QUANTIFYING THE UNCERTAINTY OF CERTAIN PROBLEMS

An Undergraduate Thesis

Submitted to the Department of Aerospace and Mechanical Engineering
of the University of Notre Dame
in Partial Fulfillment of the Requirements
for the Degree of

Bachelor of Science
in
Mechanical Engineering

by
Steven R. Brill

Dr. Joseph M. Powers, Co-Director

Dr. Waad Suber, Co-Director

Department of Aerospace and Mechanical Engineering
Notre Dame, Indiana
May 2016

QUANTIFYING THE UNCERTAINTY OF CERTAIN PROBLEMS

Abstract

by

Steven R. Brill

This thesis examines the use of Bayesian statistics for uncertainty quantification in simple engineering applications. Experimental data has uncertainty, which can make it difficult to determine the true values of parameters. Bayesian methods consider probability to be a state of knowledge, contrary to the traditional frequentist view of statistics. Leveraging this view, Bayesian inference uses prior knowledge and data to estimate parameters. It was demonstrated that Bayesian inference can reproduce the results of commonly used deterministic methods, like the method of least squares, for linear problems with Gaussian uncertainty. The Bayesian method was shown to be advantageous because it is statistical, so the degree of confidence in a solution can be found. Also, applying prior knowledge about a parameter can cause the method to converge more quickly than competing methods. Next, Bayesian inference was applied to determine the bias of a coin, the stiffness of a spring, and the radioactive decay rate constant of a material. These studies showed how the accurate priors improve convergence and how data can be used to overpower bad priors. The Monte Carlo method was used to show how Bayesian methods are implemented when there is no available analytical solution. These applications serve as examples of how Bayesian inference is implemented in engineering.

CONTENTS

FIGURES

TABLES

SYMBOLS

$P(q)$	Probability of statement q
$P(q\|r)$	Probability of statement q given statement r is true
$P(q,r)$	Probability of statements q and r
N	Number of data points in a study
d_i	The ith data point in a study
$\mathbf{D}$	The collection of all data points d_i
θ	The exact value of a parameter
ξ	A random variable
$\mathcal{N}(\mu,\sigma)$	Normal distribution with a mean of μ and a standard deviation of σ
μ	The mean of a distribution or set of data
σ	The standard deviation of a distribution or set of data
θ_μ	The estimate for parameter θ using the mean
r_i	The residual of data point i in the method of least squares
θ_{LS}	The estimate for parameter θ using the method of least squares
R	The sum of the residuals of all data points
θ_{MLE}	The estimate for parameter θ using the maximum likelihood estimation
θ_{MPE}	The estimate for parameter θ using the maximum posterior estimation
H	The bias of a coin towards heads
F	The force exerted on the spring
k	The spring constant of the spring
x	The true value of the displacement of the spring under load F

ACKNOWLEDGMENTS

I would like to thank my advisor Dr. Joseph Powers for giving me the opportunity to work with him. His guidance throughout my research has been invaluable, but his overwhelming enthusiasm for his work will inspire me for the rest of my career.

Also, I would like to thank Dr. Samuel Paolucci, Dr. Waad Subber, Dr. Temistocle Grenga and all of the members of the Center for Shock Wave-processing of Advanced Reactive Materials for their advising and support on my research projects. Additionally, I am grateful to the members of my committee, Dr. Meng Wang and Dr. Joannes Westerink. Finally, I would like to thank my parents, family, and friends for their constant support throughout my time at Notre Dame.

CHAPTER 1

INTRODUCTION

The goal of computational science is to develop models that predict phenomena observed in nature. However, these models are often based on parameters that are uncertain. For example, when making a weather prediction, simulations are created based off a temperature data at a finite number of locations. In order to create a perfect model, one would need the entire temperature field of the atmosphere [1]. Uncertainty in the initial problem formulation affects future calculations and has an impact on the final solution. Hence, it is necessary to understand how the uncertainty manifests itself in the solution and to reduce the uncertainty in applications. These two goals are the focus of the field of uncertainty quantification (UQ). UQ provides a method to develop confidence in a solution from a numerical simulation and to create models from uncertain, experimental data [2].

First, uncertainty must be defined. Uncertainty implies a partial lack of knowledge about a parameter in an analysis. This means that uncertainties are related to the model of physical systems. Colloquially, the term error is often used in a similar context, particularly when discussing measurements of an experimental setup. However, in this context, errors are defined as deficiencies due to converting a mathematical formula into a numerical algorithm in a computational code [3]. Errors could be mathematical errors in a particular numerical scheme, code bugs, or limitations in the computer, like round-off error. In these cases, the error is known. Hence, the true solution can be calculated by accounting for the error. Thus, the term error is reserved for more "certain" values. Additionally, defining errors emphasizes that errors occur in the mathematics of the problem and uncertainties occur in the physical process being modeled [2].

There are two types of uncertainties: aleatoric and epistemic uncertainties. Aleatoric uncertainties are due to randomness in the system. Some examples are manufacturing tolerances, material properties between samples of the same object, signal noise, and measurement bias [2]. This type of uncertainty is also called irreducible uncertainty, because it is always present in the system and therefore cannot be reduced, even with large numbers of trials. On the other hand, epistemic uncertainty is a lack of knowledge about the value of a parameter that does not actually vary [4]. For example, the material properties of a single sample or turbulence model assumptions are epistemic uncertainties, because there are true values of these properties. These uncertainties are called reducible uncertainties, because they can be reduced by learning more about the subject through testing and other methods.

Within UQ, there are two types of problems: forward and inverse problems. Forward problems use models to study how uncertainty propagates through a given model, while inverse problems estimate input parameters for models based off uncertain experimental data. Although the latter classification of problems will be the focus of this thesis, some commonly used methods for forward problems are: Monte Carlo simulations [5], polynomial chaos [6], and stochastic collocation methods [7]. In this thesis, I will present a study of Bayesian inference in inverse UQ problems. The advantages of the method when compared to deterministic parameter estimation methods, like the method of least squares, will be shown. Bayesian methods will be applied to estimate the bias in a coin, the spring constant for a spring in static equilibrium, and the decay rate constant of a radioactive material.

CHAPTER 2

BAYESIAN METHODS

This chapter presents the history and philosophy behind Bayesian inference and the mathematical formulation of Bayes' theorem. Bayesian methods need philosophical justification because they approach statistics in a nontraditional way. Then the main mathematical tools for Bayesian methods are defined.

2.1 History and Philosophy

In order to fully understand Bayesian inference, one must understand its place in the history of statistics. In 1713, Bernoulli questioned how to apply the deductive logic of games of chance to inductive logic that is used in science [8]. In 1763, Bayes provided a solution in Bayes' theorem, which will be detailed later [9]. In 1812, Laplace independently discovered Bayes' theorem, clarified it, and applied it to a number of scientific problems [10]. However, the work of Bayes and Laplace was rejected by most scientists because they defined a probability as a degree-of-belief: a measure of the confidence in a solution based on what is known. Most scholars of the day rejected this notion, because it seemed subjective [11] and they proceeded to define probability as the frequency of an event given infinitely many trials [12].

The frequentists' approach uses probability as a tool to study random phenomena. Hence, the methods can be used to study processes like flipping a coin or rolling a die, but not estimating the mass of Saturn, because the mass is not a random variable. In order to use the frequentists' method to study the mass of Saturn, a large number of universes would have to be considered with varying masses of Saturn. This goes to show that the frequentists' definition of probability limits its applicability. In fact, it limits it from most of the inductive cases that Bernoulli was trying to address. Regardless, the frequentists' probability theory was developed for about one hundred years until Bayes' idea reappeared [12].

Since then, Jeffreys [13] and Jaynes [14] have further developed Bayes' theory. Viewing probability as a state of knowledge enables the study of the mass of Saturn, because although it is not a random variable, it is unknown, which is a part of the state of knowledge. One could argue that this view is subjective, because two people could know different information about a problem. However, this is not subjectivity, but rather shows that all probabilities are conditional based on the information that one has available. If two people have the same information, they will still assign the same probability, which means the method is objective [12]. Additionally, applying relevant prior knowledge is natural in the human thought process [15]. When a family moves into a neighborhood and they see a green car driving back and forth up the street, they may think the driver is looking for houses to rob. This is a reasonable thought process. However, if the family knows that someone down the street drives a green car and is teaching their son to drive, then they will not worry about being robbed, because the boy is practicing his driving. In both cases, the data is the same, but having prior knowledge affects the family's decisions. This example shows how people unconsciously use prior knowledge in their choices everyday.

In order to emphasize the advantages of the Bayesian view of probability against the frequentists', consider the so-called "Monte Hall" problem from the game show "Let's Make a Deal". The contestant chooses one of three doors and wins the prize behind it. Behind two of the doors are goats and behind one is a brand new car. After the contestant chooses a door, the host reveals a goat from behind one of the other two doors and then asks if the contestant would like to switch his choice. If the contestant switches his or her guess, what is the probability that the original guess was the car? Using a frequentist view, because there are two options, there is a 50% probability. However, what if the contestant knows a goat is behind the new door? Then it is certain that the original door had the car. Although the fact that the goat is behind the second door has no effect on what is behind the first door, the knowledge of the new door affects the prediction of the first door. Hence, viewing probability as a state of knowledge and using additional knowledge, outside of the system, enables one to better predict the results [12].

2.2 Bayes' Theorem

As was stated, in Bayesian inference, all probabilities are conditional based on all of the relevant information that one has. Hence, the probability of a proposition, Z, is written $P(Z|I)$, where I is all of the relevant information. For the remainder of this thesis the conditional part of the expression will be implied, but not be written.

Mathematically, Bayes' theorem is the vehicle for Bayesian inference. It is derived as follows. First, the product rule of probability is:

$$P(A,B) = P(A|B)\,P(B), \tag{2.1}$$

$$P(B,A) = P(B|A)\,P(A), \tag{2.2}$$

where $P(A,B)$ and $P(B,A)$ are the probability that both statements A and B are true. Then, because the "and" operator is commutative, $P(A,B) = P(B,A)$. Applying the product rule to both sides gives:

$$P(A|B)\,P(B) = P(B|A)\,P(A). \tag{2.3}$$

Rearranging gives Bayes' theorem:

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}. \tag{2.4}$$

Bayes' theorem is often used to determine the probability of a hypothesis or parameter based off of data. In order to make this connection obvious, let A be the hypothesis and B be the data:

$$\underbrace{P(\text{hypothesis}|\text{data})}_{\textit{Posterior}} \propto \underbrace{P(\text{data}|\text{hypothesis})|}_{\textit{Likelihood}}\ \underbrace{P(\text{hypothesis})}_{\textit{Prior}}. \tag{2.5}$$

In this relationship, the $P(\text{hypothesis})$ was removed and the equality was replaced with a proportionality because in parameter estimation problems the term is simply a normalization constant.

Each of the terms in Eq. (2.5) have formal name: $P(\text{hypothesis})$ is called the prior probability, $P(\text{data}|\text{hypothesis})$ is the likelihood function, and $P(\text{hypothesis}|\text{data})$ is the posterior probability. Bayes' theorem determines the probability of the hypothesis by considering prior knowledge about the hypothesis and experimental data. The prior represents the probability that the hypothesis is true before any data is considered. As the name implies, the prior is used to apply previously known information to the estimation. The likelihood function factors the data into the prediction by considering the probability that the data is obtained, given the model is true. The posterior probability is the probability of the hypothesis in light of the prior knowledge and data. The ability to consider prior knowledge in the study is a result of Bayes' state of knowledge view of statistics.

In this process, the prior knowledge is used as an initial guess. Then the likelihood function is used to incorporate the experimental data into the theory. Because the prior knowledge is independent of the data, the prior and likelihood function are multiplied to determine the resulting probability. When dealing with hypotheses and data, Bayes' theorem is normalized by enforcing the condition:

$$\int_{\Gamma} P(\text{data}|\text{hypothesis})\, d(\text{data}) = 1, \tag{2.6}$$

where Γ is the support of the data. The support of a function is the set of points where the function is not zero.

CHAPTER 3

PARAMETER ESTIMATION

This chapter compares Bayesian inference to common deterministic methods for parameter estimation. The true value behind a set of measurements is estimated using the mean, the method of least squares, the maximum likelihood, and Bayesian inference with a constant prior. A similar exercise is repeated for a linear model. These exercises show that the Bayesian method not only reproduces the results from the deterministic methods, but provides a more informative solution.

3.1 Measurement

Bayesian inference is a method for parameter estimation based off data and prior knowledge. However, there are many competing methods. Consider the case when N data points, d_i, are taken to measure a single phenomena:

$$\mathbf{D} = \{d_1, d_2, ..., d_N\}\,, \tag{3.1}$$

$$\mathbf{D} = \{d_i\}_{i=1}^{i=N}\,. \tag{3.2}$$

For example, the mass of a gold nugget or the temperature of a potato at room temperature could be repeatedly measured. If multiple measurements are conducted, they will each be different due to uncertainties in measurement devices. The uncertainty in the data can be written:

$$d_i = \theta + \xi, \tag{3.3}$$

where θ is the true value and ξ is a random number sampled from some distribution. With uncertainty in the measurements, how should one find the true value of the measured quantity?

3.1.1 Average

When presented with the above problem, most people would take the average of the data to find the most accurate solution. This method is very simple. Mathematically it is written:

$$\theta_{avg} = \frac{\sum_{i=1}^{N} d_i}{N}, \tag{3.4}$$

where θ_{avg} is the average value of the data. By definition, the average is the arithmetic mean of the data.

3.1.2 Least Squares

The method of least squares is another commonly used method for interpreting experimental data. In this method, the solution is expected to be a value θ_{LS}. Define the residual: $r_i = d_i - \theta_{LS}$. Then the sum of the residuals of the data and θ_{LS} is calculated:

$$R = \sum_{i=1}^{N} r_i^2, \tag{3.5}$$

$$R = \sum_{i=1}^{N} (d_i - \theta_{LS})^2. \tag{3.6}$$

Then the residual is minimized to with respect to the data determine the value of θ_{LS} that best represents the data:

$$\begin{aligned}
\frac{\partial R}{\partial \theta_{LS}} &= 0, && (3.7)\\
-1(2)\sum_{i=1}^{N}(d_i - \theta_{LS}) &= 0, && (3.8)\\
\sum_{i=1}^{N}(d_i - \theta_{LS}) &= 0, && (3.9)\\
\left(\sum_{i=1}^{N} d_i\right) - N\theta_{LS} &= 0, && (3.10)\\
\theta_{LS} &= \frac{\sum_{i=1}^{N} d_i}{N}. && (3.11)
\end{aligned}$$

Hence, the mean of the data is the value that minimizes the least squares residual.

3.1.3 Maximum Likelihood Estimation

For this case, assume ξ in Eq. (3.3) is sampled from a normal distribution, $\xi \sim \mathcal{N}(0, \sigma)$, where $\mathcal{N}(0, \sigma)$ is the normal distribution with a mean of 0 and a standard deviation of σ. Another way to estimate the true value of the data is to determine the likelihood of the data. Because we assume the uncertainty in the data is Gaussian, the likelihood of a data point, d_i, is a Gaussian distribution:

$$P(d_i|\theta_{MLE}) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(d_i - \theta_{MLE})^2}{2\sigma^2}\right) \tag{3.12}$$

as shown in Fig. **??**. Because each data point is independent of the others, the probability of all of the data points is represented by the product of all of the likelihood functions:

$$P(\mathbf{D}|\theta_{MLE}) = \prod_{i=1}^{N} P(d_i|\theta_{MLE}), \tag{3.13}$$

$$P(\mathbf{D}|\theta_{MLE}) = \prod_{i=1}^{N} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(d_i - \theta_{MLE})^2}{2\sigma^2}\right), \tag{3.14}$$

$$P(\mathbf{D}|\theta_{MLE}) = \frac{1}{(\sigma\sqrt{2\pi})^N} \exp\left(\frac{-\sum_{i=1}^{N}(d_i - \theta_{MLE})^2}{2\sigma^2}\right). \tag{3.15}$$

The most probable value in the likelihood function is the maximum value, so the maximum value is the best estimate for θ. In order to simplify calculations, define L as the natural logarithm of the likelihood function:

$$L = \ln\left(P(\mathbf{D}|\theta_{MLE})\right), \tag{3.16}$$

$$L = \ln\left(\frac{1}{(\sigma\sqrt{2\pi})^N}\right) + \frac{-\sum_{i=1}^{N}(d_i - \theta_{MLE})^2}{2\sigma^2}. \tag{3.17}$$

Because the logarithm is monotonically increasing, the location of the maximum of L is the same as the maximum of $P(\mathbf{D}|\theta_{MLE})$. Hence, the extreme values of L exist at points where its first derivative is zero:

$$\frac{\partial L}{\partial \theta_{MLE}} = 0, \tag{3.18}$$

$$2\frac{1}{2\sigma^2}\sum_{i=1}^{N}(d_i - \theta_{MLE}) = 0, \tag{3.19}$$

$$\sum_{i=1}^{N}(d_i - \theta_{MLE}) = 0, \tag{3.20}$$

$$\left(\sum_{i=1}^{N} d_i\right) - N\theta_{MLE} = 0, \tag{3.21}$$

$$\theta_{MLE} = \frac{\sum\limits_{i=1}^{N} d_i}{N}. \tag{3.22}$$

The second derivative of L determines if the point is a minimum or maximum:

$$\frac{\partial^2 L}{\partial \theta_{MLE}^2} = -\frac{N}{\sigma^2}. \tag{3.23}$$

Hence the point is the maximum of the likelihood function. Equation (3.22) shows that the maximum likelihood estimation is exactly equal to the mean of the data.

3.1.4 Bayesian Inference with Uniform Prior

Finally, Bayes' theorem can also be used to estimate the true value of the data. The most likely value of the parameter will be the maximum of the posterior probability density function (pdf). In this case, a uniform prior will be used. A uniform prior can be used to limit the range of possible parameter values if no other information is known. For example, if it is known that the value of θ_{MPE} is between θ_{min} and θ_{max}, then the uniform prior would be:

$$P(\theta_{MPE}) = \frac{1}{\theta_{max} - \theta_{min}}, \qquad \text{for } \theta_{min} \leq \theta_{MPE} \leq \theta_{max}, \tag{3.24}$$

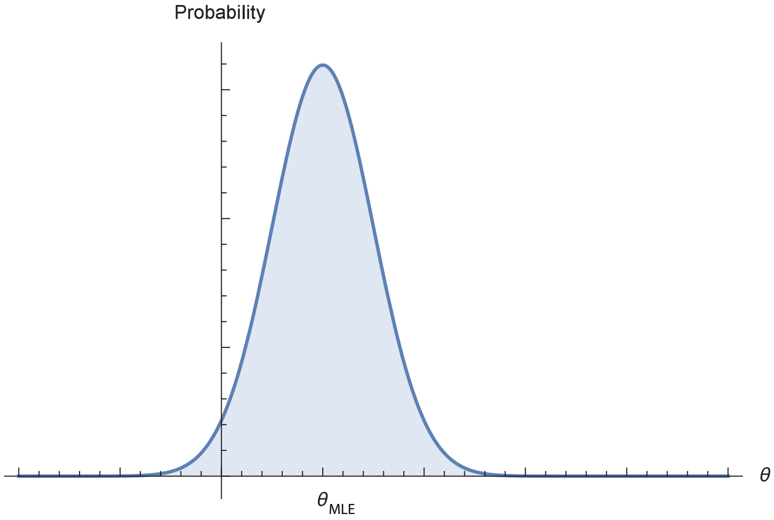

Figure 3.1. A general Gaussian likelihood function for parameter θ.

as shown in Figure 3.1.4. Note that the prior is a pdf, so it integrates to unity. Also, a likelihood function must be used. In this case, the likelihood function is the same as in Eq. (3.15). Applying Bayes' theorem gives:

$$P(\theta_{MPE}|\mathbf{D}) = \kappa P(\mathbf{D}|\theta_{MPE})P(\theta_{MPE}), \tag{3.25}$$

$$P(\theta_{MPE}|\mathbf{D}) = \frac{\kappa}{(\theta_{max} - \theta_{min})(\sigma\sqrt{2\pi})^N} \exp\left(\frac{-\sum_{i=1}^{N}(d_i - \theta_{MPE})^2}{2\sigma^2}\right)$$
$$\text{for } \theta_{min} \leq \theta_{MPE} \leq \theta_{max}, \tag{3.26}$$

where κ is a normalization constant. Similarly to the likelihood method, the most probable value of θ_{MPE} corresponds to the maximum of the function. Also, define L:

$$L = \ln\left(P(\theta_{MPE}|\mathbf{D})\right), \tag{3.27}$$

$$L = \ln\left(\frac{\kappa}{(\theta_{max} - \theta_{min})(\sigma\sqrt{2\pi})^N}\right) + \frac{-\sum_{i=1}^{N}(d_i - \theta_{MPE})^2}{2\sigma^2}. \tag{3.28}$$

Maximizing L gives:

$$\frac{\partial L}{\partial \theta_{MPE}} = 0, \tag{3.29}$$

$$2\frac{1}{2\sigma^2}\sum_{i=1}^{N}(d_i - \theta_{MPE}) = 0, \tag{3.30}$$

$$\sum_{i=1}^{N}(d_i - \theta_{MPE}) = 0, \tag{3.31}$$

$$\left(\sum_{i=1}^{N} d_i\right) - N\theta_{MPE} = 0, \tag{3.32}$$

$$\theta_{MPE} = \frac{\sum_{i=1}^{N} d_i}{N}. \tag{3.33}$$

Hence, the mean of the data is the best value of θ_{MPE} using the Bayesian inference with a uniform prior.

3.1.5 Discussion

This study shows that for cases where the true value of a parameter is estimated from data that has Gaussian uncertainty, the method of least squares, Eq. (3.11), maximum likelihood estimation, Eq (3.22), and maximum posterior estimation through Bayesian inference with a uniform prior, Eq. (3.33), all determine that the mean is the best estimate of the true value. One may ask, “If all of the

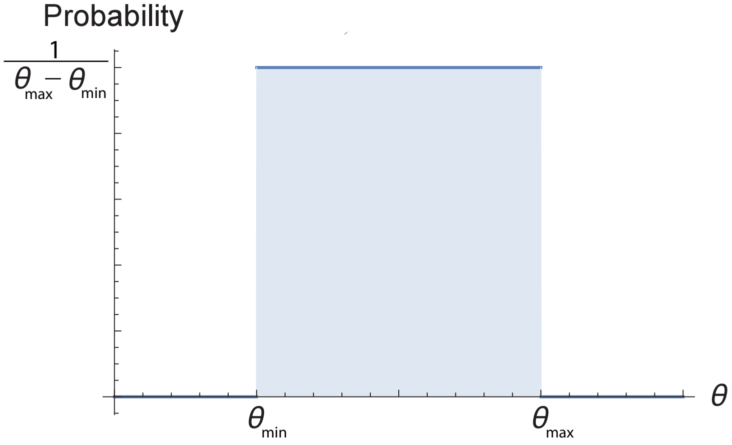

Figure 3.2. A general uninformative prior for parameter θ.

methods produce the same results, why should I use Bayesian inference over taking the mean or the method of least squares?" While all of the methods do come to the same solution, calculating the mean directly and the method of least squares do not provide as much information as the Bayesian method, because they are deterministic methods. The Bayesian method produces not just a number, but a pdf. The pdf can be further characterized with its mean, variance, and other moments. Additionally, because the pdf is a Gaussian, the mean, median, and mode will result in the same choice of θ. If the uncertainty in the data were not Gaussian, these quantities could still be calculated, but they would most likely not all be equivalent. The most important of these statistics is the confidence interval, which quantifies the confidence in the solution. Furthermore, the pdf from the Bayesian method acknowledges that there is uncertainty in the solution, but through the location of the maximum and the confidence intervals, it quantifies the uncertainty. The same cannot be said for the method of least squares, which simply yields the resulting parameter. Additionally, in this study, an uninformed prior was chosen. Using another, more informed prior could result in additional improvements.

3.2 Linear Model

Now consider a linear model $x = \theta y$, where θ is the quantity of interest. There are N data points, d_i for x corresponding to y_i values. It is assumed that the values of y_i were measured without uncertainty. Again, assume there is Gaussian uncertainty in the data, d_i with zero mean and σ standard deviation:

$$d_i = x + \xi, \qquad \xi \sim \mathcal{N}(0, \sigma). \tag{3.34}$$

Given the uncertainty in x_i the goal is to find the true value of θ.

3.2.1 Least Squares

The method of least squares is used by determining the value of θ_{LS} that minimizes the residual between the data and the model:

$$R = \sum_{i=1}^{N} (d_i - \theta_{LS} y_i)^2, \tag{3.35}$$

$$\frac{\partial R}{\partial \theta_{LS}} = 0, \tag{3.36}$$

$$-2 \sum_{i=1}^{N} (d_i - \theta_{LS} y_i) y_i = 0, \tag{3.37}$$

$$\sum_{i=1}^{N} (d_i - \theta_{LS} y_i) y_i = 0, \tag{3.38}$$

$$\sum_{i=1}^{N} (d_i y_i - \theta_{LS} y_i^2) = 0, \tag{3.39}$$

$$\theta_{LS} = \frac{\sum\limits_{i=1}^{N} d_i y_i}{\sum\limits_{i=1}^{N} y_i^2}. \tag{3.40}$$

If all of the data points are measured for the same y, then θ_{LS} is simply the average of d_i/y.

3.2.2 Maximum Likelihood Estimation

The likelihood function method is applied in a similar way to the last study.

$$f_i = \frac{1}{\sigma\sqrt{2\pi}}\exp\left(\frac{-(d_i-\theta_{MLE}y_i)^2}{2\sigma^2}\right), \tag{3.41}$$

$$F = \prod_{i=1}^{N} f_i = \frac{1}{(\sigma\sqrt{2\pi})^N}\exp\left(\sum_{i=1}^{N}\frac{-(d_i-\theta_{MLE}y_i)^2}{2\sigma^2}\right), \tag{3.42}$$

$$L = \ln(F) = \ln\left(\frac{1}{(\sigma\sqrt{2\pi})^N}\right) + \sum_{i=1}^{N}\frac{-(d_i-\theta_{MLE}y_i)^2}{2\sigma^2}, \tag{3.43}$$

$$\frac{\partial L}{\partial\theta_{MLE}} = 0, \tag{3.44}$$

$$\frac{-1}{2\sigma^2}\left(2(-1)\sum_{i=1}^{N}-(d_i-\theta_{MLE}y_i)y_i\right) = 0, \tag{3.45}$$

$$\sum_{i=1}^{N}-(d_iy_i-\theta_{MLE}y_i^2) = 0., \tag{3.46}$$

$$\theta_{MLE} = \frac{\displaystyle\sum_{i=1}^{N} d_iy_i}{\displaystyle\sum_{i=1}^{N} y_i^2}. \tag{3.47}$$

3.2.3 Bayesian Inference with Uniform Prior

When solving this problem with Bayesian inference, the same likelihood function and prior are used.

$$P(\mathbf{D}|\theta_{MPE}) = \frac{1}{(\sigma\sqrt{2\pi})^N}\text{exp}\left(\sum_{i=1}^{N}\frac{-(d_i - \theta_{MPE}y_i)^2}{2\sigma^2}\right), \tag{3.48}$$

$$P(\theta_{MPE}) = \frac{1}{\theta_{max} - \theta_{min}}, \qquad \text{for } \theta_{min} \leq \theta_{MPE} \leq \theta_{max}, \tag{3.49}$$

$$P(\theta_{MPE}|\mathbf{D}) = \kappa P(\mathbf{D}|\theta_{MPE})P(\theta_{MPE}), \tag{3.50}$$

$$P(\theta_{MPE}|\mathbf{D}) = \frac{\kappa}{(\theta_{max} - \theta_{min})(\sigma\sqrt{2\pi})^N}\text{exp}\left(\sum_{i=1}^{N}\frac{-(d_i - \theta_{MPE}y_i)^2}{2\sigma^2}\right)$$

$$\text{for } \theta_{min} \leq \theta_{MPE} \leq \theta_{max}, \tag{3.51}$$

$$L = \ln(P(\theta|\mathbf{D})) = \ln\left(\frac{\kappa}{(\theta_{max} - \theta_{min})(\sigma\sqrt{2\pi})^N}\right) + \sum_{i=1}^{N}\frac{-(d_i - \theta_{MPE}y_i)^2}{2\sigma^2}, \tag{3.52}$$

$$\frac{\partial L}{\partial \theta_{MPE}} = 0, \tag{3.53}$$

$$\frac{-1}{2\sigma^2}\left(2(-1)\sum_{i=1}^{N}(d_i - \theta_{MPE}y_i)y_i\right) = 0, \tag{3.54}$$

$$\sum_{i=1}^{N} -(d_iy_i - \theta_{MPE}y_i^2) = 0, \tag{3.55}$$

$$\theta_{MPE} = \frac{\sum_{i=1}^{N} d_iy_i}{\sum_{i=1}^{N} y_i^2}. \tag{3.56}$$

3.2.4 Discussion

In this study, it can be seen that the likelihood method and the Bayesian method with a uniform prior both come to the same conclusion as the method of least squares. This conclusion is true in the case of Gaussian uncertainty. Once again, although the solutions are the same, the Bayesian method is more informative because it makes it possible to quantify the uncertainty in the problem, which will be demonstrated in later examples. Additionally, using a more informative prior would lead to a better solution, which will be demonstrated later. The study of the linear problem shows that the Bayesian method matches and provides more information than the conventional least squares method even for a more complicated problem.

