 1976 U.S. Standard Atmosphere Altitude vs. Density

Table 3. Reentry Body Parameters

Reference Area (S_{ref})	4 m^2
Mass (*m*)	350 kg

The reentry problem is first assessed deterministically. The problem space is integrated using Matlabs® ODE45 Runge-Kutta function both forward and backward in time to ensure time symmetry of the reentry equations of motion. The simulation is run for the first 200 seconds of reentry given the initial conditions depicted in Table 4. Figures 44, 45, and 46 depict forward responses of the reentry vehicle.

Table 4. Earth Reentry Deterministic Initial Conditions

Longitude (x_1)	0	deg
Latitude (x_2)	0	deg
Radius (x_3)	6579.9e3	m
Velocity (x_4)	7589.30	m/s
Flight Path Angle (x_5)	-0.5468	deg
Heading (x_6)	45	deg

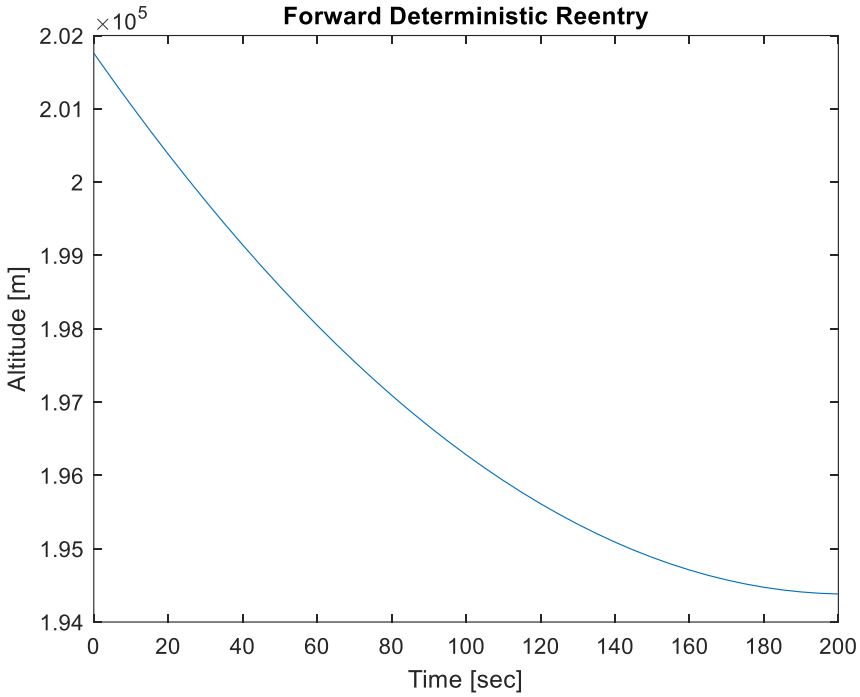

Figure 44. Forward Deterministic Earth Reentry Altitude

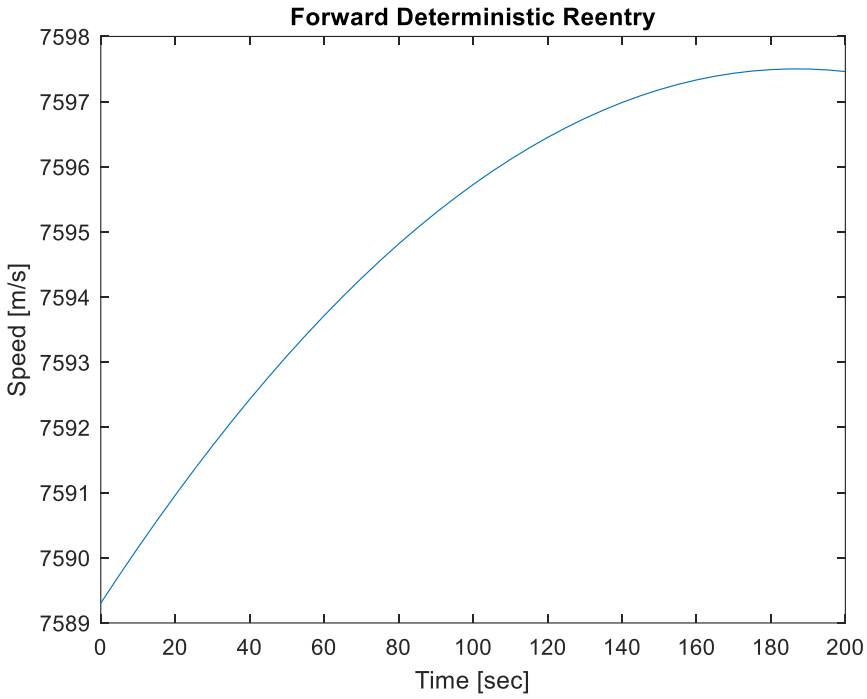

Figure 45. Forward Deterministic Earth Reentry Speed

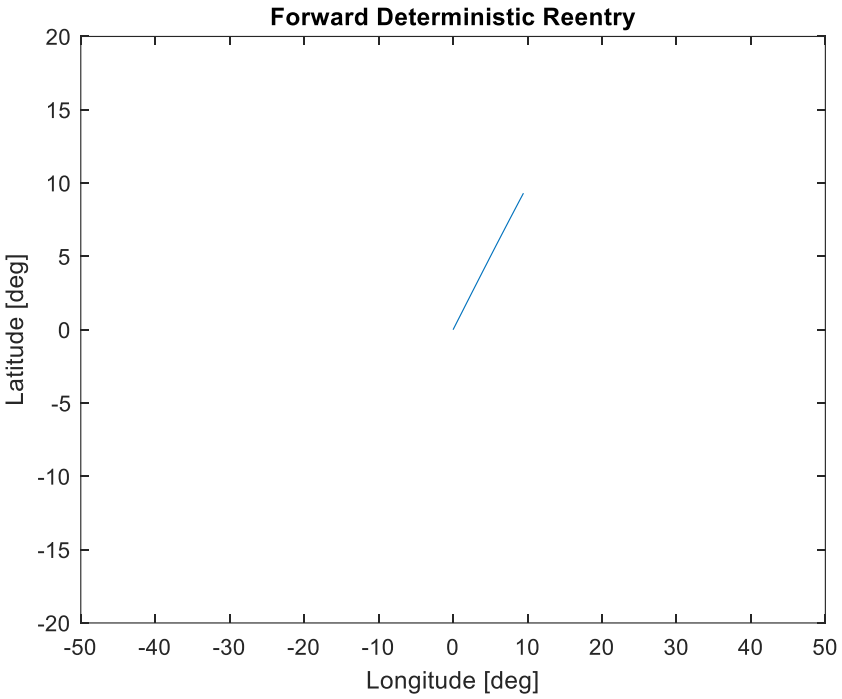

Figure 46. Forward Deterministic Earth Reentry Position

The reverse responses are computed similarly to the previous deterministic examples. The final response at 200 seconds is set as the initial condition and the equations of motion are reversed. Errors are indistinguishable when observing similar response plots for the reverse deterministic calculation. The errors for each state are on order with the integration tolerance (10^{-7}). The errors observed for each state are depicted in the figures below, demonstrating time symmetry of these reentry equations of motion.

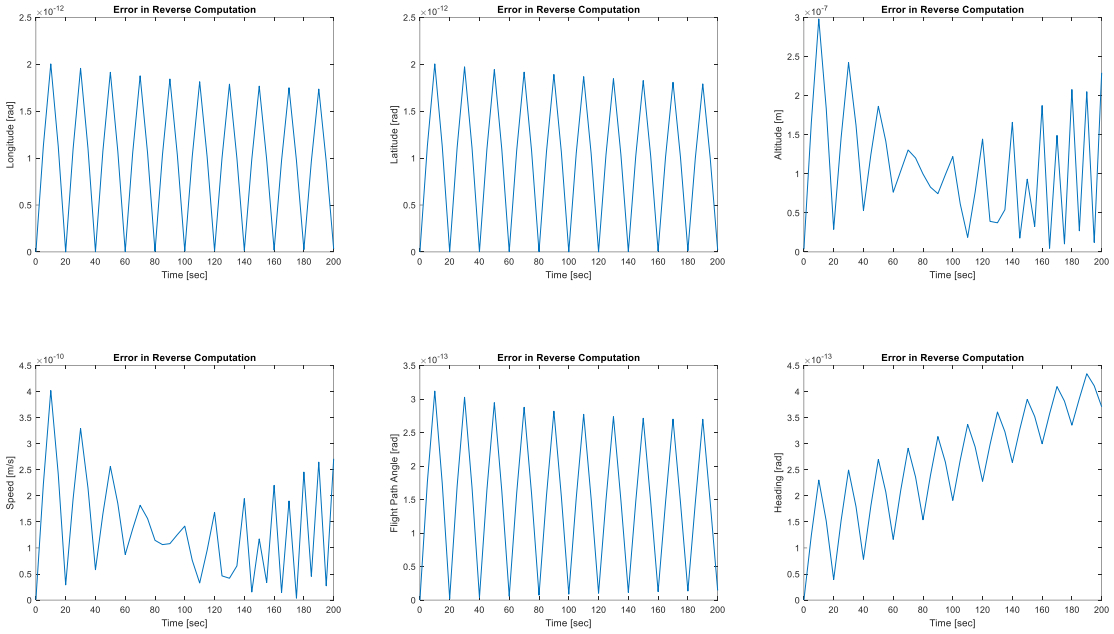

Figure 47. Reentry Deterministic Reverse Computation Errors

Uncertainty is then applied to the initial conditions. All initial conditions are treated as normal distributions with means about the deterministic values portrayed in Table 4. Standard deviations about latitude and longitude are set to approximately 3 degrees, 10 km for radius, 500 m/s for velocity, and 1 degree for flight path angle. The heading is held constant at 45 degrees. Standard deviations of 3 degrees for latitude and longitude equate to significant uncertainty within these positions. These uncertainties are purposefully overexaggerated for better visualization within resulting graphics. Monte Carlo simulations provide statistical data that are compared with Fokker-Planck results. Statistical outputs for each of the states depict high correlation between latitude and longitude. The remaining variables are weakly correlated to latitude and longitude for the variations provided above. The correlation of the variables is an important note for reducing dimensionality of the Fokker-Planck formulations discussed

later. A contour of the latitude and longitude is depicted in Figure 48. The variability surrounding the position due to uncertain initial conditions is obvious due to the overexaggerated uncertainties. Achieving convergence for each states mean standard error requires approximately 800 simulations. The high probability density region is in concurrence with the deterministic solution described above.

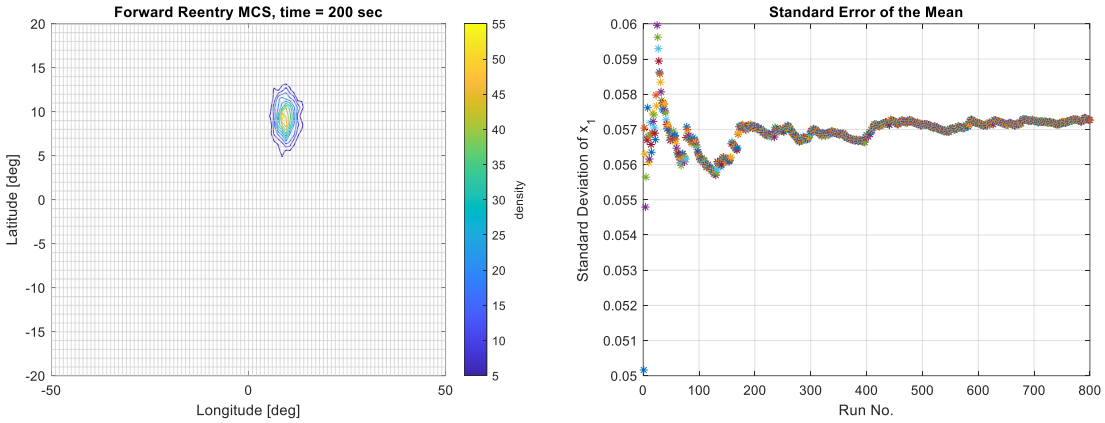

Figure 48. Forward Earth Reentry MCS

Before solving this problem set using the FP equation, additional results are presented which reinforce the correlation of each state variable to latitude and longitude. It is observed that relatively small uncertainties about the other variables have a minimal impact on the response distribution from Figure 48. Figure 49 depicts the latitude and longitude for the same system with increased standard deviations of 6 degrees about latitude and longitude, 100 km about radius, 1000 m/s about speed, and 3 degrees about the flight path angle. The larger uncertainties associated with the initial conditions here depict a much more even distribution in

the response (lower probability density) and requires a significant increase in the number of MCS to obtain convergence.

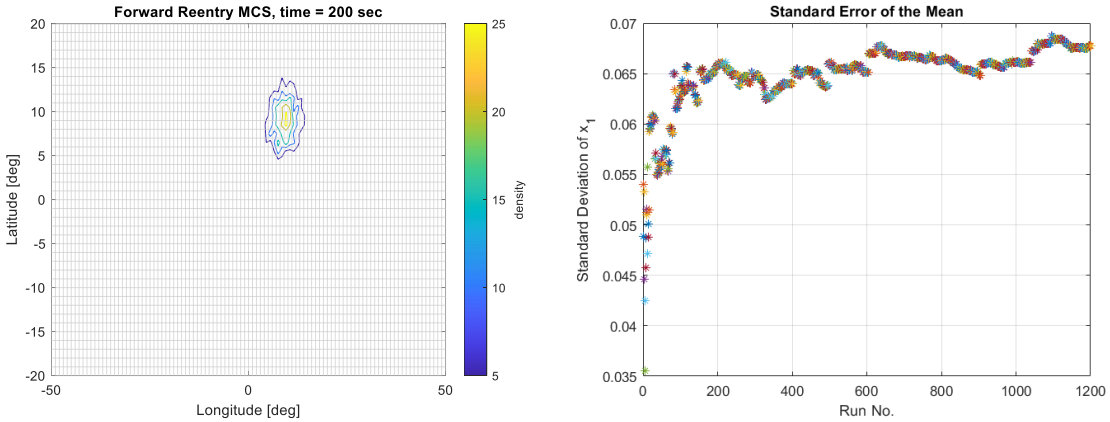

Figure 49. Forward Earth Reentry MCS, Increased Variability

Although the general position of the response distribution remains the same, the distribution overall trends toward uniformity as the variance of the other states approaches infinity. Additional MCS iterations are also required for convergence of the state means due to the increased variability.

Forward computation of the FP equation using the ADI method is depicted below. A 2-D case is first assessed where only the latitude and longitude are uncertain, resulting in an FP equation of the form:

$$\frac{\partial}{\partial t}p(\boldsymbol{x},t|\boldsymbol{x}_0,t_0)$$

$$= -\frac{\partial}{\partial x_1}\left[\frac{x_4\cos(x_5)\sin(x_6)}{x_3\cos(x_2)}p(\boldsymbol{x},t|\boldsymbol{x}_0,t_0)\right] \quad (57)$$

$$-\frac{\partial}{\partial x_2}\left[\frac{x_4\cos(x_5)\cos(x_6)}{x_3}p(\boldsymbol{x},t|\boldsymbol{x}_0,t_0)\right]$$

Grid sizes containing 500 nodes for each state is adequate to negate ADI instabilities. The uncertainty bounds are kept at 3 degrees for both latitude and longitude. All other variables have zero variability and begin at their mean values depicted in Table 4. The simulation is again run for 200 seconds of flight to compare with MCS results.

