

In presenting this thesis in partial fulfillment of the requirements for an advanced degree at Idaho State University, I agree that the Library shall make it freely available for inspection. I further state that permission for extensive copying of my thesis for scholarly purposes may be granted by the Dean of the Graduate School, Dean of my academic division, or by the University Librarian. It is understood that any copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Signature _____

Date _____

Assessing Nuclear Power Plant Component Fragility in Flooding Events Using
Bayesian Regression Modeling with Explanatory Variables

by

Alison Wells

A dissertation

submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy in Nuclear Science and Engineering

Idaho State University

Spring 2020

Copyright 2020 Alison Wells

To the Graduate Faculty:

The members of the committee appointed to examine the dissertation of Alison Wells find it satisfactory and recommend that it be accepted.

Chad L. Pope, Ph.D., P.E.,

Major Advisor

Mary Lou Dunzik-Gougar, Ph.D.,

Committee Member

Bruce Savage, Ph.D., P.E.,

Committee Member

Curtis L. Smith, Ph.D.,

Committee Member

S. Hossein Mousavinezhad, Ph.D.,

Graduate Faculty Representative

Dedication

To my family.

Thank you for believing in and supporting me throughout this endeavor.

Acknowledgments

I would like to express my deepest thanks to Dr. Pope for being my advisor and guiding me through this process. I appreciate the mentoring and have gained invaluable experience. His timely support, guidance, and encouragement has made this dissertation possible.

Next I would like to individually thank each of my committee members: Dr. Mary Lou Dunzik-Gougar for her support during my time in the nuclear engineering program, Dr. Bruce Savage for sharing his time and support on the flooding project, Dr. Curtis Smith for making this research opportunity possible and providing guidance, and Dr. S. Hossein Mousavinezhad for his support as my graduate faculty representative.

I want to thank Idaho National Laboratory (INL) and the Light Water Reactor Sustainability (LWRS) program for funding this research. It has been a pleasure failing components and working on flooding risk analysis. A thank you also to the Nuclear Regulatory Commission (NRC) for also providing funding through a fellowship grant.

To conclude, a sincere thank you to the CFEL team members for their hard work and help. Thank you Sneha Suresh for running numerous models with me and Emerald Ryan for sharing her knowledge of Neutrino.

Table of Contents

List of Figures	x
List of Tables	xiii
List of Acronyms	xviii
Abstract	xix
Chapter 1: Introduction.....	1
Chapter 2: Bayesian Data Analysis	5
2.1 Overview of Bayesian Inference Process	6
2.1.1 Bayes' Theorem	7
2.1.2 Prior Distributions.....	8
2.2 Introduction to BUGS.....	9
2.2.1 Markov Chain Monte Carlo Method	12
2.2.2 Gibbs Sampling	13
Chapter 3: Model Checking and Comparison.....	16
3.1 Model Checking Procedures.....	16
3.1.1 Posterior Predictive Checks and Bayesian P-Value.....	17
3.1.2 Residuals.....	19
3.2 Model Comparison using Deviance	21
3.2.1 Deviance Information Criterion	22
Chapter 4: Preliminary Example Analysis.....	24
4.1 Model for NASA O-ring Distress	24

Chapter 5:	Component Flooding Performance Experiments	30
5.1	Portal Evaluation Tank	30
5.2	Measurements and Procedure.....	34
5.3	Experiments and Data	35
5.3.1	Hollow Core Door Experiments.....	36
5.3.2	Steel Door Experiments.....	42
5.4	Future Experiments	51
Chapter 6:	Developing a Model to Assess Component Fragility.....	54
6.1	Generalized Linear Regression Model	54
6.1.1	Background Analysis with Binomial Distribution.....	54
6.1.2	Alternative Continuous Distribution Models	59
6.1.3	Returning to the Binomial Distribution	71
6.2	Verification Case.....	82
6.3	Model Results.....	87
6.3.1	Hollow Core Doors.....	89
6.3.2	Steel Doors.....	94
Chapter 7:	Linking a Component Fragility Model to Simulation	100
7.1	Smooth Particle Hydrodynamics	100
7.2	Simulation Build	104
7.3	Sampling the Fragility Curve.....	105
7.4	Simulation Results	108
Chapter 8:	Conclusions	112
Appendix Chapter A:	Nuclear Power Plant Flooding Events	115

Appendix Chapter B:	Basic User Guide to OpenBUGS	126
Appendix Chapter C:	OpenBugs Scripts and Results	131
Appendix Chapter D:	OpenBUGS Convergence BGR.....	179
Bibliography.....		193

List of Figures

1.1	Tsunami waves approaching the Fukushima-Daiichi Power Plant.	2
1.2	Fort Calhoun NPP surrounded by Missouri River flood water.	3
2.1	Illustration of seismic fragility for an example component.	6
2.2	(a) First iterations of the Gibb sampler for a two-parameter problem. The contours show the target posterior distribution and the full conditional distributions are above and to the right. (b) Gibbs sampler after five iterations [20]	15
3.1	Graphical representation of predictive model checking.	18
3.2	Depiction of the Bayesian p-value predictability [19].	20
4.1	Solid rocket cross section showing location of O-rings [28].	25
4.2	Fragility model curves for O-ring example.	29
5.1	The Portal Evaluation Tank.	31
5.2	Initial PET piping configuration.	32
5.3	Modified PET piping configuration.	33
5.4	Perforated center column inside modified PET configuration.	34
5.5	Modified PET piping and instrumentation diagram.	34
5.6	Flowchart for testing procedures.	36
5.7	Hollow core door set within constructed wall for installation.	37
5.8	Water depth vs. time plots for hollow core door tests.	39
5.9	Hollow core door at moment of failure in Test 3H (left) and Test 4H (right).	40

5.10	Water depth vs. time plot for inward opening hollow core door experiments, Test 5H and 6H.	41
5.11	Steel door and wall installed in the PET.	43
5.12	Water depth vs. time plot for outward opening steel door, Test 1S, 2S, and 3S.	44
5.13	Steel door at the moment of failure in Test 1S.	44
5.14	Water depth vs. time plot for outward opening steel door, Test 4S and Test 5S.	45
5.15	Water depth vs. time plot for outward opening steel door, Test 6S, 7S, and 8S.	45
5.16	Water depth vs. time plot for outward opening steel door, Test 9S, 10S, and 11S.	47
5.17	Deadbolt failure from Test 10S.	47
5.18	Test 11S steel door bending during testing (left) and post testing (right). The original position is indicated by a solid line and bent position by a dotted line.	48
5.19	Water depth vs. time plot for outward opening steel door, Test 12S and Test 13S.	49
5.20	Inward swinging Test 14S at moment of failure (left) and post testing (left).	50
5.21	Water depth vs. time plot for inward opening steel door, Test 14S. . .	51
6.1	A visual roadmap to the model development process.	55
6.2	A box plot of the residuals for the temperature regression model using steel door data.	79

6.3	A box plot of the residuals for the temperature regression model for primary O-ring distress.	88
6.4	Fragility curve showing probability of failure versus water depth plotted in OpenBUGS. Blue curves represent the 95% credible intervals.	98
6.5	Box plot of the depth coefficient estimates using normal, flat, and uniform priors.	99
7.1	Illustration of SPH approximation for the particle of interest using smoothing kernel and smoothing length [38].	102
7.2	Simplified PET setup for simulation.	104
7.3	Fragility model and Neutrino coupling flow chart.	106
7.4	Fragility model curve for steel doors plotted in Python.	107
7.5	Water depth vs. time plots for hollow core door tests.	110
7.6	Fluid height versus time frame for SPH simulation.	111