CHAPTER 4

BIASED COIN

In this chapter, a simple example of a biased coin is used to demonstrate the procedure for using Bayesian inference. Different priors are considered to show the effects of the prior and increasing the number of data points.

4.1 Problem Statement

This example will use Bayes' theorem to determine the probability that a coin is biased towards heads. An example application of this is if someone enters a casino and is betting on coin flips. If the coin results in 8 heads in 14 flips, is it a fair coin? The fairness of the coin is denoted by H, which is the bias-weighting of the coin. If $H = 0$, then the coin will result in all tails. If $H = 1$, then it will give all heads. If $H = 0.5$, then the coin is a fair coin. H could be any value between 0 and 1, so the probability of H can be represented by a continuous pdf from 0 to 1 , where a higher density indicates a confidence that a certain value of H is more likely to be the correct bias of the coin [12].

To create predictions as data points are sampled, Bayes' theorem as in Eq. (2.5) is used. In this example, the hypothesis is that the coin has a bias of H and the goal is to estimate the parameter H. Bayes' theorem becomes

$$P(H|\text{data}) \propto P(\text{data}|H)\,P(H), \tag{4.1}$$

and it is normalized by

$$\int_{-\infty}^{\infty} P(H|\text{data})\; dH = 1. \tag{4.2}$$

4.2 Prior

First, a prior must be created based on the initial knowledge. Different priors could be used depending on the perceived bias of the coin flipper. However, first, assume that one knows nothing about the coin and that all possible bias-weightings are equally likely. This is done by taking the prior to be a uniform distribution:

$$P(H) = \begin{cases} 1, & 0 \leq H \leq 1, \\ 0, & \text{otherwise.} \end{cases} \tag{4.3}$$

4.3 Likelihood Function

Next, a likelihood function that models the behavior of the coin must be selected. For this problem, the likelihood function is the chance that one would have obtained the data that was measured if the bias-weighting was known. Given a bias-weighting of H, the probability of getting a heads is

$$P(\text{Heads}|H) = H, \tag{4.4}$$

and the probability of getting a tails is

$$P(\text{Tails}|H) = 1 - H. \tag{4.5}$$

It is assumed that all of the flips are independent. Hence, the likelihood function of all of the data points is the product of the individual likelihoods. This means the probability of getting "R heads in N tosses" is represented with a binomial distribution:

$$P(\text{data}|H) \propto H^R(1-H)^{N-R}. \tag{4.6}$$

A proportionality is used instead of an equality because the pdf must be normalized.

4.4 Applying Bayes' Theorem

Finally, Eqs. (4.6) and (4.3) can be substituted into Eq. (4.1) to yield a posterior pdf, which tells us our state of knowledge based on the given data:

$$P(H|\text{data}) \propto H^R(1-H)^{N-R} \qquad \text{for } 0 \leq H \leq 1. \tag{4.7}$$

Alternatively, instead of considering all of the data at once, the data could be added sequentially as it was obtained. The details and advantages of this method are detailed in Appendix A.

4.5 Results

Initially, the posterior pdf was plotted with no data, as shown in Fig. 4.1(a). Then, a coin was flipped, and the result of a heads was recorded, and the model was updated to account for the new data as seen in Fig. 4.1(b). Then a second flip, also resulting in heads, was added to the model resulting in Fig. Figure 4.1(c). This process was repeated up to 4096 data points as shown in Fig. 4.1 to study how the posterior pdf changes with more data. The results show that the maximum posterior estimate approaches $H = 0.25$, which indicates that the coin was biased.

Now what would happen if a different prior was chosen? Let the new prior be a Gaussian distribution with a mean of 0.25 and a standard deviation of 0.05. This prior indicates that one assumes that the coin is most likely biased because the

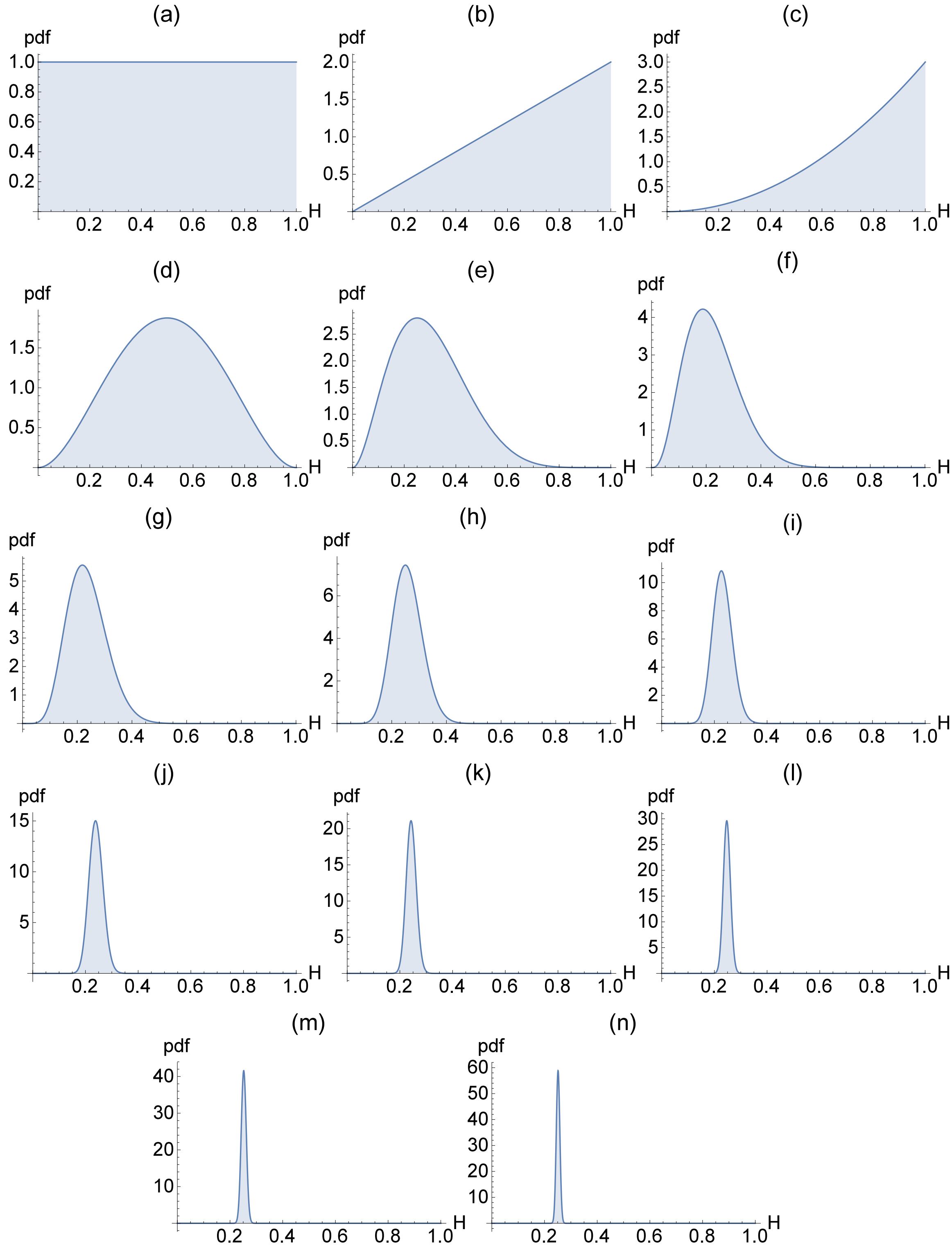

Figure 4.1. The posterior pdf for 0, 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096 flips with an uninformed prior.

highest probability is at $H = 0.25$. This belief could come from another customer at the casino telling us that the coin is rigged, so that only 25% of the flips are heads. The same exercise with the flips was repeated with the new prior and the results are shown in Figure 4.2. A third prior was chosen to be a Gaussian distribution with a mean of 0.5 and a standard deviation of 0.05. This prior indicates that we believe the coin is fair. Figure 4.3 shows the results for the flips as in the other examples.

4.6 Discussion

The results from the naive prior show how adding more data refines to the pdf to a Dirac-delta function over the value of H. In this case, the spike gets narrower and taller around $H = 0.25$, which indicates that the true bias of the coin is most likely to be $H = 0.25$.

The second prior shows how having an accurate prior causes the faster convergence to the true solution. Once again, it can be seen that $H = 0.25$ is the true bias of the coin, but it is clear after 64 sample as opposed to 128 samples for the uninformed prior.

For the third prior, Fig. 4.3(b)-(d) show that small numbers of data points do little to change the non-uniform prior. Once 32 data points were recorded, as shown in Fig. 4.3(g), it can be see that the posterior pdf is straying away from $H = 0.5$, the maximum of the prior. As more points are added, the importance of the prior relative to the data becomes small. In the end, the solution is similar to the uniform prior with $H = 0.25$ being the most probable solution. This goes to show that even if the prior does not align well with the data, increasing the number of data points can overrule the prior and push the posterior pdf to the true value of the parameter.

This example serves to show how Bayesian inference is applied to problems. It demonstrates how increasing the number of data points causes the solution to approach a specific parameter value. Additionally, it shows the power of beginning the

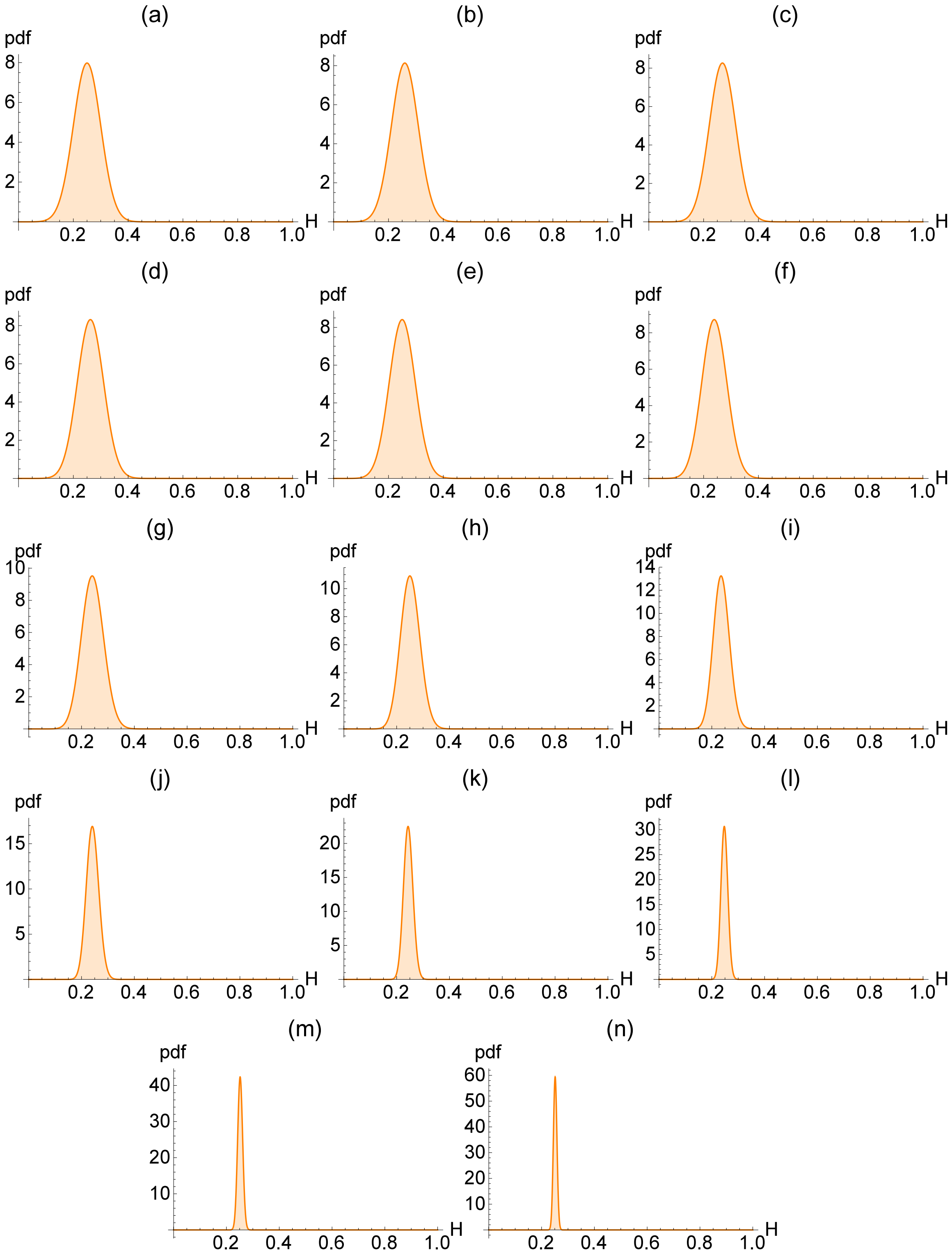

Figure 4.2. The posterior pdf for 0, 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096 flips for the Gaussian prior around $H = 0.25$.

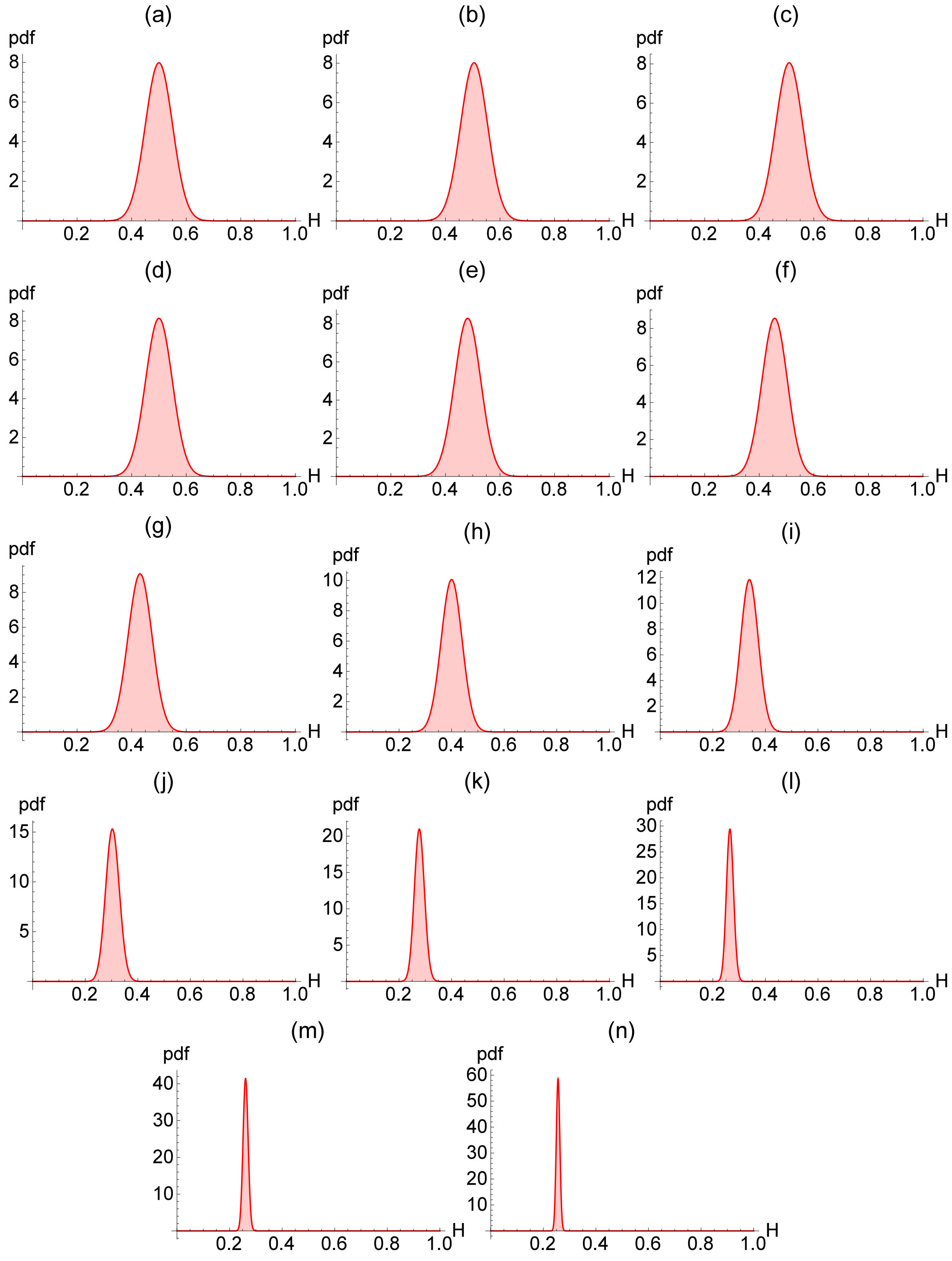

Figure 4.3. The posterior pdf for 0, 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096 flips for the Gaussian prior around $H = 0.5$.

study with an accurate prior. Finally, it demonstrated how data can overcome a prior in the case of an inaccurate prior. In this study, the true value of the bias weighting of the coin is $H = 0.25$, so the Bayesian approach determined the correct solution.

CHAPTER 5

ENGINEERING APPLICATIONS

In this chapter, Bayesian inference will be applied to the two engineering problems to demonstrate its use. First, the spring constant of a linear spring is estimated. Then the decay rate constant of a radioactive material is estimated using Bayesian inference with both analytical and numerical models.

5.1 Linear Spring

5.1.1 Problem Statement

In this problem, an engineer purchases a spring from a hardware store. However, he or she does not know its spring constant. The spring is tested in static equilibrium. To determine the spring constant, a fixed mass is hung off the spring and its displacement is measured as shown in Figure 5.1.1. This procedure is repeated N times to get data points, d_i, in order to more accurately determine the true value. The free body diagram in Figure 5.1.1 shows the applied force, F, and the reaction force, which is represented by Hooke's law kx, where k is the spring constant and x is the linear displacement. The spring's mass is neglected. Static force analysis of the spring leads to

$$\sum F = 0, \tag{5.1}$$

$$F - kx = 0, \tag{5.2}$$

$$x = \frac{F}{k}. \tag{5.3}$$

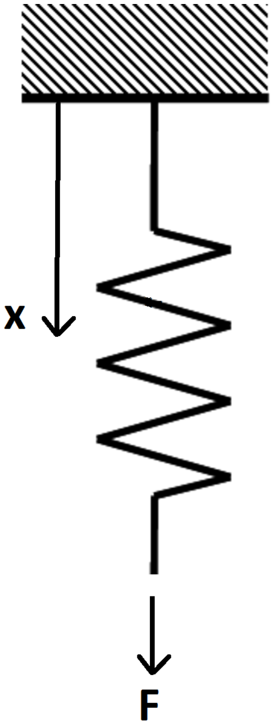

Figure 5.1. A diagram of a force, F acting on a spring showing the displacement of the spring x.

Bayesian inference is used to estimate the value of the parameter k. Hence Bayes' theorem becomes:

$$P(k|\mathbf{D}) \propto P(\mathbf{D}|k)\, P(k). \tag{5.4}$$

In this model, it is assumed that the force is measured exactly, so there is no associated uncertainty. However, there is uncertainty associated with the measurements of the displacement:

$$d_i = x + \xi, \qquad \xi \sim \mathcal{N}(0, \sigma_x), \tag{5.5}$$

where x is the true value of the displacement and σ_x is the standard deviation of the

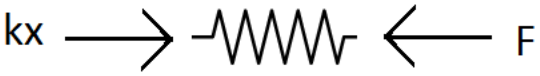

Figure 5.2. A free body diagram of the spring shows the applied force and the reaction force.

measurement error. Because the distance is measured with a ruler with a limited resolution, σ_x is equal to the smallest division on the ruler.

5.1.2 Prior

For the spring constant estimation, a prior, $P(k)$, must be selected. Initially, a naive assumption will be made, but it could be updated at a later time. Because the spring constant must be positive, let $k > 0$. Also, it will be assumed that the spring constant is less than 10 N/mm, because the engineer has outside knowledge that the spring constant is less than 10 N/mm. The uninformed prior will be constant on the interval $[0, 10]$:

$$P(k) = \begin{cases} \frac{1}{10}, & 0 \leq k \leq 10, \\ 0, & \text{otherwise}, \end{cases} \tag{5.6}$$

as shown in Figure 5.1.2.

5.1.3 Likelihood Function

The next step in the analysis is to create a likelihood function, $P(\mathbf{D}|k)$. In creating the likelihood function, one must account for the measurement uncertainty inherent in the experiment. As stated before, the ruler causes random measurement uncertainty that is sampled from a Gaussian distribution with a mean of $\mu = 0$ mm and a standard deviation of $\sigma_x = 0.1$ mm. If there is no uncertainty in the measurement

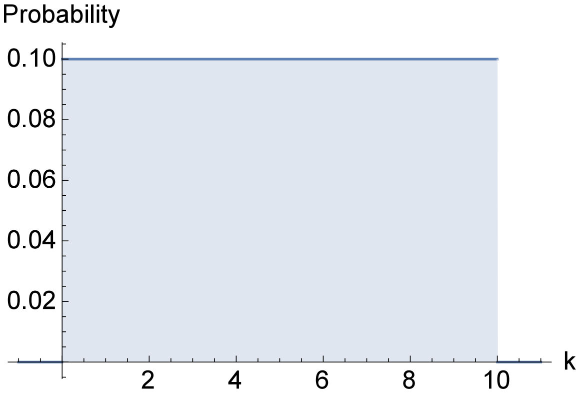

Figure 5.3. A plot of the naive prior for the linear spring problem.

of x, then the problem is deterministic. In that case, all of the data points would be equal and $x = d_1 = d_2 = ... = d_N$. The likelihood function for one data point is represented by a normal distribution around d_i with a standard deviation of σ_x, because the uncertainty in the measurements is Gaussian. In the likelihood function, the data is compared with the model in Eq. (5.3). In this case, the model is the analytical solution to the original problem, but that is not always the case. The model could also be a numerical model [16]. The probability density function for the likelihood is:

$$P(d_i|k) = \frac{1}{\sigma_x\sqrt{2\pi}} \exp\left(-\frac{(d_i - F/k)^2}{2\sigma_x^2}\right). \tag{5.7}$$

Because each data point is measured independently, the joint likelihood function is the product of all of the individual probability densities of the measurements which simplifies to:

$$P(\mathbf{D}|k) = \frac{1}{\left(\sigma_x\sqrt{2\pi}\right)^N} \exp\left(-\sum_{i=1}^{N}\frac{(d_i - F/k)^2}{2\sigma_x^2}\right). \tag{5.8}$$

5.1.4 Applying Bayes' Theorem

In order to estimate the parameter, k, Bayes' theorem will be used:

$$P(k|\mathbf{D}) = \frac{P(\mathbf{D}|k)\,P(k)}{P(\mathbf{D})}, \tag{5.9}$$

$$P(k|\mathbf{D}) = \frac{P(\mathbf{D}|k)\,P(k)}{\int P(\mathbf{D}|k)\,P(k)\,dk}, \tag{5.10}$$

$$P(k|\mathbf{D}) = \gamma P(\mathbf{D}|k)\,P(k), \tag{5.11}$$

$$P(k|\mathbf{D}) = \frac{\gamma}{10\left(\sigma_x\sqrt{2\pi}\right)^N} \exp\left(-\sum_{i=1}^{N} \frac{(d_i - F/k)^2}{2\sigma_x^2}\right) \qquad \text{for } 0 \le k \le 10 \tag{5.12}$$

where $\gamma = 1/\int P(\mathbf{D}|k)\,P(k)\,dk$ is the normalization constant.

5.1.5 Results

For this problem, a load of $F = 5$ N and $\sigma_x = 0.1$ mm were used. A total of 128 data points were taken. For this study, experimental data points were not recorded, but were simulated using the Mathematica code in Appendix B.1. All of the data points are shown in Figure 5.1.5. The data are clustered around one point because the uncertainty is Gaussian. Bayes' theorem was used to calculate posterior pdfs for the data as it was added sequentially. Figure 5.5 shows the posterior pdfs after a number of measurements. The location of the maxima of the posterior pdfs is the most likely value of the spring constant. Table 5.1 lists the maximum posterior estimation from the pdfs and confidence intervals for these solutions. The 95% confidence interval for a pdf with known mean and standard deviation is:

$$\left[k_{max} - \frac{1.96\sigma}{\sqrt{n}}, k_{max} + \frac{1.96\sigma}{\sqrt{n}}\right], \tag{5.13}$$

where k_{max} is the maximum posterior estimation, σ is the standard deviation of the data, and n is the number of data points. The constant 1.96 comes from the fact

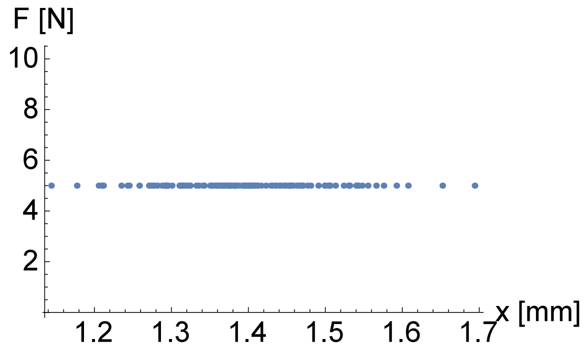

Figure 5.4. The measured data for a spring with an unknown spring constant under a 5 N force.

that for a normal distribution, $P(X > 1.96) = 0.025$ and $P(X < 1.96) = 0.975$ and so the area between the two is 0.95 or 95% of the area under the distribution. For comparison, Table 5.1 also contains the estimates that were obtained using the method of least squares.

5.1.6 Discussion

First, the results in Table 5.1 show convergence to approximately 3.59 N/mm after just 32 samples and increasing the number of samples to 128 further enforces this inference. Additionally, the confidence intervals continue to narrow as samples are added as expected. Also, the results from Bayesian inference and the method of least squares are exactly the same. Because the prior was uniform, the uncertainty is Gaussian, and the model is linear, this result further confirms the claim made in Chapter 3.2.

Although studying the spring with by the displacement after a single force converges to a result, could the value of k be found more quickly if measurements were made for different forces? The following study will explore this.

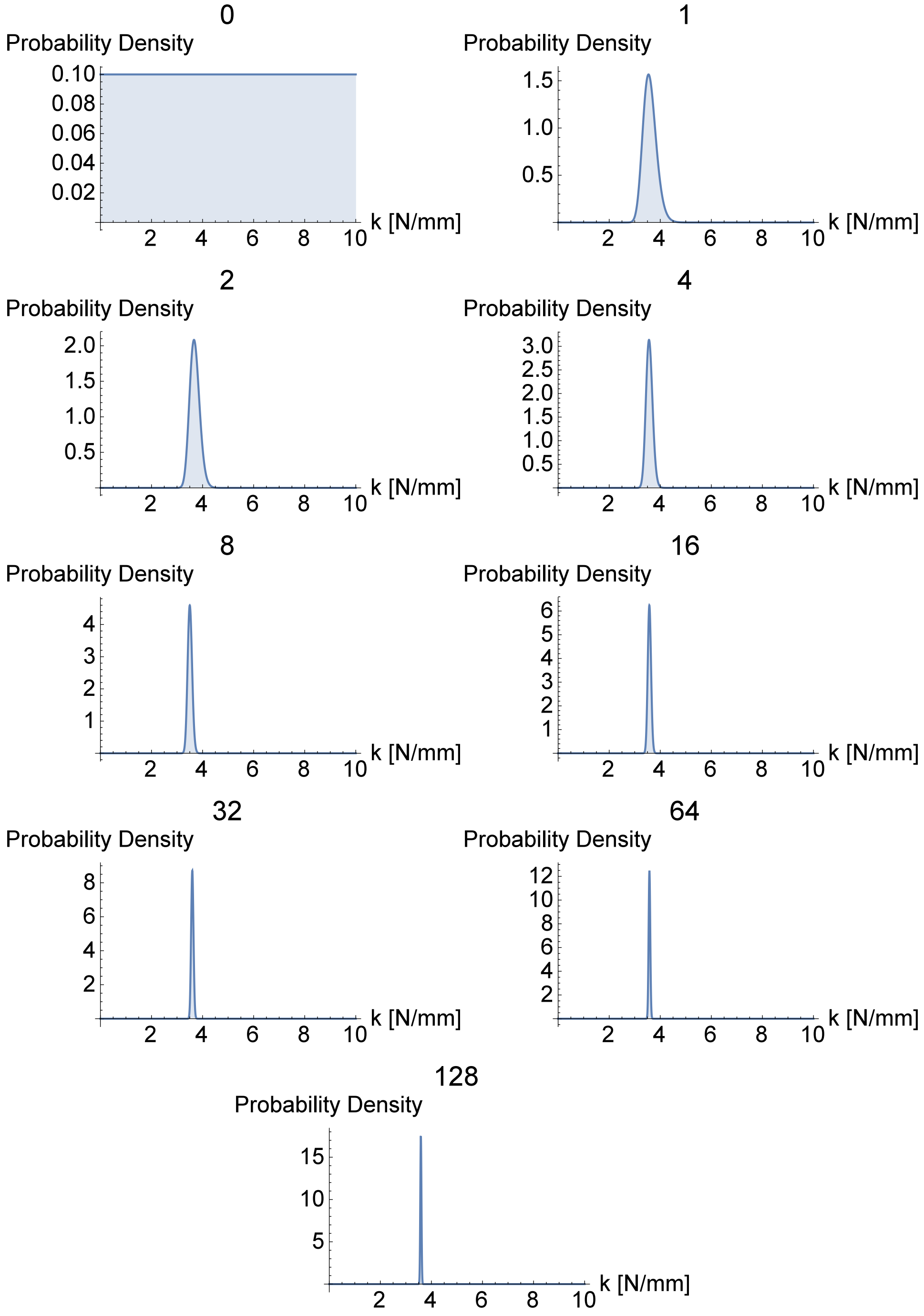

Figure 5.5. The posterior pdf for the spring constant of a linear spring with 0, 1, 2, 4, 8, 16, 32, 64, and 128 measurements of the displacement under a 5 N force.