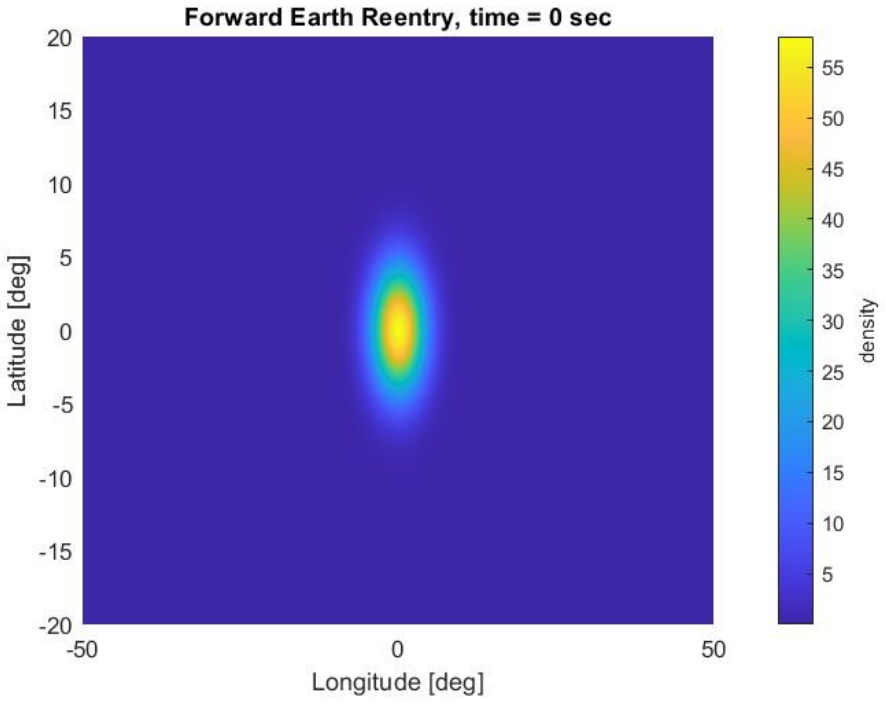

Figure 50. Forward Earth Reentry, Time = 0 sec

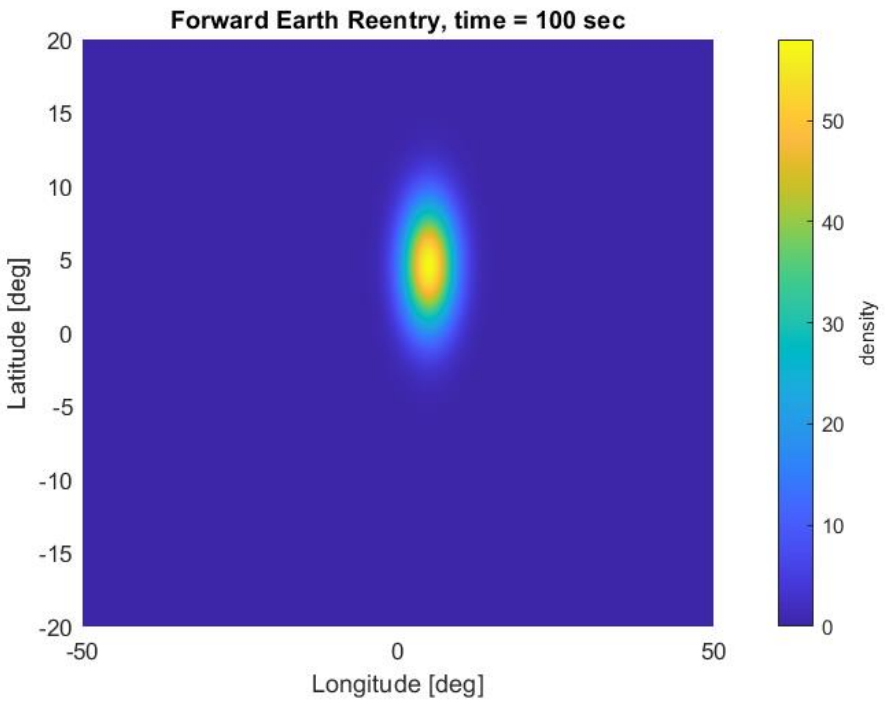

Figure 51. Forward Earth Reentry, Time = 100 sec

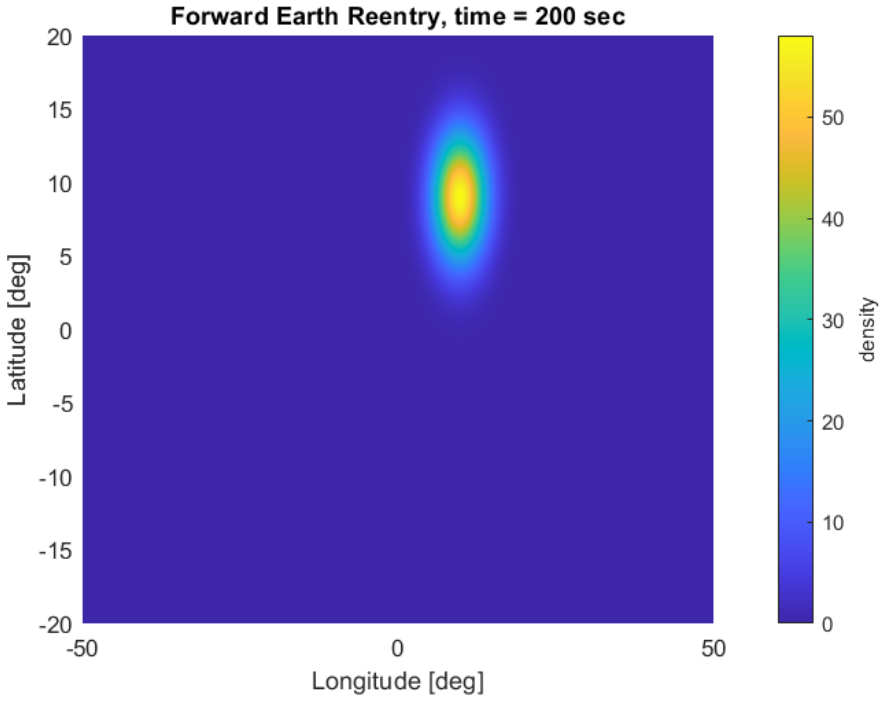

Figure 52. Forward Earth Reentry, Time = 200 sec

The resulting joint distribution observed in Figure 52 is comparable to the results depicted from MCS, demonstrating validity of the approach with this set of equations. The response in Figure 52 is then set as the initial conditions for the reverse case. The equations of motion are reversed and computed for 200 seconds. The reverse probability density is depicted inf Figures 53-55.

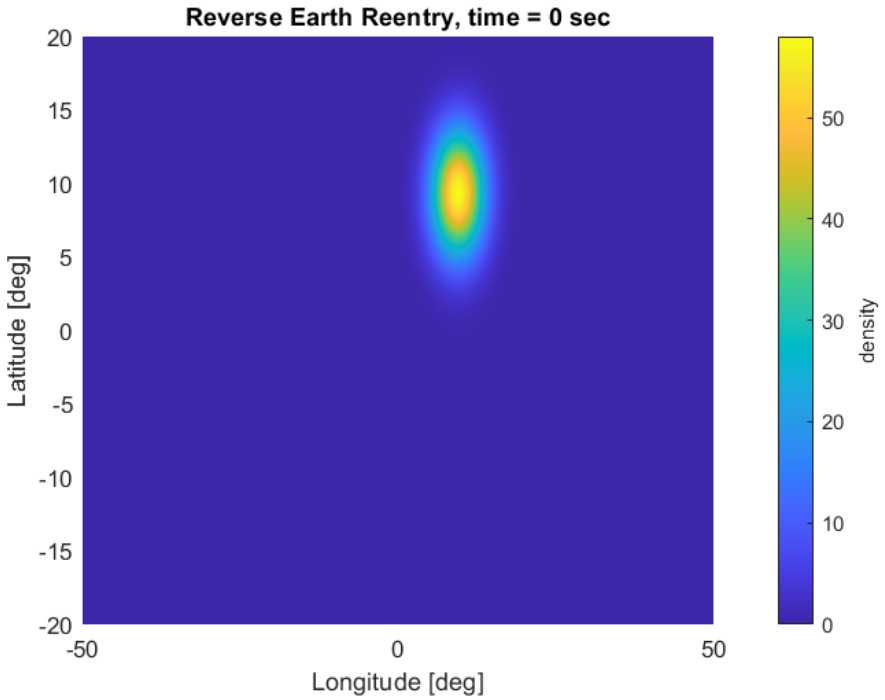

Figure 53. Reverse Earth Reentry, Time = 0 sec

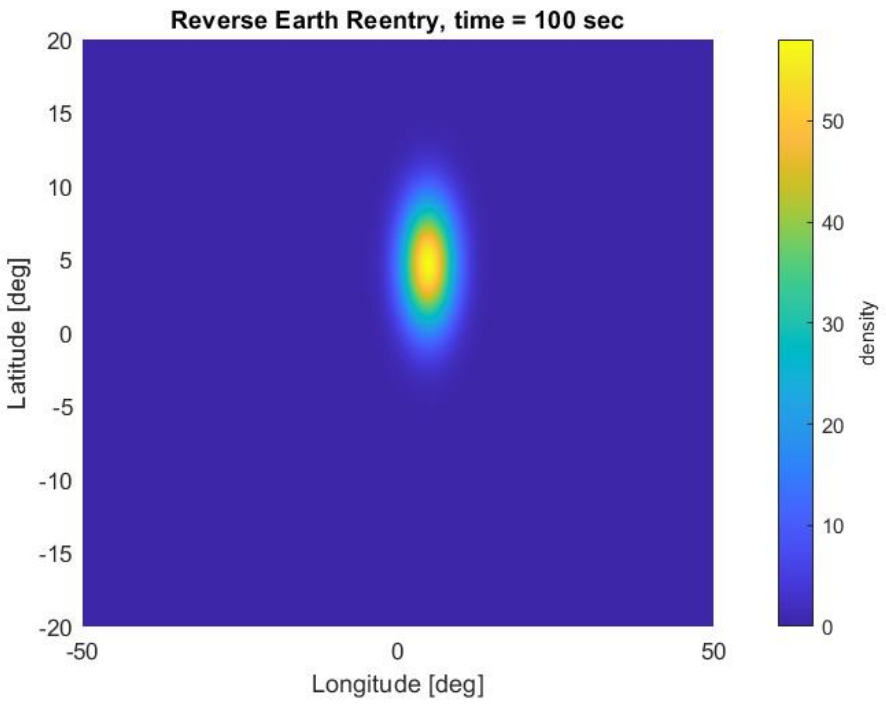

Figure 54. Reverse Earth Reentry, Time = 100 sec

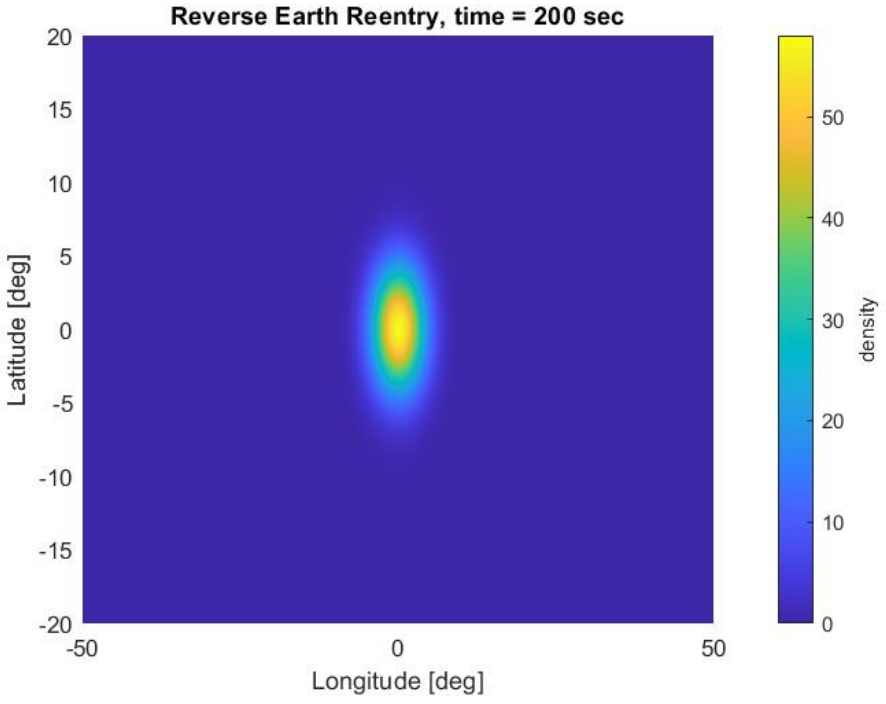

Figure 55. Reverse Earth Reentry, Time = 200 sec

Assessing the error between the actual initial conditions and the computed initial conditions is done in Figure 56. Probability density error is on the order of 10^{-11}, lower than relative tolerances set for Runge-Kutta integration of the deterministic and MCS examples.

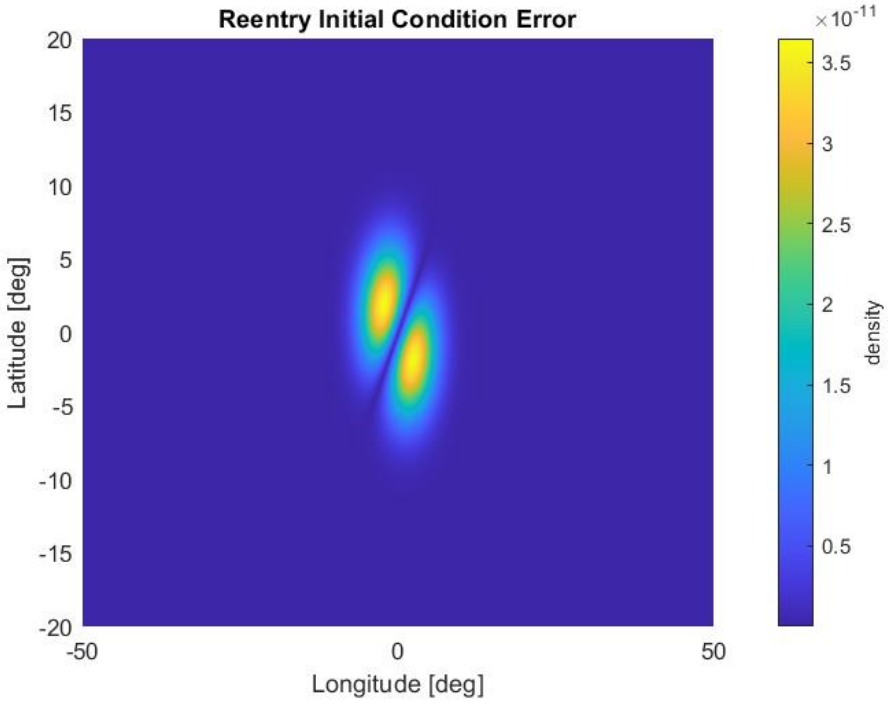

Figure 56. Earth Reentry Initial Condition Error

The two-dimensional case is then extended to 3-dimensions consisting of latitude, longitude, and radius as uncertain variables where the FP equation is now:

$$\frac{\partial}{\partial t} p(\boldsymbol{x}, t|\boldsymbol{x}_0, t_0)$$

$$= -\frac{\partial}{\partial x_1}\left[\frac{x_4 \cos(x_5)\sin(x_6)}{x_3 \cos(x_2)} p(\boldsymbol{x}, t|\boldsymbol{x}_0, t_0)\right]$$

$$-\frac{\partial}{\partial x_2}\left[\frac{x_4 \cos(x_5)\cos(x_6)}{x_3} p(\boldsymbol{x}, t|\boldsymbol{x}_0, t_0)\right] \quad (58)$$

$$-\frac{\partial}{\partial x_3}[x_4 \sin(x_5)\, p(\boldsymbol{x}, t|\boldsymbol{x}_0, t_0)]$$

The forward case is depicted below with latitude-longitude surface plots and longitude-altitude surface plots of the probability distribution. Variability about the latitude and longitude remain the same, the radius is given a standard deviation of 100 km.