List of Tables

3.1	Discrepancy functions used for Bayesian model checking.	19
4.1	O-ring thermal stress data prior to launch of Challenger in January 1986 [19].	26
4.2	OpenBUGS script for Bayesian regression model for O-ring fragility.	27
4.3	Summary posterior estimates of logistic regression parameters for primary O-ring distress.	28
5.1	Summary of results for hollow core door experiments.	41
5.2	Summary of corrected depth results for hollow core door experiments.	42
5.3	Summary of results for steel doors experiments, non-deadbolt.	52
5.4	Summary of results for steel doors experiments, deadbolt.	52
5.5	Summary of corrected depth results for steel door experiments.	53
6.1	OpenBUGS Script for binomial model using depth variable.	57
6.2	Results for Binomial model with logit function using the depth variable.	57
6.3	Summary of coefficient estimates and DIC for Binomial model using cloglog function for hollow core doors.	59
6.4	Data on inward swinging failures for hollow core doors.	60
6.5	OpenBUGS script for exponential aleatory model for door failure with Jeffreys prior.	61
6.6	Results for standard exponential script with Jeffreys prior.	61
6.7	Result of standard exponential script with diffuse normal prior.	62

6.8	OpenBUGS script for exponential aleatory model with Bayesian p-value calculation.	64
6.9	Summary of coefficient estimates, p-value, and DIC for exponential model for hollow core doors.	65
6.10	OpenBUGS script for lognormal model.	66
6.11	Summary of coefficient estimates, p-value, and DIC for lognormal model for hollow core doors.	67
6.12	Summary of coefficient estimates, p-value, and DIC for normal model for hollow core doors.	67
6.13	OpenBUGS script of a Weibull distribution with influence on the beta parameter.	69
6.14	Summary of coefficient estimates, p-value, and DIC for Weibull model with environment impact on beta parameter for hollow core door.	70
6.15	Summary of coefficient estimates, p-value, and DIC for Weibull model with environment impact on lambda parameter for hollow core door.	70
6.16	Comparison of depth coefficient posterior results for several prior cases using steel door data.	73
6.17	Comparison of depth coefficient posterior results for several prior cases using hollow core door data.	73
6.18	Coefficient results for standard logit regression models using single variables for hollow core doors.	76
6.19	Coefficient results for centered logit regression models using single variables for hollow core doors.	76

6.20	DIC results for single variable regression models using hollow core door data.	77
6.21	Coefficient results for standard logit regression models using single variables for steel doors.	77
6.22	Coefficient results for centered logit regression models using single variables for steel doors.	77
6.23	DIC results for single variable regression models using steel door data.	78
6.25	P-values and DIC of logistic regression model with an added latent variable using steel door data.	82
6.24	Script showing model portion of logistic regression model with an added latent variable.	83
6.26	Portion of developed OpenBUGS script for logistic regression on temperature using primary O-ring distress data.	85
6.27	Summary posterior estimates of logistic regression on temperature using developed model.	86
6.28	Summary posterior estimates of logistic regression parameters, temperature included as explanatory variable [19].	86
6.29	Model-checking results for logistic regression of primary O-ring distress [19].	86
6.30	Model-checking results for logistic regression on temperature using developed model.	87
6.31	Model portion of final developed OpenBUGS script for logistic regression using hollow core door data.	90
6.32	Summary posterior estimates of logistic regression parameters for hollow core doors, water depth included as explanatory variable. . .	91

6.33	Summary posterior estimates of logistic regression parameters for hollow core doors, water flow rate included as explanatory variable.	91
6.34	Summary posterior estimates of logistic regression parameters for hollow core doors, water temperature included as explanatory variable.	92
6.35	Summary posterior estimates of logistic regression parameters for hollow core doors, water depth and flow rate included as explanatory variable.	92
6.36	Summary posterior estimates of logistic regression parameters for hollow core doors, water depth and temperature included as explanatory variable.	92
6.37	Summary posterior estimates of logistic regression parameters for hollow core doors, water flow rate and temperature included as explanatory variable.	92
6.38	Summary posterior estimates of logistic regression parameters for hollow core doors; water depth, flow rate, and temperature included as explanatory variable.	93
6.39	Model checking and comparison results for logistic regression for hollow core doors using developed model.	93
6.40	Model portion of final developed OpenBUGS script for logistic regression using steel door data.	95
6.41	Summary posterior estimates of logistic regression parameters for steel doors, water depth included as explanatory variable.	96
6.42	Summary posterior estimates of logistic regression parameters for steel doors, water flow rate included as explanatory variable.	96

6.43	Summary posterior estimates of logistic regression parameters for steel doors, water depth and flow rate included as explanatory variables.	96
6.44	Model checking and comparison results for logistic regression for steel doors using developed model.	97
6.45	Summary posterior estimates of depth coefficient; normal, flat, and uniform distributions used as non-informative priors.	99
7.1	Python fragility script for determining the failure depth.	108
7.2	Python failure script for setting door failure when corresponding depth is reached.	109

List of Acronyms

AIC	Akaike Information Criteria
BIC	Bayesian Information Criteria
BUGS	Bayesian inference Using Gibbs Sampling
CDF	Cumulative Distribution Function
CFD	Computational Fluid Dynamics
CFEL	Component Flooding Evaluation Laboratory
DIC	Deviance Information Criteria
GPM	Gallons Per Minute
GUI	Graphical User Interface
IISPH	Implicit Incompressible SPH
INL	Idaho National Laboratory
ISU	Idaho State University
LWRS	Light Water Reactor Sustainability
MCMC	Markov Chain Monte Carlo
NASA	National Aeronautics and Space Administration
NPP	Nuclear Power Plant
NRC	Nuclear Regulatory Commission
PET	Portal Evaluation Tank
PRA	Probabilistic Risk Assessment
PSA	Probabilistic Safety Assessment
SPH	Smooth Particle Hydrodynamics
USD	United States Dollar

Assessing Nuclear Power Plant Component Fragility in Flooding Events Using
Bayesian Regression Modeling with Explanatory Variables

Dissertation Abstract – Idaho State University (2020)

Nuclear power plants (NPPs) are at risk from various internal and external flood hazards that can lead to extensive damage. Events such as the tsunami at Fukushima NPP or the river inundation at Fort Calhoun NPP underscore the necessity of conducting flooding probabilistic safety assessments (PSA) for NPPs. The response of components to flooding conditions can have considerable impact on the stable operation of NPPs, however, flood PSA is not common and flooding related risk analysis lacks data. To help address this shortcoming, components were tested in the Portal Evaluation Tank (PET) to provide the necessary data, initially performing experiments on non-watertight doors. A Bayesian regression model, utilizing explanatory variables, was developed for assessing NPP component fragility in flooding events. The concluding model uses the binomial distribution with a logistic regression model for probability of failure and uses non-informative priors so the experimental data takes preference in the analysis. The fragility model was validated using several metrics, including three types of Bayesian p-values. Finally, an application of the component fragility model was demonstrated to show how improvements might be made to 3D flooding simulations.