TABLE 5.1

Least squares and maximum posterior estimates for the spring constant of the linear spring with $F = 5$ N.

N	k_{LS} [N/mm]	k_{MPE} [N/mm]	95% Confidence Interval [N/mm]
1	3.54027	3.54027	$[3.40497, 3.67557]$
2	3.66501	3.66501	$[3.61246, 3.71755]$
4	3.55931	3.55931	$[3.54321, 3.57540]$
8	3.49948	3.49948	$[3.49422, 3.50473]$
16	3.57299	3.57299	$[3.57098, 3.5750]$
32	3.59771	3.59771	$[3.59699, 3.59844]$
64	3.57560	3.57560	$[3.57535, 3.57585]$
128	3.58733	3.58733	$[3.58724, 3.58742]$

5.1.7 Linear Spring Varying Force

In this study, the spring constant was estimated using the same method as in the previous example, but the applied load is varied between 2 N and 10 N. Because the force varies with each trial the model is now:

$$x_i = \frac{F_i}{k}, \tag{5.14}$$

where x_i is the true value of the displacement when load F_i is applied. The load for the first trial was 2 N and then it increased by 1 N for the next trials, until a load of 10 N was applied. Following the 10 N load, a 2 N load was used and the pattern was repeated. As in the last study, 128 trials were used, and data were added sequentially. The measured data are shown in Fig. 5.1.7. The resulting posterior pdfs are shown in Fig. 5.7. Additionally, the maximum posterior estimates and least squares solutions

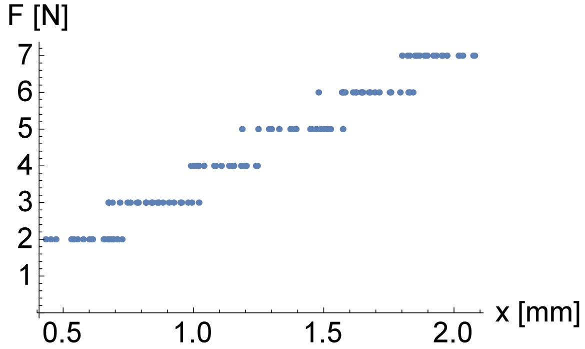

Figure 5.6. The measured data for a spring with an unknown spring constant under a varying force.

for the data are shown Table 5.2.

The maximum posterior estimates in Table 5.2 converge to a similar solution to the constant force method as one would expect. Varying the force causes faster convergence to a solution around 3.59 N/mm. In addition to the improved convergence rate, varying the force also confirms that the model is linear. When a single force is tested, it merely shows the value of k in one case and the model behavior could change for other loads. Testing the spring with different forces shows that the trend continues for a range of loads.

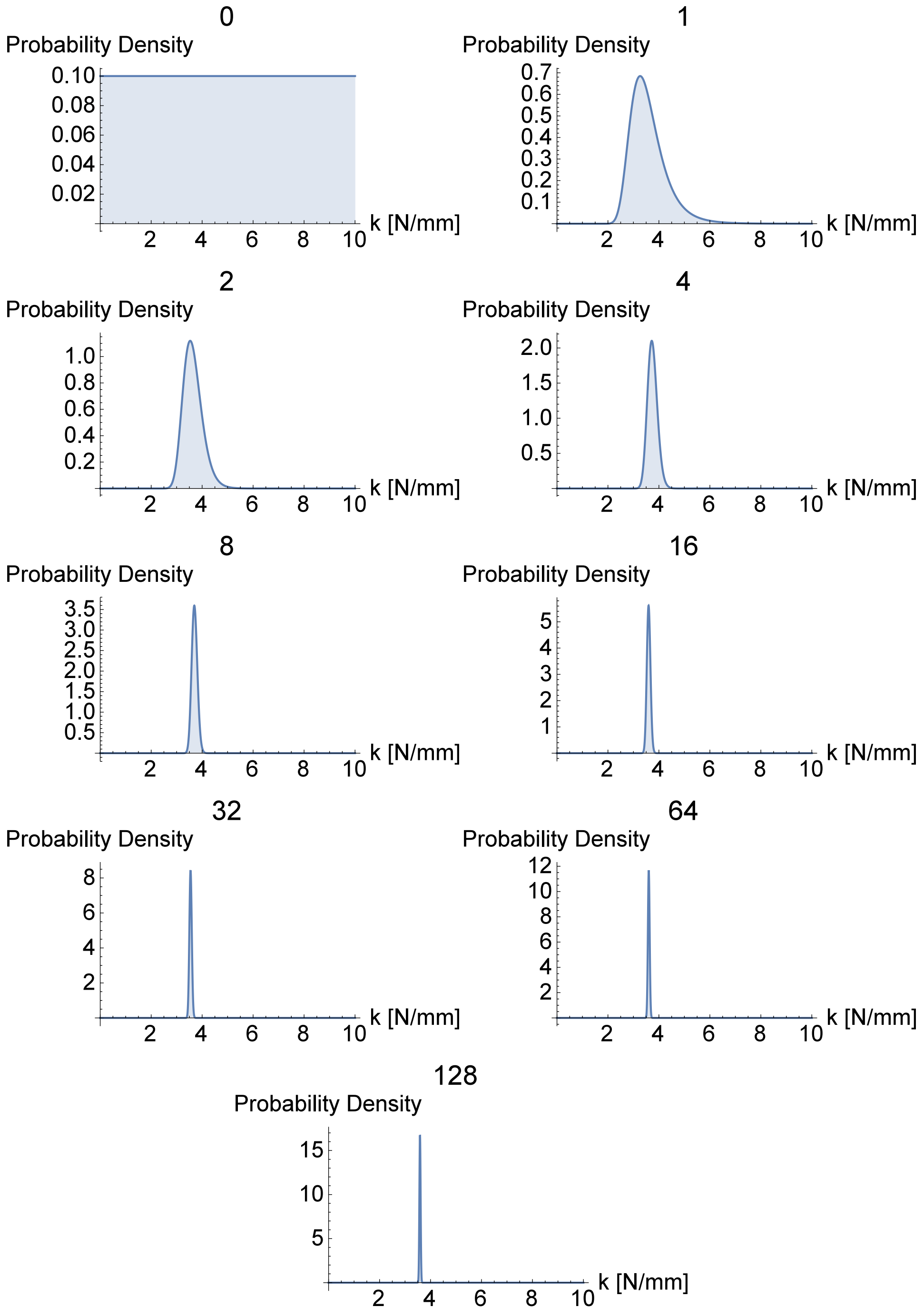

Figure 5.7. The posterior pdf for the spring constant of a linear spring with 0, 1, 2, 4, 8, 16, 32, 64, and 128 measurements of the displacement under a varying force.

TABLE 5.2

Least squares and maximum posterior estimates for the spring constant of the linear spring with varying loads.

N	k_{LS} [N/mm]	k_{MPE} [N/mm]	95% Confidence Interval [N/mm]
1	3.26466	3.26466	$[2.22695, 4.30238]$
2	3.53148	3.53148	$[3.33145, 3.73151]$
4	3.69292	3.69292	$[3.68318, 3.75600]$
8	3.59869	3.59869	$[3.6843, 3.701530]$
16	3.59467	3.59467	$[3.59219, 3.59715]$
32	3.54530	3.54530	$[3.54452, 3.54608]$
64	3.60210	3.60210	$[3.60182, 3.60239]$
128	3.58697	3.58697	$[3.58688, 3.58707]$

5.2 Radioactive Decay

5.2.1 Problem Statement

An engineer is modeling the radioactive decay of a material, but there is uncertainty in the decay rate constant. Radioactive decay is modeled with the differential equation:

$$\frac{dy}{dt} = -cy, \qquad y(0) = 1, \tag{5.15}$$

where y is the amount of material present, t is time, and c is the decay rate constant. In this study, measurements of y are taken at different times for a number of samples of the material. The uncertainty associated with the measurements of y are

$$d_i = y_i + \xi, \tag{5.16}$$

where d_i is the ith data measurement of y, y_i is the true value of y, and ξ is a random variable sampled from a Gaussian distribution with standard deviation σ: $\xi \sim \mathcal{N}(0, \sigma)$. The solution to Eq. (5.15) can be shown to be

$$y = e^{-ct}. \tag{5.17}$$

The engineer wishes to estimate the decay rate constant. Bayesian inference will be used for this task.

5.2.2 Prior

The initial prior, $P(c)$ is naive. The decay rate constant must be positive. Also, assume it is known that the decay rate is less than 10 s^{-1}, which is a conservative bound based on prior knowledge about the material. The uniform prior will have a

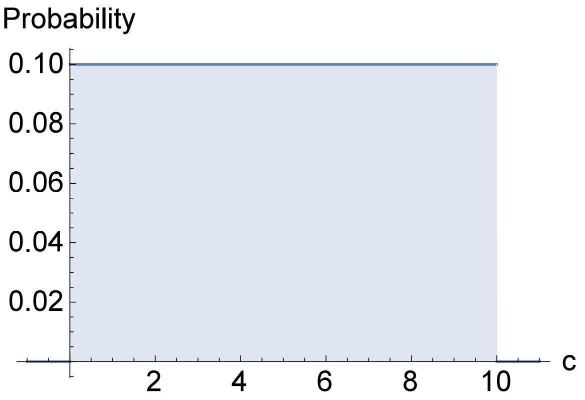

Figure 5.8. A plot of the naive prior for the radioactive decay problem.

constant value of $1/10 \text{ s}^{-1}$ on the interval $[0, 10]$:

$$P(c) = \begin{cases} \frac{1}{10}, & 0 \leq c \leq 10, \\ 0, & \text{otherwise}, \end{cases} \tag{5.18}$$

as shown in Fig. 5.2.2.

5.2.3 Likelihood Function

Next, a likelihood function, $P(\mathbf{D}|c)$, is created. Because the uncertainty in c is Gaussian, a Gaussian likelihood function is used. The data is compared to the model in Eq. (5.17). The likelihood of the ith data point at time t_i is:

$$P(d_i|c) \propto \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(d_i - \exp(-ct_i))^2}{2\sigma^2}\right). \tag{5.19}$$

Because each data point is measured independently, the overall likelihood function is the product of all of the individual measurements which simplifies to:

$$P(\mathbf{D}|c) \propto \frac{1}{\left(\sigma\sqrt{2\pi}\right)^N} \exp\left(-\sum_{i=1}^{N} \frac{(d_i - \exp(-ct_i))^2}{2\sigma^2}\right). \tag{5.20}$$

5.2.4 Applying Bayes' Theorem

In order to estimate the parameter, c, Bayes' theorem will be used:

$$P(c|\mathbf{D}) \propto P(\mathbf{D}|c)\, P(c), \tag{5.21}$$

$$P(c|\mathbf{D}) \propto \frac{1}{10\left(2\sqrt{\pi\sigma_x}\right)^N} \exp\left(-\sum_{i=1}^{N} \frac{(d_i - \exp(-ct_i))^2}{2\sigma^2}\right) \qquad \text{for } 0 \leq c \leq 10, \tag{5.22}$$

where $P(\mathbf{D}|c)$ is the likelihood function and $P(k)$ is the prior. In order for the posterior probability to be a pdf, it must be normalized by $\int_{-\infty}^{\infty} P(\mathbf{D}|c)\, P(c)\, dc$. The model was updated sequentially.

5.2.5 Results

In this problem, the engineer knows the standard deviation is $\sigma = 0.1$ based off the data sheet of the measurement device. The data was obtained from measuring 16 samples of the material at 16 different times (every 5/16 s). Once again, the data was simulated in Mathematica. When manufacturing the data, the radioactive decay constant was considered as $c = 1 + \xi$, where $\xi \sim \mathcal{N}(0, 0.05)$ to simulate variation between samples. The full code is contained in Appendix B.2. The data obtained is plotted in Fig. 5.2.5. For this simulation, the posterior pdfs after various numbers of data points are shown in Fig. 5.10. Table 5.3 lists the maximum posterior estimates and the confidence intervals of these solutions.

5.2.6 Discussion

From the results in Fig. 5.10, the maximum posterior estimate is approaching $c = 1$. The complexity of the model causes the results to converge more slowly than the linear spring model. However, after 32 data points, the maximum posterior estimate is near the final solution. This study shows that Bayesian inference is

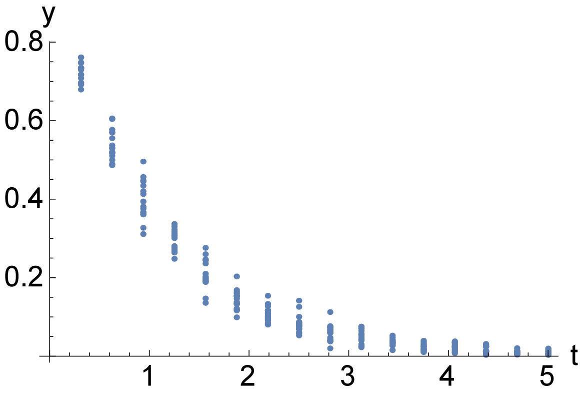

Figure 5.9. The measured data for the radioactive decay of a material.

TABLE 5.3

The maximum posterior estimates for radioactive decay rate constant of a material for varying number of data points.

N	c [1/s]	95% Confidence Interval [1/s]
1	0.915664	$[0.844205, 1.03027]$
2	0.94886	$[0.930974, 0.989734]$
4	1.02312	$[1.01867, 1.04337]$
8	1.04406	$[1.04346, 1.0592]$
16	1.03183	$[1.03224, 1.04649]$
32	1.02631	$[1.02663, 1.03369]$
64	0.984881	$[0.985087, 0.988135]$
128	0.959072	$[0.95919, 0.96059]$
256	0.982358	$[0.982384, 0.982758]$

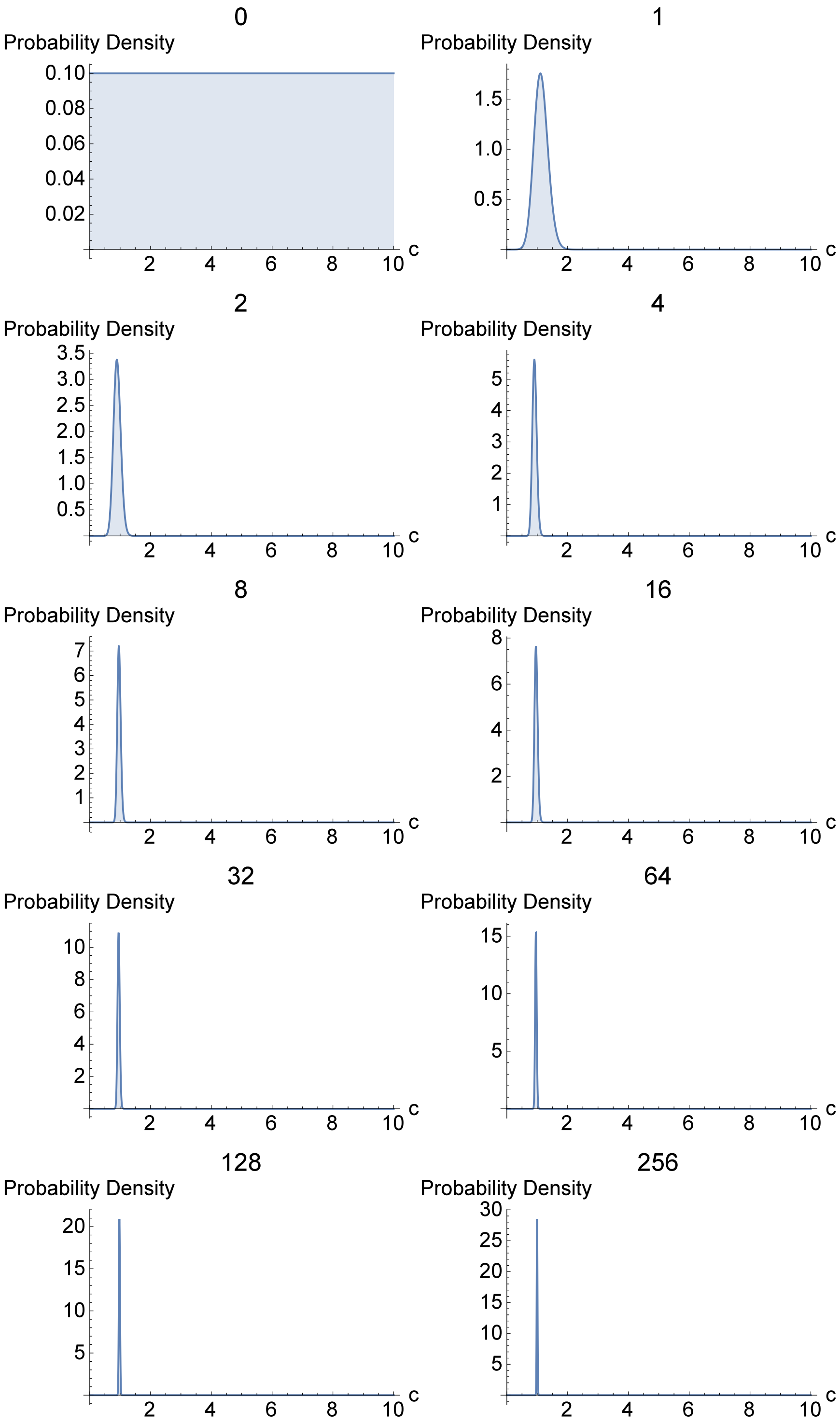

Figure 5.10. The posterior pdf for the decay rate constant of the radioactive decay of a material with 0, 1, 2, 4, 8, 16, 32, 64, and 128 measurements of the amount of remaining material.

effective for more complex problems as well.

Although Eq. (5.15) has an analytical solution, other methods could be used for the model. In particular, numerical methods could be used to as a model. The simplest stochastic method is the Monte Carlo method, however a number of other methods, like polynomial chaos, also exist. In order to demonstrate how Bayesian inference is applied to problems without analytical solutions, the Monte Carlo method will be used to estimate the decay rate constant of the material.

5.2.7 Monte Carlo Method

First, consider Bayes' theorem for the given problem:

$$P(c|\mathbf{D}) = \frac{P(c)\,P(\mathbf{D}|c)}{P(\mathbf{D})}. \tag{5.23}$$

When an analytical model is used, the maximum posterior is used as the best estimate for c, but in the Monte Carlo method, the mean of the posterior will be used as the best estimate. The mean or expected value of the posterior can be written:

$$\begin{aligned} E(c|\mathbf{D}) &= \int_\Gamma c\,P(c|\mathbf{D})\,dc, && (5.24)\\ &= \int_\Gamma c\,\frac{P(c)\,P(\mathbf{D}|c)}{P(\mathbf{D})}\,dc. && (5.25)\end{aligned}$$

The prior is the same as in Eqs. (5.18). The likelihood function is

$$P(\mathbf{D}|c) \propto \frac{1}{\left(\sigma\sqrt{2\pi}\right)^N} \exp\left(-\sum_{i=1}^{N} \frac{(d_i - y_i)^2}{2\sigma^2}\right), \tag{5.26}$$

where y_i is the value of y corresponding to data point d_i from the numerical solution of Eq. (5.15). In this case, the differential equation was solved in Mathematica using the NDsolve command, which uses the method of lines to solve differential equations. In order to calculate the mean of the posterior pdf, the Monte Carlo method is used.

In the Monte Carlo method, the integral is estimated by taking a large number of samples of the random variable, c, and substituting them into the equation. The samples of c are sampled from the prior during this method. Then the equation is solved and the final result is recorded. The mean value of the results of all of these samples is the mean of the parameter. Written mathematically, the mean is estimate by:

$$
\begin{aligned}
E(c|\mathbf{D}) &\approx \frac{1}{M}\sum_{m=1}^{M} c_m \frac{P(c_m)\,P(\mathbf{D}|c_m)}{P(\mathbf{D})}, && (5.27)\\
&= \frac{1}{M}\sum_{m=1}^{M} c_m \frac{1}{10}\frac{1}{\left(\sigma\sqrt{2\pi}\right)^N}\exp\left(-\sum_{i=1}^{N}\frac{(d_i - y_{i,m})^2}{2\sigma^2}\right)\frac{1}{P(\mathbf{D})}, && (5.28)
\end{aligned}
$$

where M is the number of sampled values of c and $y_{i,m}$ is the numerical solution to the differential equation when $c = c_m$ corresponding to data point d_i. The normalization factor, $1/P(\mathbf{D})$ is the reciprocal of the average of the likelihood function: $P(\mathbf{D}) = \sum_{m=1}^{M} P(c_m)P(\mathbf{D}|c_m)$.

In this study, $10,000$ samples of c were considered. Additionally, the method was applied to the same data as before, so the results can be directly compared. The full code for this study is in Appendix B.3. In the end, $E(c|\mathbf{D}) = 0.960842$ s^{-1}. The true value of the parameter was $c = 1$ s^{-1}, so the analytical method was slightly more accurate, but overall, both methods showed to be accurate. The Monte Carlo method is advantageous, because it is effective for models when there is no analytical solution to the underlying equations. The method's simplicity makes it easy to implement. However, its main drawback is its computational expense. The computational cost of Monte Carlo is considered compared to polynomial chaos in Appendix C. The differential equation must be numerically solved for each sample of c, so the method is computationally expensive. A number of other methods, like polynomial chaos, offer reduced order models that produce solutions more quickly than the Monte Carlo method. In conclusion, this exercise with the Monte Carlo method serves as an example of how Bayesian inference can be used in situations where no analytical solution exists.

CHAPTER 6

CONCLUSIONS

Bayesian statistics offer a simple method to quantify the uncertainty in a wide range of engineering problems. As demonstrated, they are able to reproduce the results of deterministic methods, but are more informative, because the statistical results provide additional information like the confidence interval. Additionally, modifying the prior offers a quicker convergence to the true value of a parameter. These methods were shown to provide reliable results for estimating the bias in a coin, the linear spring constant, and the radioactive decay rate constant and could be applied to a wide variety of other problems.

Presently, the major areas of interest in Bayesian statistics are decreasing computational costs. When there is no explicit expression for the measured quantity as a function of the parameters, numerical methods must be used, like the one shown in Section 5.2.7. In these methods, the mean value of the parameter is determined by approximating the integral. Additionally, integrals of pdfs appear when normalizing. When there are only a few variables, deterministic methods, like the trapezoidal rule, are effective for numerial integration. However, when there are more dimensions, these methods cannot be used and other strategies, like the Monte Carlo method must be used. As was stated, the Monte Carlo method is computationally expensive, so surrogate models are created using methods like polynomial chaos [17]. Bayesian methods and uncertainty quantification are especially important in the petascale and future exascale simulations, because the simulations are often so computationally expensive that they can only be run once a year. Surrogate models must be created and used and the uncertainty in these models must be quantified [18]. In summary, methods for approximating integrals and developing surrogate models are two of the main areas of research in Bayesian analysis today.

In addition to these areas of interest within Bayesian methods, UQ is likely to become more critical in the coming years. As computational power increases, computational methods are being used more frequently in engineering. Computational models are considered a third mode of engineering discovery alongside theory and experiment. Verification, validation, and uncertainty quantification must be used to create trustworthy simulations for discovery. Verification and validation ensure that the model solves the mathematical model correctly and that the mathematical model correctly represents the physics respectively. An example of verification can be seen in Appendix E. Uncertainty quantification is critical in the creation of predictive simulations because it gives a measure of the reliability. Reliability will be key as numerical models are trusted with higher consequence decisions and UQ will be at the forefront of the decisions.

APPENDIX A

SEQUENTIAL BAYES' THEOREM

If N data points are considered to determine a parameter using Bayes' theorem. All of the data can be added at once using a likelihood model that fits all of the data. Alternatively, data could be added into the model sequentially as new data arrives. Once the posterior pdf is calculated for the first set of data, D_1, then that pdf can be used as the prior for the second set of data D_2. The sequential method considers new data as an entirely new problem. Furthermore, this new problem uses the posterior pdf from the old data as the prior, because it is now knowledge that one knew before the new data. Both the full data method and the sequential method produce exactly the same results. Consider a problem where two sets of data are obtained at different times. If Bayes' theorem was applied to both sets of data at once the result would be:

$$\text{prob}(H|D_2, D_1) \propto \text{prob}(D_2, D_1|H) \times \text{prob}(H). \tag{A.1}$$

If the model was updated sequentially, the posterior pdf after the considering the second set of data would be

$$\text{prob}(H|D_2, D_1) \propto \text{prob}(D_2|H, D_1) \times \text{prob}(H|D_1). \tag{A.2}$$

If the data obtained was independent from each other, $\text{prob}(D_2|H, D_1) = \text{prob}(D_2|H)$. Manipulating Eq. (A.2) gives:

$$\begin{aligned}
\text{prob}(H|D_2, D_1) &\propto \text{prob}(D_2|H) \times \text{prob}(H|D_1), && \text{(A.3)}\\
&\propto \text{prob}(D_2|H) \times (\text{prob}(D_1|H) \times \text{prob}(H)), && \text{(A.4)}\\
&\propto \text{prob}(D_2, D_1|H) \times \text{prob}(H). && \text{(A.5)}
\end{aligned}$$

Hence, adding data sequentially results in the same solution as adding the data at once [12].

Sequentially adding data is advantageous, because it is computationally less expensive than the non-sequential method. The posterior pdf from the previous step can be saved and reused instead of calculating and entirely new pdf each time more data is added.

A.1 Coin Example

The sequential method was also used for the coin flipping problem in Chapter 4. For this example, the uniform prior was chosen to be the initial prior. Figure A.1 shows the results of the study for various numbers of flips. The results are exactly the same as in the non-sequential method in Fig. 4.1 as expected.

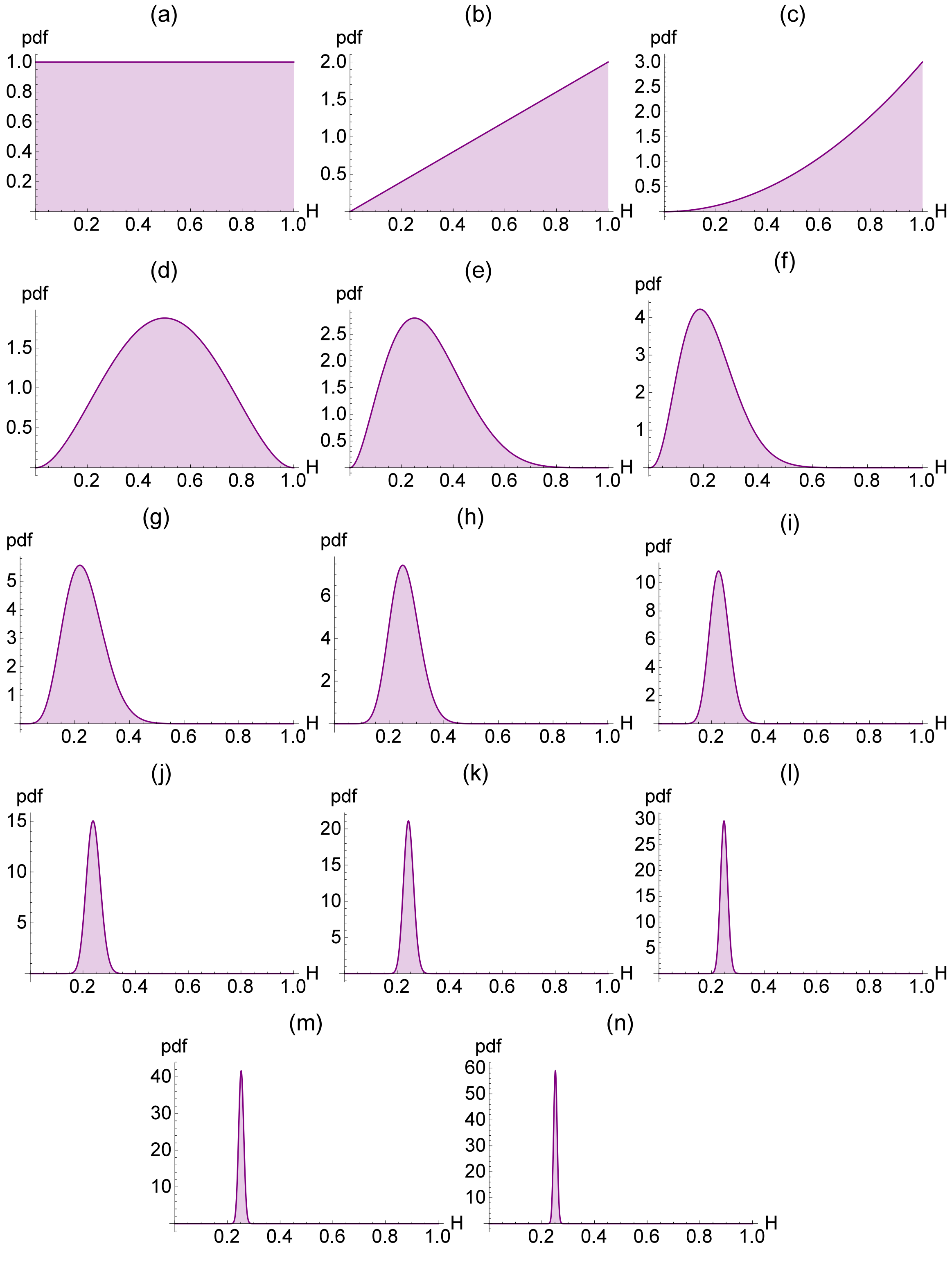

Figure A.1. The posterior pdf for 0, 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096 flips for the uniform prior were found sequentially.