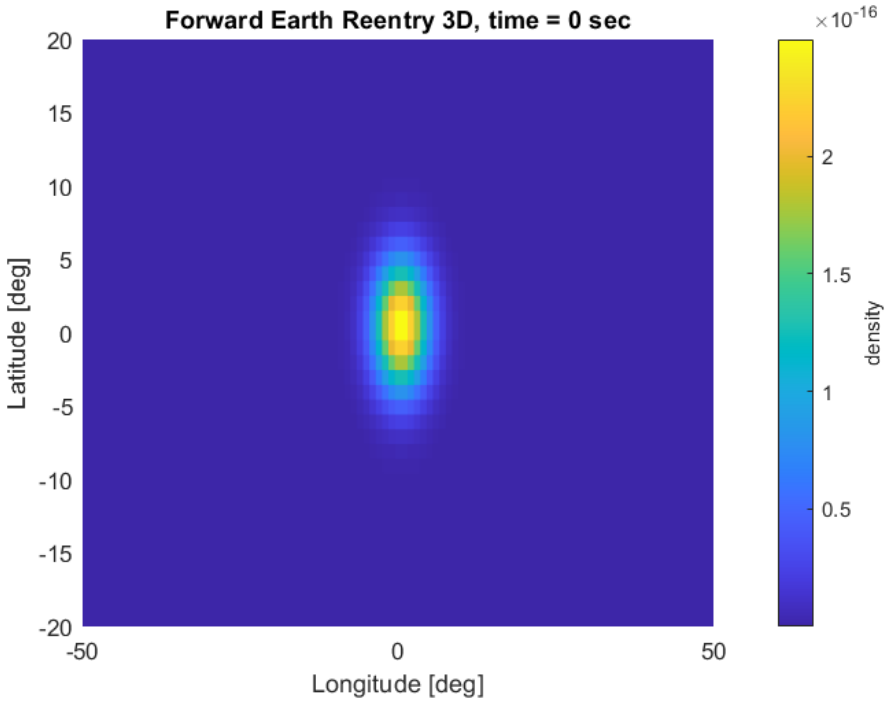

Figure 57. Forward 3D Earth Reentry Latitude, Time = 0 sec

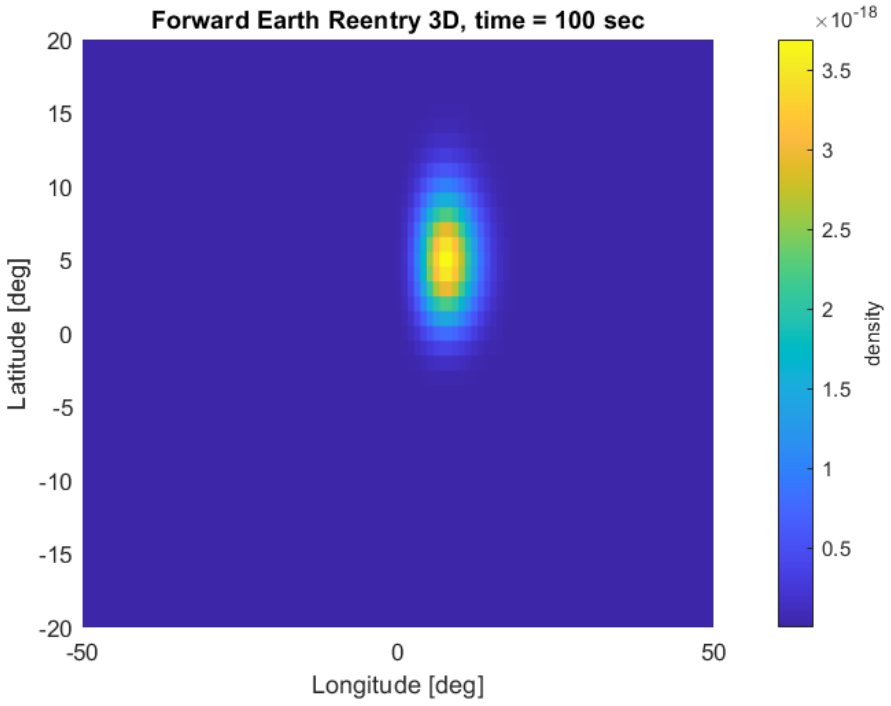

Figure 58. Forward 3D Earth Reentry Latitude, Time = 100 sec

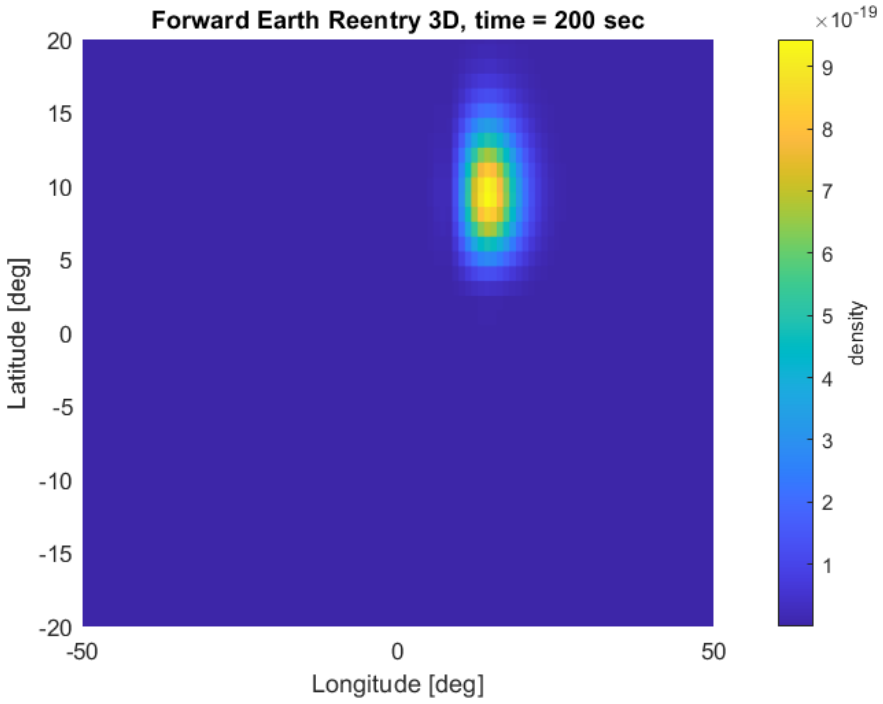

Figure 59. Forward 3D Earth Reentry Latitude, Time = 200 sec

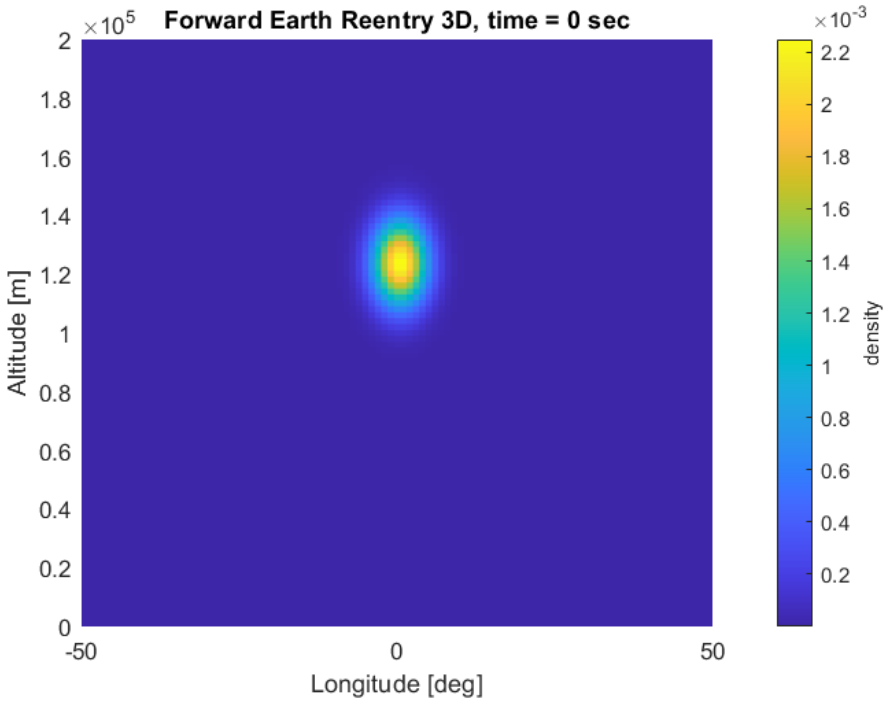

Figure 60. Forward 3D Earth Reentry Altitude, Time = 0 sec

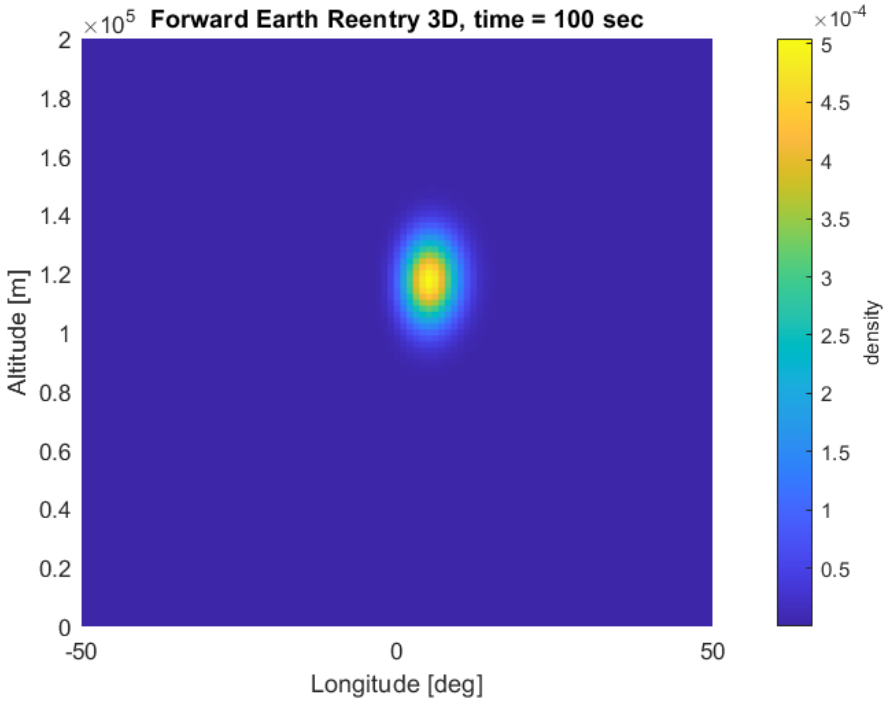

Figure 61. Forward 3D Earth Reentry Altitude, Time = 100 sec

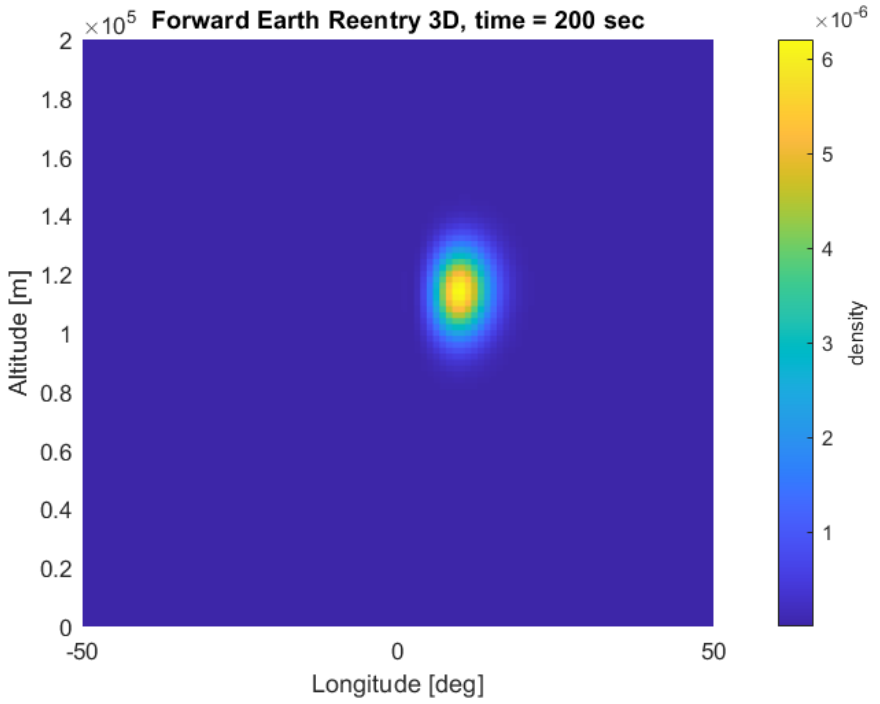

Figure 62. Forward 3D Earth Reentry Altitude, Time = 200 sec

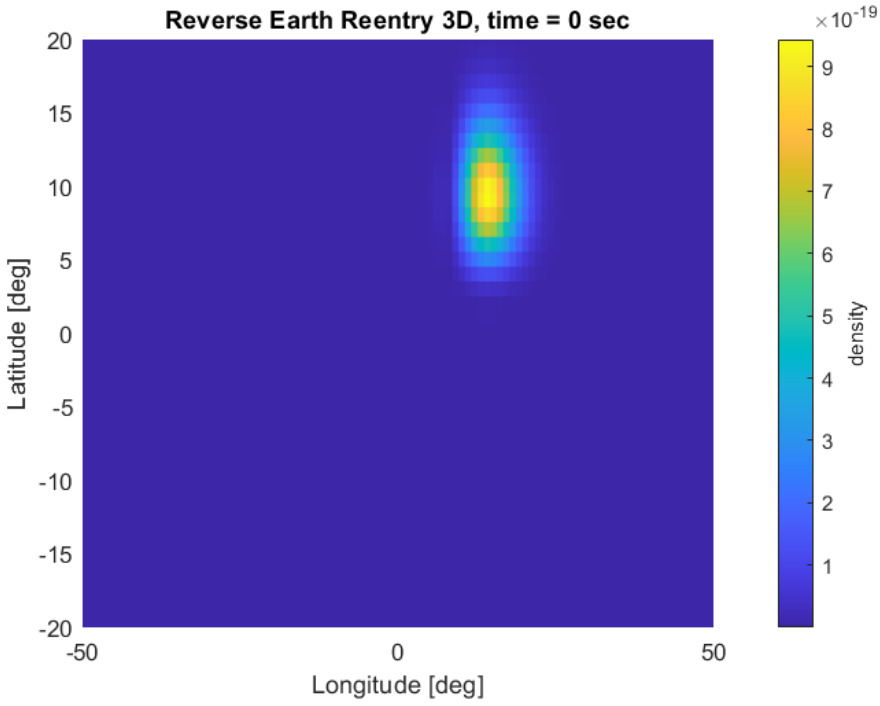

Figure 63. Reverse 3D Earth Reentry Latitude, Time = 0 sec

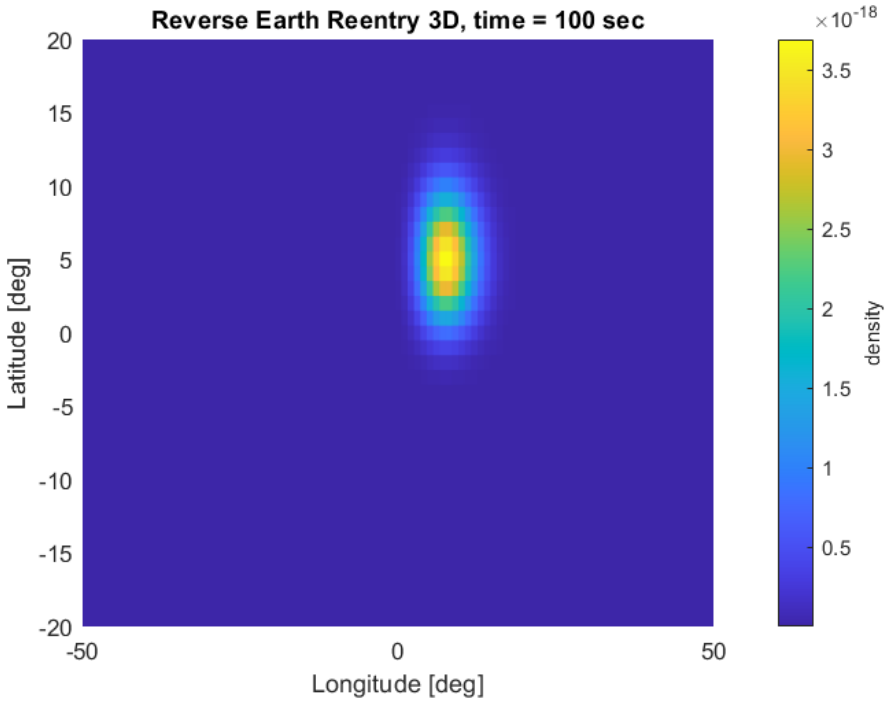

Figure 64. Reverse 3D Earth Reentry Latitude, Time = 100 sec

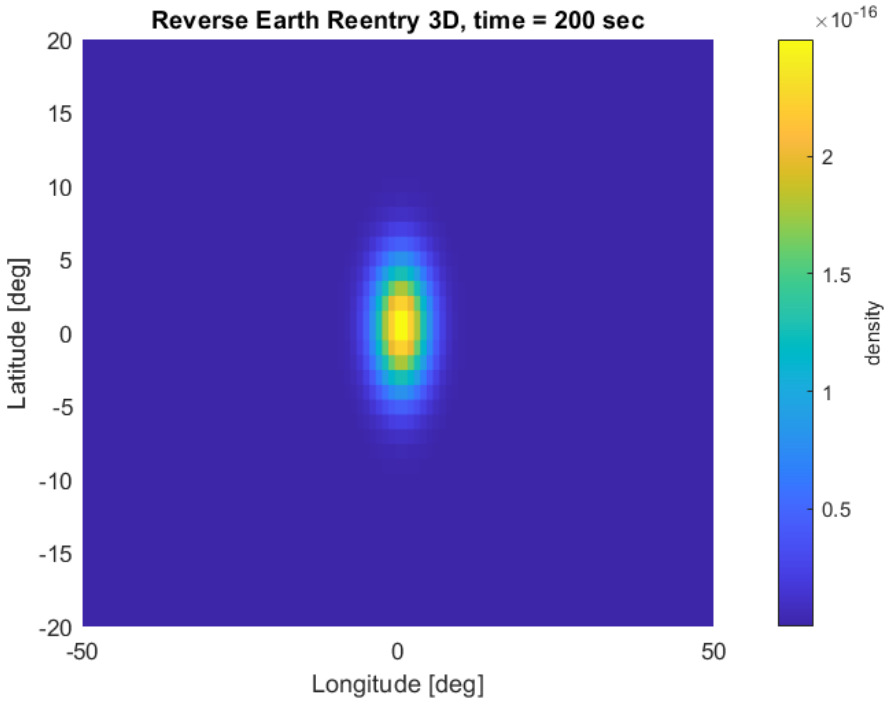

Figure 65. Reverse 3D Earth Reentry Latitude, Time = 200 sec

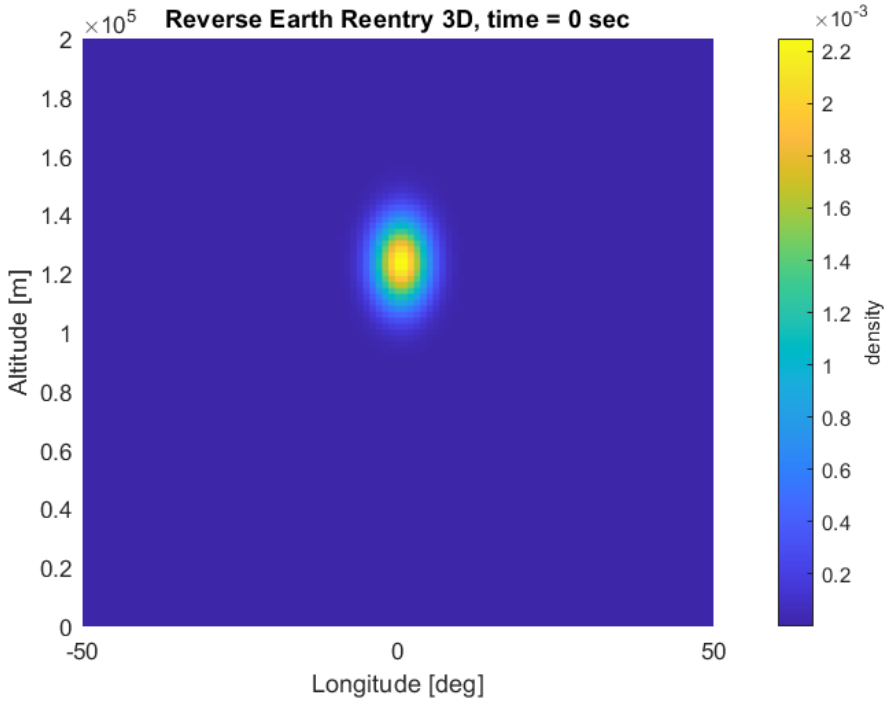

Figure 66. Reverse 3D Earth Reentry Altitude, Time = 0 sec

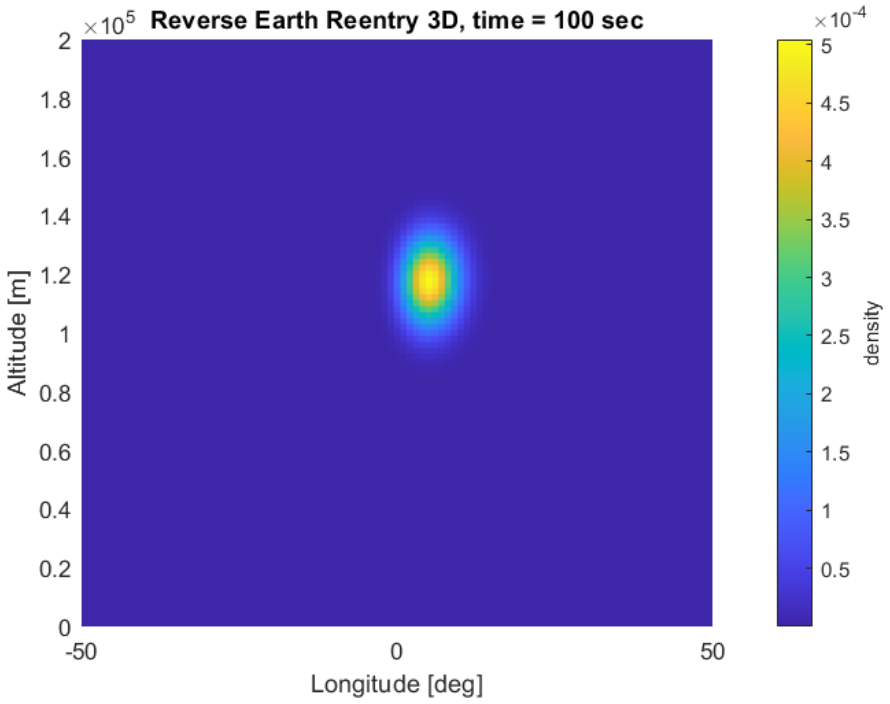

Figure 67. Reverse 3D Earth Reentry Altitude, Time = 100 sec

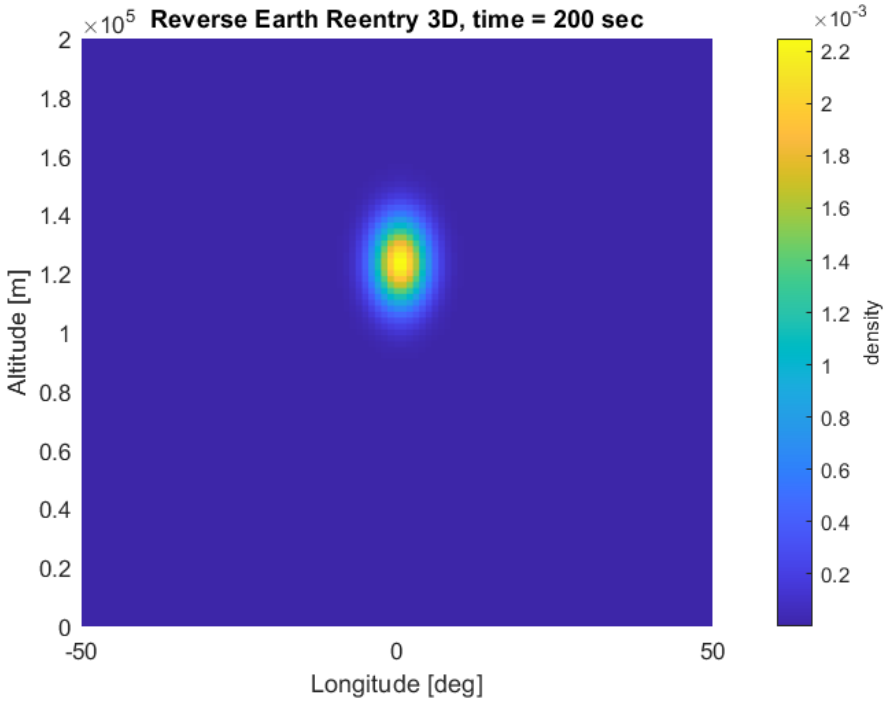

Figure 68. Reverse 3D Earth Reentry Altitude, Time = 200 sec

Error between the true initial condition and reverse computed initial conditions is depicted in the following two figures. Figure 69 depicts the error for the joint distribution of latitude and longitude. Joint distribution for altitude and longitude is illustrated in Figure 70.

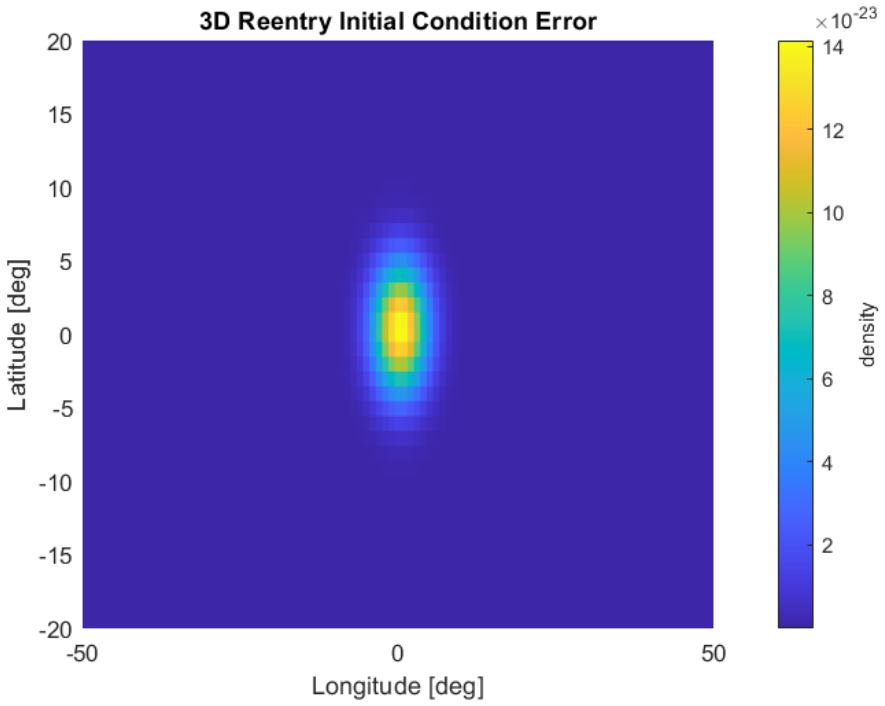

Figure 69. 3D Earth Reentry Position Error

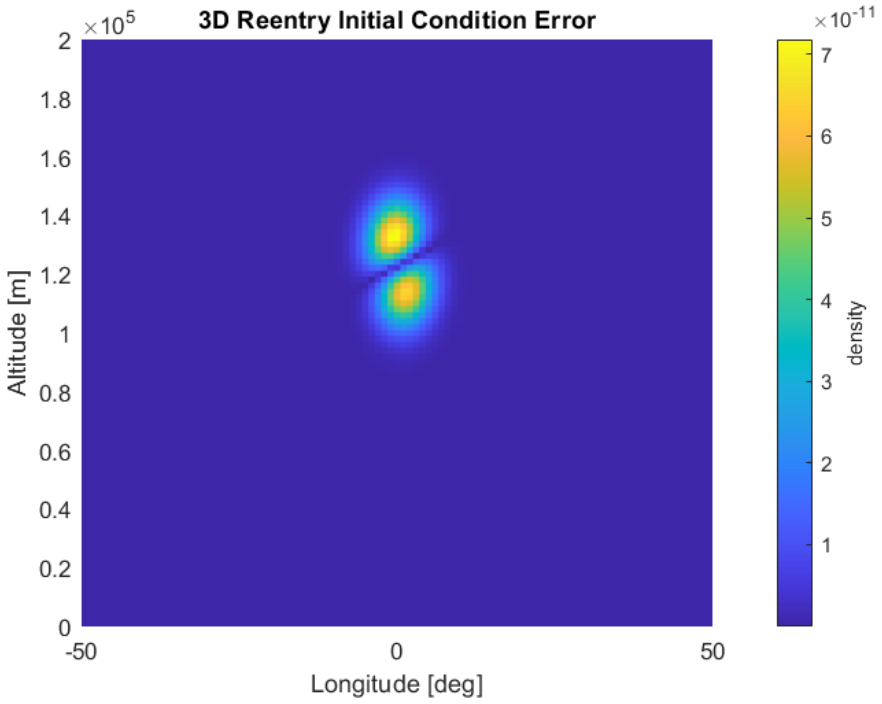

Figure 70. 3D Earth Reentry Altitude Error

5.3 MARS REENTY

For the Martian atmosphere, a NASA exponential model [56] depicted in Equations (59)-(62) is used, where h is altitude from the Martian surface, T is temperature in degree Celsius, p is pressure in kiloPascals, and ρ is density in kilograms per cubic meter.

$$h > 7000: T = -23.4 - 0.00222h \tag{59}$$

$$h < 7000: T = -31 - 0.000998h \tag{60}$$

$$pr = 0.699e^{-0.00009h} \tag{61}$$

$$\rho = \frac{pr}{0.1921(T + 273.1)} \tag{62}$$