Chapter 1

Introduction

As of September 2019, nuclear energy provides approximately 10% of the world's electricity from about 450 power reactors operational in 30 countries, with nearly 100 residing in the United States to generate 1/5 of the country's electrical capacity [1]. These nuclear power plants (NPP) withdraw vast amounts of water, $\sim 25,000\text{-}60,000 \frac{\text{gal}}{\text{min}}$, primarily for cooling to prevent reactor core overheating [2]. Due to these high water demands, NPPs are typically situated near reliable bodies of water—rivers, lakes, estuaries, or oceans—and contain miles of water filled pipes. This puts NPPs at an increased risk for various internal (e.g. broken pipes) and external (e.g. heavy rainfall and swollen rivers) flooding events.

Flooding has long been recognized as a risk to manage with an array of protection measures [3] [4] [5]. However, the tsunami that struck the Japanese Fukushima Daiichi NPP complex in March 2011 served as a severe reminder of the impact flooding hazards can have on nuclear power plants [6] [7] [8]. An $\sim 43\text{-}49$ foot tsunami struck the Fukushima site, as seen approaching in Figure 1.1, causing the meltdown of three reactors and the release of radioactive materials into the environment [9].

While tsunamis may not be a significant risk for all reactor locations, there are other internal and external events that can lead to flooding. In June 2011, the Fort Calhoun NPP in Nebraska declared a "Notification of Unusual Event" due to flooding [10]. Heavy spring rains and snowmelt caused the Missouri River to overflow and inundate the plant, as seen in Figure 1.2. This event highlights the economic and reliability impact of flooding risks since the total recovery cost associated with this single event exceeded \$500 million USD and drove the plant's



Figure 1.1: Tsunami waves approaching the Fukushima-Daiichi Power Plant.

production costs to \$71/MWhr, approximately double the industry average at the time [11] [12]. This eventually led to plant closure, with decommissioning costs estimated at \$1.2 billion USD [13].

The above are just two examples of flooding events at NPPs, which forced off-normal conditions and had significant economic impacts. NPP flooding events are more common than may be thought. In the U.S., NPPs have faced various internal and external flooding hazards. Further literature review and brief summaries of some U.S. nuclear plant flooding may be found in Appendix A.

These events underscore the necessity of conducting flooding probabilistic safety assessment (PSA) for NPPs. The four basic steps of external event PSA are: 1) hazard analysis, 2) component fragility evaluation, 3) plant response analysis, and 4) risk quantification. The ASME/ANS Probabilistic Risk Analysis (PRA)



Figure 1.2: Fort Calhoun NPP surrounded by Missouri River flood water.

Standard [14] outlines the technical requirements for performing the analysis. An important task is the determination of component fragility. The fragility of the component is the probability of it reaching a limit state condition at a given demand [15]. However, flood PSA is not very common and flooding related risk analysis lacks data. An improved and deeper understanding of the reliability of NPP components under flooding conditions is needed to better inform risk models.

To help address the shortcomings of component characterization, the Component Flooding Evaluation Laboratory (CFEL) was developed at Idaho State University (ISU). The CFEL research group focuses on individual component performance when subjected to flooding scenarios, beginning with rising water experiments conducted in a custom designed tank. These experiments provide the necessary data to develop component fragility mathematical models.

Improved characterization of component fragility can be integrated into comprehensive NPP risk models to enhance the overall risk understanding. The following work presents the development of a Bayesian regression model for assessing NPP component fragility in flooding events. This fragility model utilizes component failure data, generated as part of the research from full-scale flooding experiments, to introduce explanatory variables into the failure probability. The developed fragility model was validated using several metrics, including three types of Bayesian p-values. An application of the fragility model is provided by coupling the work to a flooding simulation. By analyzing the probabilistic risk associated with component flooding, data informed flooding fragility can improve simulation to realistically model component behavior before a flooding event occurs.

Chapter 2

Bayesian Data Analysis

The approach selected to analyze component failure in flooding scenarios are fragility models. Most fragility modeling in the nuclear industry has been focused on seismic component fragility determination. In a seismic fragility model, the single vertical ground acceleration variable is used to completely characterize the failure probability of structures or components of interest. These fragility models have been simple monotonically-increasing functions that define the probability of a certain failure mode as:

$$P_f(A) = \phi\left(\frac{\ln\left(\frac{A}{A_m}\right)}{\beta_R}\right) \quad (2.1)$$

where ϕ is the standard Gaussian cumulative distribution function (CDF), A is the peak ground acceleration, A_m is the median ground acceleration, and β_R is the log standard deviation of randomness [16]. An example of a seismic fragility model's plot is shown in Figure 2.1.

However, other observable parameters may be better indicators for the potential of failure. Expanding upon the seismic example, these observables could include the detailed characteristics of the earthquake such as X, Y, and Z components of the ground motion; frequency of the waves; the age of the component; the anchorage of the component; the specifics of the component type; or any combination of the above.

Limitations found in these traditional fragility models include simplistic (single driving parameter) and excessive conservatism. Some flooding scenarios, such as submersion or slow water-rise, may be sufficiently modeled with a simple one-

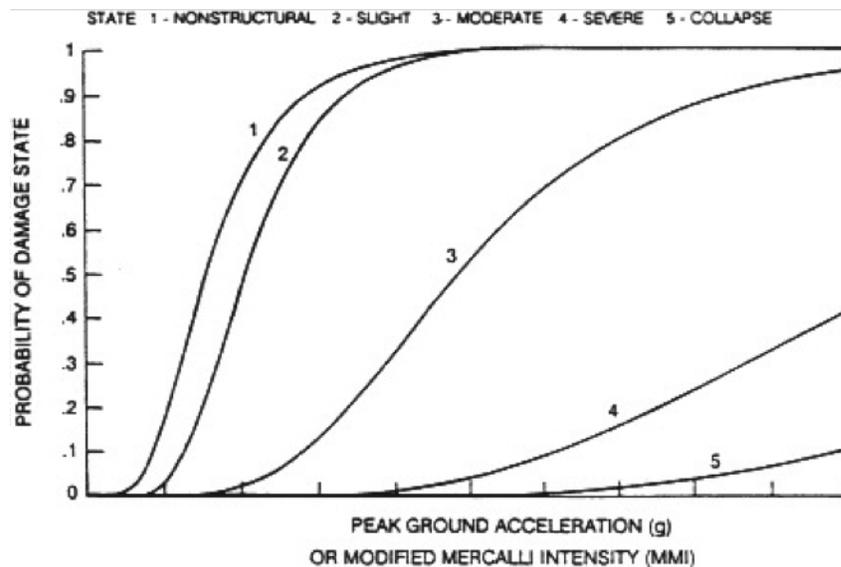


Figure 2.1: Illustration of seismic fragility for an example component.

dimensional approach [16]. For complex flooding fragility modeling requiring more observables, these issues will be avoided by moving to a more flexible, data-informed approach—Bayesian fragility modeling through phenomena-driven regression modeling. As stated by Box and Tiao [17], “Bayesian inference alone seems to offer the possibility of sufficient flexibility to allow reaction to scientific complexity free from impediment from purely technical limitation.”

2.1 Overview of Bayesian Inference Process

Reverend Thomas Bayes’ famous paper, published in 1763, provides the basis for the Bayesian inference method. He opened with the following [18]:

Given the number of times in which an unknown event has happened and failed: *Required* the chance that the probability of its happening in a single trial lies somewhere between any two degrees of probability that can be named.