APPENDIX B

CODE

B.1 Linear Spring

This is a copy of the Mathematica code used for the linear spring problem in Section 5.1.5.

```
SetOptions[Plot,
  BaseStyle -> {FontFamily -> "CMU Serif", 18, GrayLevel[0]}];
SetOptions[Plot3D,
  BaseStyle -> {FontFamily -> "CMU Serif", 12, GrayLevel[0]},
  AxesStyle -> Black];
SetOptions[ParametricPlot3D,
  BaseStyle -> {FontFamily -> "CMU Serif", 12, GrayLevel[0]},
  AxesStyle -> Black];
SetOptions[ContourPlot,
  BaseStyle -> {FontFamily -> "CMU Serif", 16, GrayLevel[0],
    AxesStyle -> Black}];
SetOptions[DiscretePlot,
  BaseStyle -> {FontFamily -> "CMU Serif", 16, GrayLevel[0],
    AxesStyle -> Black}];
SetOptions[ListPlot,
  BaseStyle -> {FontFamily -> "CMU Serif", 18, AxesStyle -> Black}];
SetOptions[ParametricPlot,
  BaseStyle -> {FontFamily -> "CMU Serif", 12, GrayLevel[0],
    AxesStyle -> Black}];
SetOptions[ListLogLogPlot,
  BaseStyle -> {FontFamily -> "CMU Serif", 12, GrayLevel[0],
    AxesStyle -> Black}];
SetOptions[DensityPlot,
  BaseStyle -> {FontFamily -> "CMU Serif", 12, GrayLevel[0],
    AxesStyle -> Black}];

(* Set up random Values *)
Print["Seed"]
```

```
seed = 848(*RandomInteger[{1,10000}]*)
SeedRandom[seed];

(* Number of data points *)
m = 7;
numPts = 2^m;

(* Problem Parameters *)
F1 = 5;
kexact = 3.5983;
xexact1 = F1/kexact;
exactPlot1 = {{xexact1, F1}};
kCov = 0;
xerr = 0.1;

(* x=F/k *)
data1 = Table[0, {i, 1, numPts}]; (* Initialize *)
Do[
 data1[[i]] =
  F1/(kexact*(1 + RandomVariate[NormalDistribution[0, 1]]*kCov)) +
   RandomVariate[NormalDistribution[0, 1]]*xerr,
 {i, 1, numPts}
 ];

(* Optional Plot Data *)
plotData1 = Table[0, {i, 1, numPts}];
Do[
  plotData1[[i]] = {data1[[i]], F1},
  {i, 1, numPts}
  ];
dataPlot =
 ListPlot[{plotData1(*,exactPlot1*)},
  AxesLabel -> {"x [mm]", "F [N]"}, PlotLegends -> {"Data"(*,"Exact"*)}]

(* Bayesian Method *)
(* Create Prior *)
priorA = PDF[
   UniformDistribution[
    {0, 10}
    ],
   k];
priorB = PDF[
   NormalDistribution[
    kexact, 0.5
    ],
   k];
priorC = PDF[
   NormalDistribution[
    kexact, 0.25
    ],
   k];
priorD = PDF[
   NormalDistribution[
    kexact, 0.1
    ],
   k];

prior = priorA;

(* Standard Deviation for Likelihood *)
\[Sigma]1 = xerr(*Sqrt[xerr]*);

(* Initialize posteriod pdfs *)
posterior1 = Table[0, {i, 1, numPts + 1}];

posterior1[[1]] = prior;

(* Make Table of Least Squares Solutions *)
lsSoln = Table[0, {i, 1, numPts + 1}];

(*Integration Bounds for problem*)
intL = -\[Infinity];
intR = \[Infinity];
Do[{
  likelihood1 =
   1/(\[Sigma]1*Sqrt[2*\[Pi]])*
    Exp[-(data1[[i - 1]] -
          F1/k)^2/(2*\[Sigma]1^2)]; (*Gaussian likehood*)
  (*xunit=2*xerr; (* Smallest unit on the ruler *)
  likelihood1=Piecewise[{{0,Abs[F1/k-data1[[i-1]]]>xunit},{1/xunit,
  Abs[F1/k-data1[[i-1]]]>=xunit}}];*)

  posterior1[[i]] = prior*likelihood1;

  norm = NIntegrate[posterior1[[i]], {k, intL, intR}];
  posterior1[[i]] = posterior1[[i]]/norm;

  prior = posterior1[[i]];
  (*Least Squares for the First i data points*)
  Fvec = Table[{F1}, {j, 1, i - 1}];
  dataVec = Table[data1[[j]], {j, 1, i - 1}];
  lsSoln[[i]] = 1/LeastSquares[Fvec, dataVec];
  }, {i, 2, numPts + 1}]

max = FindMaximum[posterior1[[1]], {k, kexact}];
mean = NIntegrate[k*posterior1[[1]], {k, intL, intR}];
stDev = NIntegrate[(k - mean)^2*posterior1[[1]], {k, intL, intR}];
Print["------------------------------------------------------------"];
Print[0, " Points"];
Print["Mean = ", mean];
Print["Standard Deviation = ", stDev];
Print["95% Confidence Interval = [", mean - 2*stDev, ",",
 mean + 2*stDev, "]"]
Print[];
Print["Location of Maximum = ", k /. max[[2]]];
Print[];
Plot[posterior1[[1]], {k, 0, 10}, PlotRange -> All,
  AxesLabel -> {"k [N/mm]", "Probability Density"}, PlotLabel -> 0,
  Filling -> Axis] // Print

For[
 i = 0, i <= m, i++,
 max = FindMaximum[posterior1[[2^i + 1]], {k, kexact}];
 mean = NIntegrate[k*posterior1[[2^i + 1]], {k, intL, intR}];
 stDev = NIntegrate[(k - mean)^2*posterior1[[2^i + 1]], {k, intL,
    intR}];
 Print["------------------------------------------------------------"];
 Print[2^i, " Points"];
 Print["Mean = ", mean];
 Print["Standard Deviation = ", stDev];
 Print["95% Confidence Interval = [", mean - 2*stDev, ",",
  mean + 2*stDev, "]"];
 Print[];
 Print["Location of Maximum = ", k /. max[[2]]];
 Print["Least Squares Solution = ", lsSoln[[2^i + 1]][[1]]];
 Print[];
 Plot[posterior1[[2^i + 1]], {k, 0, 10}, PlotRange -> All,
   AxesLabel -> {"k [N/mm]", "Probability Density"}, PlotLabel -> 2^i,
    Filling -> Axis] // Print
 ]
```

B.2 Radioactive Decay Analytical Solution

This section contains the Mathematica code for the radioactive decay problem using the analytical solution of the underlying equations in Section 5.2.5.

```
SetOptions[Plot,
  BaseStyle -> {FontFamily -> "CMU Serif", 18, GrayLevel[0]}];
SetOptions[Plot3D,
  BaseStyle -> {FontFamily -> "CMU Serif", 18, GrayLevel[0]},
  AxesStyle -> Black];
SetOptions[ParametricPlot3D,
  BaseStyle -> {FontFamily -> "CMU Serif", 18, GrayLevel[0]},
  AxesStyle -> Black];
SetOptions[ContourPlot,
  BaseStyle -> {FontFamily -> "CMU Serif", 18, GrayLevel[0],
    AxesStyle -> Black}];
SetOptions[DiscretePlot,
  BaseStyle -> {FontFamily -> "CMU Serif", 18, GrayLevel[0],
    AxesStyle -> Black}];
SetOptions[ListPlot,
  BaseStyle -> {FontFamily -> "CMU Serif", 18, AxesStyle -> Black}];
SetOptions[ParametricPlot,
  BaseStyle -> {FontFamily -> "CMU Serif", 18, GrayLevel[0],
    AxesStyle -> Black}];
SetOptions[ListLogLogPlot,
  BaseStyle -> {FontFamily -> "CMU Serif", 18, GrayLevel[0],
    AxesStyle -> Black}];
SetOptions[DensityPlot,
  BaseStyle -> {FontFamily -> "CMU Serif", 18, GrayLevel[0],
    AxesStyle -> Black}];


(* Set up random Values *)
Print["Seed"]
seed = RandomInteger[{1, 10000}]
SeedRandom[seed];

(* Number of data points *)
ss = 4;
ts = 4;
numSamples = 2ˆss;
numTimes = 2ˆts;
m = ss + ts;
numPts = 2ˆm;
```

```
(* Standard Deviation of Measurement Device*)
\[Sigma]measure = 0.1;

(* Problem Parameters *)
\[Mu] = 1;
\[Sigma] = 0.05;
model = Exp[-c*t];

exactSoln = Exp[-\[Mu]*t];
Plot[exactSoln, {t, 0, 6}, PlotLabel -> "Exact Solution",
 AxesLabel -> {"t", "y"}]

func = Table[0, {i, 1, numSamples}];
Do[
  ktemp = \[Mu] + \[Sigma]*RandomVariate[NormalDistribution[0, 1]];
  If[ktemp > 0,
   func[[i]] =
    Exp[-(\[Mu] + \[Sigma]*RandomVariate[NormalDistribution[0, 1]])*t],
   i = i - 1
   ],
  {i, 1, numSamples}
  ];

tEval = Table[0, {j, 1, numTimes}];
Do[
 {
  temp = 5/numTimes;
  tEval[[j]] = temp*j;
  },
 {j, 1, numTimes}
 ]

data = Table[0, {i, 1, numPts}];
modelEval = Table[0, {i, 1, numPts}];
plotData = Table[0, {i, 1, numPts}];
Do[
  Do[
   {
    data[[j + (i - 1)*numTimes]] =
     func[[i]] /.
      t -> tEval[[
         j]]*(1 +
          RandomVariate[NormalDistribution[0, \[Sigma]measure]]);
    modelEval[[j + (i - 1)*numTimes]] = model /. t -> tEval[[j]];
    plotData[[j + (i - 1)*numTimes]] = {tEval[[j]],
      data[[j + (i - 1)*numTimes]]};
    },
   {j, 1, numTimes}
   ],
  {i, 1, numSamples}
  ];

Plot[{func[[1]], func[[2]], func[[3]], func[[4]], func[[5]],
  func[[6]], func[[7]], func[[8]]}, {t, 0, 6},
 PlotLabel -> "Error Solutions", AxesLabel -> {"t", "y"}]

ListPlot[plotData, AxesLabel -> {"t", "y"}]

priorA = PDF[
   UniformDistribution[
    {0, 10}
    ],
   c];

(* Choose a prior*)
prior = priorA;

(* Initialize posteriod pdfs *)
intL = -\[Infinity];
intR = \[Infinity];
posterior = Table[0, {i, 1, numPts + 1}];

posterior[[1]] = prior;

Do[{
  likelihood =
   1/(\[Sigma]*Sqrt[2*\[Pi]])*
    Exp[-(data[[i - 1]] -
          modelEval[[i - 1]])^2/(2*\[Sigma]^2)]; (*Gaussian likehood*)


  posterior[[i]] = prior*likelihood;

  norm = NIntegrate[posterior[[i]], {c, intL, intR}];
  posterior[[i]] = posterior[[i]]/norm;

  prior = posterior[[i]];
  }, {i, 2, numPts + 1}]
```

```
mean = NIntegrate[c*posterior[[1]], {c, intL, intR}];
stDev = NIntegrate[(c - mean)^2*posterior[[1]], {c, intL, intR}];
max = FindMaximum[posterior[[1]], {c, mean}];
Print["-------------------------------------------------------------"];
Print[0, " Points"];
Print["Mean = ", mean];
Print["Standard Deviation = ", stDev];
Print["95% Confidence Interval = [", mean - 2*stDev, ", ",
 mean + 2*stDev, "]"]
Print[];
Print["Location of Maximum = ", c /. max[[2]]];
Print[];
Plot[posterior[[1]], {c, 0, 10}, PlotRange -> All,
  AxesLabel -> {"c", "Probability Density"}, PlotLabel -> 0,
  Filling -> Axis] // Print

pointsErr = Table[0, {i, 1, m + 1}];
For[
 i = 0, i <= m, i++,
 mean = NIntegrate[c*posterior[[2^i + 1]], {c, intL, intR}];
 stDev = NIntegrate[(c - mean)^2*posterior[[2^i + 1]], {c, intL,
    intR}];
 max = FindMaximum[posterior[[2^i + 1]], {c, mean}];
 maxVal = c /. max[[2]];
 pointsErr[[i + 1]] = {2^i, Abs[maxVal - \[Mu]]};
 Print["-------------------------------------------------------------"];
 Print[2^i, " Points"];
 Print["Mean = ", mean];
 Print["Standard Deviation = ", stDev];
 Print["95% Confidence Interval = [", mean - 2*stDev, ", ",
   mean + 2*stDev, "]"]
  Print[];
 Print["Location of Maximum = ", c /. max[[2]]];
 (*Print["Least Squares Solution = ",lsSoln[[2^i+1]][[1]]];*)
 Print[];
 Plot[posterior[[2^i + 1]], {c, 0, 10}, PlotRange -> All,
   AxesLabel -> {"c", "Probability Density"}, PlotLabel -> 2^i,
   Filling -> Axis] // Print
 ]
ListPlot[pointsErr, AxesLabel -> {"Points", "Error [1/s]"}]
```

B.3 Radioactive Decay Monte Carlo Method

This section contains the Mathematica code for the radioactive decay problem using the Monte Carlo method as in Section 5.2.7.

```
SetOptions[Plot,
  BaseStyle -> {FontFamily -> "CMU Serif", 18, GrayLevel[0]}];
SetOptions[Plot3D,
  BaseStyle -> {FontFamily -> "CMU Serif", 18, GrayLevel[0]},
  AxesStyle -> Black];
SetOptions[ParametricPlot3D,
  BaseStyle -> {FontFamily -> "CMU Serif", 18, GrayLevel[0]},
  AxesStyle -> Black];
SetOptions[ContourPlot,
  BaseStyle -> {FontFamily -> "CMU Serif", 18, GrayLevel[0],
    AxesStyle -> Black}];
SetOptions[DiscretePlot,
  BaseStyle -> {FontFamily -> "CMU Serif", 18, GrayLevel[0],
    AxesStyle -> Black}];
SetOptions[ListPlot,
  BaseStyle -> {FontFamily -> "CMU Serif", 18, AxesStyle -> Black}];
SetOptions[ParametricPlot,
  BaseStyle -> {FontFamily -> "CMU Serif", 18, GrayLevel[0],
    AxesStyle -> Black}];
SetOptions[ListLogLogPlot,
  BaseStyle -> {FontFamily -> "CMU Serif", 18, GrayLevel[0],
    AxesStyle -> Black}];
SetOptions[DensityPlot,
  BaseStyle -> {FontFamily -> "CMU Serif", 18, GrayLevel[0],
    AxesStyle -> Black}];

(* Set up random Values *)
Print["Seed"]
seed = 4736(*RandomInteger[{1,10000}]*)
SeedRandom[seed];

(* Things we know in experiment*)
(* Number of data points *)
ss = 4;
ts = 4;
numSamples = 2^ss;
numTimes = 2^ts;
m = ss + ts;
numPts = 2^m;
```

```
(* Standard Deviation of Measurement Device*)
\[Sigma]measure = 0.1;

(* vvv This is random data that is created.  Ignore all of this when \
using method vvv *)
(* Problem Parameters *)
\[Mu] = 1;
\[Sigma] = 0.05;(*1/3;*)
model = Exp[-c*t];

exactSoln = Exp[-\[Mu]*t];
Plot[exactSoln, {t, 0, 6}, PlotLabel -> "Exact Solution",
 AxesLabel -> {"t", "y"}]

func = Table[0, {i, 1, numSamples}];
Do[
  ktemp = \[Mu] + \[Sigma]*RandomVariate[NormalDistribution[0, 1]];
  If[ktemp > 0,
   func[[i]] =
    Exp[-(\[Mu] + \[Sigma]*RandomVariate[NormalDistribution[0, 1]])*t],
   i = i - 1
   ],
  {i, 1, numSamples}
  ];

tEval = Table[0, {j, 1, numTimes}];
Do[
 {
  temp = 5/numTimes;
  tEval[[j]] = temp*j;
  },
 {j, 1, numTimes}
 ]

data = Table[0, {i, 1, numPts}];
modelEval = Table[0, {i, 1, numPts}];
plotData = Table[0, {i, 1, numPts}];
Do[
  Do[
   {
    data[[j + (i - 1)*numTimes]] =
     func[[i]] /.
      t -> tEval[[j]]*(1 +
          RandomVariate[NormalDistribution[0, \[Sigma]measure]]);
    modelEval[[j + (i - 1)*numTimes]] = model /. t -> tEval[[j]];
    plotData[[j + (i - 1)*numTimes]] = {tEval[[j]],
      data[[j + (i - 1)*numTimes]]};
    },
   {j, 1, numTimes}
   ],
  {i, 1, numSamples}
  ];

Plot[{func[[1]], func[[2]], func[[3]], func[[4]], func[[5]],
  func[[6]], func[[7]], func[[8]]}, {t, 0, 6},
 PlotLabel -> "Error Solutions", AxesLabel -> {"t", "y"}]

ListPlot[plotData, AxesLabel -> {"t", "y"}]
(* ^^^ This is random data that is created.  Ignore all of this when \
using method ^^^ *)




(* Bayesian *)
(* Make Prior *)
priorDistr = UniformDistribution[{0, 10}];
priorA = PDF[
   priorDistr,
   c];
Plot[priorA, {c, -0.5, 11.5}];

Nsamples = 10000; (* Number of samples *)
sum = 0;
LA = 0;
ci = Table[0, {i, 0, Nsamples}];
likelihoodi = Table[0, {i, 0, Nsamples}];
Do[{
  ci[[i]] = RandomVariate[priorDistr];
  priori = priorA /. c -> ci[[i]];
  likelihoodi[[i]] = 1;
  model =
   NDSolve[{D[y[t], t] == -ci[[i]]*y[t], y[0] == 1}, y[t], {t, 0, 6}];
  modely = y[t] /. model;
  Do[
   Do[
    {
     likelihoodi[[i]] =
      likelihoodi[[i]]*1/(\[Sigma]measure*Sqrt[2*\[Pi]])*
        Exp[-(data[[j + (k - 1)*numTimes]] -
              modely)^2/(2*\[Sigma]measure^2)] /.
       t -> tEval[[j]]; (*Gaussian likehood*)
     },
    {j, 1, numTimes}
    ],
   {k, 1, numSamples}
   ];
  LA = LA + likelihoodi[[i]]; (*For Normalization*)
  sum = sum + ci[[i]]*likelihoodi[[i]];
  }
 , {i, 1, Nsamples}
 ]
mean = sum/Nsamples;
area = LA/Nsamples;
normal = mean/area
```

APPENDIX C

STOCHASTIC HEAT EQUATION

This appendix shows work done studying the use of the Monte Carlo method and polynomial chaos to model heat transfer in heterogeneous materials [19]. This study of UQ is different from Bayesian inference, because this the forward problem, so the goal is to study how uncertainty propagates through the problem. This study demonstrates the method of polynomial chaos and shows how it is much less computationally expensive than the Monte Carlo method.

C.1 Introduction

One class of problems that contains uncertainty is the study of heterogeneous materials. A heterogeneous material is a material made of different materials. Heterogeneous materials are commonly found in daily life. For example, Swiss cheese is a heterogeneous material because it is a mixture of cheese and voids. When solving a problem with a heterogeneous material, it is uncertain how the material is mixed. For example, if one were to select a random point in the Swiss cheese, it is uncertain whether one would select cheese or a void. The degree of mixture of the materials will also affect the bulk material properties of the body. The material properties will more closely resemble the material in higher concentration. However, even if two bodies had the same concentrations of materials, it is possible that the materials would be more poorly mixed in one of the bodies that would also affect the material properties. In summary, uncertainty in the heterogeneity in a material affects the material properties, that will affect later predictions. Hence the results of the uncertainty due to material heterogeneity must be quantified. In this report, the effect of material heterogeneity on conduction heat transfer will be studied, because heat transfer through heterogeneous materials has implications in a wide range of fields, like the creation of new materials from heterogeneous mixtures.

C.2 Heat Equation

Conduction heat transfer through a material is often well modeled by

$$\frac{\partial T}{\partial t} = \alpha \nabla^2 T, \tag{C.1}$$

where T is the temperature of the material, t is time, α is the thermal diffusivity of the material, and ∇^2 is the Laplacian operator. Equation (C.1) is a parabolic partial differential equation (PDE) with respect to space and time. It can be solved by a variety of analytic and numerical methods. In this study, the heat equation was solved using the NDsolve function in Wolfram Mathematica. This software uses standard numerical methods to discretize the equations in space and time and generate an approximate solution [20]. The thermal diffusivity of a pure material material is defined as

$$\alpha = \frac{k}{\rho c_p}, \tag{C.2}$$

where k is the thermal conductivity, ρ is the density, and c_p is the specific heat capacity of the material. It is assumed that the thermal diffusivity is a constant material property, because in solids k, ρ, and c_p do not vary much with temperature. Also, α must be positive, because physically, a negative thermal diffusivity would violate the second law of thermodynamics by allowing temperature to flow from a cold body to a hot body. Also, mathematically, solutions to Eq. (C.1) are unstable when α is negative.

C.3 Method

Equation (C.1) is deterministic. It does not contain any uncertainty. Because α is a material property, the uncertainty due to the material heterogeneity is added to the heat equation by defining the thermal diffusivity

$$\alpha(\xi) = \mu_\alpha + \sigma_\alpha \xi, \tag{C.3}$$

where μ_α is the mean of thermal diffusivity, σ_α is the standard deviation of the thermal diffusivity, and $\xi \in (-\infty, \infty)$ is a random variable with a probability distribution function (pdf) of a Gaussian distribution with a mean of 0 and a standard deviation of 1. Hence, $\alpha(\xi)$ has a pdf of a Gaussian with mean of μ_α and a standard deviation σ_α. Each ξ sampled from the Gaussian represents a different thermal diffusivity. However, because there is no spatial component in the formulation of $\alpha(\xi)$, each sample of ξ is equivalent to sampling a new material. For example, if the heterogeneity of a material is described by a Gaussian distribution, like in Eq. (C.3), each ξ value would represent a different mixture of the materials. However, it assumes the same thermal diffusivity throughout the whole body. In order to simplify the problem, it is assumed that the problem is one dimensional. Hence, the stochastic heat equation is modeled as

$$\frac{\partial T(x,t,\xi)}{\partial t} = \alpha(\xi) \frac{\partial^2 T(x,t,\xi)}{\partial x^2}. \tag{C.4}$$

The set of initial and boundary conditions is chosen to be

$$T(0,t) = T_0, \tag{C.5}$$

$$T(1,t) = T_0, \tag{C.6}$$

$$T(x,0) = T_0 + T_0 \sin(\pi x), \tag{C.7}$$

for a domain of $x \in (0, 1)$ and $T_0 = 300$. Because T is now a function of ξ that is described by a pdf, there is also a pdf that represents T at each point in space and time. Equation (C.4) is not a deterministic problem, so it cannot be solved using traditional numerical methods. However, if a specific ξ_0 was sampled from the pdf, Eq. (C.4) could be solved deterministically like Eq. (C.1) [21]. In order to develop a general solution to the stochastic heat equation, three techniques are used: the Monte Carlo method, the intrusive polynomial chaos method, and the non-intrusive polynomial chaos method.

C.3.1 Monte Carlo

The first method used to solve the stochastic heat equation is the Monte Carlo method. This method involves sampling a large number of values of ξ and substituting them into Eq. (C.4). Once a ξ value is substituted, the stochastic heat equation becomes deterministic. Hence for each sampled ξ, the heat equation is solved, and the temperature is evaluated at a specific point (x_0, t_0). Once a large number of ξs are sampled and the resulting values for $T(x_0, t_0)$ are calculated, a histogram of the $T(x_0, t_0)$ is created. As the number of samples increases, the histogram of $T(x_0, t_0)$ approaches the probability density function of $T(x_0, t_0)$. Then the mean and standard deviation of $T(x_0, t_0)$ are calculated to characterize the distribution [22].

The Monte Carlo method is used throughout UQ as a standard to compare other methods against. Because the Monte Carlo method simply involves sampling random variables and solving the PDE numerically, it provides a reliable solution, because the numerical methods used to solve the PDE are chosen so that the errors due to spatial and temporal discretization are negligible. Although the solutions to the PDEs are reliable, an infinite number of ξs must be sampled in order to obtain the true pdf of the temperature at a point. However, a large number of samples will lead to a mean and standard deviation against which other methods can be compared. Although the Monte Carlo method provides reliable solutions, it is computationally expensive because it relies on solving a large number of partial differential equations. Hence, less computationally expensive methods are desirable.

C.3.2 Intrusive Polynomial Chaos

The uncertainty in the thermal diffusivity is approximated using a $P + 1$ term Fourier expansion

$$\alpha(\xi) = \sum_{p=0}^{P} \alpha_p \phi_p(\xi), \tag{C.8}$$

where α_p is a set of coefficients and $\phi_p(\xi)$ is a set of known basis functions. As ξ varies, the value of $\alpha(\xi)$ varies, that causes the output T to vary. Hence, the output temperature is also approximated with a Fourier expansion

$$T(x, t, \xi) = \sum_{p=0}^{P} T_p(x, t) \phi_p(\xi), \tag{C.9}$$

where $T_p(x, t)$ is a set of amplitudes that are a function of space and time. It is worth noting that the term amplitude is not a traditional amplitude that usually connotes the difference between the maximum and minimum of a function. With this approximation, a separation of variables technique can be performed so that the uncertainty in ξ is contained in the basis functions and the spatial and temporal dependence is contained in the amplitudes.

First, the basis functions are chosen to be orthogonal so that

$$\langle \phi_p(\xi), \phi_m(\xi) \rangle = 0, \qquad p \neq m. \tag{C.10}$$

The inner product is defined to be the Gaussian inner product

$$\langle \phi_p(\xi), \phi_m(\xi) \rangle = \int_{-\infty}^{\infty} \phi_p(\xi) \phi_m(\xi) w(\xi) d\xi, \tag{C.11}$$

where $w(\xi)$ is a weighting function. For the Gaussian inner product the weighting function is

$$w(\xi) = \frac{1}{\sqrt{2\pi}} e^{-\xi^2/2} d\xi. \tag{C.12}$$

The basis functions are chosen to be the probabilists' Hermite polynomials because ξ dwells on the domain from negative to positive infinity and they are orthogonal with respect to the Gaussian inner product. The probabilists' Hermite polynomials are

$$\begin{aligned} He_0(\xi) &= 1, && \text{(C.13)} \\ He_1(\xi) &= \xi, && \text{(C.14)} \\ He_2(\xi) &= \xi^2 - 1, && \text{(C.15)} \\ He_3(\xi) &= \xi^3 - 3\xi, && \text{(C.16)} \\ &\vdots && \text{(C.17)} \\ He_p(\xi) &= (-1)^p e^{\xi^2/2} \frac{d^p e^{-\xi^2/2}}{d\xi^p}. && \text{(C.18)} \end{aligned}$$

Hence, the inner product between two basis functions is

$$\begin{aligned} \langle \phi_p(\xi), \phi_m(\xi) \rangle &= \int_{-\infty}^{\infty} He_p(\xi) He_m(\xi) \frac{1}{\sqrt{2\pi}} e^{-\xi^2/2} d\xi, && \text{(C.19)} \\ &= p! \delta_{pm}, && \text{(C.20)} \end{aligned}$$

where δ_{pm} is the Kronecker delta [22].