This atmospheric model begins to break down at an altitude of 112 kilometers. This occurs due to Equations (59) and (60) approaching and dropping below absolute zero. For many Earth applications, a 100-kilometer altitude relative to mean sea-level is used as the defining line between the atmosphere and space called the Karman line [57]. The Karman line is derived from computations on winged bodies where the velocity required to generate lift reaches orbital speeds. Although the Martian atmosphere is thinner than that of Earth, reduced gravitational pull on the atmosphere results in a larger scale height (approximately 11.1 kilometers more than Earth) on Mars [58]. Due to this phenomenon, it is assumed that the Karman line is similar on Mars. Figure 71 depicts the atmospheric density of Mars up to the Martian Karman line.

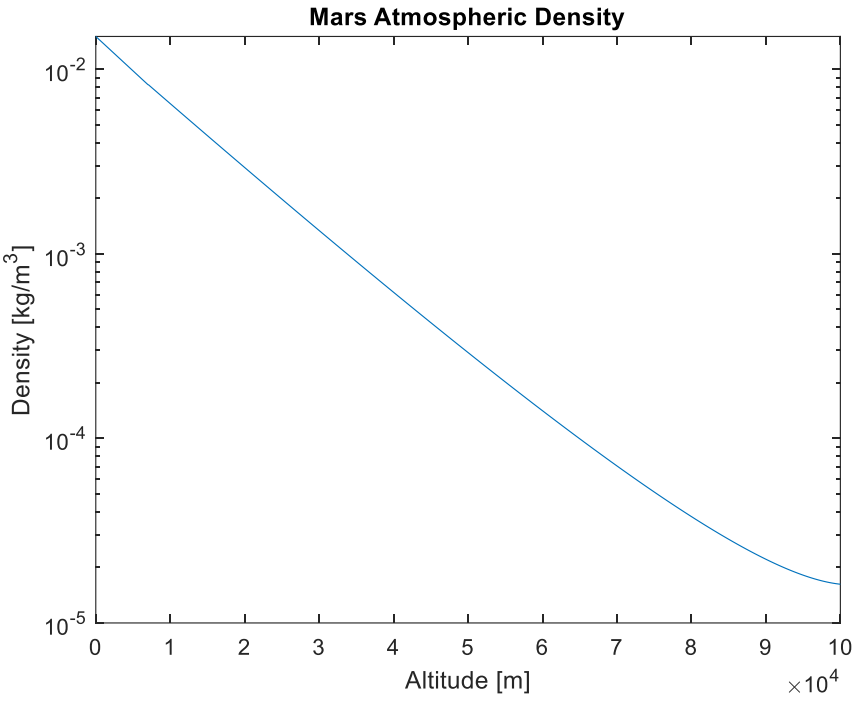

Figure 71. Mars Atmospheric Density

The thin atmosphere requires a scaling of the Mach dependent drag coefficients to attain feasible reentry computations. The original drag coefficients are increased by a factor of five for Mars reentry computations. Martian celestial properties are also different from Earth. Table 5 lists the Martian properties for use in the reentry equations of motion [58].

Table 5. Mars Parameters

Standard Gravitational Parameter (μ)	4.2828e13	m^3/s^2
Mean Radius (R)	3389500	m
Angular Velocity (ω)	7.094834e-5	rad/s
Acceleration Due to Gravity (g)	3.73	m/s^2

As described in Chapter 1, the desired response of the reentry vehicle is to land within the Holden crater given some variability about a central landing location. It is assumed that at an altitude of 11 kilometers and velocity of 420 meters per second is required for a parachute apparatus to deploy and slow the vehicle down to a safe landing speed [59]. The desired system response to reach these conditions over the Holden crater is overlayed on a Mars map in Figure 72, along with ending states before parachute deployment in Table 6. Again, a 3-degree standard deviation is defined as the desired response conditions for the final joint probability of the landing zone. The 3-degree standard deviation allows for ease of visualizing the transient distribution on a map of Mars.

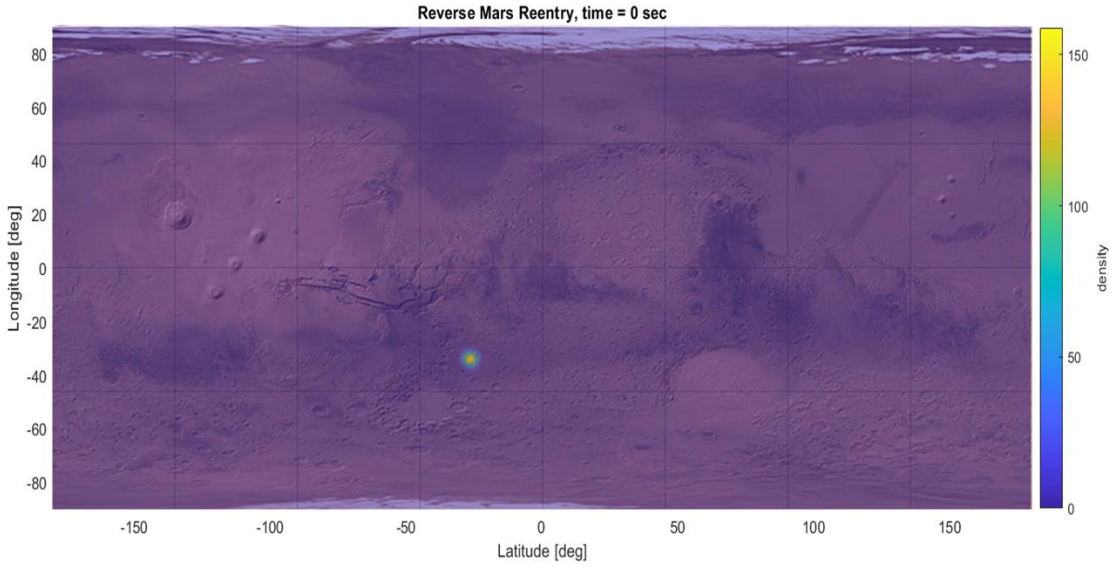

Figure 72. Mars Desired Response

Table 6. Mars Desired Response States at Parachute Deployment

Longitude (x_1)	-34	deg
Latitude (x_2)	-26.4	deg
Radius (x_3)	3400500	m
Velocity (x_4)	420	m/s
Flight Path Angle (x_5)	-28	deg
Heading (x_6)	315	deg

The desired response depicted above is deterministically solved first. The reverse equations of motion terminate at a reentry altitude of 100 kilometers. Figures 73-74 depict the reverse deterministic altitude and speed for the Mars Reentry vehicle.