The problem contains two key parts. The first is the use of probability as a means of expressing uncertainty about an unknown quantity of interest. The other

is evaluating a conditional probability of failure in a single trial, given data on previous failures. This implies one can learn about failure probability on the basis of observed data. Bayes went on to propose a theorem combining conditional and marginal probabilities of random variables in order to update a prior state of knowledge.

2.1.1 Bayes' Theorem

From the Bayesian perspective, both data and parameters can have probability distributions, and the task of Bayesian analysis is to build a model for the relationship between parameters (θ) and observables (y), and then calculate the posterior probability. The Bayesian method, therefore, relies on three items: an aleatory model, a prior distribution for the parameter(s) of the aleatory model, and data associated with the aleatory model. An aleatory model pertains to stochastic or non-deterministic events, the outcome of which is described using probability. The posterior distribution for the model output function is developed in accordance with Bayes' Theorem [19], which is generally written as:

$$p(\theta|y) = \frac{p(\theta)p(y|\theta)}{p(y)} \quad (2.2)$$

where,

- $p(\theta|y)$: Posterior distribution, which is conditional upon data (y) that is known related to the hypothesis (θ);
- $p(\theta)$: Prior distribution, for knowledge of the hypothesis (θ) that is independent of data (y);
- $p(y|\theta)$: Likelihood, or aleatory model, representing the process or mecha-

nism that provides data (y);

- $p(y)$: Marginal distribution, which serves as a normalization constant.

In Bayesian computation, it is not usually necessary to calculate the denominator $p(y)$ to evaluate properties of the posterior, because it is a fixed constant that does not affect inference. Bayes' Theorem can be expressed simply as:

$$p(\theta|y) \propto p(\theta)p(y|\theta) \quad (2.3)$$

where proportionality is in relation to θ [20].

In summary, the above equation takes our prior knowledge about the parameters and updates this knowledge with the likelihood to observe the data for particular parameter values and gives the posterior probability. It essentially states:

$$\text{posterior} \propto \text{prior} \times \text{likelihood} \quad (2.4)$$

This process combines everything that is known about a particular data set and model response to produce a posterior estimate of the output function's probability distribution.

2.1.2 Prior Distributions

A defining feature of Bayesian inference is the prior information taken into account in the analysis. Therefore, a fundamental part of any Bayesian analysis is the prior distribution. This distribution represents all that is known or assumed, including uncertainty, about a parameter θ before the observed data is analyzed. Prior distributions can be categorized several ways, with the basic division being between informative and non-informative priors [21].

An informative prior is not dominated by the likelihood and has an impact on the resulting posterior distribution. Informative prior distributions can be either objective, subjective, or a combination of both [20]. Objective input to the prior distribution could include operational data, observational data from a previous comparable experiment, or manufacturer provided information. In contrast, subjective input can be based off of an analyst's personal experience or judgements, stating one's beliefs regarding the parameter, the experiment, or opinions.

If the prior has minimal impact on the posterior distribution, it is non-informative. The prior is also described as vague, flat, or diffuse. Non-informative priors are intended to allow Bayesian inference when relatively limited information is known about a parameter rather than there being a state of complete ignorance as the name may suggest. The rationale for using a non-informative prior is that a parameter's prior information is not considered substantial relative to the information provided by observed data [22]. In other words, inferences are unaffected by information external to the current data.

2.2 Introduction to BUGS

Integration of functions plays an important role in Bayesian statistical analysis, however, explicit evaluation of these integrals is only possible for a limited number of special cases. When the posterior belongs to the same family of distributions as the prior, meaning they are conjugate, the posterior distribution can be computed analytically [20]. For example, a normally distributed prior on mean (μ) is conjugate with respect to a normal likelihood distribution [23]. The mathematics for this Normal-Normal case are demonstrated below.

Assume the variance σ^2 is known, while the mean μ is unknown. For a Bayesian analysis, a prior is specified for μ , $p(\mu)$. Suppose a normal prior,

$N(\mu_0, \sigma_0^2)$, such that

$$p(\mu) = \frac{1}{\sqrt{2\pi\sigma_0^2}} \exp\left(-\frac{(\mu - \mu_0)^2}{2\sigma_0^2}\right) \quad (2.5)$$

where the mean and variance of the prior are specified by μ_0 and σ_0 . Additionally, assume the data, y , is normally distributed with known variance such that the likelihood function is given by

$$p(y|\mu) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - \mu)^2}{2\sigma^2}\right). \quad (2.6)$$

Using Bayes' Theorem, this gives a posterior, $p(\mu|y)$, of

$$p(\mu|y) \propto \exp\left(-\frac{(\mu - \mu_0)^2}{2\sigma_0^2} - \frac{\sum_{i=1}^n (y_i - \mu)^2}{2\sigma^2}\right). \quad (2.7)$$

up to a constant of proportionality, which has been dropped for simplicity. In fact, any term that does not include μ can be viewed as a proportionality constant, factored out of the exponential, and dropped. Rewriting the terms inside the exponential using algebra gives

$$-\frac{1}{2} \left[\frac{\mu^2 - 2\mu\mu_0 + \mu_0^2}{\sigma_0^2} + \frac{\sum y^2 - 2n\bar{y}\mu + n\mu^2}{\sigma^2} \right] \quad (2.8)$$

$$-\frac{1}{2} \left[\frac{\mu^2\sigma^2 - 2\sigma^2\mu\mu_0 - 2\sigma_0^2n\bar{y}\mu + \sigma_0^2n\mu^2}{\sigma^2\sigma_0^2} \right] \quad (2.9)$$

$$-\frac{1}{2} \left[\frac{(n\sigma_0^2 + \sigma^2)\mu^2 - 2(\sigma^2\mu_0 + \sigma_0^2n\bar{y})\mu}{\sigma^2\sigma_0^2} \right] \quad (2.10)$$

$$-\frac{1}{2} \left[\frac{\mu^2 - 2\mu \frac{(\sigma^2\mu_0 + \sigma_0^2 n\bar{y})}{(n\sigma_0^2 + \sigma^2)}}{\frac{\sigma^2\sigma_0^2}{(n\sigma_0^2 + \sigma^2)}} \right] \quad (2.11)$$

$$-\frac{1}{2} \left[\frac{\left(\mu - \frac{(\sigma^2\mu_0 + \sigma_0^2 n\bar{y})}{(n\sigma_0^2 + \sigma^2)}\right)^2}{\frac{\sigma^2\sigma_0^2}{(n\sigma_0^2 + \sigma^2)}} \right]. \quad (2.12)$$

Therefore, the posterior function of μ is normally distributed with mean

$$\mu_n = \frac{(\sigma^2\mu_0 + \sigma_0^2 n\bar{y})}{(n\sigma_0^2 + \sigma^2)} \quad (2.13)$$

and variance

$$\sigma_n^2 = \frac{\sigma^2\sigma_0^2}{(n\sigma_0^2 + \sigma^2)}. \quad (2.14)$$

Some additional information can be gained from the mean and variance. Rearranging the variance gives

$$\frac{1}{\sigma_n^2} = \frac{n}{\sigma^2} + \frac{1}{\sigma_0^2}. \quad (2.15)$$

Thus, the inverse of the variance, known as precision λ , of the posterior is the precision of the prior plus the precision of the data scaled by the number of observations n . Likewise, applying the following substitutions for precision

$$\lambda = \frac{1}{\sigma^2} \quad \lambda_0 = \frac{1}{\sigma_0^2} \quad (2.16)$$

the mean can be rewritten as

$$\mu_n = \frac{\lambda_0\mu_0}{n\lambda + \lambda_0} + \frac{n\lambda\bar{y}}{n\lambda + \lambda_0} \quad (2.17)$$

The mean of the posterior is a weighted average of the prior mean μ_0 and the data mean \bar{y} . The weight on the prior mean is proportional to its precision. Also, the weight on the data mean is proportional to its precision. Therefore, if the prior mean is very precise relative to the data mean, it will be highly weighted. Alternatively, if the data mean is more precise relative to the prior mean, it is given a larger weight.