In order to calculate the coefficients α_p, the Galerkin method is performed by taking the inner product of both sides of Eq. (C.8) with respect to the basis functions.

$$\begin{aligned}
\alpha(\xi) &= \sum_{p=0}^{P} \alpha_p \phi_p(\xi), & \text{(C.21)}\\
\langle \phi_m(\xi), \alpha(\xi) \rangle &= \langle \phi_m(\xi), \sum_{p=0}^{P} \alpha_p \phi_p(\xi) \rangle \qquad m = 0, 1, .., P, & \text{(C.22)}\\
&= \sum_{p=0}^{P} \alpha_p \langle \phi_m(\xi), \phi_p(\xi) \rangle \qquad m = 0, 1, .., P, & \text{(C.23)}\\
&= \sum_{p=0}^{P} \alpha_p p! \delta_{pm} \qquad m = 1, 2, .., P, & \text{(C.24)}\\
&= m! \alpha_m \qquad m = 0, 1, .., P. & \text{(C.25)}
\end{aligned}$$

Then, algebraic manipulation and substituting Eq. (C.3) gives

$$\begin{aligned}
\alpha_p &= \frac{\langle \phi_p(\xi), \alpha(\xi) \rangle}{p!} \qquad p = 0, 1, .., P, & \text{(C.26)}\\
&= \frac{\langle \phi_p(\xi), \mu_\alpha + \sigma_\alpha \xi \rangle}{p!} \qquad p = 0, 1, .., P, & \text{(C.27)}\\
&= \frac{1}{\sqrt{2\pi} p!} \int_{-\infty}^{\infty} He_p(\xi)(\mu_\alpha + \sigma_\alpha \xi) e^{-\xi^2/2} d\xi \qquad p = 0, 1, .., P. & \text{(C.28)}
\end{aligned}$$

Because Eq. (C.28) is an integral of a known function, it can be calculated analytically or numerically. However, because $\mu_\alpha + \sigma_\alpha \xi$ is a polynomial and can be rewritten:

$$\alpha(\xi) = \mu_\alpha + \sigma_\alpha \xi = \mu_\alpha(1) + \sigma_\alpha(\xi) = \mu_\alpha He_0(\xi) + \sigma_\alpha He_1(\xi), \qquad \text{(C.29)}$$

because $He_0(\xi) = 1$ and $He_1(\xi) = \xi$. It is clear that $\alpha(\xi)$ is orthogonal to all Hermite basis functions with $n > 1$, because the Hermite basis functions are orthogonal. It is also clear from Eq. (C.29) that $\alpha_0 = \mu$ and $\alpha_1 = \sigma$. Hence, the integral did not have to be calculated for this case and actually could have been determined by inspection [22].

Next, Eqs. (C.9,C.8) are substituted into the Eq. (C.4):

$$\frac{\partial}{\partial t}\left(\sum_{p=0}^{P} T_p(x,t)\phi_p(\xi)\right) = \left(\sum_{p=0}^{P} \alpha_p \phi_p(\xi)\right)\frac{\partial^2}{\partial x^2}\left(\sum_{m=0}^{P} T_m(x,t)\phi_m(\xi)\right). \tag{C.30}$$

The boundary conditions must also be approximated using a series, so

$$T(0,t,\xi) = 300 = \sum_{p=0}^{P} T_p(0,t)\phi_p(\xi), \tag{C.31}$$

$$T(1,t,\xi) = 300 = \sum_{p=0}^{P} T_p(1,t)\phi_p(\xi), \tag{C.32}$$

$$T(x,0,\xi) = 300 + 300\sin(\pi x) = \sum_{p=0}^{P} T_p(x,0)\phi_p(\xi). \tag{C.33}$$

The initial amplitudes, $T_p(x,t)$ could be calculated using the same process as Eq. (C.28), but by inspection because $\phi_0(\xi) = 1$, it is clear by inspection that

$$T_0(0,t) = 300, \qquad T_1(0,t) = 0, \qquad ..., \qquad T_P(0,t) = 0, \tag{C.34}$$

$$T_0(1,t) = 300, \qquad T_1(1,t) = 0, \qquad ..., \qquad T_P(1,t) = 0, \tag{C.35}$$

$$T_0(x,0) = 300 + 300\sin(\pi x), \qquad T_1(x,0) = 0, \qquad ..., \qquad T_P(x,0) = 0. \tag{C.36}$$

Now, manipulating Eq. (C.30) gives

$$\sum_{p=0}^{P} \frac{\partial T_p(x,t)}{\partial t}\phi_p(\xi) = \sum_{p=0}^{P}\sum_{m=0}^{P} \alpha_p \frac{\partial^2 T_m(x,t)}{\partial x^2}\phi_p(\xi)\phi_m(\xi). \tag{C.37}$$

Once again, the dependence on the random variable, ξ, must be removed in order to calculate the amplitudes $T_p(x,t)$ to approximate the temperature. Once again, the Galerkin method is used by taking the inner product of both sides of Eq. (C.37) with respect to the set of the basis functions, $\phi_l(\xi)$ for $l = 0, 1, ..., P$:

$$\begin{aligned}
\langle \phi_l(\xi), \sum_{p=0}^{P} \frac{\partial T_p(x,t)}{\partial t} \phi_p(\xi) \rangle &= \langle \phi_l(\xi), \textstyle\sum_{p=0}^{P} \sum_{m=0}^{P} \alpha_p \frac{\partial^2 T_m(x,t)}{\partial x^2} \phi_p(\xi)\phi_m(\xi) \rangle, && \text{(C.38)} \\
\sum_{p=0}^{P} \frac{\partial T_p(x,t)}{\partial t} \langle \phi_l(\xi), \phi_p(\xi) \rangle &= \textstyle\sum_{p=0}^{P} \sum_{m=0}^{P} \alpha_p \frac{\partial^2 T_m(x,t)}{\partial x^2} \langle \phi_l(\xi), \phi_p(\xi)\phi_m(\xi) \rangle, && \text{(C.39)} \\
\sum_{p=0}^{P} \frac{\partial T_p(x,t)}{\partial t} (l! \delta_{lp}) &= \textstyle\sum_{p=0}^{P} \sum_{m=0}^{P} \alpha_p \frac{\partial^2 T_m(x,t)}{\partial x^2} \langle \phi_l(\xi), \phi_p(\xi)\phi_m(\xi) \rangle, && \text{(C.40)} \\
l! \frac{\partial T_l(x,t)}{\partial t} &= \textstyle\sum_{p=0}^{P} \sum_{m=0}^{P} \alpha_p \frac{\partial^2 T_m(x,t)}{\partial x^2} \langle \phi_l(\xi), \phi_p(\xi)\phi_m(\xi) \rangle, && \text{(C.41)} \\
\frac{\partial T_l(x,t)}{\partial t} &= \textstyle\frac{1}{l!} \sum_{p=0}^{P} \sum_{m=0}^{P} \alpha_p \frac{\partial^2 T_m(x,t)}{\partial x^2} \langle \phi_l(\xi), \phi_p(\xi)\phi_m(\xi) \rangle . && \text{(C.42)}
\end{aligned}$$

The dependence on ξ has been successfully removed because inner product expression is simply the integral

$$\langle \phi_l(\xi), \phi_p(\xi)\phi_m(\xi) \rangle = \int_{-\infty}^{\infty} \phi_l(\xi)\phi_p(\xi)\phi_m(\xi) \frac{1}{\sqrt{2\pi}} e^{-\xi^2/2} d\xi, \qquad \text{(C.43)}$$

that can be calculated and tabulated. Because there is no longer dependence on ξ, Eq. (C.42) is a system of $P+1$ coupled partial differential equations [22]. The system can be solved using numerical schemes, like NDSolve in Mathematica.

The two-term formulation ($P = 1$) was performed. Evaluating the inner products leads to the following coupled PDEs:

$$\begin{aligned}
T_0(0,t) = 300, \qquad T_1(0,t) = 300, \qquad T_0(1,t) = 300, \qquad T_1(1,t) &= 300, && \text{(C.44)} \\
T_0(x,0) = 300 + 300 \sin(\pi x), \qquad T_1(x,0) &= 0, && \text{(C.45)} \\
\frac{\partial T_0(x,t)}{\partial t} &= \mu \frac{\partial^2 T_0(x,t)}{\partial x^2} + \sigma \frac{\partial^2 T_1(x,t)}{\partial x^2}, && \text{(C.46)} \\
\frac{\partial T_1(x,t)}{\partial t} &= \sigma \frac{\partial^2 T_0(x,t)}{\partial x^2} + \mu \frac{\partial^2 T_1(x,t)}{\partial x^2}. && \text{(C.47)}
\end{aligned}$$

The system of PDEs is solved numerically in Mathematica for the amplitudes $T_0(x,t)$

and $T_1(x,t)$, but these equations may have an analytic solution. Then $T_0(x,t)$ and $T_1(x,t)$ are substituted into Eq. (C.9) to give the solution

$$T(x,t,\xi) = T_0(x,t) + T_1(x,t)\xi. \tag{C.48}$$

In a similar manner to the Monte Carlo method, once the expression for $T(x,t,\xi)$ is found, a large number of ξ values are sampled and substituted into it. Then the results at a specific x_0 and t_0 could be calculated and statistically analyzed. The term $T(x_0,t_0,\xi)$ is a polynomial of ξ because the basis functions are Hermite polynomials. Although both the intrusive polynomial chaos and the Monte Carlo methods sample a large number of ξs, the polynomial chaos method involves calculating the values of polynomials for each ξ, and the Monte Carlo method involves solving a PDE for each ξ value. Also, in the polynomial chaos method the PDEs are solved once, which is advantageous because solving the PDEs is the most computationally expensive part of the method. Hence, one would expect that the intrusive polynomial chaos method is a more efficient method of quantifying the uncertainty [22]. This method is called an intrusive method, because it only requires the formulation of a stochastic version of the original model and not multiple solutions of the original model [2]. However, this method only requires one to solve $P+1$ PDEs, so one would expect that it is less computationally expensive than the Monte Carlo method.

C.3.3 Non-Intrusive Polynomial Chaos

Non-intrusive polynomial chaos is another method that manipulates stochastic problems to solve them more efficiently than the Monte Carlo method. Like in the intrusive polynomial chaos method, the temperature is approximated by a set of linearly independent basis functions and amplitudes to separate the random variable from the deterministic variables:

$$T(x,t,\xi) = \sum_{p=0}^{P} T_p(x,t)\phi_p(\xi). \tag{C.49}$$

Once again, the basis functions were chosen to be the probabilists' Hermite polynomials because they are orthogonal with respect to the Gaussian weighting function. Then a Galerkin formulation is used, and the inner product of both sides of Eq. (C.49) is taken with respect to the basis functions. Then the orthogonality of the basis functions is used to solve for the amplitudes [23]:

$$\langle T(x,t,\xi), \phi_m(\xi)\rangle = \langle \sum_{p=0}^{P} T_p(x,t)\phi_p(\xi), \phi_m(\xi)\rangle \qquad m = 1, 2, .., P, \tag{C.50}$$

$$\langle T(x,t,\xi), \phi_m(\xi)\rangle = \sum_{p=0}^{P} T_p(x,t)\langle \phi_p(\xi), \phi_m(\xi)\rangle \qquad m = 1, 2, .., P, \tag{C.51}$$

$$\langle T(x,t,\xi), \phi_m(\xi)\rangle = \sum_{p=0}^{P} T_p(x,t) m!\delta_{pm} \qquad m = 1, 2, .., P, \tag{C.52}$$

$$\frac{\langle T(x,t,\xi), \phi_m(\xi)\rangle}{m!} = T_m(x,t) \qquad m = 1, 2, .., P. \tag{C.53}$$

Because the amplitudes $T_m(x,t)$ are simply equal to an inner product, the inner product can be expanded to show that the amplitudes are equal to integrals:

$$T_p(x,t) = \frac{\int_{-\infty}^{\infty} T(x,t,\xi)\phi_p(\xi)w(\xi)d\xi}{p!} \qquad p = 1, 2, .., P, \tag{C.54}$$

$$w(\xi) = \frac{1}{\sqrt{2\pi}} e^{-\xi^2/2}. \tag{C.55}$$

In the integral in Eq. (C.54), $\phi_p(\xi)$ and $w(\xi)$ are known, but the integral cannot be evaluated outright because $T(x,t,\xi)$ is unknown. However, the integral can be approximated using Gaussian quadrature. Gaussian quadrature is a method of approximating an integral as a series of weights, W_i and function values at specific points, ξ_i:

$$\int_{-\infty}^{\infty} f(\xi)d\xi \approx \sum_{i=0}^{I} W_i f(\xi_i). \tag{C.56}$$

For the $I+1$ term approximation of the integral, the values for W_i and ξ_i are taken from Appendix D. For this specific problem, the Gaussian quadrature approximation of the integral is

$$T_p(x,t) = \frac{\int_{-\infty}^{\infty} T(x,t,\xi)\phi_p(\xi)w(\xi)d\xi}{p!} \approx \frac{\sum_{i=0}^{I} W_i T(x,t,\xi_i)\phi_p(\xi_i)w(\xi_i)}{p!}. \tag{C.57}$$

Gaussian quadrature is used to approximate the integral. The values of $T(x,t,\xi_i)$ are evaluated for all Gauss points ξ_i values by substituting the values into Eq. (C.4), the stochastic heat equation, and it can be solved deterministically. Hence, for a specific number of terms, P, the heat equation is solved deterministically for each of I ξ_i values and $T_p(x,t)$ is formed by performing a weighted summation. Hence, through Gaussian quadrature, the integral is approximated, and the resulting amplitudes are multiplied by the bases to approximate the temperature distribution like in Eq. (C.49). Once a specific point in space and time, (x_0, t_0) is chosen, the resulting approximation is simply a polynomial function of ξ. Like in the intrusive polynomial chaos method, once the polynomial approximation is created, a large number of ξ values are sampled and the resulting temperature values can be statistically analyzed. This method is non-intrusive polynomial chaos, because the solution is dependent on solving the original PDE multiple times for each ξ_i value for Gaussian quadrature [2].

C.4 Results

In this study, the initial conditions, boundary conditions, and thermal diffusivity approximation were arbitrarily defined. The initial conditions in Eqs. (C.5-C.7) were arbitrarily chosen. Also, the thermal diffusivity as defined in Eq. (C.3) was arbitrarily selected with $\mu = 1$, $\sigma = 0.75$, and ξ as a random variable from a Gaussian distribution with a mean of 0 and a standard deviation of 1. The ratio of standard deviation to mean for α was selected to be 0.75 so there would be a large range of α values. In this formulation, it was assumed that α is described by a Gaussian distribution, but when modeling a real material, one would characterize its heterogeneity to determine the correct distribution to model the material. However, because the Gaussian distribution is fully described by the mean and standard deviation, the heterogeneity in a material could be approximated by a Gaussian by calculating the mean and standard deviation of the thermal diffusivity of the material.

The solution was studied by calculating the temperature at a specific point in space and time. At a specific spatial and temporal point, the temperature is simply a function of ξ. In each method, a large number of ξ values were sampled and the resulting solutions were plotted in a histogram. The mean and standard deviation of the solutions at the specific point were calculated in order to characterize the solution. From these properties, an approximating Gaussian could be created. If this approximation has a large amount of error, other distributions could also be used to approximate the solution by calculating the higher moments of the solution. After an approximate pdf is created, a confidence interval is calculated.

C.4.1 Monte Carlo Method

It was arbitrarily chosen that the temperature would be analyzed at the point $x_0 = 0.3$, $t_0 = 0.3$. However, because the Monte Carlo method involves sampling a large number of ξ values, there is a chance that a negative thermal diffusivity value is selected. Specifically, when $\xi < -4/3$ is chosen, $\alpha < 0$. As previously stated, a negative thermal diffusivity is both physically impossible and mathematically unstable. When such a value is chosen, a non-physical temperature value is also predicted.

For example, with a Gaussian thermal diffusivity, 100 ξ values were sampled and the Monte Carlo method was used to solve the heat equation. The resulting histogram in Fig. C.1 shows that while the majority of the α values produced physical results, one negative thermal diffusivity causes the non-physical results that greatly skew the results. A temperature of 8000 at a point is non-physical when the maximum initial temperature in the domain was 600. The 100 sample ξ values chosen for this trial were relatively small, so solving the same problem with even more samples would cause more non-physical results. One possible remediation would be to change the values of μ_α and σ_α so that it would be less likely that a negative thermal diffusivity was chosen. This method could work, but because a large sample size is needed to correctly model the pdf of the solution, there is still a chance that a non-physical value of α would be realized. Another remedy is to approximate the thermal diffusivity with a distribution that guarantees a positive value. Because the distribution representing α is arbitrary, a positive Gaussian distribution could still approximate the heterogeneity in a real material. The distribution of the heterogeneity of a physical material is guaranteed to be positive because the thermal diffusivity of a material is always positive, so the issue of non-physical results would not occur with a real material. To this end, the Monte Carlo method was solved with a log-normal distribution so that

$$\alpha = e^{\mu_\alpha + \sigma_\alpha \xi}. \tag{C.58}$$

Using a log-normal distribution ensures that the thermal diffusivity is always positive.

In order to get physical results, the thermal diffusivity was modeled with a log-normal distribution as in Eq. (C.58) with $\mu_\alpha = 1$, $\sigma_\alpha = 0.75$ and ξ modeled by a Gaussian with mean 0 and standard deviation 1 as before. In order to evaluate the solution well before it is fully relaxed to a steady state, the temperature was evaluated at $x_0 = 0.3$ and $t_0 = 0.01$. Using the Monte Carlo method, the stochastic heat equation was solved for $50,000$ random ξ values. The resulting histogram in Fig.

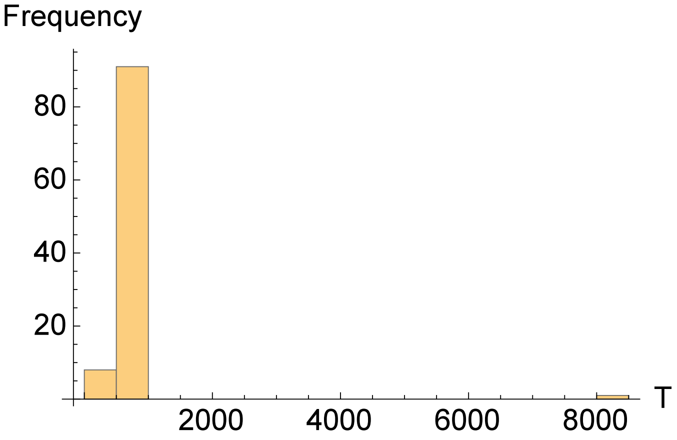

Figure C.1. The Monte Carlo method was used to solve the heat equation. The solution at $T(x,t)$ was sampled at $x_0 = 0.3$, $t_0 = 0.3$ for 100 trials with a Gaussian α and it shows the non-physical results of negative thermal diffusivity values.

C.2 shows that the resulting temperature pdf resembles a log-normal distribution. The mean of these results is $\mu = 477$ and the standard deviation is $\sigma = 40$. There is 95% confidence that $397 \leq T(0.3, 0.01, \xi) \leq 557$. The total run time for the Monte Carlo method was 4.27 hours.

A convergence study was also conducted on the Monte Carlo method by measuring the solution for varying N, the number of ξ values sampled. Then the solution for $50,000$ ξ values was treated as the exact solution and the percentage of error of each solution was calculated. Figure C.3 shows the results of the convergence study for the mean and standard deviation. As the number of samples increases, the mean and standard deviation both converge. The slope of the log-log plot of the mean is -0.57 and the slope of the log-log plot of the standard deviation is -0.58. Also, Figure C.4 shows that as the number of samples increases, the computational time increases. The slope of the log-log plot is about 1, so the computational time varies linearly with the number of samples. This result was expected because the program

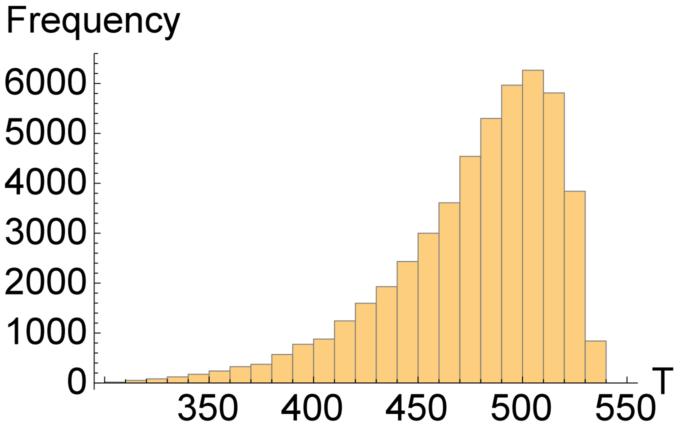

Figure C.2. The Monte Carlo method was used to solve the heat equation. The solution at $T(x,t)$ was sampled at $x_0 = 0.3$, $t_0 = 0.01$ for $50,000$ trials with a log-normal α and it shows that the resulting pdf also resembles a log-normal distribution.

should solve each PDE in the same amount of time and increasing the number of samples should not affect the time of to solve the PDE once. Hence, increasing the number of terms further would cause additional increase in the computation time. In conclusion, the mean and standard deviation of the temperature converge with increasing the number of samples and the computational time required increases with the number of samples.

C.4.2 Intrusive Polynomial Chaos

The intrusive polynomial chaos method was used to solve the heat equation with Gaussian thermal diffusivity. Although the Monte Carlo method is unable to solve the problem with negative α values, intrusive polynomial chaos has no issues. The Gaussian thermal diffusivity must be used for this method because α is represented by a series of polynomial basis functions as in Eq. (C.8). Because the Gaussian representation of α is also a polynomial, the series in Eq. (C.8) has a finite number of terms. However, if a log-normal distribution were used, an infinite number of

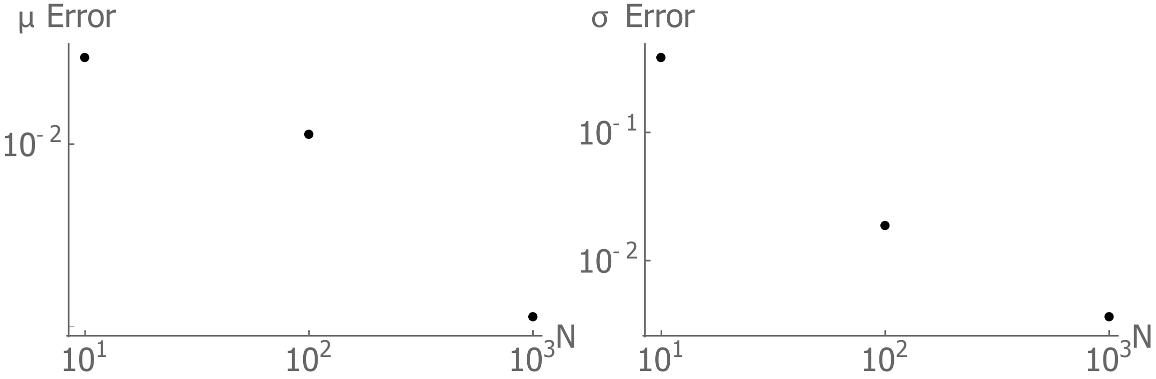

Figure C.3. The convergence study shows that the slope of the best fit lines of the mean and standard deviation are −0.57 and −0.58 respectively.

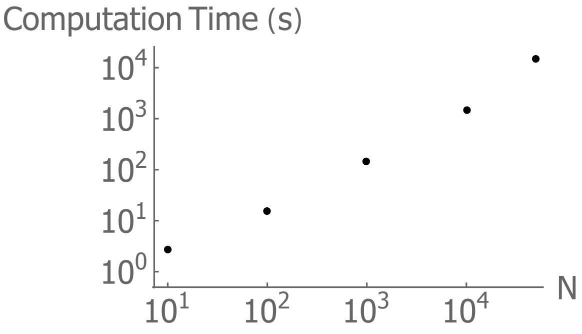

Figure C.4. The Monte Carlo convergence study shows that increasing the number of ξ values sampled causes an exponential increase in the computational time. The slope is approximately unity.

terms in the series would be required to approximate the thermal diffusivity. Unless a large number of terms were used, the solution to the stochastic heat equation would not be accurate. Hence, an α that can be approximated by a finite number of Hermite polynomials is desirable. However, because the representation of the thermal diffusivity is different in the intrusive polynomial chaos method than in the Monte Carlo method, there is no solution to which to compare the results.

A Gaussian thermal diffusivity was used with a mean of 1 and a standard deviation of 0.75 was used to solve the stochastic heat equation using intrusive polynomial chaos. The arbitrary point $x_0 = 0.3$ and $t_0 = 0.3$ was chosen so that the temperature had not reached steady state. A two term approximation of the solution was used. Once the amplitudes were evaluated as described previously, $50,000$ values of ξ were sampled and substituted into the resulting polynomial determine the temperature at the (x_0, t_0). Figure C.5 shows the resulting histogram form the trials. The pdf resembles a normal distribution as expected. The mean and standard deviation were 359 and 57 respectively. There is 95% confidence that $244 \leq T(0.3, 0.3, \xi) \leq 473$. The total run time was 2.28 seconds.

C.4.3 Non-Intrusive Polynomial Chaos

Because the non-intrusive polynomial chaos method relies upon solving the stochastic heat equation deterministically for certain ξ_i values, the thermal diffusivity could be approximated with a Gaussian distribution assuming the $\xi_i \not< -4/3$. However, using a log-normal distribution lifts these restrictions and ensures that the results are directly comparable to the Monte-Carlo results. Hence, the log-normal distribution as described in Eq. (C.58) was used with $\mu_\alpha = 1$, $\sigma_\alpha = 0.75$, and ξ as a random variable from at Gaussian distribution with a mean of 0 and a standard deviation of 1. The temperature was evaluated at $x_0 = 0.3$ and $t_0 = 0.01$. The solutions were calculated with $I = P$. $50,000$ ξ values were sampled and a histogram of the resulting

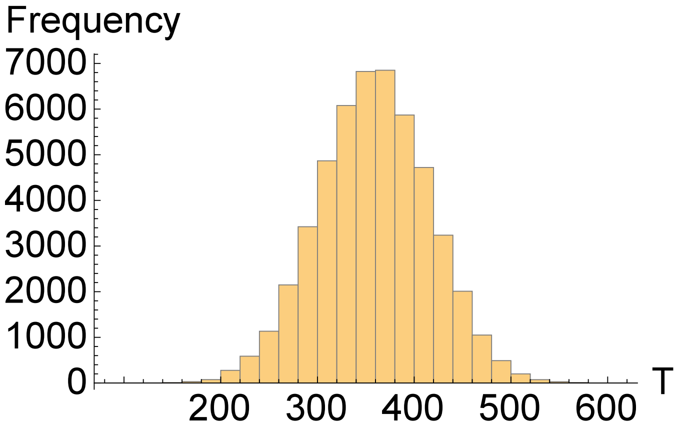

Figure C.5. Intrusive polynomial chaos with 2 terms was used to solve the heat equation. The solution at $T(x,t)$ was sampled at $x_0 = 0.3$, $t_0 = 0.3$ for $50,000$ trials with a Gaussian α and it shows that the resulting pdf also resembles a normal distribution.

temperature for a ten-term approximation is shown in Figure C.6. The resulting pdf appears to be a log-normal distribution. It has a mean of 476.7, a standard deviation of 40, and 95% confidence that $397 \leq T(0.3, 0.01, \xi) \leq 557$. The run time was 1.54 seconds.

The convergence of the non-intrusive polynomial chaos method was studied with increasing the number of terms in the approximation. Figure C.7 shows that both the mean and standard deviation converge as the number of terms in the approximation is increased. The error of the solutions relative to the ten term solution. Figure C.8 shows that the slope of the error of the mean is -0.41 and the slope of the error of the standard deviation is -0.33. The mean has less than 0.5% error after five terms and the standard deviation has less than 0.5% error after nine terms. Hence the mean converges more quickly than the standard deviation. One would expect the mean to converge more quickly because the higher moments of a distribution, like the standard deviation, are finer details that require more resolution to resolve.

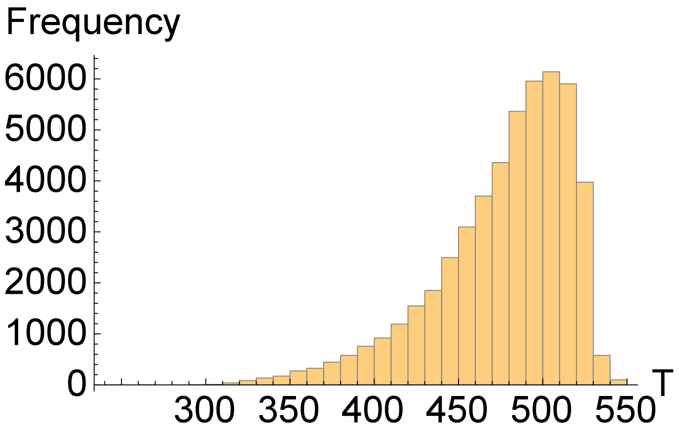

Figure C.6. Non-intrusive polynomial chaos with 10 terms was used to solve the heat equation. The solution at $T(x,t)$ was sampled at $x_0 = 0.3$, $t_0 = 0.01$ for $50,000$ trials with a log-normal α and it shows that the resulting pdf also resembles a log-normal distribution.