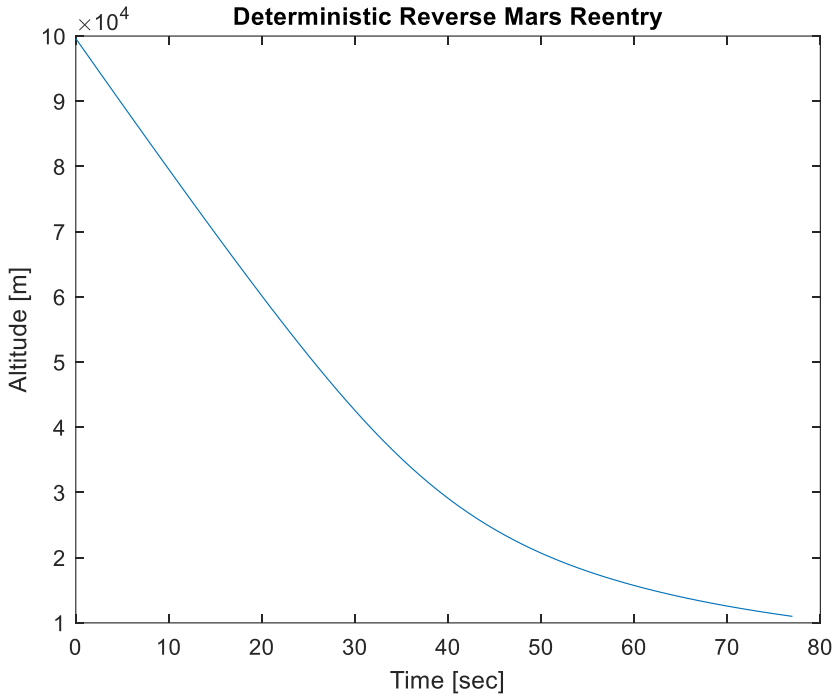

Figure 73. Mars Reverse Deterministic Reenty Altitude

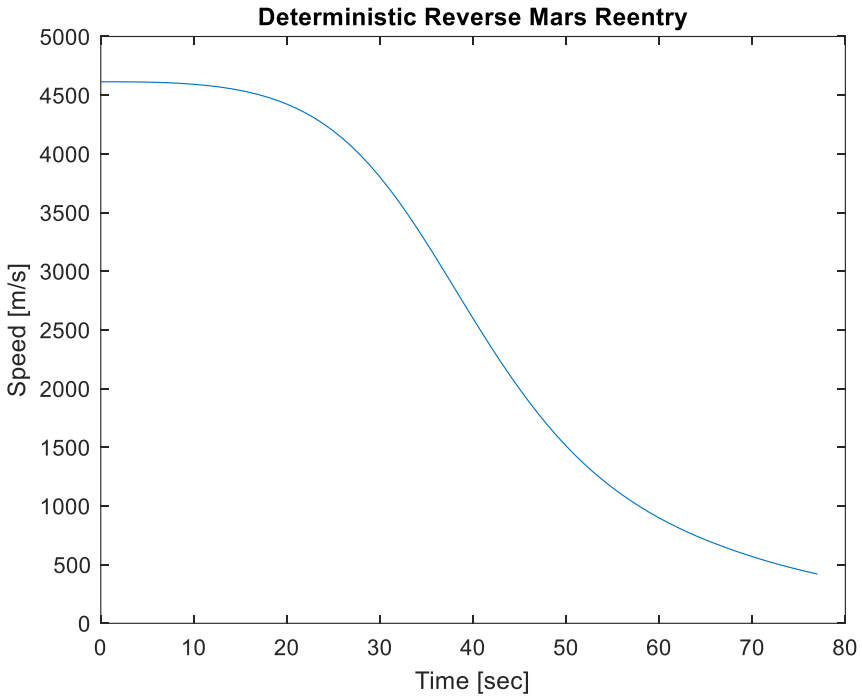

Figure 74. Mars Reverse Deterministic Reenty Speed

The reverse computation yields a time of 77 seconds to attain an altitude of 100 kilometers. The 4.5 kilometers per second entry speed observed at 77 seconds is a comparable atmospheric reentry speed relative to other models [59].

Verification of the forward and reverse reentry problem using the ADI method has been demonstrated in previous examples. Using Mars inputs in lieu of Earth and leveraging the same reentry body parameters and Equation (57), the ADI method is applied to a desired response of the reentry body. An end time (as determined by the deterministic solution) is set at 77 seconds with a time step of 0.5 seconds. The two-dimensional case where only latitude and longitude are uncertain is assessed. Grid sizes of 1500 nodes are used for both latitude and longitude. Initial conditions for all other variables are listed in Table 6. The short timeline and high speed

of the reentry vehicle results in a relatively small haversine ground range of approximately 209 kilometers. Haversine distance is a well-documented algorithm in literature using the following equations to compute the shortest spherical distance between two points, where ϕ is latitude in radians, λ is longitude in radians, and R is the sphere radius:

$$d\phi = \phi_2 - \phi_1 \tag{63}$$

$$d\lambda = \lambda_2 - \lambda_1 \tag{64}$$

$$a = sin^2\left(\frac{d\phi}{2}\right) + \cos(\phi_1)\cos\,(\phi_2)sin^2\left(\frac{d\lambda}{2}\right) \tag{65}$$

$$c = 2\tan^{-1}\left(\sqrt{a}, \sqrt{1-a}\right) \tag{66}$$

$$d = Rc \tag{67}$$

Zooming in on the area of interest provides a better top-view visualization of the changes occurring about the uncertain latitude and longitude. Figures 75-78 depict the transient joint distribution of latitude and longitude from the desired response to atmospheric entry.

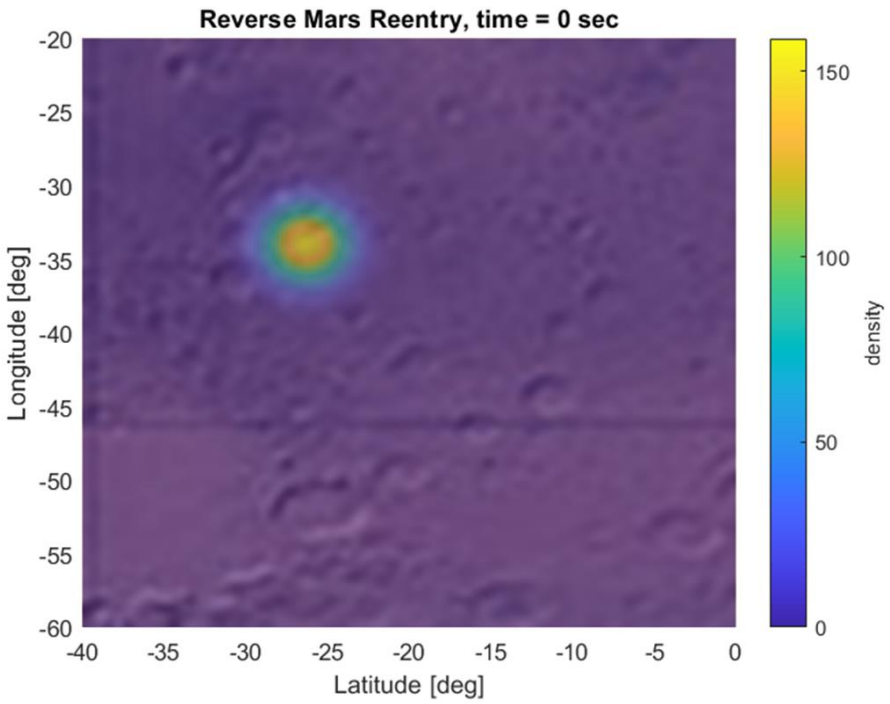

Figure 75. Reverse Mars Reentry, Time = 0 sec

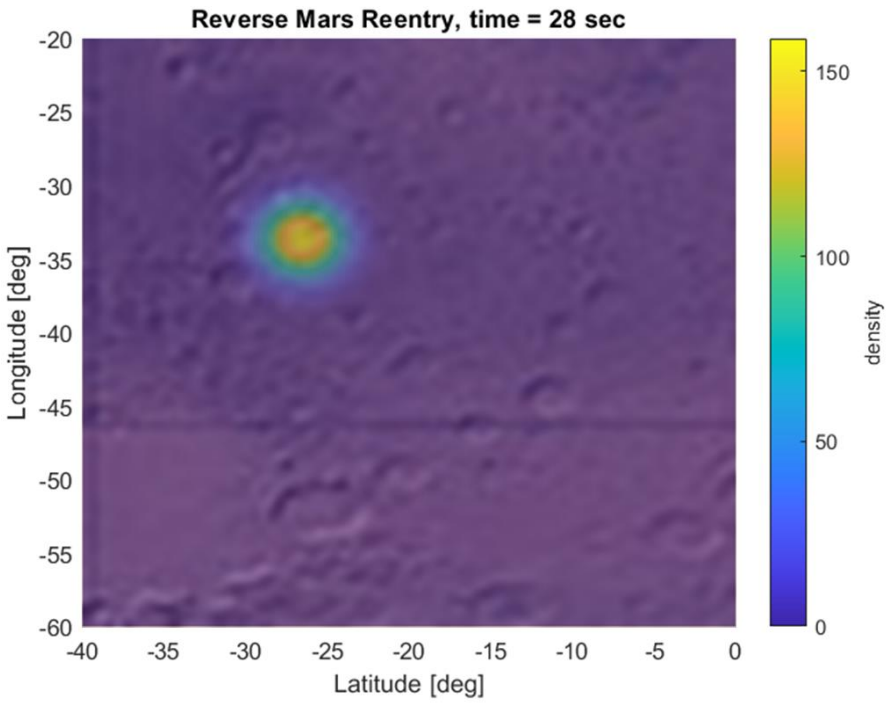

Figure 76. Reverse Mars Reentry, Time = 28 sec

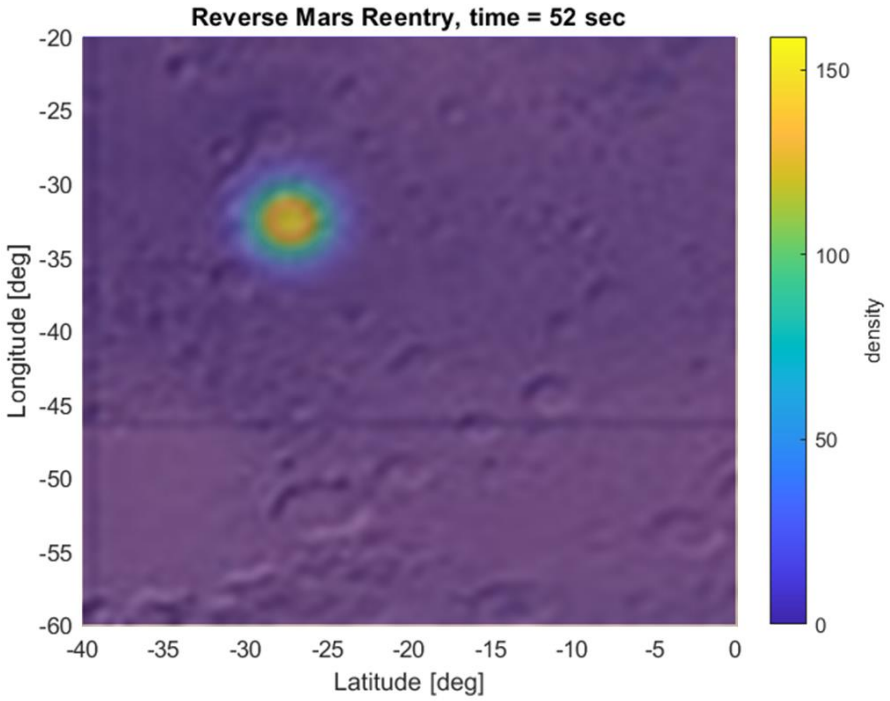

Figure 77. Reverse Mars Reentry, Time = 52 sec

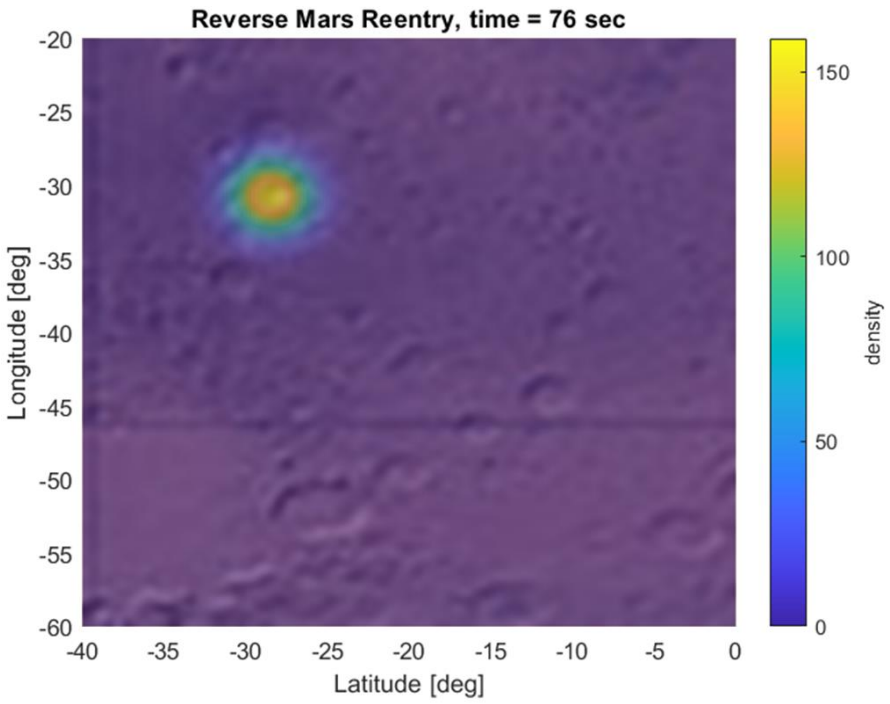

Figure 78. Reverse Mars Reentry, Time = 76 sec

CHAPTER 6

CONCLUSION AND FUTURE WORK

This final chapter discusses findings from the results to include computation time comparison of ADI and MCS, ADI grid sizing comparison and accuracy, and limitations of the presented methodology. Concluding remarks summarize the capabilities and limitations of the presented approach. Future work is discussed in areas of finite difference optimization, uncertainty in controlled systems, high dimension optimization of FP equation, and metamodeling for reduction of high order systems.

6.1 RESULTS DISCUSSION

Two-dimensional cases leveraging the ADI method for computing the FP equations of forward and reverse dynamic problems proved to be as computationally efficient as running Monte Carlo simulations of the same equations of motion. The MCS are run until convergence of the mean square error in each output is achieved, providing solution sets of the uncertainty that are treated as truth. The truth solutions support verification of ADI results, providing evidence of ADI accuracy. Small errors are observed between the MCS results and ADI results. As the number of dimensions (uncertain values) increases, the efficiency of the ADI numerical method drastically decreases. Table 7 depicts computational times for varying MCS run numbers and ADI grid sizes. Highlighted rows depict MCS run numbers required for solution convergence and ADI nodes size that eliminates response instabilities.

Table 7. Linear Oscillator Computational Times

Linear Oscillator				
MCS Runs	**MCS time [sec]**	**ADI Nodes**	**ADI Time Step**	**ADI Time [sec]**
100	0.96	200x200	0.01	1.37
200	1.75	200x200	0.001	15.04
300	2.85	200x200	0.005	2.82
500	5.23	300x300	0.005	6.52
700	7.31	400x400	0.005	14.83
1000	12.47			

For the linear oscillator, time to compute the solutions presented above are slightly faster (near negligible) using the ADI method. Nonlinear 2-D problem spaces do not significantly change the difference between MCS and ADI as depicted in Table 8, although MCS is slightly faster. In the final row of the reentry problem where an additional uncertain variable is applied, a significant change in computational time is observed for ADI.

Table 8. Trim and Earth Reentry Computational Times

Nonlinear Trim				
MCS Runs	**MCS time [sec]**	**ADI Nodes**	**ADI Time Step**	**ADI Time [sec]**
500	4.63	300x300	0.01	4.83
Earth Reentry				
MCS Runs	**MCS time [sec]**	**ADI Nodes**	**ADI Time Step**	**ADI Time [sec]**
800	20.5	500x500	1	22
1200	28	100x100x100	1	156

Due to computational limitations, a 500x500x500 grid was not feasible (ADI tables exceeded computer memory). A 100x100x100 grid demonstrates a significant increase in computational time, while displaying some instabilities within the solution set. The addition of extra dimensions to ADI quickly becomes inefficient from a computational standpoint,

especially in a case where all six states of the reentry problem are uncertain. Not only are 6-D tables a cause for concern computationally but are also a problem both intuitively and functionally. The finite difference grids required at each state for response stability is not feasible for standard memory specifications. Fine grids in each dimension scale each other, rapidly consuming computational resources. In a 6-D case where each dimension is composed of 100 nodes, the tables required to solve the FP equation via an ADI method are 100 to the sixth power! Leveraging Matlabs® sparse matrix function is not enough to reduce these massive tables to manageable memory sizes. Now consider the extra dimension of time. Each of the 100^6 nodes must be solved at each time step over the span of interest. For reentry problems where times can extend to thousands of seconds, the infeasibility of the presented method becomes even more apparent.