While solving for the posterior distribution analytically is possible for conjugate cases, usually problems will involve complex distributions and explicit evaluation is not possible. Traditionally, statisticians would be forced to use numerical integration or analytical approximation techniques. However, there are now several powerful software programs that exist for full Bayesian inference. One of the most widely used by statistical practitioners is the BUGS (Bayesian inference Using Gibbs Sampling) family of programs. The most popular packages from the BUGS family are WinBUGS and OpenBUGS. Both software packages are freely available online, with the later being an open source version. A basic guide to running OpenBUGS is provided in Appendix B. There are several methods devised for construction and sampling complex Bayesian posterior distributions. BUGS software utilize Markov chain Monte Carlo (MCMC) methods.

2.2.1 Markov Chain Monte Carlo Method

MCMC is a general method based on randomly sampling values from a prior distribution to approximate the posterior distribution $p(\theta|y)$. The sampling is done sequentially, with the distribution of the sampled parameter depending on the value from the previous step only, forming a Markov chain [22].

As defined in probability theory, a sequence of random vectors $\theta^0, \theta^1, \theta^2, \dots$ forms a Markov chain if, for any iteration t , the distribution of θ^{t+1} given all

previous θ 's is given by

$$\theta^{t+1} \sim p_{trans}(y|\theta^{(t)} = y^{(t)}). \quad (2.18)$$

where the right hand side is called the transition distribution of the Markov chain [20]. It defines the conditional probability of moving to any particular new value given the current value of the chain. With appropriate transition probabilities, as $t \rightarrow \infty$, the Markov chain will converge to a unique stationary distribution, the posterior distribution. Therefore, the key to MCMC method is the approximate distributions are improved at each step in the simulation, and after running the simulation long enough, converging to the posterior distribution. An additional feature of Markov chains is that they are memoryless. Each variable in the chain depends directly on its predecessor and no new knowledge comes from knowing the history of events.

Many algorithms exist for performing sampling for a wide range of applications. OpenBUGS uses the following: Gibbs, Metropolis-Hasting, and slice sampling [24]. Of these three, the Gibbs sampler is one of the mostly widely used algorithms for simulating Markov chains.

2.2.2 Gibbs Sampling

The Gibbs sampler is a special case of the Metropolis-Hastings algorithms [20] used in simulating multidimensional Markov chains by dividing the parameter vector θ into k subvectors, $\theta = (\theta_1, \theta_2, \dots, \theta_k)$. For each iteration t , the subvector is sampled, conditional on the most recent values for all other subvectors. The algorithm proceeds as follows [22]:

1. Choose initial ($t = 0$) starting values for each component, $\theta_1^{(0)}, \theta_2^{(0)}, \dots, \theta_k^{(0)}$.

2. Sample, from the full conditional distribution, new values for each element of θ by cycling through the following steps:

- (a) Sample a new value of θ_1 , given the most recent values of all other elements of θ and the data.

$$\theta_1^{(1)} \sim p(\theta_1 | \theta_2^{(0)}, \theta_3^{(0)}, \dots, \theta_k^{(0)}, y) \quad (2.19)$$

- (b) Sample a new value of θ_2 , given the most recent values of all other elements of θ and the data.

$$\theta_2^{(1)} \sim p(\theta_2 | \theta_1^{(1)}, \theta_3^{(0)}, \dots, \theta_k^{(0)}, y) \quad (2.20)$$

Note that the most recent value for θ_1 is now $\theta_1^{(1)}$.

- (c) ...

- (d) Sample a new value of θ_k .

$$\theta_k^{(1)} \sim p(\theta_k | \theta_1^{(1)}, \theta_2^{(1)}, \dots, \theta_{k-1}^{(1)}, y) \quad (2.21)$$

This completes one iteration of the Gibbs sampler.

3. Repeat step 2 for many iterations ($t = 2, 3, \dots, T$), updating conditional on the most recent values of all other components of θ .

The power behind Gibbs sampling is that simulation of a complex, multivariate posterior distribution is reduced to a sequence of algorithms for sampling a simpler conditional distribution [20]. Figure 2.2 provides a graphical illustration of the Gibbs sampler for a simple two-parameter case.

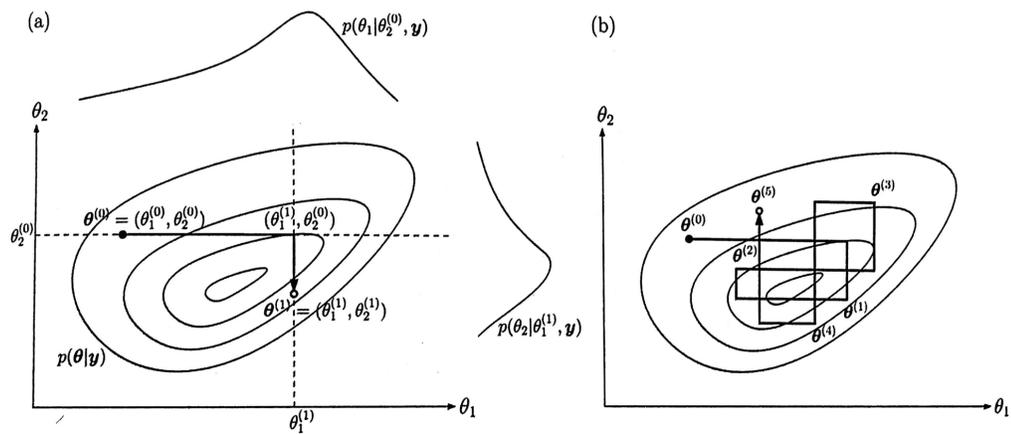


Figure 2.2: (a) First iterations of the Gibbs sampler for a two-parameter problem. The contours show the target posterior distribution and the full conditional distributions are above and to the right. (b) Gibbs sampler after five iterations [20]

Chapter 3

Model Checking and Comparison

After constructing a probability model and computing posterior distributions for all estimated parameters, the next step of a Bayesian analysis includes checking that the model adequately represents the data and is plausible for the purpose for which the model will be used. Note, however, it is not a matter of selecting the "correct" model or asking whether a model is "true." Probability models in most data analysis use assumptions to reasonably approximate reality and will not be perfectly true. British statistician George Box wrote the famous aphorism [25], "All models are wrong, but some are useful." Instead, model checking is used to identify failings in the probability model that may limit its utility [19]. Model criticism and comparison, therefore, involves a degree of judgement by the analyst and cannot be strictly reduced to a set of formal rules [20].