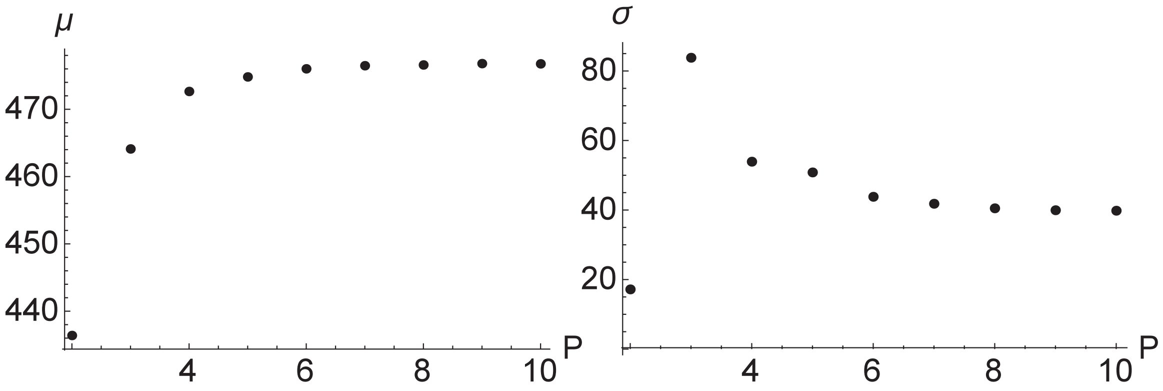

Figure C.7. The convergence study shows that the mean converges more quickly than the standard deviation when increasing the number of terms in the approximation

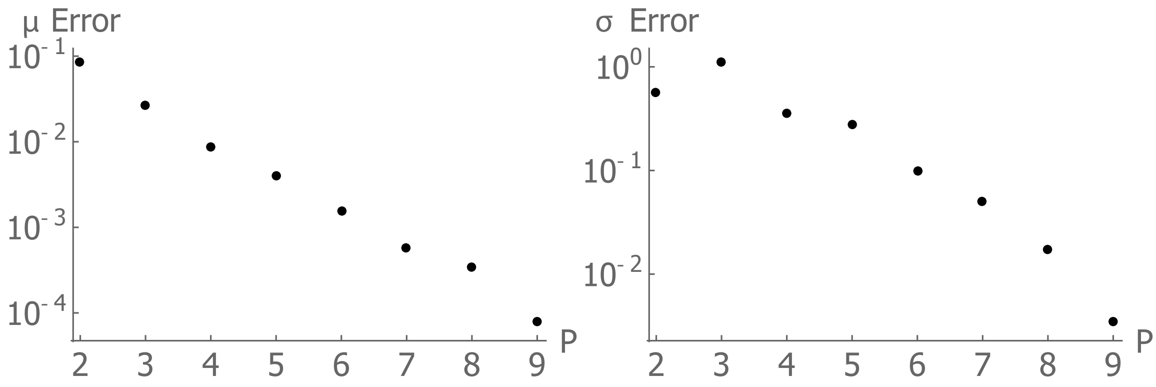

Figure C.8. The slope of the log linear plot of the mean and standard deviation errors are −0.41 and −0.33 respectively.

C.5 Discussion and Conclusions

This study has shown that the Monte Carlo method does provide a solution to the stochastic heat equation. Because the numerical methods used to solve the deterministic PDE are chosen so that errors due to spatial and temporal discretization are negligible, the solutions can be trusted and used for comparison. However, in order to get solutions for $50,000$ samples of ξ, the Monte Carlo method took 4.27 hours. In order to obtain more accurate solutions with the method, additional samples must be taken that further increases the computational expense. However, the Monte Carlo method could be sped up by using parallel programming. Because the method is based on solving a large number of computationally intense problems, and then combining the results at the end, it lends well to parallel programming because separate processors could sample ξ values and solve the problem independently and then combine the results in a histogram at the end. Hence, if $1,000$ processors were used, each processor would only have to solve 50 PDEs to reach $50,000$ total samples, that would greatly reduce the wall time used.

The thermal diffusivity was modeled with a log-normal distribution when using the Monte Carlo and non-intrusive polynomial chaos methods in order to ensure positive values for α. If the heterogeneity of a physical material were characterized, the thermal diffusivity would always be positive. Therefore, all sampled thermal diffusivity values would be positive, so regardless of the type of distribution the values would not cause mathematical issues in the model.

The intrusive polynomial chaos method provided reasonable solutions, but the solution could not be compared to the Monte Carlo method because the Monte Carlo method was unable to solve the heat equation with a Gaussian thermal diffusivity. However, although the solution could not be confirmed, it could be used to model a physical material, because a physical material could have a Gaussian distribution. If a material were characterized as a Gaussian, this method could be compared to the Monte Carlo method and validated by experiment. Intrusive polynomial chaos also showed large performance improvements relative to the Monte Carlo method. Without any parallelization, the intrusive polynomial chaos method took only a matter of seconds as opposed to the hours the Monte Carlo method took.

The non-intrusive polynomial chaos method provided a solution that could be compared with the Monte Carlo method. From a visual comparison of Figs. C.2 and C.6, it is clear that the methods produced similar solutions. The non-intrusive polynomial chaos mean was 0.04% different from the Monte Carlo method. The standard deviations were 0.4% different. The larger difference in standard deviations is due to the slower convergence of the higher moments. The small difference in the solutions from these methods verifies the polynomial chaos method. One may also notice that it is possible for the non-intrusive polynomial chaos method to produce non-physical results. For example, Fig. C.6 shows temperatures lower than 300, which is not possible for the given boundary conditions. These cases can occur for ξ values far from the mean. However, because the log-normal distribution has compact support, the frequency of cases at the extremes are of the pdf are 0 [21]. Hence, the non-physical solutions do not interfere with the accuracy of the model. Also, non-intrusive polynomial chaos produced similar results to the Monte Carlo method in much shorter time.

Overall, all three methods are able to solve the stochastic heat equation. The Monte Carlo method produces reliable results but takes a large amount of time. Also, it cannot solve the problem with a Gaussian thermal diffusivity. The intrusive polynomial chaos method provides a solution for a Gaussian α, but could not be compared with the Monte Carlo results. The non-intrusive polynomial chaos solution converges to the same solution as the Monte Carlo method. Hence it can be trusted.

APPENDIX D

GAUSSIAN QUADRATURE

Gaussian quadrature is a method to approximate an integral as a sum, similar to the trapezoidal rule. In general the quadrature formula is

$$I = \int_a^b f(\xi) d\xi \approx \sum_{i=1}^{I} W_i f(\xi_i), \tag{D.1}$$

where $f(\xi)$ is a general function, W_i are weights, and ξ_i are specific abscissas. The weights and abscissas are chosen based on the method. For the trapezoidal rule these values are chosen so that a line is integrated between the two adjacent points in the approximation. However, in Gaussian quadrature, the W_i and ξ_i values are chosen for the most accurate approximation for a specific number of terms. On an infinite interval, $(a, b) \to (-\infty, \infty)$, the abscissa values are chosen to be the zeros of the nth physicists' Hermite polynomial. The physicists' Hermite polynomials are different than the probabilists' Hermite polynomials discussed in this report:

$$\begin{aligned} H_0(\xi) &= 1, && \text{(D.2)} \\ H_1(\xi) &= 2\xi, && \text{(D.3)} \\ H_2(\xi) &= 4\xi^2 - 2, && \text{(D.4)} \\ H_3(\xi) &= 8\xi^3 - 12\xi, && \text{(D.5)} \\ &\vdots && \text{(D.6)} \\ H_n(\xi) &= (-1)^n e^{\xi^2} \frac{d^n e^{-\xi^2}}{d\xi^n}. && \text{(D.7)} \end{aligned}$$

However, with a small effort, one could also formulate the Gaussian quadrature for the probabilists' Hermite polynomials. The weights for an infinite domain are

$$W_i = \frac{2^{n-1} n! \sqrt{\pi}}{n^2 [H_{n-1}(\xi_i)]^2}. \tag{D.8}$$

The abscissas and weights for Gaussian quadrature for up to $I = 10$ are tabulated in Tables D.1-D.9 [24].

TABLE D.1

Gaussian quadrature ξ_i and W_i values for $I = 2$

$\pm\xi_i$	W_i
0.70710 67811 86548	1.46114 11826 611

TABLE D.2

Gaussian quadrature ξ_i and W_i values for $I = 3$

$\pm\xi_i$	W_i
0.00000 00000 00000	1.18163 59006 037
1.22474 48713 31589	1.32393 11752 136

TABLE D.3

Gaussian quadrature ξ_i and W_i values for $I = 4$

$\pm\xi_i$	W_i
0.52464 76232 75290	1.05996 44828 950
1.65068 01238 85785	1.24022 58176 958

TABLE D.4

Gaussian quadrature ξ_i and W_i values for $I = 5$

$\pm\xi_i$	W_i
0.00000 00000 00000	0.94530 87204 829
0.95857 24646 13819	0.98658 09967 514
2.020184 28704 56086	1.18148 86255 360

TABLE D.5

Gaussian quadrature ξ_i and W_i values for $I = 6$

$\pm\xi_i$	W_i
0.43607 74119 27617	0.87640 13344 362
1.33584 90740 13697	0.93558 05576 312
2.35060 49736 74492	1.13690 83326 745

TABLE D.6

Gaussian quadrature ξ_i and W_i values for $I = 7$

$\pm\xi_i$	W_i
0.00000 00000 00000	0.81026 46175 568
0.81628 78828 58965	0.82868 73032 836
1.67355 16287 67471	0.89718 46002 252
2.65196 13568 35233	1.10133 07296 103

TABLE D.7

Gaussian quadrature ξ_i and W_i values for $I = 8$

$\pm\xi_i$	W_i
0.38118 69902 07322	0.76454 41286 517
1.15719 37124 46780	0.79289 00483 864
1.98165 67566 95843	0.86675 26065 634
2.93063 74202 57244	1.07193 01442 480

TABLE D.8

Gaussian quadrature ξ_i and W_i values for $I = 9$

$\pm\xi_i$	W_i
0.00000 00000 00000	0.72023 52156 061
0.72355 10187 52838	0.73030 24527 451
1.46855 32892 16668	0.76460 81250 946
2.26658 05845 31843	0.84175 27014 787
3.19099 32017 81528	1.04700 35809 767

TABLE D.9

Gaussian quadrature ξ_i and W_i values for $I = 10$

$\pm\xi_i$	W_i
0.34290 13272 23705	0.68708 18539 513
1.03661 08297 89514	0.70329 63231 049
1.75668 36492 99882	0.74144 19319 436
2.53273 16742 32790	0.82066 61264 048
3.43615 91188 37738	1.02545 16913 657

APPENDIX E

VERIFICATION OF THE WAVELET ADAPTIVE MULTIRESOLUTION REPRESENTATION METHOD

E.1 Introduction

A goal of many computational methods is to predict results observed in nature. The Wavelet Adaptive Mutliresolution Representation (WAMR) method is a computational method of solving problems that guarantees error less than a user-prescribed error, which is advantageous because many other methods only provide increased accuracy with a finer grid. As with any method, it must be verified so that it may be trusted on new problems. As Roache said [25], "The identification, elimination, and engineering 'proof' or demonstration of the absence of coding errors are the concern of Code Verification." Verification is a concept that is commonly confused with validation. Validation is solving the correct mathematical model, while verification is solving the mathematical model correctly. This study is only concerned with verification. In order to accurately verify the WAMR method, it was used to solve many test cases with accurate reference solutions. The test problems used are benchmark problems from the Enhanced Verification Test Suite (EVTS) [26] that are commonly used for verification of simulation codes. The goal of this verification is to allow the WAMR method to be used predictively and in cases where producing a solution to an automatically verified amount of error is advantageous. Unlike most codes, the evaluation of the WAMR method's success is not on decreasing error as the grid is refined. Instead, the method's success is measured by whether the solution meets the user-prescribed error criteria. These test problems will show that the WAMR method does guarantee that the solution meets a user-prescribed error; hence, the solutions are automatically verified.

The problems selected from the EVTS were selected because they each present different challenges for numerical codes. In general, the problems solved are multi-scale hydrodynamics problems. It is particularly important to find effective methods to solve multi-scale problems because they are computationally expensive [27]. The WAMR method is particularly useful for solving multi-scale problems because it provides and automatically verified solution, so once the problem is solved, no more effort will be necessary to verify the solution. The problems solved in this report all have analytical solutions or well-verified numerical solutions, so that the error of the solution from the WAMR method can be quantitatively verified.

E.2 Mathematical Model

The problems modeled are from hydrodynamics. More specifically, they are shock hydrodynamics problems with solutions that are the weak solutions of partial differential equations. In the EVTS the problems are modeled by the compressible Euler equations. However, the WAMR method cannot solve problems with embedded mathematical discontinuities, so viscosity and heat conduction terms were introduced into the model. In reality, the Euler equations are non-physical because there will always be diffusion at a sufficiently small scale. Hence the WAMR method solves more physically relevant problems than the problems listed in the EVTS. Since the method is effective for multidimensional and multi-species problems, the general form of the Navier-Stokes equations solved for the problems are

$$
\begin{aligned}
\frac{\partial \rho}{\partial t} &= -\frac{\partial}{\partial x_i}\left(\rho u_i\right) && \text{(E.1)}\\
\frac{\partial \rho u_i}{\partial t} &= -\frac{\partial}{\partial x_j}\left(\rho u_j u_i + p\delta_{ij}\right) + \frac{\partial \tau_{ij}}{\partial x_j} && \text{(E.2)}\\
\frac{\partial \rho E}{\partial t} &= -\frac{\partial}{\partial x_j}\left(u_j(\rho E + p)\right) + \frac{\partial}{\partial x_i}\left(u_j \tau_{ji} - q_i\right) && \text{(E.3)}\\
\frac{\partial \rho Y_k}{\partial t} &= -\frac{\partial}{\partial x_i}\left(u_i \rho Y_k\right) - \frac{\partial j_{ik}}{\partial x_i}, \qquad k = 1, \ldots, K-1, && \text{(E.4)}
\end{aligned}
$$

where t is the time, x_i is the position in the i direction, ρ is the density, u_i is the component of the velocity vector in the i direction, p is the pressure, δ_{ij} is the Kronecker delta, τ_{ij} is the viscous stress tensor for the i face and in the j direction, E is the specific total energy, q_i is the heat flux component in the i direction, Y_k is the mass fraction of species k, j_{ik} is the species mass flux of species k in the i direction, and M_k is the molecular mass of species k. The problems model ideal mixtures of calorically perfect ideal gas which are Newtonian fluids satisfying the Fick's law and

Fourier's law, so the constitutive relations are

$$\tau_{ij} = \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) + \left(\kappa - \frac{2}{3}\mu\right)\frac{\partial u_l}{\partial x_l}\delta_{ij} \tag{E.5}$$

$$q_i = -\lambda\frac{\partial T}{\partial x_i} + \sum_{k=1}^{K}\left(h_k j_{ik} - \frac{RT}{M_k X_k} D_k^T d_{ik}\right) \tag{E.6}$$

$$j_{ik} = \frac{\rho Y_k}{\overline{M} X_k}\sum_{\substack{j=1\\ j\neq k}}^{K} M_j D_{kj} d_{ij} - \frac{D_k^T}{T}\frac{\partial T}{\partial x_i} \tag{E.7}$$

$$d_{ik} = \frac{\partial X_k}{\partial x_i} + (X_k - Y_k)\frac{1}{p}\frac{\partial p}{\partial x_i} \tag{E.8}$$

$$e = c_v T + e_0 \tag{E.9}$$

$$E = e + \frac{1}{2}u_i u_i \tag{E.10}$$

$$p = \frac{\rho R T}{\overline{M}} \tag{E.11}$$

$$\sum_{k=1}^{K} Y_k = 1, \tag{E.12}$$

where μ is the shear mixture viscosity, κ is the bulk mixture viscosity, λ is the mixture thermal conductivity, T is the absolute mixture temperature, h_k is the enthalpy of species k, R is the universal gas constant, X_k is the mole fraction of species k, D_k^T are the thermal diffusion coefficients, d_{ik} is the mixture diffusion driving force of species k in the i direction, $\overline{M}$ is the mixture molecular mass as defined in as:

$$\overline{M} = \sum_{k=1}^{K} Y_k m_k \tag{E.13}$$

where $\overline{M}$ is the mixture molecular mass, Y_k is the mass fraction of species k, and m_k is the molecular mass of species k, D_{kj} are the ordinary multicomponent diffusion coefficients, e is the mixture specific internal energy, c_v is the mixture specific heat at constant volume, and e_0 is the initial mixture specific internal energy.

The governing equations can be simplified in the one-dimensional, single species case. In the simplified case Eq. (1) becomes the same equation with $i = 1$. In the one-dimensional case $\delta_{ij} = 1$, $i = 1$, and $j = 1$ in Eq. (2) and Eq. (3). Eq. (4) is unnecessary with one component. The values of $j_{i,k}$ and $d_{i,k}$ are 0 in the single component case, so many terms from Eq. (5) and Eq. (6) can be removed along with the subscripts. Eq. (7), Eq. (8), and Eq. (12) become unnecessary when the problem is simplified. Eq. (9) remains unchanged. Eq. (10) has $i = 1$ and Eq. (12) can use the mass of the single species for $\overline{M}$. Also, since the gases are assumed to be calorically perfect ideal gases

$$e = c_v T, \tag{E.14}$$

$$\gamma = 1 + \frac{R}{c_v}, \tag{E.15}$$

$$\gamma = \frac{c_p}{c_v}, \tag{E.16}$$

where c_v is the specific heat of the mixture at constant volume, γ is the ratio of specific heats, and c_p is the specific heat of the mixture at constant pressure. Hence, the governing equations become much simpler in the one-dimensional, one species case. The problems are described dimensionally in the EVTS, so they were solved dimensionally in order to verify the WAMR method using the same method from the EVTS. The WAMR method can also solve the non-dimensionalized equations. One advantage of solving the non-dimensional equations is that solving the problem once would verify infinitely many other problems. Also, the EVTS unfortunately is restricted to problems with artificially low temperatures. With the non-dimensional equations one could solve a physical problem with the same solution as the non-physical problem in the EVTS. However, in order to verify the WAMR method according to the EVTS, the problems were solved dimensionally.

The EVTS initially models the shock as a perfect jump using the Heaviside step function. Once viscosity is introduced into the model, it is impossible to model the shock with the Heaviside step function. Instead, shocks were initially modeled using hyperbolic tangents as shown. This approximation also makes the model more physical than the model in the EVTS, because step functions do not occur in nature. An example of a function used to initially model the pressure in a one-dimensional shock is

$$p = \frac{1}{2}\left[(p_H + p_L) - (p_H - p_L)\tanh\left(\frac{x - x_0}{\delta}\right)\right], \tag{E.17}$$

where p_H is the high pressure, p_L is the low pressure, x is the position, x_0 is the position of the shock, and δ is the thickness of the shock. For each problem δ is chosen to be small relative to the domain in order to most closely resemble the Heaviside step function.

The specific conditions for each problem will be explained later. The Sod problem, Shu-Osher problem, and Woodward-Colella problem all model one-dimensional, single species shocks, while the Richtmyer-Meshkov instability problem has multiple species and was solved in two-dimensions. The Sod problem models a shock caused by a jump in pressure [28]. The Shu-Osher problem models a shock with a sinusoidally varying pre-shock density [29]. The Woodward-Colella problem models the interactions of two shocks caused by two jumps in pressure [30]. The Richtmyer-Meshkov instability problem models a shock wave passing through a gas curtain in two dimensions [31].

E.3 Method

The WAMR method provides an accurate solution that meets a user-prescribed error value, ε. The method uses wavelet and scaling functions to form a multiscale basis. The first scaling function represents the function at the coarsest level. Then each of the following wavelets provide finer representations of the function. The wavelet amplitudes are the difference between the actual function value at a point and the value interpolated from the next coarsest grid. Hence the wavelet amplitudes provide a measure for the local approximation error at each collocation point. Then wavelets with amplitudes smaller than the user-prescribed error threshold are discarded. This method allows the WAMR method to limit the number of basis vectors while also guaranteeing that the solution meets the user-prescribed error. Representing a state variable $u\left(\mathbf{x}\right)$ as a basis of wavelets with a threshold parameter, ε is given by the equation:

$$u^J(\mathbf{x}) = \underbrace{\sum_{\mathbf{k}} u_{0,\mathbf{k}}\Phi_{0,\mathbf{k}}(\mathbf{x}) + \sum_{j=0}^{J-1} \sum_{\{\boldsymbol{\lambda}\,:\,|d_{j,\boldsymbol{\lambda}}|\geq\varepsilon\}} d_{j,\boldsymbol{\lambda}}\Psi_{j,\boldsymbol{\lambda}}(\mathbf{x})}_{u_\varepsilon^J} \quad + \underbrace{\sum_{j=0}^{J-1} \sum_{\{\boldsymbol{\lambda}\,:\,|d_{j,\boldsymbol{\lambda}}|<\varepsilon\}} d_{j,\boldsymbol{\lambda}}\Psi_{j,\boldsymbol{\lambda}}(\mathbf{x}),}_{R_\varepsilon^J}$$

where u_ε^J represents the wavelets with amplitude greater than the threshold and R_ε^J represents the wavelets with amplitude less than the threshold. Then the R_ε^J term is discarded in order to make a sparse grid. Figures E.1 and E.2 show an example function and the wavelet representation of the function [32]. These plots show how the WAMR method limits the number of necessary points by only putting the finest grid on a small portion of the domain. The WAMR method is also able to limit the time discretization error by using the smallest time step throughout the entire domain. The WAMR method is also able to adapt the grid with respect to space based on the details of the problem, which reduces the computation time necessary without affecting the detail of the solution. The test cases in this report will demonstrate that the WAMR method produces verified results based on the user-prescribed error criteria. [32]

Since the goal is to show that the WAMR method is automatically verified, the error of the method must be evaluated. The solution for every parameter in the problem will have less error than that threshold. When an analytical solution is available, the error would be evaluated using the analytical solution, but for many

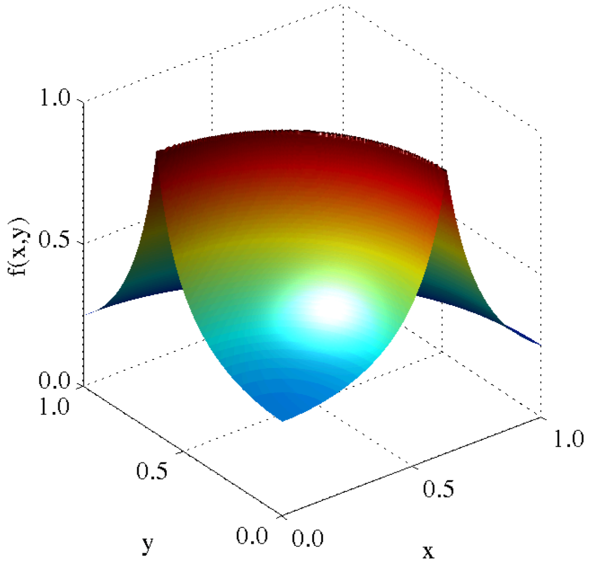

Figure E.1. Example function $f(x, y) = 0.2/(|0.4 - x^2 - y^2| + 0.2)$

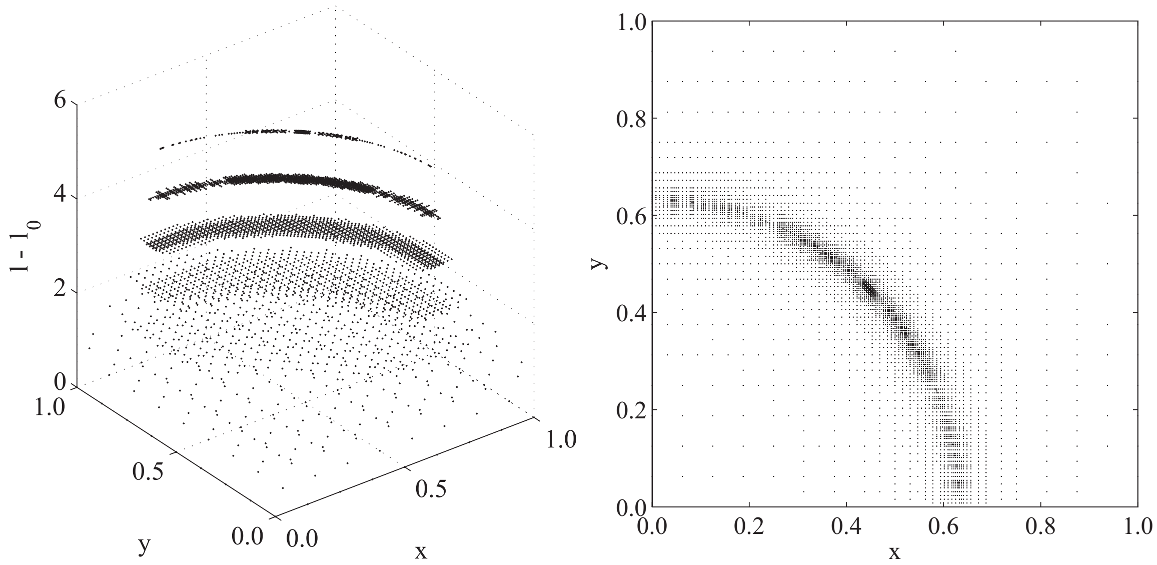

Figure E.2. Collocation points of the example function for $\varepsilon = 5 \times 10^{-3}$

problems there is no analytical solution. In those cases a reference solution was calculated using a finer grid than the grid created by the highest wavelet level of any of the solutions. The reference solutions were created by solving the problem using the same method, but with a very fine uniform grid instead of using the wavelet interpolation. Since the reference solution is created using a grid finer than any portion of the grid of the other solutions, the reference solution is guaranteed to be more accurate than the other solutions. Once an analytical or reference solution was found or created, the error was evaluated at every point on the grid for a specific variable at a fixed time. Then the maximum error was taken for each variable to verify that it was less than the user-prescribed error. The error formula is

$$error = \left\| \frac{U_n - U_a}{U_a} \right\|_\infty , \tag{E.18}$$

where U_n is the calculated solution value of field variable U and U_a is the actual value for field variable U. The error is evaluated at a fixed time at all points on the grid. Each problem was run for multiple ε values so that it could be confirmed that the error is automatically verified to be less than the user-prescribed error. The error for multiple different parameters was evaluated for each ε. The resolution of a solution can be calculated by dividing the size of the domain by the number of grid points used to approximate the solution.

E.4 Results

Each of the problems was solved using multiple values of ε so that it can be verified that the error in the solution is automatically less than the used prescribed error threshold. Once the solutions for different ϵ values were complete, a reference solution was created using a uniform grid that is finer than the grid produced from the highest wavelet level of the other solutions. Then the error was evaluated based on the reference solutions. As the WAMR method solves the problem until a set final time, a set number of snapshots of the solution are saved in order to show that the method is correctly modeling all of the hydrodynamic interactions.

E.4.1 Sod Shock Tube Problem

The Sod shock tube problem is a problem modeled by the one-dimensional Euler equations. It is commonly used to verify computational hydrodynamics codes' abilities to handle hyperbolic equations. Initially two chambers of the gas are separated by an infinitely thin diaphragm. Each section of the chamber contains different initial conditions. Many different sets of initial conditions can be used in the Sod problem, and the choice of initial conditions affects the computational challenges of the problem. For this verification two sets of initial conditions from the EVTS were used [28]. The problem could be physically modeled in a shock tube by removing a diaphragm or by puncturing a diaphragm with a plunger. Then the resulting shock could be observed.

Since the Sod problem is a one-dimensional, single species problem, the simplifications to the governing equations were used to solve the problem. The gas in the problem is a calorically perfect, ideal gas. The calorically perfect equation of state for an ideal gas is $p = (\gamma - 1)\rho e$ where γ is the ratio of specific heats. The problem statement assumed that the $\gamma = 7/5$ for the gas in use. In order to add viscosity to the inviscid problem defined in the EVTS, the gas was assumed to be diatomic nitrogen based on the given value of γ. However, the given parameters are so small that the initial temperature would be less than 1 K, which would mean the nitrogen is a liquid. Since gases are being modeled, the coefficient of viscosity and the coefficient of thermal expansion were assumed the be the coefficients for N_2 at 300 K instead of at the temperature prescribed in the problem. The coefficients were found to be $\mu = 1.813 \times 10^{-4}$ gm/(cm s) and $\lambda = 2618.7$ erg/(cm s K) using the CHEMKIN database. For this problem κ was taken to be 0. Since the problem contains a single shock, the shock was modeled in the same form as Eq. (16). For the Sod problems $\delta = 10^{-2}$ cm was chosen. For both sets of initial conditions the initial domain was divided into 100 grid points so that the initial grid accurately portrayed the solution while still allowing for the adaptive algorithm to efficiently display the solution.

Although the Sod problem as described in the EVTS has an analytical solution, that solution could not be used for error evaluation because the added viscosity makes it impossible to find an analytical solution. Instead, a reference solution was found by solving the problem using a fine grid. A fine grid of 102400 points was use to create the reference solution for each set of initial conditions.

E.4.1.1 Sod Problem 1

The first set of initial conditions is a simple test that is used to find problems resolving wave structure. There is an 8:1 density ratio and a 10:1 pressure ratio between the left and right sides of the shock. Initially, all of the gas starts at zero velocity. The problem is designed such that the ends of the tube do not affect motion of the shock.

The initial conditions were given in dimensional form in the EVTS [26], so the problem was solved dimensionally in order to verify the WAMR method using the method as stated in the EVTS. The dimensional initial conditions for the first Sod Problem are listed in Table E.1 where x is the domain of the problem, x_i is the initial location of the shock, and t_f is the final time of the problem. The boundaries were prescribed to maintain the initial conditions. The boundaries did not interact with the solution. In order to prevent boundary interaction the both boundaries were treated as zero stress boundaries.