Intuitively, a 6-D probability density table is not easy to observe or extract data from. Consider the 3-D case only involving longitude, latitude, and velocity. If the means maintain the deterministic values of Table 4 and standard deviation is set to 3 degrees and 500 m/s, the joint multivariate normal distributions may be constructed as a 3-D table of probabilities. Visualizing these distributions can only be done for two dimensions at a time. Grid sizes of 500 are set for the latitude and longitude. Velocity grid size is set to 1000 (to capture the much larger span of velocity points). Assessing surface plots for latitude and longitude now requires a sweep of plots over all 1000 velocity nodes. Fortunately for the initial condition it is known that the velocity starts off centered about 7589 m/s. The latitude and longitude may then be assessed at that node of the velocity grid as seen in Figure 79.

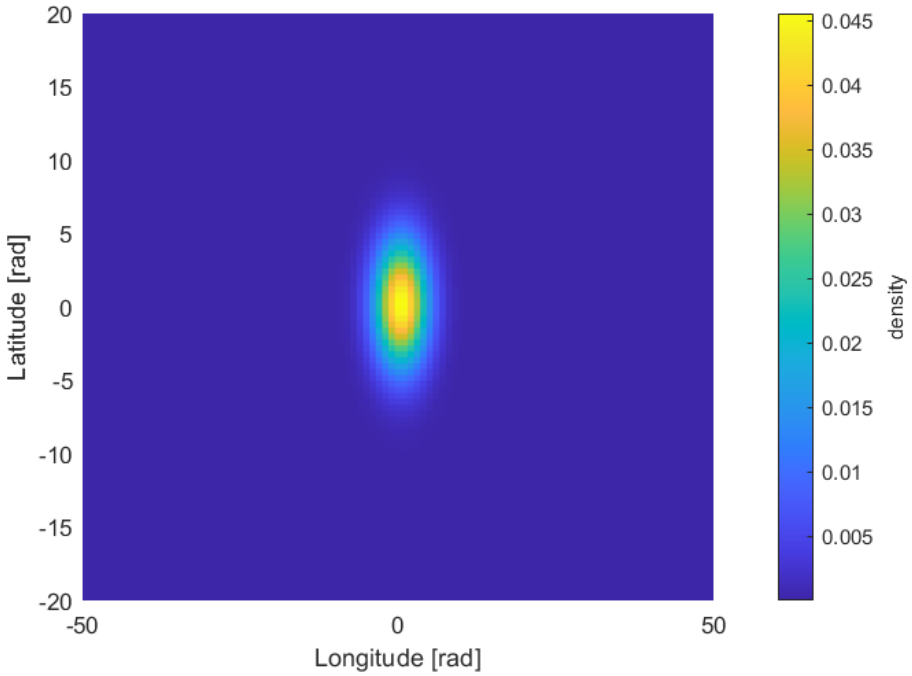

Figure 79. 3-D Initial Condition at 7589 m/s

Assessment of latitude and longitude at the velocity mean immediately depicts probability densities that are lower than values obtained in previous 2-D examples (by orders of magnitude). This is due to a change in the scale of the probability distribution function with the addition of velocity (which has much higher values than latitude and longitude). The scale of these values trends toward extremely small probability density values with the addition of altitude uncertainty. This problem can be resolved with appropriate scaling techniques, but this adds additional complexity to the problem space. Observing the latitude and longitude probability distribution at the same time over different velocity points is illustrated in Figures 80 and 81.

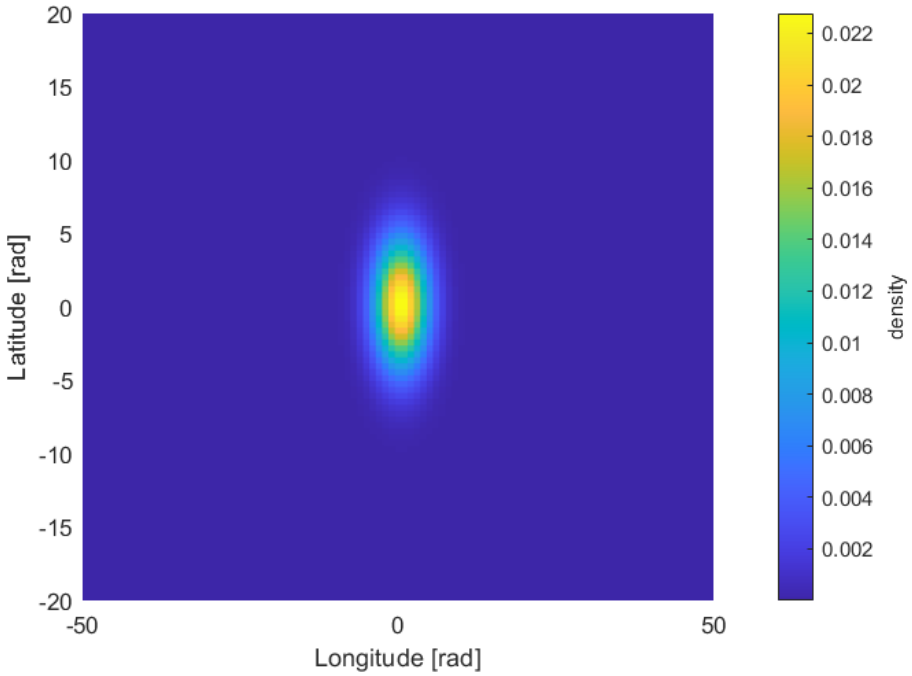

Figure 80. Initial Conditions at 7000 m/s

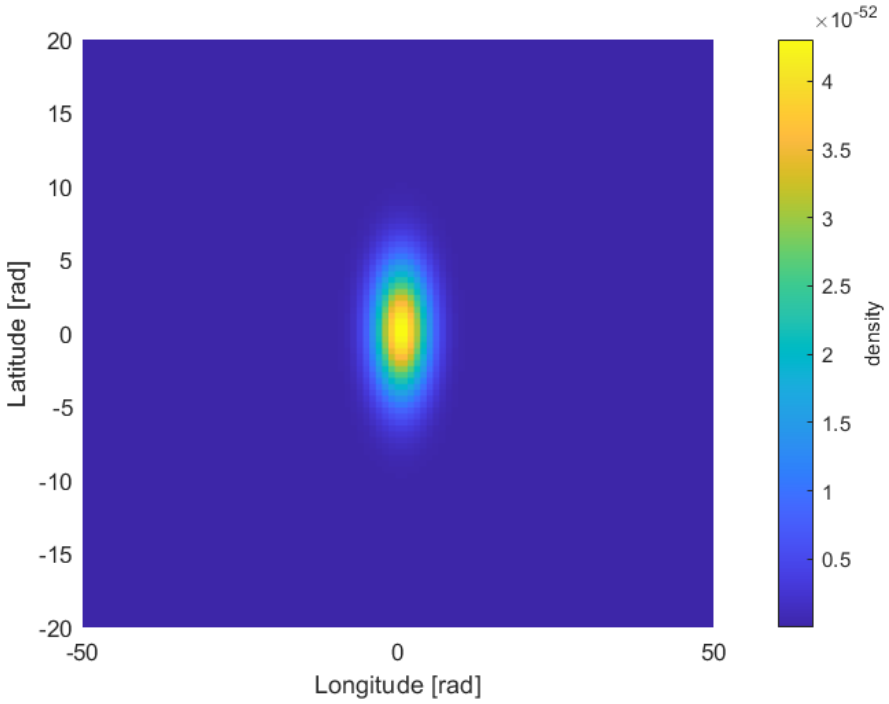

Figure 81. Initial Conditions at 0 m/s

The magnitude of the probability density changes over each velocity node. As the velocity trends further from its higher probability density, the latitude and longitude density decrease but does not change shape or position. Now, extend this to an all-up problem set with 6 uncertain variables. Already high computational times are further increased to find peak densities among uncertain parameters for visualization of 2 variables. Correlations also develop among 2, 3, and even 4 factors that have a significant effect on output relationships between the variables. The computational and intuitive barriers are major drivers for the limited assessment of only 3 uncertain variables for reentry problems.

Additional verification of the ADI methodology is presented in the forward and backward computation of results. Forward computation is performed for each case first, as seen in academic and reentry problems presented in Chapters 4 and 5. Realistic and representative responses are found from the forward case, lending initial conditions that are applied to the reverse computations of each dynamic system. Minimal errors are observed in the predicted initial conditions as compared to actual initial conditions used to generate a desired response of each system.

6.2 CONCLUSIONS/FUTURE WORK

An uncertainty quantification scheme is derived and applied to a series of example problems demonstrating its applicability toward solving uncertain initial conditions given the desired response of a dynamic system. This method is demonstrated for 2-dimensional academic problems and extended to higher dimensional atmospheric reentry dynamics. A use case for this method is applied for a Mars reentry mission knowing final conditions necessary for parachute deployment and uncertainty bounds required about latitude and longitude. The

initial entry conditions are computed for a FP equation using the ADI method applied to reverse reentry equations of motion that are shown to be time symmetric. The uniqueness of this research resides in the study of uncertainty bounds about reversible dynamic systems. MCS and ADI methods are used to quantify the transient nature of probabilistic uncertainty allowable in a system response. The ability to predict uncertainty bounds about the initial conditions of dynamic systems quickly and accurately provides additional insights necessary for system design and mission planning.

The MCS and ADI methods are shown to be comparable for 2-dimensional cases in both solution space and computational time. As the number of dimensions are increased, the ADI method quickly becomes infeasible. A 3-dimensional reentry example is demonstrated using ADI, showing an exponential increase in computational time. In addition to computational resources, the ability to extract and observe data from ADI multi-dimensional tables becomes a limitation of the method as well. This is primarily due to the sheer number of nodes that must be assigned for solution stability within the finite differencing of each variable space.

Many realistic engineering problems require the assessment of more than 2 uncertain variables. Not only is this prevalent in system modeling and simulation but becomes much more dominant in system-of-systems modeling or when assessing future systems that lack specification requirements. As the demand signal for uncertainty quantifications grows, the presented methodologies must be improved upon and applied in other novel ways. A few areas that could be improved include ADI finite difference and time step optimization for solution stability, in-line running of ADI with controlled systems, higher dimensional optimization for solving FP equation, and metamodeling for reduction of high order systems.

The ability to optimize the ADI finite difference sizes and time step provide efficient methods for efficient setup of computational scripts that result in stable solutions while minimizing required computational resources. For the work presented above, heuristics were applied to grid sizes and time steps that achieved stable solutions. Incrementally adding to table sizes eventually results in solution stability but takes a significant amount of time for algorithm setup. These grids are most likely not the most effective to achieve solution stability, particularly with higher dimensional problems that not only need to weight significant factors but also consider significant factor interactions. As problem spaces become more complex and of higher dimensions, the demand for optimal/variable grid sizes and time steps increases [60]. The alternatives to researching this topic further include long algorithm setup times and high computational resource demand. Computational resources and setup time may be alleviated with the use of high-performance computing centers and a setup procedure that overestimates grid size requirements.

Controlled dynamic systems generally have a feedback loop that influences an input to the equations of motion. This is inherent in stochastic model predictive control [61] where uncertainty and random noise make control difficult. Exploration of running the methodologies presented in line with controlled dynamic systems could present solutions for predictive control algorithms or real-time uncertainty characterization. This research could add value to adaptive control schemes, sensitivity analysis, and instrument calibrations.

Solving for higher dimensional problems is a vital aspect of this research. It is possible using the ADI method but proves to be difficult and computationally expensive. The use of the ADI method is shown to be efficient in handling large variable spaces (as compared to other

numerical finite difference methods) but is still limited as the number of uncertain variables increases. Efficient handling of additional uncertain variables is an obvious point of additional research stemming from the presented effort. Obtaining feasible run times using ADI as compared to MCS allows for efficient transient computation of probability distributions per each variable. For reasons discussed above, these probability distributions are useful in characterizing the statistics of a system at any given point in time. Alternatively, other methods may be developed that more efficiently solve the FP equation for higher dimensional equations of motion. Ultimately, the alternatives to MCS (which also becomes computationally expensive with higher orders of uncertainty) is a prominent area for continued research in the discipline of uncertainty quantification.

Mathematical models are composed of numerous inputs that characterize a specific system. Simulations then instantiate these models over time using a variety of methods such as the MCS and ADI methods presented in this research. As was discussed previously, high order models are observed to be computationally expensive pending the research of new methodologies or optimization schemes. One way to reduce this computational load is to reduce the order of the models being instantiated. This can be done through meta-models, defined as “a model about a model” [62]. A meta-model provides a characterization of an entire space given only critical inputs (or inputs that have been found to be major contributors to outputs of interest). Further research of meta-modeling complex, multi-dimensional spaces that reduce the order or a model may save significant computational resources. This could result in uncertainty assessments of large trade spaces efficiently without the loss of accuracy.

Many exciting prospects remain for further research in the field of uncertainty quantification. The application of uncertainty methods for solving forward and reverse dynamic systems is demonstrated in this work, fulfilling the objective of this research effort. Uncertainty characterization will continue to be a major topic of discussion as engineers work toward more realistic representations and analysis of the world around us.

REFERENCES

[1] Helmholtz UQ Community. “Types of Uncertainty,” Retrieved from Uncertainty Quantification: https://dictionary.helmholtz-uq.de/content/types.html [Cited February 2022].

[2] Castillo, F. U., “Probabilistic Analysis of Structures Using Stochastic Finite Elements,” Manizales: Universidad Nacional de Colombia, 2015.

[3] Alexander Litvinenko, H. G., “Inverse Problems and Uncertainty Quantification,” Institute of Scientific Computing, Technische Universitat Braunschweig.

[4] Chandrupatla, Tirupathi R. and Ashok D. Belegundu. "Penalty Approach." Upper Saddle River, NJ, Pearson Prentice Hall, 2012, Chap. 1.

[5] Shampine, L.F. and M.W. Reichelt. "The MATLAB ODE Suite" *SIAM Journal on Scientific Computing*, 1997, pp. 1-22. Doi: 10.1137/S1064827594276424

[6] Bogacki, P. and Shampine, L.F. "A 3(2) Pair of Runge-Kutta Formulas" *Appl. Math. Lett.*, Vol. 2, 1989, pp. 321-325.

[7] Goicolea, J. M. and Garcia Orden J. C. , “Dynamic Analysis of Rigid and Deformable Multibody Systems with Penalty methods and Energy-Momentum Schemes”, *Computer Methods in Applied Mechanics and Engineering,* Vol. 188, 2000, pp. 789-804. Doi: 10.1016/S0045-7825(99)00362-X

[8] Arnold, L., *Stochastic Differential Equations: Theory and Applications*, John Wiley & Sons, Inc., 1974.

[9] NASA, Picking a Landing Site for NASA’s Mars 2020 Rover, NASA Archived Content, https://mars.nasa.gov/mars2020/timeline/prelaunch/landing-site-selection/eight-potential-sites/. [Cited April 2023].

[10] Soong T.T., *Random Differential Equations in Science and Engineering*, Academic Press, 1973.

[11] Pinsky, M.A. and Karlin, S., *An Introduction to Stochastic Modeling 4th Edition,* Academic Press, 2011, Chap. 3.

[12] Li, C., Yang, W., *Time-Dependent Reliability Theory and Its Applications*, Woodhead Publishing Series in Civil and Structural Engineering, pp. 183-253, 2023, Chap. 4.

[13] Marmarelis, V. Z., *Nonlinear Dynamic Modeling of Physiological Systems*, IEEE Press, 2004.

[14] Wirsching, P.H., *Random Vibration Theory and Practice*, Dover Publications Inc., 2006.

[15] Weisstein, E.W., *Probability Density Function*, MathWorld—A Wolfram Web Resource. https://mathworld.wolfram.com/ProbabilityDensityFunction.html. [Cited May 2023].

[16] Gelman, A., *Bayesian Data Analysis (Third Edition)*, CRC Press, 2013.

[17] Saltelli, A., et al., *Global Sensitivity Analysis. The Primer*, John Wiley & Sons, Inc., 2008, Chap. 1.

[18] Haldar, A., and Mahadevan, S., Probability, Reliability, and Statistical Methods in Engineering Design, John Wiley & Sons, Inc., 2000, Chap. 1.

[19] Haupt, S.E., et al., "Principles of Meteorology and Numerical Weather Prediction," *Renewable Energy Forcasting*, Woodhead Publishing, 2017, pp. 3-28. Doi: 10.1016/B978-0-08-100504-0.00001-9

[20] Werndl, C., "Initial-Condition Dependence and Initial-Condition Uncertainty in Climate Science," The British Journal for the Philosophy of Science, 2019, pp. 953-976. Doi: 10.1093/bjps/axy021

[21] Hui Huo, W. X., "New Non-Intrusive Stachastic Finite Element Method for Plate Structures", *Computers and Structures*, Vol. 268, 2022. Doi: 10.1016/j.compstruc.2022.106812

[22] Lio, W., and Liu, B., "Initial Value Estimation of Uncertain Differential Equations and Zero-Day of COVID=19 Spread in China," Fuzzy Optimization and Decision Making, 2021, pp. 177-188. doi: 10.1007/s10700-020-09337-6.