3.1 Model Checking Procedures

There are multiple ways of assessing a model's performance. Available model checking procedures include pivotal discrepancy measures, prior and posterior predictive checks, sampled predictive p-values, and predictive probability integral transform tests, to just name a few. Each approach possesses good statistical properties, but many have limits in application. Some procedures require additional data from previous studies (prior predictive checks), while several other approaches utilize double-sampling methods that increase the computational intensity of a Bayesian analysis [26]. The approach selected in the presented work for model checking is posterior predictive checking, a useful direct way of assessing the fit of a model to various aspects of the data. A general approach to posterior

predictive checking is discussed in the following section. Additionally, residual tests are presented for informal model criticism and outlier identification.

3.1.1 Posterior Predictive Checks and Bayesian P-Value

Posterior predictive checks are a primary form of Bayesian model checking used to assess the fit of the model to various aspects of the data. The procedure is based upon the following assumption: if a given model fits, then data simulated or replicated under the model should be comparable to the real-world observed data the model was fitted to [22]. In other words, the observed data should be plausible under the posterior predictive distribution. If any systematic differences occur between simulations and the data, it potentially indicates that model assumptions are not being met.

Let θ be the parameter of the aleatory model that generates the observed data y . The notation y^{rep} is used to denote a replicated data set under the model being checked. This replicated data could have been observed or would be observed if the experiment that produced y was replicated with the same model and θ that produced the observed data. The posterior predictive distribution is then defined as [22]:

$$p(y^{rep}|y) = \int p(y^{rep}|\theta)p(\theta|y)d\theta. \quad (3.1)$$

The above equation averages the likelihood function for the replicated data values over the posterior distribution for θ in order to obtain the posterior distribution for the replicated data, given the observed values.

The model is checked for deviations from an assumed parameter form by means of test quantities or discrepancy functions, $T(y, \theta)$, that depend on both

data and parameters. A check is made whether $T(y, \theta)$ is compatible with the simulated distribution of $T(y^{rep}, \theta)$ by calculating a Bayesian p-value [22]:

$$\begin{aligned} p_B &= Pr(T(y^{pred}, \theta) \geq T(y, \theta) | y) \\ &= \iint I(T(y^{pred}, \theta) \geq T(y, \theta)) \times p(y^{pred} | \theta) p(\theta | y) dy^{pred} d\theta, \end{aligned} \quad (3.2)$$

where I is the indicator function. The p-value is obtained by seeing how often the discrepancy function, based on predictive data, is less than the observed discrepancy, as illustrated in Figure 3.1.

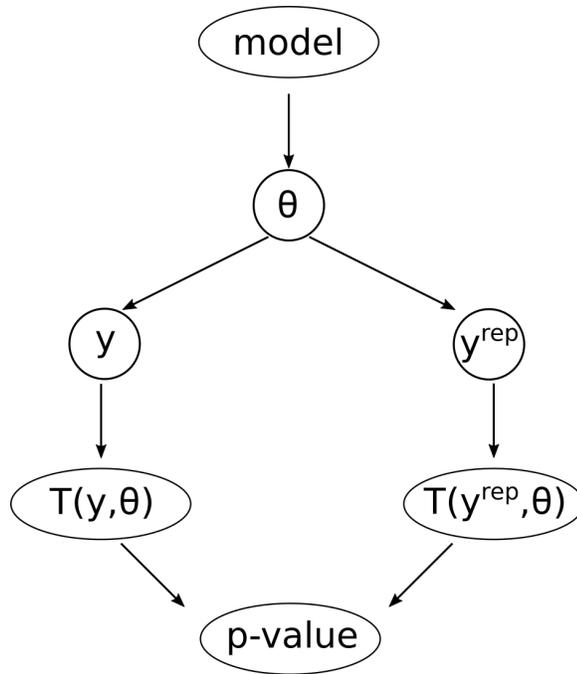


Figure 3.1: Graphical representation of predictive model checking.

Regarding the choice of discrepancy functions, focus is given to diagnosing global lack of fit rather than discovering outliers [26]; a task given to residual calculations. A brief summary of candidate discrepancy functions considered is

provided in Table 3.1

Table 3.1: Discrepancy functions used for Bayesian model checking.

Name	Definition
χ^2	$T(y, \theta) = \sum_i \frac{(y_i - E(y_i \theta))^2}{Var(y_i \theta)}$
Likelihood ratio statistic	$T(y, \theta) = 2 \sum_i y_i \log\left(\frac{y_i}{E(y_i \theta)}\right)$
Freeman-Tukey statistic	$T(y, \theta) = \sum_i (\sqrt{y_i} - \sqrt{E(y_i \theta)})^2$
Cramer-von Mises statistic	$T(y) = \sum_i^n (F(y_i) - \frac{2i-1}{2n})^2$

Note that ideally model checking should be based on new data, although in practice the same data is generally used for both developing and checking the model. This means Bayesian p-values based on these checks tend to be conservative [20]. However, this does not imply that posterior predictive checks lack value. Given that tests are conservative, small (less than 0.05) and large (greater than 0.95) p-values strongly suggest lack of fit. P-values closest to 0.5 indicate a high degree of predictive capability [19]. This concept of Bayesian p-value is graphically represented in Figure 3.2. Again, the checks are used to understand a model's limits of its applicability in reality and not for acceptance or rejection.

3.1.2 Residuals

Residuals measure the discrepancy between the observed data and an assumed model. Informal tests based on Pearson and deviance residuals can be used to identify obvious assumption violations. Note that these analyses are generally carried out informally in Bayesian application, since all residuals depend on θ and have posterior distributions [26]. Therefore, they are not truly independent as re-

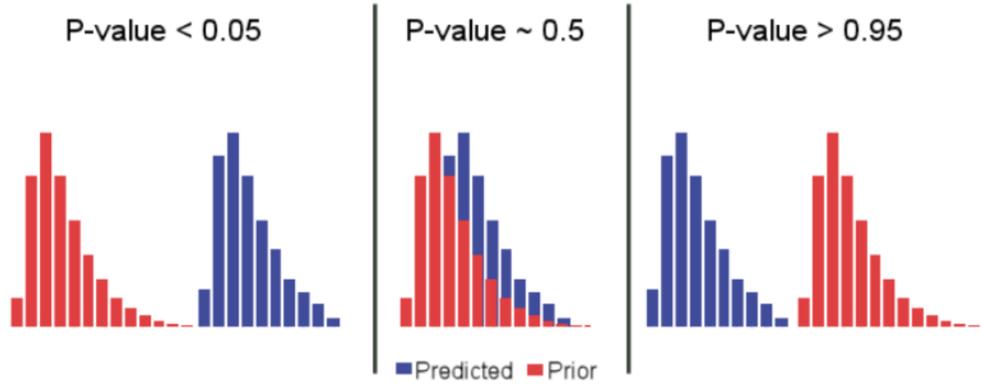


Figure 3.2: Depiction of the Bayesian p-value predictability [19].

quired in unbiased application of goodness-of-fit tests.

A standardized Pearson residual is defined as:

$$r_i = \frac{y_i - E(y_i|\theta)}{\sqrt{\text{Var}(y_i|\theta)}}, \quad (3.3)$$

where $E(y_i|\theta)$ is the expected value and $\text{Var}(y_i|\theta)$ is the variance. Since it is considered a function of random y_i for a fixed θ , Pearson residuals should generally take on values between -2.0 and 2.0 [26]. Values falling outside of this range would represent outliers.