For the first set of initial conditions, 10^3 solution snapshots were saved. A snap-

TABLE E.1

Sod problem 1 initial conditions

	ρ [g/cm^3]	u [cm/s]	p [dyne/cm^2]
Left	1.0	0.0	1.0
Right	0.125	0.0	0.1
$0 \leq x \leq 1$ cm; $x_s = 0.5$ cm; $t_f = 0.25$ s			

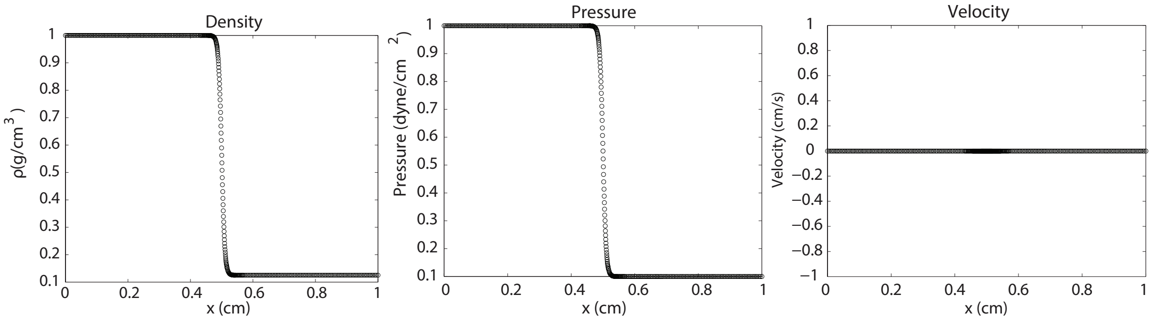

Figure E.3. Sod problem 1 initial conditions

shot was taken every 250 μs. This problem was solved for $\varepsilon = 10^{-1}$, $\varepsilon = 10^{-2}$, $\varepsilon = 10^{-3}$, $\varepsilon = 10^{-4}$, $\varepsilon = 10^{-5}$, and $\varepsilon = 10^{-6}$. The maximum wavelet level used for the smallest ε was 6 so for the reference solution a 102400 point grid was used to ensure that the reference grid was finer than any of the grids from the WAMR method. The resolution scale level of the problem was 10. The size of the uniform grid was $\Delta x = 9.766 \times 10^{-6}$ cm. The initial conditions of the first Sod problem are plotted in Figure E.3. There is a pressure and density drop across the shock, and all of the gas is initially at zero velocity.

The 500th saved solution was recorded at $t = 0.125$ s. This solution accurately

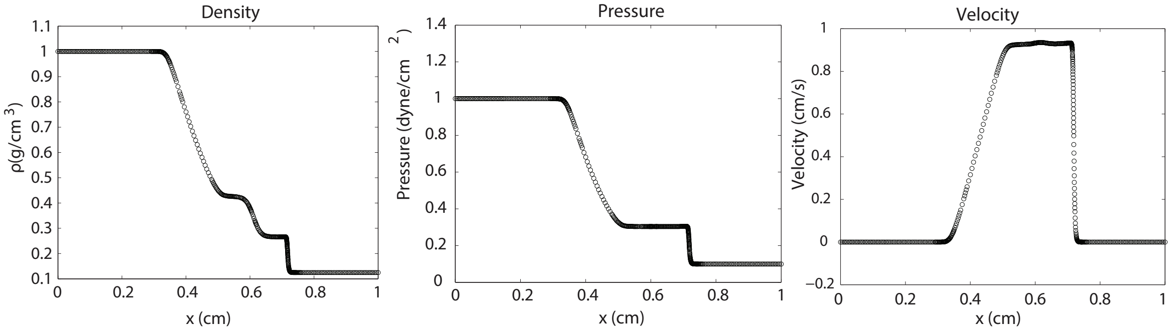

Figure E.4. Sod problem 1 solution at $t = 0.125$ s

portrays the structure of a shock wave in Figure E.4 for $\varepsilon = 10^{-6}$. At this time the shock has moved to $x = 0.715$ cm, and the contact discontinuity is visible around $x = 0.61$ cm. The rarefaction is visible to the left of the contact discontinuity. The initial conditions are preserved at both boundaries, and the boundaries do not affect the solution, which is as expected. There is a slight bump in the velocity graph around the location of the contact discontinuity. This bump would not exist in a physical or analytical solution. It is a result of assuming too large of a viscosity. Adding viscosity to the inviscid problem caused the sharp edges around the shock be become more smooth. However, the large assumed viscosity caused the contact to become smooth instead of a discontinuity as the name implies. The smoothness of the contact discontinuity also caused the bump on the velocity graph.

The final solution to the first Sod problem is shown in Figure E.5. The form of shock in the final solution is similar to the solution in Figure E.4. The shock has continued moving right and is now located around $x = 0.94$ cm. The contact discontinuity is also moving left and is located at $x = 0.74$ cm. It is also clear that the rarefaction is moving left as one would expect. The initial conditions are still preserved at the boundaries, and the boundaries have had no effect on the solution as prescribed in the problem definition.

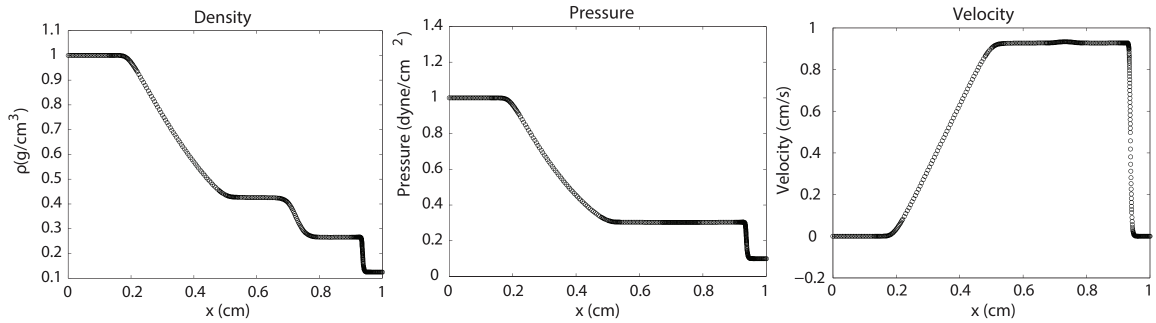

Figure E.5. Sod problem 1 final solution

Now the error must be evaluated for varying values of ε. The maximum error for pressure, density, and temperature at $t = 0.875 \times 10^{-2}$ s was plotted as a function of ε in Figure E.6. Comparing the reference line with a slope of 1 to the error plots show that the maximum error for each variable as a function of ε has about a slope near 1. The graph of the error for the temperature intercepts the point $[10^{-4}, 4 \times 10^{-4}]$. Hence the measured error meets the prescribed error for each ε. Figure E.7 shows that the error decreases as the resolution decreases and that the resolution decreases as ε decreases.

E.4.1.2 Sod Problem 2

The second Sod Problem has a transonic rarefaction wave. The problem has an 8:1 density ratio and a 10:1 pressure ratio between the left and right sides like the first problem. However, this problem has velocity on the left side of the shock and zero velocity on the right side. As with the first problem, the boundary does not affect the wave evolution.

The initial conditions for the second version of the Sod problem were also given dimensionally. The dimensional initial conditions for the second Sod Problem are listed in Table E.2 where x is the domain of the problem, x_s is the initial location

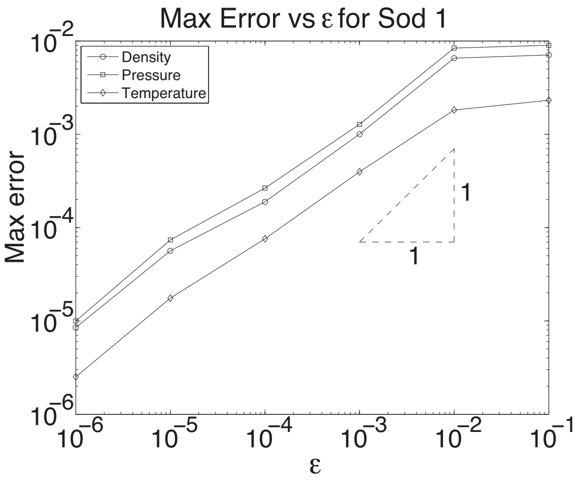

Figure E.6. Sod 1 error vs ε at $t = 0.875 \times 10^{-2}$ s

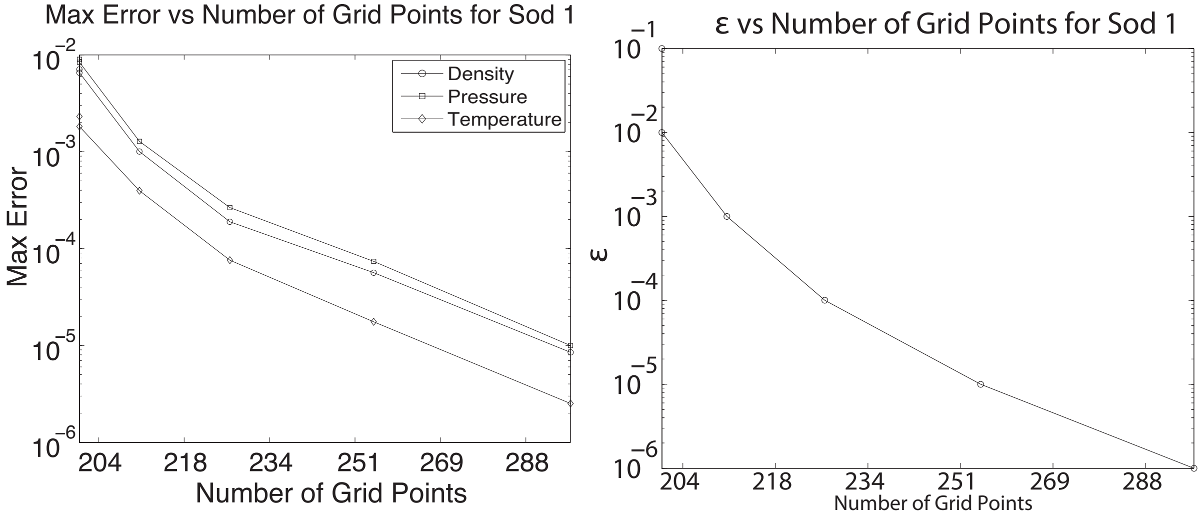

Figure E.7. Sod 1 error vs ε at $t = 0.875 \times 10^{-2}$ s

TABLE E.2

Sod problem 2 initial conditions

	ρ [g/cm^3]	u [cm/s]	p [dyne/cm^2]
Left	1.0	0.75	1.0
Right	0.125	0.0	0.1
$0 \leq x \leq 1$ cm; $x_s = 0.3$ cm; $t_f = 0.2$ s			

of the shock, and t_f is the final time of the problem. The boundaries were set to maintain the initial conditions and not to interfere with the solution. The left boundary was treated as an inflow boundary so that the velocity was always set to the initial velocity and the right boundary was a zero stress boundary.

For the second set of initial conditions, 10^3 solution snapshots were saved. A snapshot was taken every 200 μs. This problem was solved for $\varepsilon = 10^{-1}$, $\varepsilon = 10^{-2}$, $\varepsilon = 10^{-3}$, $\varepsilon = 10^{-4}$, $\varepsilon = 10^{-5}$, and $\varepsilon = 10^{-6}$. The maximum wavelet level for the smallest ε was 7 so for the reference solution a 102400 point grid was used to ensure that the reference grid was finer than any of the grids from the WAMR method. The resolution scale level of the problem was 10. The size of the uniform grid was $\Delta x = 9.766 \times 10^{-6}$ cm. The initial conditions of the second Sod problem are plotted in Figure E.8. There is a pressure, density and velocity drop across the shock.

The 500^{th} saved solution was recorded at $t = 0.1$ s. This solution accurately portrays the structure of a shock wave in Figure E.9 for $\varepsilon = 10^{-6}$. The solution portrays similar dynamics to the first Sod problem. At this time the shock has moved to $x = 0.51$ cm and the contact discontinuity is visible around $x = 0.43$ cm. The rarefaction is visible to the left of the contact discontinuity. The initial conditions are preserved at both boundaries are still preserved and the boundaries do not affect

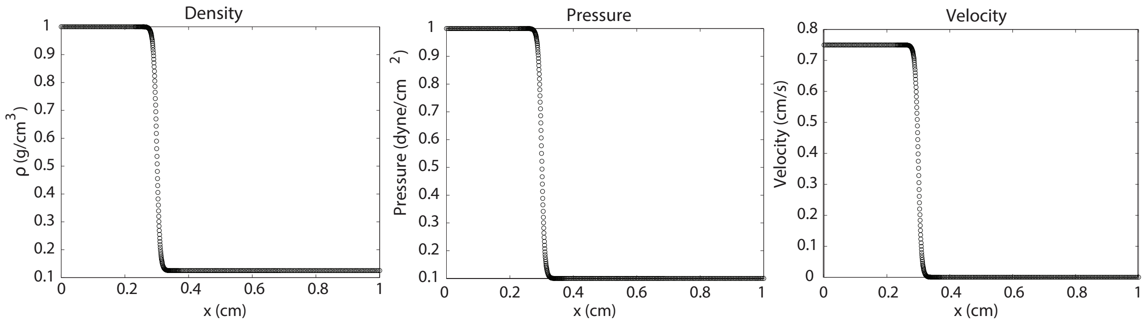

Figure E.8. Sod problem 2 initial conditions

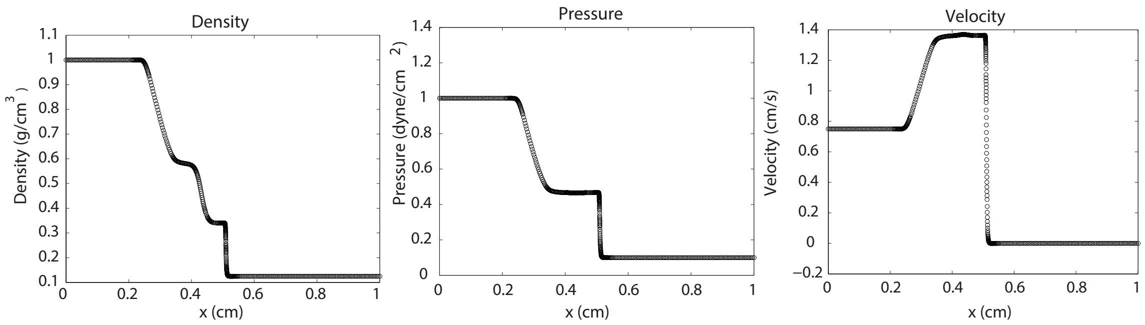

Figure E.9. Sod problem 2 solution at $t = 0.1$ s

the solution, as expected. As with the first Sod problem, there is a slight bump in the velocity graph around the location of the contact discontinuity. This is a result of assuming too much viscosity, like in the first problem.

The final solution to the second Sod problem is shown in Figure E.10. The form of shock in the final solution is similar to the solution in Figure E.9. The shock has continued moving right and is now located around $x = 0.725$ cm. The contact discontinuity is also moving left and is located at $x = 0.56$ cm. It is also clear that the rarefaction has moved left since the solution at the the halfway time. This shows that the rarefaction is able to travel through the gas, even though the gas has a velocity

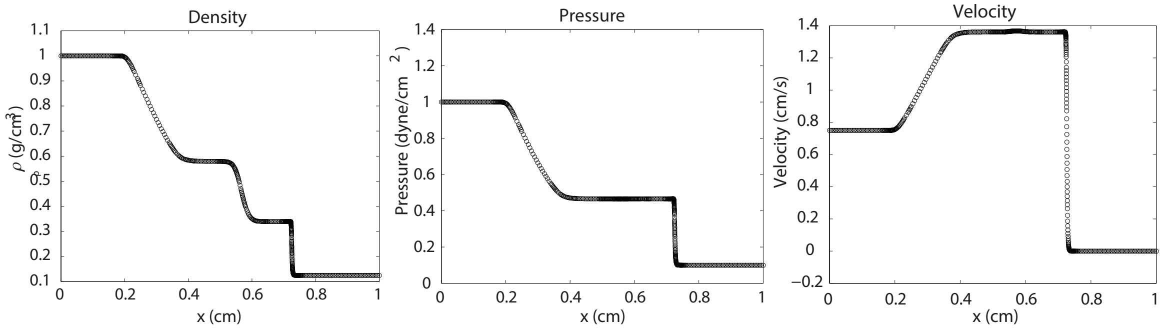

Figure E.10. Sod problem 2 final solution

in the opposite direction. The boundaries did not affect the solution as prescribed by the EVTS.

The error from these solutions is the most important result. The error is plotted as a function of ε at $t = 0.26 \times 10^{-2}$ s in Figure E.11. The error vs ε plot has a slope of about 1 and the temperature plot intercepts the point $[10^{-4}, 2 \times 10^{-5}]$. Hence the maximum error for each variable is below the error threshold. Figure E.12 shows that the error decreases as the resolution decreases and that the resolution decreases as ε decreases.

E.4.2 Shu-Osher Entropy Wave Problem

The Shu-Osher Entropy Wave Problem is a hydrodynamics problem used to test an algorithm's ability to handle complex flow structures in the presence of a shock wave. The problem also tests that a numerical method can handle fine scale flow structures and non-oscillatory solutions in the presence of shocks. The problem models a one-dimensional tube filled with an ideal gas with two sections separated by an infinitely thin diaphragm to simulate a Mach 5 shock. The post-shock conditions have a larger pressure and density and a velocity resulting from an inflow. The pre-shock conditions have zero velocity and a sinusoidally varying density. [29]

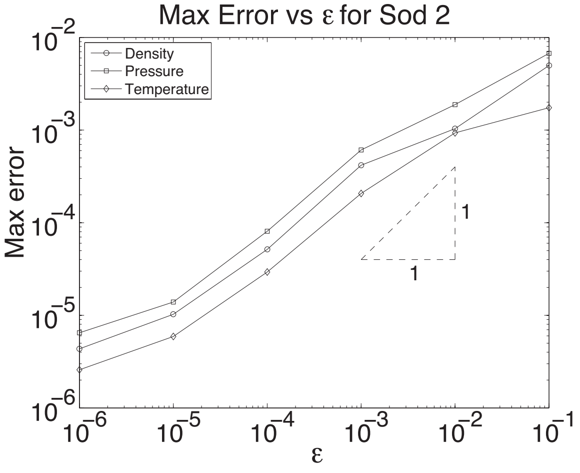

Figure E.11. Sod 2 error vs ε at $t = 0.26 \times 10^{-2}$ s

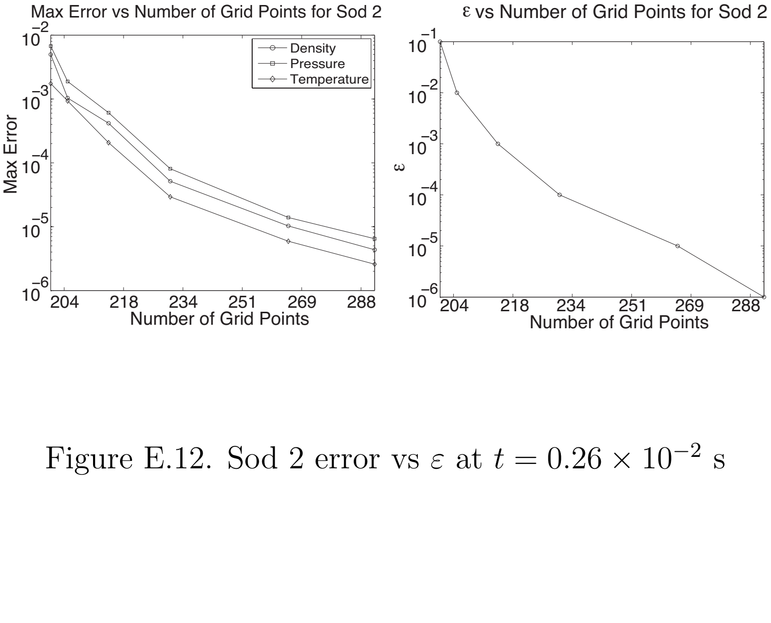

Figure E.12. Sod 2 error vs ε at $t = 0.26 \times 10^{-2}$ s

TABLE E.3

Shu-Osher problem initial conditions

	ρ [g/cm^3]	u [cm/s]	p [dyne/cm^2]
Left	3.857143	2.629369	10.33333
Right	1+ 0.2sin($5x$)	0.0	1.0
$0 \leq x \leq 9$ cm; $x_s = 4.5$ cm; $t_f = 1.8$ s			

Since the Shu-Osher problem is a one-dimensional, single species problem, the simplifications to the governing equations were used to solve the problem. The same equation of state and gas were the same as in the Sod problem. The initial shock was modeled the same form as Eq. (16), but in this case the low pressure is a sinusoidally varying function. For the Shu-Osher problem $\delta = 10^{-2}$ cm was chosen. Similar to the Sod problem, the initial domain was divided into 100 grid points.

The initial conditions were given dimensionally in the EVTS [26] and are contained in Table E.3 where x is the position, x_s is the initial location of the shock, and t_f is the final time of the problem. The initial velocity of the left side of the shock was originally calculated using the Rankine-Hugoniot jump conditions of the nominal pre-shock state. The left boundary condition is an inflow boundary condition that is always at the post-shock initial conditions. No boundary conditions were set on the right side, because the right side boundary is to be fixed at the pre-shock state, but the waves do not reach the right boundary when solving the problem. Hence, the lack of boundary conditions did not interfere with the solution.

For the Shu-Osher problem, 6×10^3 solutions were saved. A snapshot was take every 300 μs. The problem was solved for only $\varepsilon = 10^{-2}$, $\varepsilon = 10^{-3}$, and $\varepsilon = 10^{-4}$ because smaller ε values would be very computationally expensive, and larger ε causes

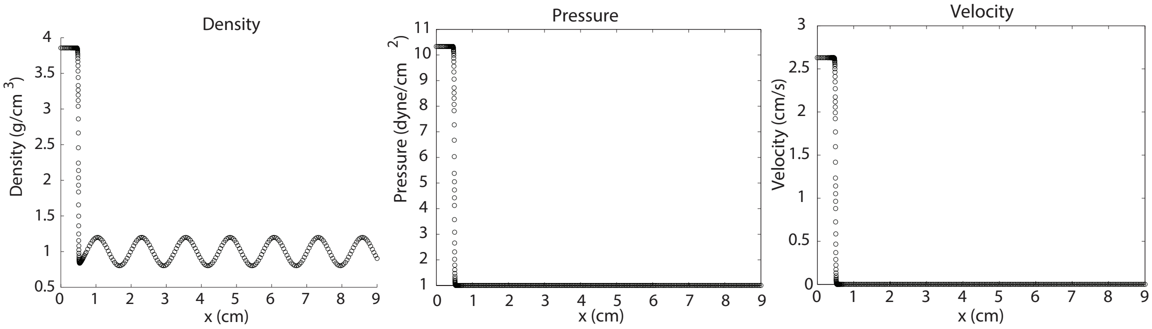

Figure E.13. Shu-Osher problem initial conditions

errors in the code. Larger ε values create a grid with too few points so it does not capture the physics of the problem. Neither the inviscid nor viscous Shu-Osher problems have analytical solutions, so a fine grid reference solution was created. The maximum wavelet level used during the $\varepsilon = 10^{-4}$ solution was 13, so a grid of 6553600 points was used for the uniform grid. The resolution scale level of the problem was 16. The size of the uniform grid was $\Delta x = 1.373 \times 10^{-6}$ cm. The initial conditions contains a jump in pressure, velocity and density at the shock. Also, the density on the right side of the shock varies sinusoidally as shown in Figure E.13.

An intermediate solution at $t = 0.9$ s is shown for $\varepsilon = 10^{-4}$ in Figure E.14. At this time the shock has moved to $x = 3.7$ cm. High amplitude and high velocity density waves follow the shock. However as the sinusoidal density gets further away from the shock, its amplitude and frequency decrease to a value similar to the initial pre-shock density profile. Also there is a contact discontinuity at $x = 1.05$ cm.

The final solution has a similar structure to the intermediate solution. Figure E.15 shows a comparison between the final density profile as given in the EVTS and the final density results using the WAMR method for different ε values. The solution provided in the EVTS is the solution to the inviscid problem, so the solutions should not match perfectly. However the solutions are very similar, so it is clear that the

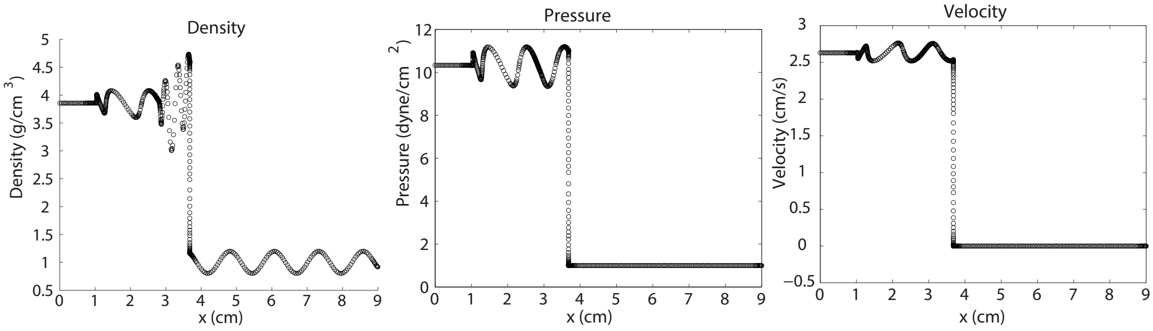

Figure E.14. Shu-Osher problem at $t = 0.9$ s

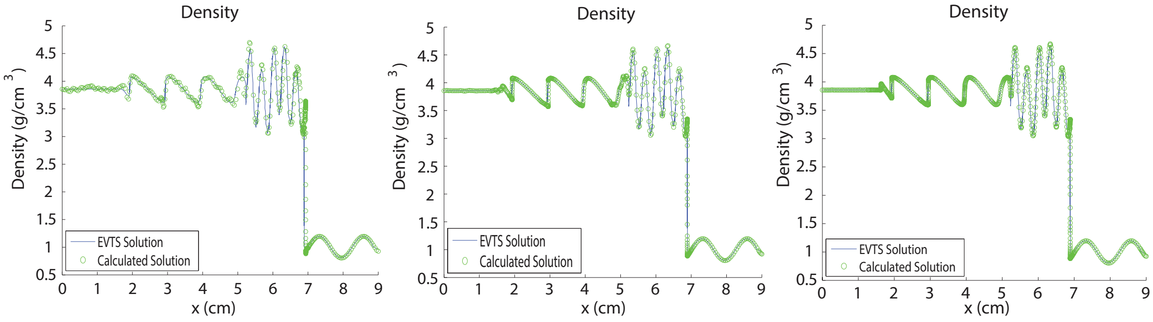

Figure E.15. Shu-Osher solution vs. EVTS result

added viscosity had little effect on the problem.

The error as a function of ε has about a slope of 1 as shown in Figure E.16. Also note that the error from the solution for each variable has error less than the user-prescribed error threshold. Hence, the method is verified. Also, Figure E.17 shows that for the Shu-Osher problem as the ε decreases, the number of points increase. Also, as the number of points increases the error decreases as expected. Hence the solution is much more compact when using the WAMR method.

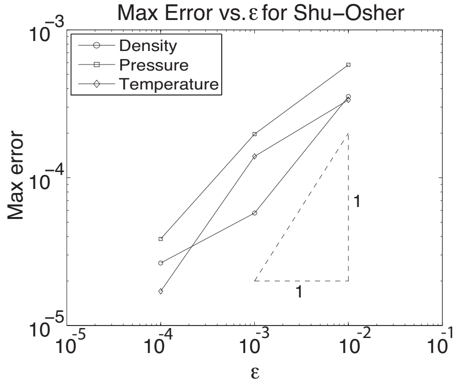

Figure E.16. Shu-Osher error vs. ε at $t = 0.3 \times 10^{-3}$

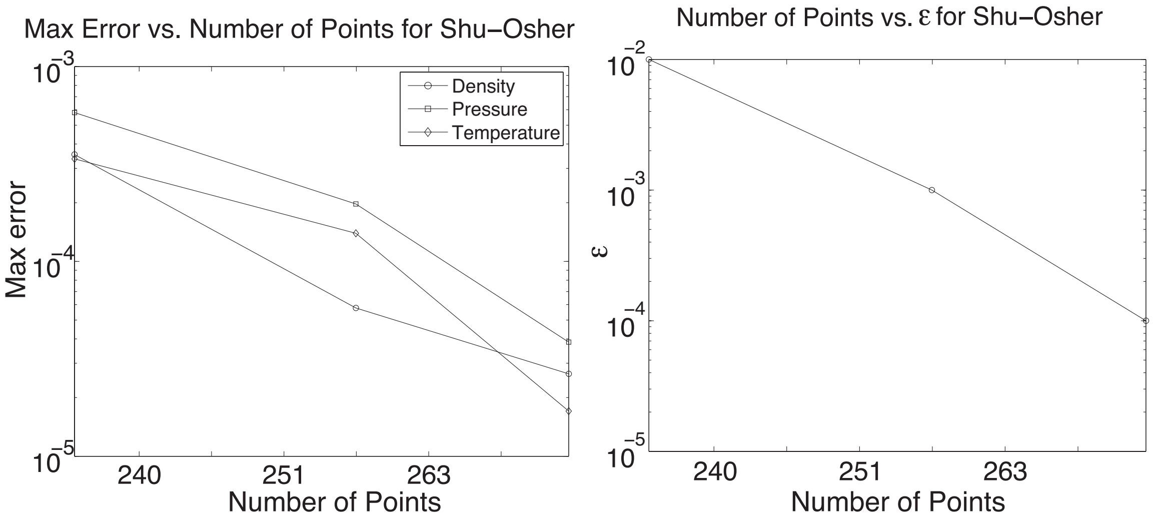

Figure E.17. Shu-Osher grid size $t = 0.3 \times 10^{-3}$

E.4.3 Woodward-Colella Blast Wave Problem

The Woodward-Colella problem is a hydrodynamics problem that tests a method's ability to track contact surfaces, capture shocks, and handle strong shock interactions. It tests an algorithm's ability to numerically handle complex wave interactions and strong shocks. The Woodward-Colella problem models a one-dimensional tube filled with a calorically perfect ideal gas. The gas is separated into three sections with two infinitely thin diaphragms all with zero velocity and constant density. The left section takes up 10% of the tube and has the highest pressure. The middle section contains 80% of the tube and has the lowest pressure. The right section takes up 10% of the tube and has an intermediate pressure. The ends of the tube are sealed. Once the diaphragms are broken, two strong shocks approach the center causing complex wave interactions [30].