[23] Montecinos, A., Davis, S., and Peralta, J., "Deterministic Physical Systems Under Uncertain Initial Conditions: The Case of Maximum Entropy Applied to Projectile Motion," *European Journal of Physics*, 2018, pp. 12. doi: 10.1088/1361-6404

[24] Acciarini, G., Greco, C., and Vasile, M., "On the Solution of the Fokker-Planck Equation Without Diffusion for Uncertainty Propagation in Orbital Dynamics," *Proceedings of the AAS/AIAA Astrodynamics Specialist Conference*, AAS, Virtual Event, 2020, pp. 3659-3676.

[25] Bellomo, N., Brzezniak, Z., and de Socio, L. M., *Nonlinear Stochastic Evolution Problems in Applied Sciences*, Kluwer Academic Publishers, 1992.

[26] Trisolini, M. and Colombo, C., "Propagation and Reconstruction of Reentry Uncertainties Using Continuity Equation and Simplicial Interpolation," *Journal of Guidance, Control, and Dynamics*, Vol. 44, No. 4, April 2021. doi: 10.2514/1.G005228.

[27] Gondelach, D.J., Armellin, R., and Lidtke, A., “Ballistic Coefficient Estimation for Reentry Prediction of Rocket Bodies in Eccentric Orbits Based on TLE Data,” *Mathematical Problems in Engineering*, Vol. 2017, 10 December 2017, pp. 1-13. Doi: 10.1155/2017/7309637

[28] Lin, Y.K., Cai, G.Q., “Stochastic Analysis of Nonlinear Systems,” *Encyclopedia of Vibration*, 2001, pp. 1238-1246. Doi: 10.1006/rwvb.2001.0044

[29] Huo, H., Xu, W., Wang, W., Chen, G., Yang, D., New Non-Intrusive Stachastic Finite Element Method for Plate Structures, *Computers and Structures*, Vol.268, August 2022. Doi: 10.1016/j.compstruc.2022.106812

[30] IBM Cloud Education, *Monte Carlo Simulation*, August 2020. https://www.ibm.com/cloud/learn/monte-carlo-simulation [Cited February 2023]

[31] Driels, M. R., *Determining the Number of Iterations for Monte Carlo Simulations of Weapon Effectiveness*, Naval Postgraduate School, Dissertation, 2004.

[32] Budynas, R.G., and Nisbett, J.K., “Shigley’s Mechanical Engineering Design - Ninth Edition”, McGraw-Hill, 2011, Chap. 6.

[33] Smith, C.L., *Uncertainty Propagation Using Taylor Series Expansion and a Spreadsheet*, Idaho National Engineering Laboratory, 1999.

[34] Sudret, B. and Der Kiureghian, A., *Stochastic Finite Element Methods and Reliability: A State-of-the-Art Report*, Department of Civil and Environmental Engineering, University of California, 2000.

[35] Faes, M., Broggi, M., Patelli, E., Govers, Y., Mottershead, J., Beer, M., and Moens, D., “A Multivariate Interval Approach for Inverse Uncertainty Quantification With Limited Experimental Data,” *Mechanical Systems and Signal Processing*, March 2019. Doi: 10.1016/j.ymssp.2018.08.050

[36] Yang, X., and Shen, Y., “Runge-Kutta Method for Solving Uncertain Differential Equations,” *Journal of Uncertainty Analysis and Applications*, pp. 3-17, 2015. doi: 10.1186/s40467-015-0038-4.

[37] Weisse, A.Y., Middleton, R.H., and Huisinga, W., “Quantifying Uncertainty, Variability, and Likelihood for Ordinary Differential Equation Models,” *BMC Systems Biology*, Vol. 4, No. 144, 2010. Doi: 10.1186/1752-0509-4-144.

[38] Risken, H. and Frank, T*., The Fokker-Planck Equation Methods of Solution and Applications*, Springer, Chap. 1, 1996.

[39] Gardiner, C.W., *Handbook of Stochastic Methods for Physics, Chemisty, and Natural Sciences*, Springer, 1997.

[40] Marmarelis, V. Z., *Nonlinear Dynamic Modeling of Physiological Systems*, IEEE Press, 2004.

[41] Sun, X., Duan, J., Liu, H., Wang, X., and Zheng, Y., *Derivation of Fokker-Planck Equations for Stochastic Dynamical Systems Under Excitation of Multiplicative Non-Gaussian White Noise*, Cornell University, 2016.

[42] Canor, T. and Denoël, V., "Transient Fokker-Planck Equation with SPH Method Application in Seismic Design," *Proceedings of the 4th ECCOMAS Thematic Conference on Computational Methods in Structural Dynamics and Earthquake Engineering*, Greece, 2013.

[43] Kumar, P., Narayanan, S., "Solution of Fokker-Planck Equation by Finite Element and Finite Difference Methods for Nonlinear Systems," *Sadhana* 31(3), pages 445–461, 2006.

[44] Wijker, J., *Random Vibrations in Spacecraft Structures Design*, Springer, 2009.

[45] Pichler, L., Masud, A., and Bergman, L. A., "Numerical Solution of the Fokker-Planck Equation by Finite Difference and Finite Element Methods – A Comparative Study," *Proceedings of the 3rd ECCOMAS Thematic Conference on Computational Methods in Structural Dynamics and Earthquake Engineering*, Greece, 2011.

[46] Kis, T., *Tridiagonal Matrix Algorithm*, https://tamaskis.github.io/files/Tridiagonal_Matrix_Algorithm.pdf, 2021.

[47] Yoon, J., *Nonlinear Bayesian Estimation Via Solution of the Fokker-Planck Equation*, University of Oklahoma, 2009.

[48] Chen, J., Yang, J., Shen, K., Zheng, Z., and Chang, Z., "Stochastic Dynamic Analysis of Rolling Ship in Random Wave Condition by Using Finite Element Method," Ocean Engineering, Vol. 250, 2022.

[49] Liu, W., "Solving the inverse problem of time independent Fokker-Planck Equation with Self Supervised Neural Network Method," *Scientific Reports*, 2021. doi: 10.1038/s41598-021-94712-5.

[50] Strogatz, S.H., *Nonlinear Dynamics and Chaos With Applications to Physics, Biology, Chemistry, and Engineering*, CRC Press, pp.164-168, 2015.

[51] Sinha, N. and Ananthkrishnan, N., *Advanced Flight Dynamics with Elements of Flight Control*, CRC Press, Chap. 2, pp. 79, 2017.

[52] Gundersen, G., Understanding Moments, https://gregorygundersen.com/blog/2020/04/11/moments/, 2020.

[53] Weisstein, Eric W., "Gaussian Integral", *MathWorld-A Wolfram Web Resource*, https://mathworld.wolfram.com/GaussianIntegral.html.

[54] LeVeque, R.J., *Finite Difference Methods for Ordinary and Partial Differential Equations*, Society for Industrial and Applied Mathematics, 2007.

[55] Tewari, A., *Atmospheric and Space Flight Dynamics,* Chapter 12, Birkhäuser, 2007.

[56] Hall, N., "Mars Atmosphere Model", *NASA* [online webpage], https://www.grc.nasa.gov/www/k-12/airplane/atmosmrm.html [cited 9 December 2023].

[57] McDowell, J.C., *The edge of space: Revisiting the Karman Line,* Acta Astronautica, Vol. 151, pp. 668-677, October 2018.

[58] Williams, D.R., "Mars Fact Sheet", *NASA* [online webpage], https://nssdc.gsfc.nasa.gov/planetary/factsheet/marsfact.html, 5 December 2023, [cited 9 December 2023].

[59] Platt, J., "Entry, Descent, and Landing", *NASA* [online webpage], https://mars.nasa.gov/mars2020/timeline/landing/entry-descent-landing/#:~:text=Parachute%20Deployment&text=The%20parachute%2C%20which%20is%2070.5,940%20mph%20(1%2C512%20kph), [cited 9 December 2023].

[60] Sun, G. and Trueman, C. W., "A simple method to determine the time-step size to achieve a desired dispersion accuracy in ADI-FDTD" *Microwave and Optical Technology Letters*, March 2004, pp. 487-490. Doi: 10.1002/mop.20012.

[61] Buehler, E., *Efficient Uncertainty Propagation for Stochastic Model Predictive Control,* Department of Chemical Engineering, University of California, Berkeley, 2017.

[62] Haskins, C., *Systems Engineering Analyzed, Synthesized, and Applied to Sustainable Industrial Park Development,* Norwegian University of Science and Technology, 2008.

APPENDIX A

COMPUTER CODES

MATLAB® by Mathworks was utilized as the base coding platform. Computations were run on an AMD Ryzen 7 4800H 2.90 GHz processor with Radeon Graphics and 8.0 GB of RAM. The MATLAB® version utilized is R2022a. The following sections depict MATLAB® scripts and functions created to produce results illustrated in this report.

Appendix A.1 leverages the primary run file (solveFPE.m) to run both the forward and reverse dynamic case for the nonlinear trim example problem. Additional functions required to run this example include ADI_fwd.m, ADI_rvs.m, matrix_fwd.m, matrix_rvs.m, matrixTridiag.m, and solveTridiag.m. Addition comments are provided within scripts to guide users on where in apply inputs and algorithm uses.

Appendix A.2 uses a primary run file named MCS_NonlinearTrim.m to run a specific number of Monte Carlo Simulations for the same nonlinear trim problem solved using Appendix A.1. The function MCS_contour.m is called to produce contour plots from the aggregate MCS data.

A.1. NONLINEAR TRIM FP EQUATION USING THE ADI METHOD

solveFPE.m

```
%% Forward/Backward Trim Flight Characterization
% AUTHOR: Troy Newhart

% DATE: 4 April 2023
% ORGANIZATION: Old Dominion University

% DESCRIPTION: Uses an Alternating Directions Implicit method to solve the
% Fokker-Planck PDE, finding the transient joint probability distribution
% of the response.

% REFERENCES:
```

```
% [1] "Numerical Solution of the Fokker-Planck Equation by Finite
% Difference and Finite Element Methods - A Comparative Study," L. Pichler,
% A. Masud, and L.A. Bergman.

% FUNCTION CALLS:
% matrix_fwd.m, matrix_rvs.m ADI_fwd.m,, ADI_rvs.m matrixTridiag.m, solveTridiag.m

%% Initialize
clear
clc

%% User Inputs
dt = 0.005; % Finite difference timestep
T_end = 3; % End time
M1 = 0.5; M2 = 0.5; % Grid bounds for y(1) = [-M1,M1] and y(2) = [-M2, M2]
N1 =990; N2 = 990; % Number of nodes along each variable axis

%% Solve the forward problem
[D1,D2,x1,x2]=matrix_fwd(M1,N1,M2,N2); % Formulate the tridiagonal matrices of each
state variable
[t,p]=ADI_fwd(D1,D2,x1,x2,dt,T_end,N1,N2); % Alternating Directions Implicit Method
to solve FP PDE for p

%% Solve the backward problem using response PDF from above as initial conditions
p_response = p(:,:,end);
clearvars -except p_response t dt T_end N1 N2 M1 M2 p
[D1,D2,x1,x2]=matrix_rvs(M1,N1,M2,N2); % Formulate the tridiagonal matrices each
state variable
[t_rvs,p_rvs]=ADI_rvs(p_response,D1,D2,dt,T_end,N1,N2); % Alternating Directions
Implicit Method to solve FP PDE for p


%% Plot desired time steps
time = 3; % Specify the time step of interest
t_ind = time/dt + 1; % Time index of the specified time of interest

surf(x2,x1,p(:,:,t_ind),'edgecolor','none')
view(90,90)
xlabel('x_2')
ylabel('x_1')
title("Reverse Nonlinear Trim, time = "+time+" sec")
cb = colorbar();
ylabel(cb,'density')
```

ADI_fwd.m

```
function [t,p]=ADI_fwd(D1,D2,x1,x2,dt,T_end,N1,N2)

% INPUTS: Tridiagonal matrices for each variable (D1, D2), variable grid
% (x1, x2), timestep (dt), end time (T_end)
% OUTPUT: time points and joint transient PDF
% FUNCTION CALLS: solveTridiag.m

t=(0:dt:T_end)';
```

```
p=zeros(N1,N2,length(t)); % Initialize for speed

% Create A matrix and b vector for solving A*p = b
rhs_first = sparse(speye(N1*N2)+.5*dt*D1.matrix); % First A matrix
rhs_second = sparse(speye(N1*N2)+.5*dt*D2.matrix); % Second A matrix

a_first =-.5*dt*D2.a;
b_first =ones(N1*N2,1)'-.5*dt*D2.b;
c_first =-.5*dt*D2.c;

a_second =-.5*dt*D1.a;
b_second =ones(N1*N2,1)'-.5*dt*D1.b;
c_second =-.5*dt*D1.c;

[X1,X2] = meshgrid(x1,x2); % Mesh the discretized spatial domain

%% Assign Initial Conditions
mu = [0, 0.4]; sigma = 0.001*eye(2); % Uncertain Initial conditions
p(:,:,1) = reshape(mvnpdf([X1(:) X2(:)],mu,sigma),length(x2),length(x1))'; % Joint
PDF of uncertain initial conditions

%% Compute the numerical solution to the Fokker-Planck Equation
p = reshape(p,N1*N2,length(t)); % Shape p into a vector for computations

tic()
for j=2:length(t)

    % Track progress of the computations
    if rem(j,100)==0
        fprintf('Number of Iteration %d out of %d finished...\n',j,length(t));
    end

    % Solve for p^(n+1/2)
    reshaped_r = reshape(reshape(rhs_first*p(:,j-1),N1,N2)',N1*N2,1);
    p_half = solveTridiag(a_first,b_first,c_first,reshaped_r);

    % Solve for p^(n+1)
    reshaped_r = reshape(reshape(rhs_second*p_half,N2,N1)',N1*N2,1);
    p(:,j) = solveTridiag(a_second,b_second,c_second,reshaped_r);

end
toc()

p = reshape(p,N1,N2,length(t)); % Reshape into matrix for surface plotting
```

ADI_rvs.m

```
function [t,p_rvs]=ADI_rvs(p_response,D1,D2,dt,T_end,N1,N2)

% INPUTS: Tridiagonal matrices for each variable (D1, D2), variable grid
% (x1, x2), timestep (dt), end time (T_end)
% OUTPUT: time points and joint transient PDF
% FUNCTION CALLS: solveTridiag.m
```

```
t=(0:dt:T_end)';
p_rvs=zeros(N1,N2,length(t)); % Initialize for speed

% Create A matrix and b vector for solving A*p = b
rhs_first = sparse(speye(N1*N2)+.5*dt*D1.matrix); % First A matrix
rhs_second = sparse(speye(N1*N2)+.5*dt*D2.matrix); % Second A matrix

a_first =-.5*dt*D2.a;
b_first =ones(N1*N2,1)'-.5*dt*D2.b;
c_first =-.5*dt*D2.c;

a_second =-.5*dt*D1.a;
b_second =ones(N1*N2,1)'-.5*dt*D1.b;
c_second =-.5*dt*D1.c;

%% Assign Initial Conditions
p_rvs(:,:,1) = p_response; % Joint PDF of uncertain initial conditions

%% Compute the numerical solution to the Fokker-Planck Equation
p_rvs = reshape(p_rvs,N1*N2,length(t)); % Shape p into a vector for computations

tic()
for j=2:length(t)

    if rem(j,100)==0
        fprintf('Number of Iteration %d out of %d finished...\n',j,length(t));
    end

    % Solve for p^(n+1/2)
    reshaped_r = reshape(reshape(rhs_first*p_rvs(:,j-1),N1,N2)',N1*N2,1);
    p_half_rvs = solveTridiag(a_first,b_first,c_first,reshaped_r);

    % Solve for p^(n+1)
    reshaped_r = reshape(reshape(rhs_second*p_half_rvs,N2,N1)',N1*N2,1);
    p_rvs(:,j) = solveTridiag(a_second,b_second,c_second,reshaped_r);
end
toc()

p_rvs = reshape(p_rvs,N1,N2,length(t)); % Reshape into matrix for surface plotting
```

matrix_fwd.m

```
function [D1,D2,x1,x2]=matrix_fwd(M1,N1,M2,N2)

% INPUTS: Discretized the spatial domain: x1 = [-M1, M1], x2 = [-M2, M2]
%   with number of nodes N1 and N2 (matrix size = (N1*N2)x(N1*N2) and vectors
%   (N1*N2)x1
% OUTPUT: D1,D2 tridiagnal matrices and x1, x2 grid points
% FUNCTION CALLS: matrixTridiag.m

%% Initialize finite difference grid
dx1=2*M1/(N1-1);
if(dx1<0)
    disp('Discretization of x1 cannot be negative.');
```

```
    return;
end
x1=(-M1:dx1:M1)';

dx2=2*M2/(N2-1);
if(dx2<0)
    disp('Discretization of x2 cannot be negative');
    return;
end
x2=(-M2:dx2:M2)';

%% Compute tridiagonal matrix over x1 span
a = []; b = zeros(N1*N2,1)'; c = [];
for i=1:N2
    a_coeff = (x1(1:end-1)*0 + x2(i))./(2*dx1);
    c_coeff = (x1(1:end-1)*0 - x2(i))./(2*dx1);

    a = [a, 0, a_coeff'];
    c = [c, c_coeff', 0];
end
D1.a = a; D1.b = b; D1.c = c;
D1.matrix = matrixTridiag(a,b,c);

%% Compute tridiagonal matrix over x2 span
a = []; b = zeros(N1*N2,1)'; c = [];
for i=1:N1
    a_coeff = (-5*0.1*(1-x1(i).^2)*x2(1:end-1)-9*sin(x1(i)))./(2*dx2);
    c_coeff = (5*0.1*(1-x1(i).^2)*x2(1:end-1)+9*sin(x1(i)))./(2*dx2);

    a = [a, 0, a_coeff'];
    c = [c, c_coeff', 0];
end
D2.a = a; D2.b = b; D2.c = c;
D2.matrix = matrixTridiag(a,b,c);
```