Residuals can also be based on a saturated version of the deviance, defined as:

$$D_S(\theta) = -2 \log p(y|\theta) + 2 \log p(y|\hat{\theta}_S(y)), \quad (3.4)$$

where $\hat{\theta}_S(y)$ are the saturated estimates. For example, the saturated deviance for a binomial distribution is

$$\text{Binomial} : D_S(\theta) = 2 \sum_i \left\{ y_i \log \left[\frac{y_i/n_i}{\theta_i} \right] + (n_i - y_i) \log \left[\frac{1 - y_i/n_i}{1 - \theta_i} \right] \right\}. \quad (3.5)$$

Models for which saturated deviance is appropriate, such as Poisson and binomial, the rule of thumb for a rough assessment of the fit is the mean saturated deviance should approximately equal sample size n [20].

3.2 Model Comparison using Deviance

Following model checking, comparisons can be made on the performance of alternative hypothesized models. It is not an uncommon occurrence for more than one probability model to provide an adequate fit to the data. These models may differ in prior specification, link function selection, or which explanatory variables are included in the regression, to name a few. Therefore, an analysis should not only examine models to see how they fail to fit reality, but compare how sensitive the resulting posterior distributions are to arbitrary specifications using any number of model comparison or performance metrics.

There are a variety of Bayesian model comparison methods that can be broadly divided into two different approaches. The first is based on Bayes factors, a likelihood ratio of two contending hypotheses, and the second is based on information criteria, which are measures of the relative fit. Commonly discussed information criteria are Bayesian Information Criteria (BIC) for non-hierarchical models and Akaike Information Criteria (AIC) or Deviance Information Criteria (DIC) for selecting among hierarchical models. Bayes factors require proper prior distributions, which are not required for DIC or AIC. Additionally, Bayes factors are known to be difficult to compute in MCMC. DIC is a generally straight forward computation and no additional scripting is needed to calculate it in OpenBUGS, making it the comparison approach selected for this work and discussed further below.

3.2.1 Deviance Information Criterion

Deviance Information Criteria is a measure of model fit that can be applied to Bayesian models and is applicable when the parameter estimation is done using techniques such as Gibbs sampling. It is particularly useful in Bayesian model selection problems where the posterior distributions of the model have been obtained by MCMC simulation. DIC is recommended for selection among hierarchical models. A hierarchical model, also referred to as a multilevel model, have mutual dependence on the selected parameters that affect the modeling [27].

Mathematically, DIC is calculated as

$$DIC = Dbar + pD \quad (3.6)$$

$$pD = Dbar - Dhat \quad (3.7)$$

where pD is the effective number of parameters, $Dhat$ is a point estimate of the deviance, and $Dbar$ is the posterior mean deviance [20]. In OpenBUGS, $Dbar$ is automatically monitored by the node called deviance and requires no additional scripting [24].

As a rule of thumb, the model with the smallest DIC usually indicates the better fitting model. Note, however, only differences between models in DIC are important, not strictly absolute values. While it is not easy to define what constitutes an important difference, the following rough guide can be used for DIC comparison [20]:

- Differences greater than 10 can be used to rule out the model with the higher DIC.

- Differences between 5 and 10 are substantial.
- Differences less than 5, there is uncertainty about choice of model. Other methods may need to be considered, especially if models make different inferences

Note that these considerations include negative values for the DIC, which occur in cases where the deviance is negative. It must also be noted that since DIC is a measure of relative fit, a model with the smallest DIC can still be a poor fit for the data [19].

Chapter 4

Preliminary Example Analysis

To illustrate the possibility of a Bayesian approach for fragility modeling, the following preliminary example is provided from reference [19], which estimates the probability of O-ring failure in the solid-rocket booster motors of a space shuttle. For this example, pressure and temperature may be leading indicators of failure and information about these explanatory variables is incorporated into the Bayesian inference.

4.1 Model for NASA O-ring Distress

An O-ring is a torus shaped mechanical gasket designed to seal field joints between each fuel segment, as shown in Figure 4.1. Each shuttle had three primary and three secondary O-rings with one of each type having to fail to cause a disastrous failure at launch like the Challenger in 1986. Table 4.1 shows data taken from the NASA Space Shuttle program related to O-ring thermal stress impacts. The column labeled "distress" represents erosion of the O-ring or blow-by of hot gases. There are six field O-rings on a shuttle, so the number of distress events is an integer in the interval $[0,6]$.

To carry out the quantitative model analysis, two parts are required. First, a model that represents the failure of an O-ring during a launch of the spacecraft. A commonly-used model for failures-on-demand is the binomial. Second, the key variable, p in a binomial model, is the failure probability on demand. However, for the purposes of fragility modeling, it is desirable to determine if and what observable phenomena drives failure. Thus, the parameter p is turned into its own model.

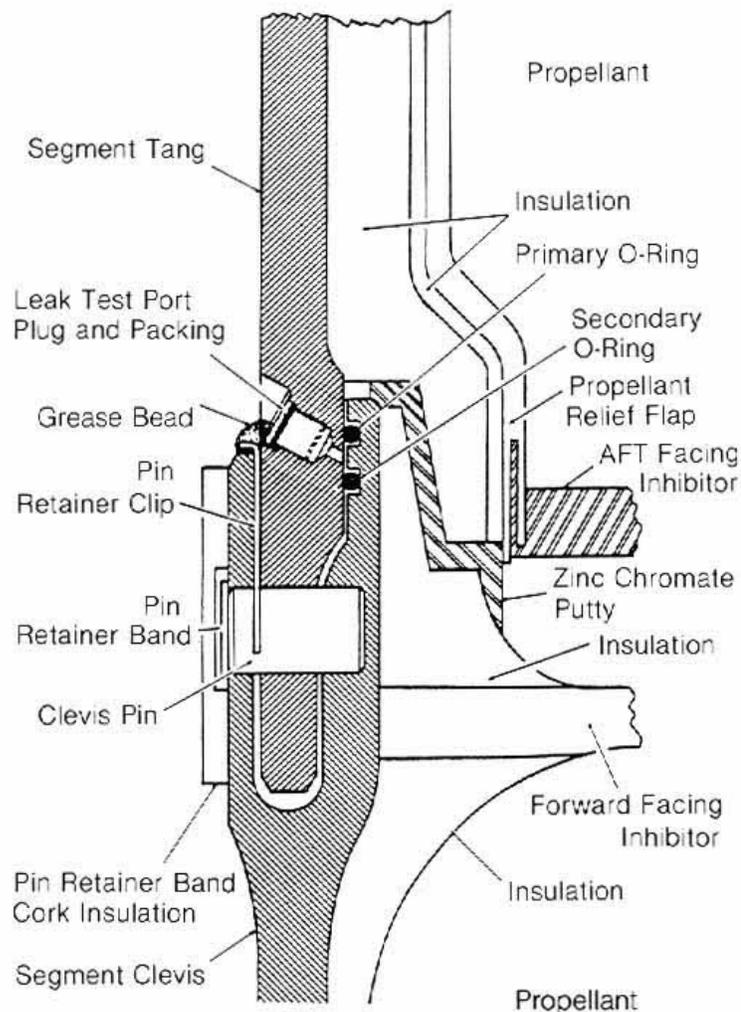


Figure 4.1: Solid rocket cross section showing location of O-rings [28].