Like the first two problems, the Woodward-Colella problem could also be modeled in a shock tube. However, for this case, three regions of gas would need to be created using two diaphragms. The initial conditions in each chamber would be different and then the shock would begin when both diaphragms were removed or punctured with plungers.

Since the Woodward-Colella problem is a one-dimensional, single species problem, the simplifications to the governing equations were used to solve the problem. The same equation of state and gas assumptions as the Sod problem and the Shu-Osher problem were used for this problem. Since the Woodward-Colella problem contains two shocks, Eq. (16) will not apply for this problem. Instead, a different function was developed to contain two shocks. The initial pressure is described by the function

$$p = \frac{p_L}{4}\left[\left(1+\frac{p_M}{p_L}\right)-\left(1-\frac{p_M}{p_L}\right)\tanh\left(\frac{x-x_2}{\delta}\right)\right]\left[\left(\frac{p_H}{p_L}+1\right)-\left(\frac{p_H}{p_L}-1\right)\tanh\left(\frac{x-x_1}{\delta}\right)\right], \tag{E.19}$$

where p_L is the low pressure, p_M is the medium pressure, p_H is the high pressure, x

TABLE E.4

Woodward-Colella problem initial conditions

	ρ [g/cm^3]	u [cm/s]	p [dyne/cm^2]
Left	1.0	0.0	10^3
Middle	1.0	0.0	10^{-2}
Right	1.0	0.0	10^2
$0 \leq x \leq 1$ cm; $x_1 = 0.1$ cm; $x_2 = 0.9$ cm; $t_f = 0.038$ s			

is the position, x_1 is the position of the first shock, x_2 is the position of the second shock, and δ is the thickness of the shock. The thickness of both shocks was set at $\delta = 10^{-3}$ cm in order to be sufficiently small relative to the domain. Also, the initial grid was divided into 100 grid points, like the first two problems.

The initial conditions were given dimensionally in the EVTS [26] and are contained in Table E.4 where x is the position, x_1 is the initial location of the first diaphragm, x_2 is the initial location of the second diaphragm, and t_f is the final time of the problem. The left and right boundaries of the tube are both walls, so the boundary conditions of the problem set the velocity at both walls to be zero. No other boundary conditions were used to solve this problem.

For the Woodward-Colella Blast Wave problem, 100 solutions were saved. A snapshot was saved every 380 μs. As with the Shu-Osher problem, this problem was also solved for $\varepsilon = 10^{-2}$, $\varepsilon = 10^{-3}$, and $\varepsilon = 10^{-4}$, because larger ε values would cause the problem to become much more computationally expensive. Neither the inviscid nor viscous Woodward-Colella problems have analytical solutions, so a fine grid reference solution was created. The maximum wavelet level used in the smallest ε solution was found to be 13 so a uniform grid of 6553600 points was used in order

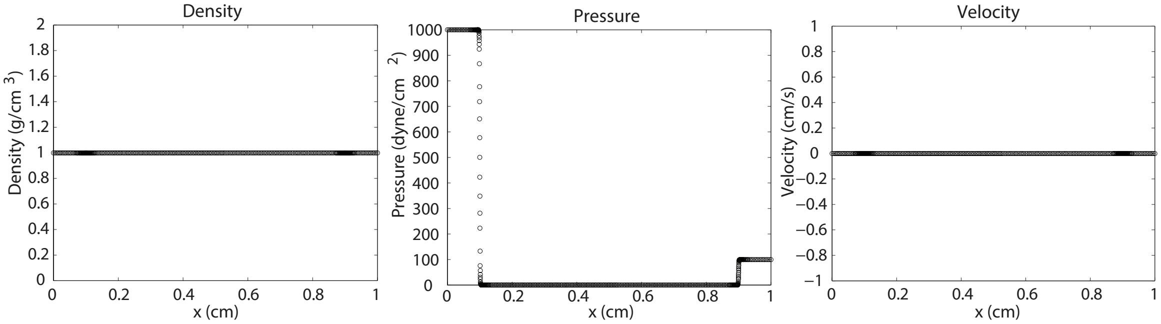

Figure E.18. Woodward-Colella initial conditions

to make a uniform grid sufficiently finer than the solutions using the WAMR method. The resolution scale level of the problem was 10. The size of the uniform grid was $\Delta x = 1.526 \times 10^{-7}$ cm. This uniform grid was used to calculate the reference solution. The initial conditions plotted in Figure E.18 show the two initial jumps in pressure.

The two shocks approach each other as seen in the plots at $t = 0.019$ s in Figure E.19. Then the shocks hit each other causing complex interactions which are accurately modeled as shown in Figure E.20 at the final time of $t = 0.038$ s. The EVTS included an image of density the final solution to the inviscid Woodward-Colella problem. When that solution is superimposed on the final solution using the WAMR method with an error threshold of $\varepsilon = 10^{-4}$, one can see that the result is very similar. Figure E.21 shows that the WAMR solution is very similar to the inviscid solution, but they are slightly different around the corners and at the discontinuities. These differences were expected when viscosity was added to the problem.

The error for the Woodward-Colella problem will be reported at a later date, because the computational intricacies of the problem cause it to be computationally more expensive. It is currently in process and error will be reported as soon as a solution is saved. However, one should expect that the error will be less than the user-prescribed error criteria.

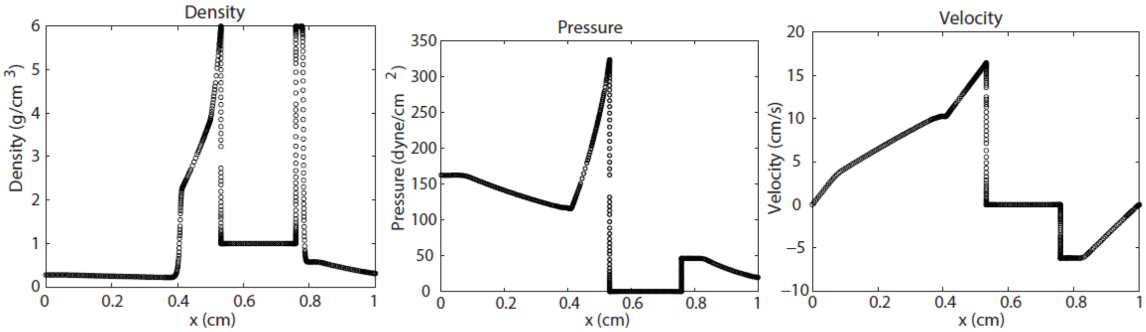

Figure E.19. Woodward-Colella at $t = 0.019$ s

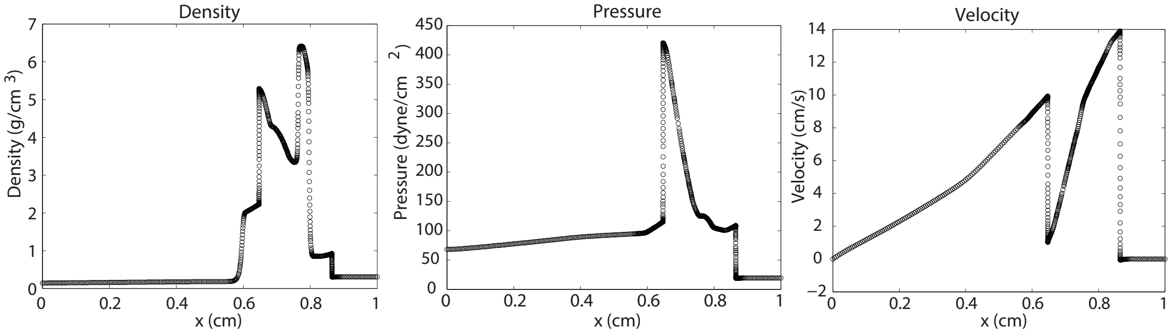

Figure E.20. Woodward-Colella at $t = 0.038$ s

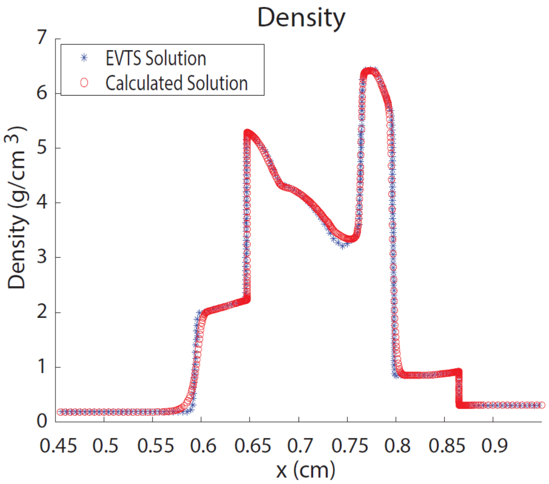

Figure E.21. Woodward-Colella at $t = 0.038$ s compared to EVTS solution

E.4.4 Richtmyer-Meshkov Instability Problem

The Richtmyer-Meshkov Instability (RMI) problem is a hydrodynamics problem that is used to test how codes handle multiple fluid interactions in the presence of shocks. It also tests how a method captures large and fine scale dynamics of the mixing layer. The problem was modeled in two dimensions. In general, the problem discusses the stability of an accelerated interface separating two incompressible fluids. Initially a sinusoidal shaped gas curtain of SF_6 is at rest in a shock tube of air. Then a shock of high pressure air moves through gas curtain, which causes instability. Next, the shock bounces off the end of the tube and hits the initial instability causing complex instabilities to occur. The vortex dynamics of the RMI are detailed in other sources [33, 34, 35].

The Richtmyer-Meshkov problem can be tested in a shock tube, like the first four problems. The RMI is relevant many different situations in engineering; it is important in the mixing in ramjet engines [36, 37]; it helps model the mixing in supernovæ[38, 39]; it plays a key role in describing the deflagration-to-detonation transition [40].

The RMI problem is a two dimensional, multi-fluid problem so the general form of the governing equations were used to solve the problem. The quiescent fluids chosen for the problem were air for the main fluid and SF_6 for the gas curtain. For this problem, air mixture was defined as 23.2% mass O_2 and 76.8% mass N_2. The problem was designed to have a single mode sinusoidal perturbation at the interface between the gases. The gas curtain was described by

$$Y_{SF_6} = 10^{-12} + \frac{A\left(1 + B\cos\left(k\left(y - 0.18\text{cm}\right)\right)\right)}{1 + B} e^{\frac{-2\alpha(x-x_o)^2}{(1+\beta\cos(k(y-0.18\text{cm})))^2}}, \tag{E.20}$$

where Y_{SF_6} is the mass fraction of SF_6, y is the position in the y direction, x is the position in the x direction, x_0 is the location of the gas curtain in the x direction, and A, B, k, α, and β are constant parameters defined in Table E.5. The shock occurred in the air region and was defined in the same form at Eq (16) for pressure, temperature, and velocity. The initial conditions are described in Table E.6 where a_L is the acoustic speed to the left of the shock, a_R is the acoustic speed to the right of the shock, t_s is the location of the shock in the x direction, and t_f is the final time. The shock thickness was defined to be $\delta = 10^{-4}$ cm so that δ is sufficiently small relative to the domain. The density is much higher in the gas curtain than in the surrounding error because the curtain is made of SF_6. All boundaries were set as zero stress boundaries. The left end of the tube allowed an inflow of air at the post shock conditions. On the left and right walls, the velocity is set to 0 in the x direction and on the top and bottom walls the velocity in the y directional velocity is zero.

For the Richtmyer-Meshkov instability problem, 250 solutions were saved. Each solution was saved every 2.5 μs. The problem was solved for $\varepsilon = 10^{-1}$, $\varepsilon = 10^{-2}$,

TABLE E.5

RMI problem gas curtain coefficients

A	B	k (rad/cm)	α	β
0.97	0.2	$2\pi/0.36$	8.36	-0.04
$0 \leq x \leq 15$ cm; $0 \leq y \leq 1.1$; $x_0 = 4.3$ cm				

TABLE E.6

Richtmyer-Meshkov problem initial conditions

	p [dyne/cm^2]	T [K]	u [cm/s]
Left	1.202×10^6	338.4	$1.2a_L - 0.8422a_R$
Right	7.95×10^5	300.0	0.0
$0 \leq x \leq 15$ cm; $0 \leq y \leq 1.1$; $x_s = 3.0$ cm; $t_f = 6.25 \times 10^{-4}$ s			

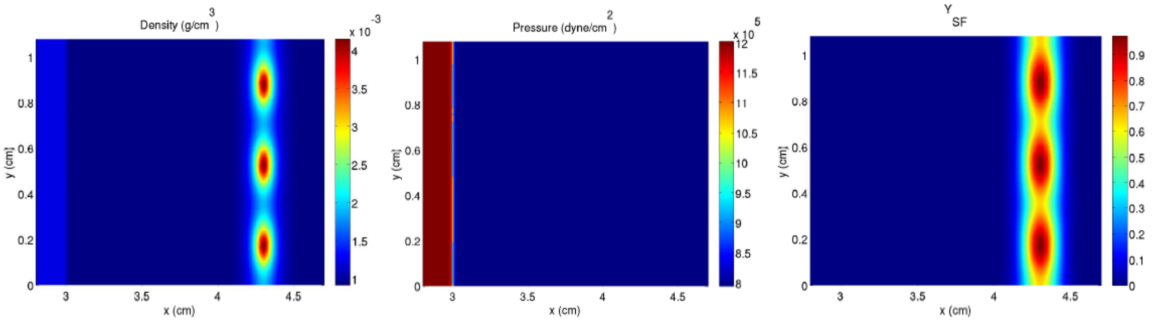

Figure E.22. Richmyer-Meshkov initial conditions

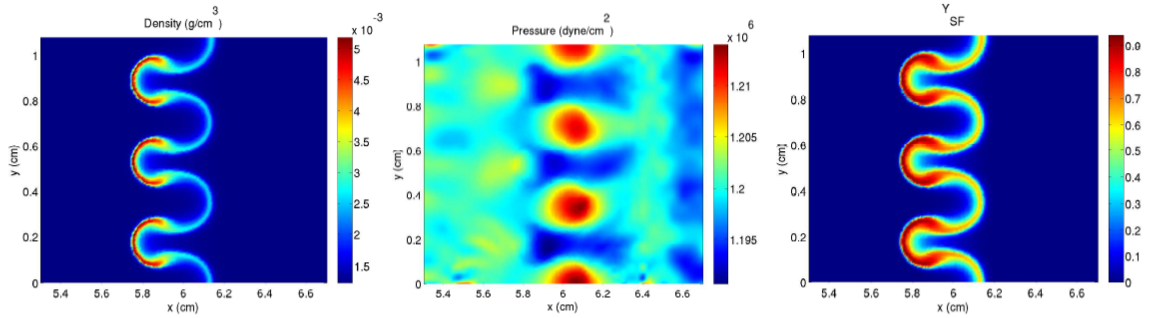

Figure E.23. Richmyer-Meshkov at $t = 1.925 \times 10^{-4}$ s

and $\varepsilon = 10^{-3}$. Due to the large computation time associated with this 2-dimensional problem there was no solution on a uniform grid. Instead the $\varepsilon = 10^{-3}$ solution was used as the reference solution for error evaluation. The initial gas curtain of SF_6 can be seen in Figure E.22. Another solution at $t = 1.925 \times 10^{-4}$ s was recorded showing how the shock initially disturbs the gas curtain in Figure E.23. Then the shock continues to hit the wall as shown in Figure E.24.

The error could not be effectively computed because the saved solutions were created with a dynamic time-step. The dynamic time-step caused solutions to be saved at different times, so error could not be measured. Comparing two solutions

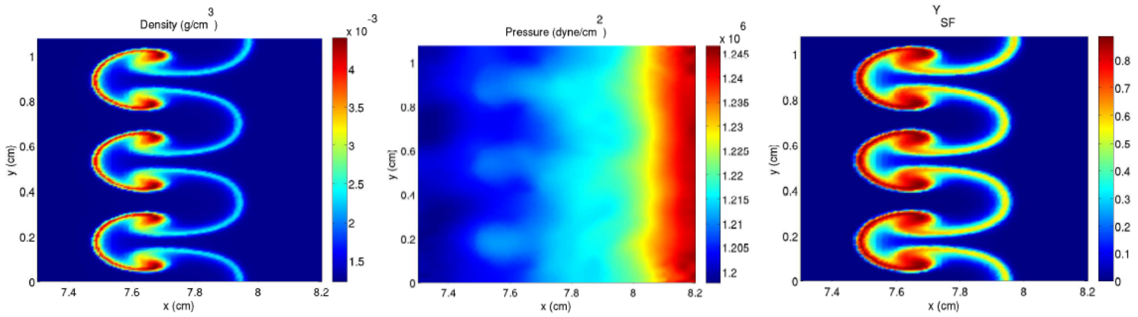

Figure E.24. Richmyer-Meshkov at $t = 3.6 \times 10^{-4}$ s

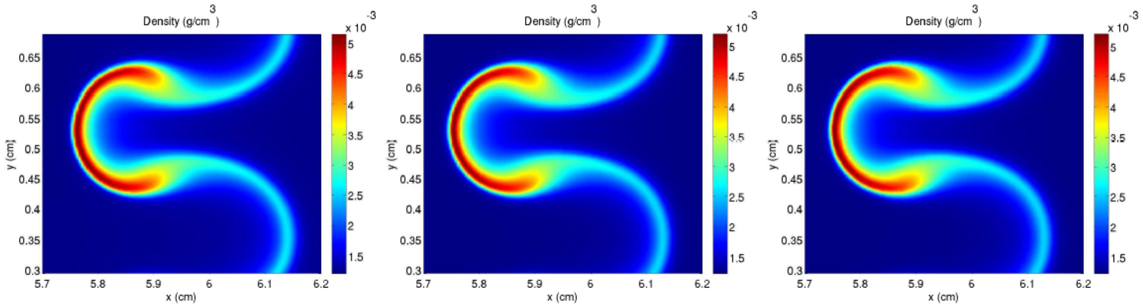

Figure E.25. Richmyer-Meshkov detailed comparison

at even slightly different time steps would make the solution appear to have a large amount of error, which may not be the case. However, the accuracy of the solutions can be seen in the detail of the solution. High magnification solutions of the RMI problem at $t = 1.925 \times 10^{-4}$ are shown in Figure E.25 for $\varepsilon = 10^{-1}$, $\varepsilon = 10^{-2}$, and $\varepsilon = 10^{-3}$. As the the ε value gets smaller, the area of the high density becomes more defined and the general shape of the instability becomes more round. Hence the solutions with lower ε values are more accurate.

E.5 Discussion and Conclusions

The error from both Sod problems show that the error initially meets the user-prescribed error criteria, but over time the error gets worse. These results imply there is error introduced during the time discretization. The error introduced in time is a result of using too large of a time step. The time-step used for the second Sod problem was $\Delta t = 0.1 \times 10^{-7}$ s. In order to ensure the accuracy of a numerical solution with diffusion one must insist that $\Delta t \leq (\Delta x)^2 / (\mu/\rho)$. Rerunning the problem with a lower time-step should eliminate most of the error introduced in time.

The slope of the error vs. ε plot should be 1 because for each ε the error should be of the same order of magnitude. For example, Figure E.16 has about a slope of 1. However, Figure E.16 also shows that the error is over an order of magnitude below the user-prescribed error. The WAMR method produced results with less error than prescribed because the initial grid fine enough that it eliminates some of the error allowed by the WAMR method. They show that while the WAMR method is guaranteed to meet the user-prescribed error, it can produce results that have less error than prescribed.

The error plots for each solution show that the WAMR method is verified. In each case it provides an error value that is less than the user-prescribed error. This is not just true for one ε but for a range of values. Hence the solution for any value of ϵ chosen by the user. Verification that the WAMR method shows that the method provids an automatically verified solution.

The problems were taken from the EVTS so that the WAMR method could be verified in the same manner as other computational codes, but diffusion was added to the problems in order to solve them with the WAMR method. The viscosity and the thickness of hyperbolic tangent functions was set low in order to reduce the impact of the modifications on the final solution. Figure E.15 shows that modifications have little effect on solution. Figure E.21 shows that the viscosity makes corners appear more round in the solution, but the overall shape of the solution is preserved. The approximations for viscosity and thermal conductivity were effective to maintain the shape of the solution, but they could be improved. Figure E.5 shows that the contact discontinuity is too gentle and it is not a discontinuity. The viscosity also influences the velocity plot by causing a small bump to occur at the same location as the contact discontinuity. These issues can be mended by assuming a coefficient of viscosity. Since the problem is at a very low temperature and pressure assuming a very small viscosity will not negatively affect the solution. Instead it will make the jumps more closely resemble the discontinuities in the inviscid problem.

The verification of the method implies that the WAMR method can be used to solve new problems to a set error criteria. Most other computational methods do not allow a user to set the error in advance, so the WAMR method has a unique advantage over other methods. The compression of the function with the wavelets also provides advantages over other methods because it helps to solve the problem quickly. For example, the first Sod problem took about a day to run at $\varepsilon = 10^{-6}$ while the uniform grid without the wavelet interpolation was running for two weeks and still did not reach the final time. The compression created by the wavelets makes the WAMR method much less computationally expensive than other methods, like the uniform grid.

In conclusion, the WAMR method provides automatically verified results. Five problems from the EVTS were used to test the verification. The error from each solution met the user-prescribed error criteria. The WAMR method is unique because it produces an automatically verified solution and because it is computationally less expensive than other methods. The WAMR method produces a more compact grid for each problem which save computational time.

BIBLIOGRAPHY

1. R. C. Smith. *Uncertainty Quantification Theory, Implementation, and Applications.* Society for Industrial and Applied Mathematics, Philadelphia, 2014.

2. G. Iaccarino. Introduction to uncertainty quantification. *Mechanical Engineering and Institute for Computational Mathematical Engineering, Stanford University (course)*, 2011. See also URL `http://web.stanford.edu/group/uq/events/UQLS_stanford/Iaccarino.pdf`.

3. W. L. Oberkampf and T. G. Trucano. Verification and validation in computational fluid dynamics. *Progress in Aerospace Sciences*, 38.3(3): 209–272, 2002.

4. H. N. Najm. Uncertainty quantification and polynomial chaos techniques in computational fluid dynamics. *Annual Review of Fluid Mechanics*, 41: 35–52, 2009.

5. P. Doubilet, C. B. Begg, M. C. Weinstein, P. Braun, and B. J. McNeil. Probabilistic sensitivity analysis using Monte Carlo simulation. A practical approach. *Medical Decision Making: An International Journal of the Society for Medical Decision Making*, 5(2): 157–177, 1984.

6. D. B. Xiu and G. E. Karniadakis. Modeling uncertainty in flow simulations via generalized polynomial chaos. *Journal of Computational Physics*, 187(1): 137–167, 2003.

7. I. Babuska, F. Nobile, and R. Tempone. A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM Journal of Numerical Analysis*, 25(3): 1005–1034, 2007.

8. J. Bernoulli. *Ars Conjectandi.* Thurnisiorum, Basel, 1713.

9. T. Bayes. An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions The Royal Society of London*, 53: 370–418, 1763.

10. P. S. Laplace. *Théorie Analytique des Probabilités.* Courcier Imprimeur, Paris, 1812.

11. A. Gelman. Objections to Bayesian statistics. *Bayesian Analysis*, 3(3): 445–450, 2008.

12. D. S. Sivia. *Data Analysis A Bayesian Tutorial,* 2nd edition. Oxford University Press, New York, 2006.

13. H. Jeffereys. *Theory of Probability.* Clarendon Press, Oxford, 1939.

14. E. T. Jaynes. Clearing up mysteries-the original goal. In *Maximum Entropy and Bayesian Methods,* J. Skilling, editor. Kluwer, Dordrecht, 1989.

15. E. T. Jaynes. *Probability Theory: The Logic of Science.* Cambridge University Press, New York, 2003.

16. M. Allmaras, Bangerth W., J. M. Linhart, J. Polanco, F. Wang, K. Wang, and S. Zedler. Estimating parameters in physical models through Bayesian inversion: a complete example. *SIAM Review*, 55(1): 149–167, 2013.

17. R. G. Ghanem and P. D. Spanos. *Stochastic Finite Elements: A Spectral Approach.* Dover Publications, Mineola, NY, 1991.

18. T. Bui-Thanh, C. Burstedde, O. Ghattas, J. Martin, G. Stadler, and L. C. Wilcox. Extreme-scale UQ for Bayesian inverse problems governed by PDEs. *Proceedings of the 2012 International Conference for High Performance Computing, Networking, Storage and Analysis,* Salt Lake City, 2012.

19. S. R. Brill. Stochastic modeling of heat transfer through heterogeneous materials. *University of Notre Dame AME 48491, Undergraduate Research*, 2015.

20. Numerical solution of partial differential equations. `http://reference.wolfram.com/language/tutorial/NDSolvePDE.html`. Accessed: 2015-05-25.

21. C.W. Gardiner. *Handbook of Stochastic Methods for Physics, Chemistry and the Natural Sciences,* 2nd edition. Springer-Verlag, Berlin, 1985.

22. J.M. Powers and M. Sen. *Mathematical Methods in Engineering.* Cambridge University Press, New York, 2015.

23. S. Hosder, R.W. Walters, and R. Perez. A non-intrusive polynomial chaos method for uncertainty propagation in CFD simulations. 44th AIAA Aerospace Sciences Meeting and Exhibit, AIAA-2006-0891, January 2006. Reno, Nevada.

24. S. Paolucci. Numerical Methods. University of Notre Dame AME 60614, Lecture Notes, 2011.

25. P. J. Roache. *Fundamentals of Verification and Validation.* Hermosa, Sorroco, NM, 2009.

26. J. R. Kamm, J. S. Brock, S. T. Brandon, D. L. Cotrell, B. Johnson, P. Knupp, W. J. Rider, T. G. Trucano, and V. G. Weirs. Enhanced verification test suite for physics simulation codes. *Los Alamos National Laboratory*, (SAND 2008-7813), 2008.

27. T. J. Barth, T. Chan, and R. Haimes. *Multiscale and Multiresolution Methods: Theory and Applications*. Springer, Berlin, 2002.

28. G. A. Sod. A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *Journal of Computational Physics*, 27(1): 1–31, 1978.

29. C. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of Computational Physics*, 77(2): 439–471, 1988.

30. P. Woodward and P. Colella. The numerical simulation of two-dimensional fluid flow with strong shocks. *Journal of Computational Physics*, 54(1): 115–173, 1984.

31. R.D. Richtmyer. Taylor instability in shock acceleration of compressible fluids. *Communications on Pure and Applied Mathematics*, 13(2): 297–319, 1960.

32. S. Paolucci, Z. J. Zikoski, and D. Wirasaet. WAMR: An adaptive wavelet method for the simulation of compressible reacting flow. part I. accuracy and efficiency of algorithm. *Journal of Computational Physics*, 272: 814–841, 2014.

33. D. L. Cotrell and A. W. Cook. Scaling the incompressible Richtmyer-Meshkov instability. *Physics of Fluids*, 19(7): 078105, 2007.

34. D. Lee, G. Peng, and N. J. Zabusky. Circulation rate of change: A vortex approach for understanding accelerated inhomogeneous flows through intermediate times. *Physics of Fluids*, 18(9): 097102, 2006.

35. N. J. Zabusky. Vortex paradigm for accelerated inhomogeneous flows: Visiometrics for the Rayleigh-Taylor and Richtmyer-Meshkov environments. *Annual Review of Fluid Mechanics*, 31: 495–536, 1999.

36. E. T. Curran, W. H. Heiser, and D. T. Pratt. Fluid phenomena in scramjet combustion systems. *Annual Review of Fluid Mechanics*, 28: 323–360, 1996.

37. J. Yang, T. Kubota, and E. E. Zukoski. Applications of shock-induced mixing to supersonic combustion. *AIAA Journal*, 31(5): 854–862, 1993.

38. W. D. Arnett, J. N. Bahcall, R. P. Kirshner, and S. E. Woosley. Supernova 1987a. *Annual Review of Astronomy and Astrophysics*, 27: 629–700, 1989.

39. A. Burrows, J. Hayes, and B. Fryxell. On the nature of core collapse supernova explosions. *Astrophysics Journal*, 450(2): 830–850, 1995.

40. A. M. Khokhlov, E. S. Oran, and G. O. Thomas. Numerical simulation of deflagration-to-detonation transition: the role of shock–flame interactions in turbulent flames. *Combustion and Flame*, 117(1): 323–339, 1999.

This document was prepared & typeset with pdfLaTeX*, and formatted with* ND$\text{diss}2_{\varepsilon}$ *classfile (v3.2013[2013/04/16]) provided by Sameer Vijay and updated by Megan Patnott.*