matrix_rvs.m

```
function [D1,D2,x1,x2]=matrix_rvs(M1,N1,M2,N2)

% INPUTS: Discretized the spatial domain: x1 = [-M1, M1], x2 = [-M2, M2]
%   with number of nodes N1 and N2 (matrix size = (N1*N2)x(N1*N2) and vectors
%   (N1*N2)x1
% OUTPUT: D1,D2 tridiagnal matrices and x1, x2 grid points
% FUNCTION CALLS: matrixTridiag.m

%% Initialize finite difference grid
dx1=2*M1/(N1-1);
if(dx1<0)
    disp('Discretization of x1 cannot be negative.');
    return;
end
x1=(-M1:dx1:M1)';

dx2=2*M2/(N2-1);
```

```
if(dx2<0)
    disp('Discretization of x2 cannot be negative');
    return;
end
x2=(-M2:dx2:M2)';

%% Compute tridiagonal matrix over x1 span
a = []; b = zeros(N1*N2,1)'; c = [];
for i=1:N2
    a_coeff = -(x1(1:end-1)*0 + x2(i))./(2*dx1);
    c_coeff = -(x1(1:end-1)*0 - x2(i))./(2*dx1);

    a = [a, 0, a_coeff'];
    c = [c, c_coeff', 0];
end
D1.a = a; D1.b = b; D1.c = c;
D1.matrix = matrixTridiag(a,b,c);

%% Compute tridiagonal matrix over x2 span
a = []; b = zeros(N1*N2,1)'; c = [];
for i=1:N1
    a_coeff = -(-5*0.1*(1-x1(i).^2)*x2(1:end-1)-9*sin(x1(i)))./(2*dx2);
    c_coeff = -(5*0.1*(1-x1(i).^2)*x2(1:end-1)+9*sin(x1(i)))./(2*dx2);

    a = [a, 0, a_coeff'];
    c = [c, c_coeff', 0];
end
D2.a = a; D2.b = b; D2.c = c;
D2.matrix = matrixTridiag(a,b,c);
```

matrixTridiag.m

```
function [ Mat ] = matrixTridiag( a,b,c )
% OUTPUT: a sparse tridiagonal matrix given a, b, c

%% Ensure vectors are the same length
N = length(a);
if (length(b) ~= N || length(c) ~= N)
    fprintf('Length of a, b, c not the same.\n');
end

%% Formulate tridiagonal matrix
x_indx = [2:N,1:N,1:N-1]; % Index in the first dimension
y_indx = [1:N-1,1:N,2:N]; % Index in the second dimension
trid = [a(2:end),b(1:end),c(1:end-1)]; % Tridiagonal matrix

Mat = sparse(x_indx,y_indx,trid,N,N); % Remove zeroes to save space on memory

end
```

solveTridiag.m

```
function d_prime = solveTridiag(a,b,c,d) % Use Thomas algorithm to efficiently solve
inverse of tridiagonal matrix
% INPUTS: LHS tridiagonal vectors and RHS matrix
% OUTPUT: Probability density
% REFERENCES: https://en.wikipedia.org/wiki/Tridiagonal_matrix_algorithm

% Initialize variables
N = length(a);
d_prime = zeros(N,1);
c_prime = zeros(N,1);

b_tilde = b(1) ;
d_prime(1) = d(1)/b_tilde ;
% Decomposition of the special case of Gaussian elimination
for j = 2:N
    c_prime(j)=c(j-1)/b_tilde;
    b_tilde=b(j)-a(j)*c_prime(j);
    d_prime(j)=(d(j)-a(j)*d_prime(j-1))/b_tilde;
end
% Backward substitution to find all of the unknowns
for j = 1:(N-1)
    k = N-j ;
    d_prime(k) = d_prime(k) - c_prime(k+1)*d_prime(k+1);
end
```

A.2. NONLINEAR TRIM MCS

MCS_NonlinearTrim.m

```
%% Forward Trim Flight Characterization
% Author: Troy Newhart

% Date: 4 April 2023
% Organization: Old Dominion University

% Description: Using Matlabs ODE45 function, solve and plot the forward
% solution of the differential equation provided from ref. 1.

% References
% [1] "Advanced Flight Dynamics with Elements of Control," Chapter 2, Box
% 2.2.

% Function Calls
% ODE45.m

%% Initialize
clear
clc

%% Monte Carlo Simulation
tic
for i = 1:1:800 %% SEM = s/sqrt(n), want SEM to be 1%, s is 1/9
```

```
    %% Constants for integrating the forward and reverse dynamics
    tspan = 0:0.005:3;
    opts = odeset('RelTol',1e-6,'AbsTol',1e-10);

    %% Compute the forward and reverse dynamics
    y0 = [normrnd(0,sqrt(0.001)); normrnd(0.4,sqrt(0.001))]; % Forward initial
conditions
    ode = @(t,y) linOsc(t,y);
    [t,y] = ode45(ode, tspan, y0, opts); % Forward dynamics

    y_mcs{i} = y;

    if i > 1
        y_total = [y_total;y];
        stddev = std(y_total(:,1));
        plot(i,stddev,'*')
        grid on
        hold on
        title ('Standard Error of the Mean')
        xlabel('Run No.')
        ylabel('Standard Deviation of x_1')
    else
        y_total = y;
        stddev = std(y_total(:,1));
        plot(i,stddev,'*')
        grid on
        hold on
        title ('Standard Error of the Mean')
        xlabel('Run No.')
        ylabel('Standard Deviation of x_1')
    end

    clearvars -except y_mcs i y_total tspan
end
toc
%% Post process the MCS data
for j = 1:1:numel(y_mcs)
    ytar = y_mcs{j};
    y1_t0(j,1) = ytar(1,1);
    y2_t0(j,1) = ytar(1,2);
    y1_t10(j,1) = ytar(end,1);
    y2_t10(j,1) = ytar(end,2);
end

MCS_contour(y2_t10,y1_t10,-0.5:0.025:0.5,-0.5:0.025:0.5)
xlabel('x_1')
ylabel('x_2')
title('Nonlinear Trim MCS, time = 3 sec')

%% Equation of motion functions
% Forward Dynamics
function dydt = linOsc(t,y)
    dydt = [y(2); -9*sin(y(1))-5*0.1*(1-y(1)^2)*y(2)];
end
```

MCS_contour.m

```
function [] = MCS_contour(y2, y1, sizey2, sizey1)
    [N,Xedges,Yedges] = histcounts2(y2,y1,sizey2,sizey1,'Normalization','pdf');
    % Compute bin centers
    Xcnt = Xedges(2:end) - abs(diff(Xedges(1:2))/2);
    Ycnt = Yedges(2:end) - abs(diff(Yedges(1:2))/2);
    figure()
    contour(Xcnt,Ycnt, N)
    grid off
    set(gca,'YDir','reverse')
    % show bins

    arrayfun(@(x)xline(x,'Color',[.8 .8 .8]),Xedges)
    arrayfun(@(y)yline(y,'Color',[.8 .8 .8]),Yedges)
    % colorbar
    cb = colorbar();
    ylabel(cb,'density')
end
```

A.3. NONLINEAR TRIM WITH DIFFUSION

solveFPE.m

```
%% Forward/Backward Trim Flight Characterization
% AUTHOR: Troy Newhart

% DATE: 4 April 2023
% ORGANIZATION: Old Dominion University

% DESCRIPTION: Uses an Alternating Directions Implicit method to solve the
% Fokker-Planck PDE with diffusion, finding the transient joint probability
distribution
% of the response.

% REFERENCES:
% [1] "Numerical Solution of the Fokker-Planck Equation by Finite
% Difference and Finite Element Methods - A Comparative Study," L. Pichler,
% A. Masud, and L.A. Bergman.

% FUNCTION CALLS:
% matrix_fwd.m, matrix_rvs.m ADI_fwd.m,, ADI_rvs.m matrixTridiag.m, solveTridiag.m

%% Initialize
clear
clc

%% User Inputs
dt = 0.005; % Finite difference timestep
T_end = 3; % End time
M1 = 0.5; M2 = 0.5; % Grid bounds for y(1) = [-M1,M1] and y(2) = [-M2, M2]
N1 =990; N2 = 990; % Number of nodes along each variable axis
Ds = [0, 0.01]; % Scaling factor for Gaussian white noise on diffusion
```

```
%% Solve the forward problem
[D1,D2,x1,x2,DF1,DF2]=matrix_fwd(M1,N1,M2,N2,Ds); % Formulate the tridiagonal
matrices of each state variable
[t,p]=ADI_fwd(D1,D2,x1,x2,dt,T_end,N1,N2,DF1,DF2); % Alternating Directions Implicit
Method to solve FP PDE for p

%% Plot desired time steps
time = 3; % Specify the time step of interest
t_ind = time/dt + 1; % Time index of the specified time of interest

surf(x2,x1,p(:,:,t_ind),'edgecolor','none')
view(90,90)
xlabel('x_2')
ylabel('x_1')
title("Reverse Nonlinear Trim, time = "+time+" sec")
cb = colorbar();
ylabel(cb,'density')
```

matrix_fwd.m

```
function [D1,D2,x1,x2,DF1,DF2]=matrix_fwd(M1,N1,M2,N2,Ds)

% INPUTS: Discretized the spatial domain: x1 = [-M1, M1], x2 = [-M2, M2]
%   with number of nodes N1 and N2 (matrix size = (N1*N2)x(N1*N2) and vectors
%   (N1*N2)x1
% OUTPUT: D1,D2 tridiagnal matrices and x1, x2 grid points
% FUNCTION CALLS: matrixTridiag.m

%% Initialize finite difference grid
dx1=2*M1/(N1-1);
if(dx1<0)
    disp('Discretization of x1 cannot be negative.');
    return;
end
x1=(-M1:dx1:M1)';

dx2=2*M2/(N2-1);
if(dx2<0)
    disp('Discretization of x2 cannot be negative');
    return;
end
x2=(-M2:dx2:M2)';

%% Compute tridiagonal drift matrix over x1 span
a = []; b = zeros(N1*N2,1)'; c = [];
for i=1:N2
    a_coeff = (x1(1:end-1)*0 + x2(i))./(2*dx1);
    c_coeff = (x1(1:end-1)*0 - x2(i))./(2*dx1);

    a = [a, 0, a_coeff'];
    c = [c, c_coeff', 0];
end
D1.a = a; D1.b = b; D1.c = c;
```

```
D1.matrix = matrixTridiag(a,b,c);

%% Compute tridiagonal drift matrix over x2 span
a = []; b = zeros(N1*N2,1)'; c = [];
for i=1:N1
    a_coeff = (-5*0.1*(1-x1(i).^2)*x2(1:end-1)-9*sin(x1(i)))./(2*dx2);
    c_coeff = (5*0.1*(1-x1(i).^2)*x2(1:end-1)+9*sin(x1(i)))./(2*dx2);

    a = [a, 0, a_coeff'];
    c = [c, c_coeff', 0];
end
D2.a = a; D2.b = b; D2.c = c;
D2.matrix = matrixTridiag(a,b,c);

%% Compute tridiagonal diffusion matrix over x1 span
a = []; b = []; c = [];
for i=1:N2
    a = [a, 0, Ds(1)*(ones(N1-1,1)./(dx1^2))'];
    c = [c, Ds(1)*(ones(N1-1,1)./(dx1^2))',0];
    b = [b, -2*Ds(1)*(ones(N1,1)./(dx1^2))'];
end
DF1.a = a; DF1.b = b; DF1.c = c;
DF1.matrix = matrixTridiag(a,b,c);

%% Compute tridiagonal diffusion matrix over x2 span
a = []; b = []; c = [];
for i=1:N1
    a = [a, 0, Ds(2)*(ones(N2-1,1)./(dx2^2))'];
    c = [c, Ds(2)*(ones(N2-1,1)./(dx2^2))',0];
    b = [b, -2*Ds(2)*(ones(N2,1)./(dx2^2))'];
end
DF2.a = a; DF2.b = b; DF2.c = c;
DF2.matrix = matrixTridiag(a,b,c);
```

ADI_fwd.m

```
function [t,p]=ADI_fwd(D1,D2,x1,x2,dt,T_end,N1,N2,DF1,DF2)

% INPUTS: Tridiagonal matrices for each variable (D1, D2), variable grid
% (x1, x2), timestep (dt), end time (T_end)
% OUTPUT: time points and joint transient PDF
% FUNCTION CALLS: solveTridiag.m

t=(0:dt:T_end)';
p=zeros(N1,N2,length(t)); % Initialize for speed

% Create A matrix and b vector for solving A*p = b
rhs_first = sparse(speye(N1*N2)+.5*dt*(D1.matrix+DF1.matrix)); % First A matrix
rhs_second = sparse(speye(N1*N2)+.5*dt*(D2.matrix+DF2.matrix)); % Second A matrix

a_first =-.5*dt*(D2.a+DF2.a);
b_first =ones(N1*N2,1)'-.5*dt*(D2.b+DF2.b);
c_first =-.5*dt*(D2.c+DF2.c);

a_second =-.5*dt*(D1.a+DF1.a);
```

```
b_second =ones(N1*N2,1)'-.5*dt*(D1.b+DF1.b);
c_second =-.5*dt*(D1.c+DF1.c);

[X1,X2] = meshgrid(x1,x2); % Mesh the discretized spatial domain

%% Assign Initial Conditions
mu = [0, 0.4]; sigma = 0.001*eye(2); % Uncertain Initial conditions
p(:,:,1) = reshape(mvnpdf([X1(:) X2(:)],mu,sigma),length(x2),length(x1))'; % Joint
PDF of uncertain initial conditions

%% Compute the numerical solution to the Fokker-Planck Equation
p = reshape(p,N1*N2,length(t)); % Shape p into a vector for computations

tic()
for j=2:length(t)

    % Track progress of the computations
    if rem(j,100)==0
        fprintf('Number of Iteration %d out of %d finished...\n',j,length(t));
    end

    % Solve for p^(n+1/2)
    reshaped_r = reshape(reshape(rhs_first*p(:,j-1),N1,N2)',N1*N2,1);
    p_half = solveTridiag(a_first,b_first,c_first,reshaped_r);

    % Solve for p^(n+1)
    reshaped_r = reshape(reshape(rhs_second*p_half,N2,N1)',N1*N2,1);
    p(:,j) = solveTridiag(a_second,b_second,c_second,reshaped_r);

end
toc()

p = reshape(p,N1,N2,length(t)); % Reshape into matrix for surface plotting
```

matrixTridiag.m

```
function [ Mat ] = matrixTridiag( a,b,c )
% OUTPUT: a sparse tridiagonal matrix given a, b, c

%% Ensure vectors are the same length
N = length(a);
if (length(b) ~= N || length(c) ~= N)
    fprintf('Length of a, b, c not the same.\n');
end

%% Formulate tridiagonal matrix
x_indx = [2:N,1:N,1:N-1]; % Index in the first dimension
y_indx = [1:N-1,1:N,2:N]; % Index in the second dimension
trid = [a(2:end),b(1:end),c(1:end-1)]; % Tridiagonal matrix

Mat = sparse(x_indx,y_indx,trid,N,N); % Remove zeroes to save space on memory

end
```

solveTridiag.m

```
function d_prime = solveTridiag(a,b,c,d) % Use Thomas algorithm to efficiently solve
inverse of tridiagonal matrix
% INPUTS: LHS tridiagonal vectors and RHS matrix
% OUTPUT: Probability density
% REFERENCES: https://en.wikipedia.org/wiki/Tridiagonal_matrix_algorithm

% Initialize variables
N = length(a);
d_prime = zeros(N,1);
c_prime = zeros(N,1);

b_tilde = b(1) ;
d_prime(1) = d(1)/b_tilde ;
% Decomposition of the special case of Gaussian elimination
for j = 2:N
    c_prime(j)=c(j-1)/b_tilde;
    b_tilde=b(j)-a(j)*c_prime(j);
    d_prime(j)=(d(j)-a(j)*d_prime(j-1))/b_tilde;
end
% Backward substitution to find all of the unknowns
for j = 1:(N-1)
    k = N-j ;
    d_prime(k) = d_prime(k) - c_prime(k+1)*d_prime(k+1);
end
```

VITA

Troy S. Newhart

Mechanical & Aerospace Engineering
Old Dominion University
214A Kaufman Hall, Norfolk, VA 23529

Email:
tsn5048@gmail.com

Education:

M.S. Aerospace Engineering, Old Dominion University, December 2019
B.S. Aerospace Engineering, The Pennsylvania State University, May 2015