For the O-ring example, the number of stress events is modeled as binomial with parameters p and $n = 6$ for the six O-rings potentially challenged during each launch. In this model p is possibly a function of both temperature and applied leak-test pressure. Additionally, since parameter p represents a probability, it must be restricted between the interval $[0,1]$. A common approach to constrain this is to use the logit relationship for p :

Table 4.1: O-ring thermal stress data prior to launch of Challenger in January 1986 [19].

Flight	Distress	Temperature (°F)	Pressure (psig)
1	0	66	50
2	1	70	50
3	0	69	50
5	0	68	50
6	0	67	50
7	0	72	50
8	0	73	100
9	0	70	100
41-B	1	57	200
41-C	1	63	200
41-D	1	70	200
41-G	0	78	200
51-A	0	67	200
51-C	2	53	200
51-D	0	67	200
51-B	0	75	200
51-G	0	70	200
51-F	0	81	200
51-I	0	76	200
51-J	0	79	200
61-A	2	75	200
61-B	0	76	200
61-C	1	58	200

$$\text{logit}(p) = \ln\left(\frac{p}{1-p}\right) \quad (4.1)$$

The fragility model in this case will look at three possibilities: that p is determined by both pressure and temperature, that temperature alone drives failure, or pressure alone drives failure of the O-ring. The above cases are modeled as:

$$\text{logit}(p) = a + bT + cP \quad (4.2)$$

In addition to the parameter estimate calculations, the Bayesian p-value for each model is obtained. The Bayesian chi-squared statistic is used for this example to calculate the p-value as a quantitative measure of the predictive model validity [19]. Again, p-values closest to 0.5 indicate a high degree of predictive capability.

The mean values calculated for the applicable parameters in the O-ring fragility models and corresponding Bayesian p-values are shown in Table 4.3. P-values indicate that Equation 4.4, which is only a function of pressure, is slightly better than the other two models. However, these p-values are all similar, which suggests they all perform adequately. Equation 4.2 could be selected to afford additional flexibility in case additional data was ever to become available.

Table 4.3: Summary posterior estimates of logistic regression parameters for primary O-ring distress.

Parameter	Equation 4.2	Equation 4.3	Equation 4.4
a (intercept)	2.1	5.2	-5.0
b (temperature coefficient)	-0.10	-0.12	n/a
c (pressure coefficient)	0.012	n/a	0.013
Bayesian p-value	0.19	0.21	0.26

The parameters and their corresponding coefficients in Table 4.3 can be used with the fragility model to calculate the failure probability for an O-ring as a function of pressure and/or temperature. For example, using the Equation 4.2 fragility model and setting it equal to the logit Equation 4.5, the probability p can be solved to determine:

$$p = \frac{1}{e^{-(a+bT+cP)} + 1} \quad (4.5)$$

Figure 4.2 shows the plot of the failure probability, or fragility probability, ver-

sus temperature for the three fragility models, also known as the O-ring fragility curves. At low launch temperatures, the probability that one of the six O-rings experiences thermal stress is higher. Note that the Equation 4.4 fragility model is invariant to changes in temperature ($p = 0.024$), since the regression model in that case does not have the temperature parameter in the model. Also, full uncertainty on the fragility model can also be determined, since the uncertainties on all parameters are quantified via OpenBUGS.

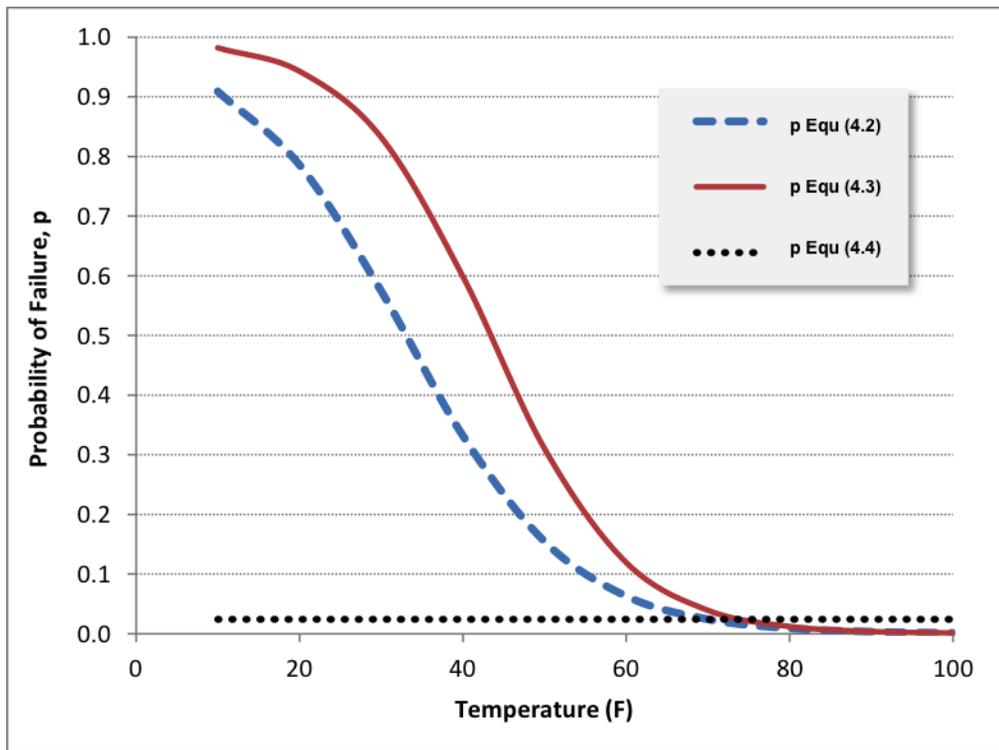


Figure 4.2: Fragility model curves for O-ring example.

Chapter 5

Component Flooding Performance Experiments

The objective of component flooding experiments is to acquire data that can be incorporated into models representing component fragility. Initial experiments performed in a custom designed tank provide both failure data and experience to inform future testing and apparatus modifications to expand the range and type of data collected. The following sections provide detail on experiment apparatuses, procedures, and results, as well as future experiments that could be undertaken.

5.1 Portal Evaluation Tank

To better understand the failure characteristics of doors and conduct door performance experiments, the Portal Evaluation Tank (PET) was designed and built to facilitate testing of doors under flooding conditions [29]. The PET, shown in Figure 5.1, is a steel semi-cylindrical tank with a height and diameter of 8 ft. Its design includes a 62.4 ft² opening to the outside for installation of components to be tested, a front water tray with a 90-degree v-notch weir to determine leakage rates, and the ability to hold up to 2,000-gal of water. The PET was initially connected through 3 in. PVC pipes to a 5-HP submersible pump, which is located inside a ~8,000-gal water reservoir. The initial PET piping configuration is shown in Figure 5.2. Accompanying instrumentation initially included an inlet electromagnetic flow-meter, an ultrasonic depth sensor, and a pressure transducer. The top of the PET is also equipped with pressure and air relief valves and a digital pressure gauge.

Multiple hollow core door tests were conducted using the initial PET piping configuration. This allowed for testing and data recording methodology to be de-



Figure 5.1: The Portal Evaluation Tank.

veloped for door and future component experiments. These tests also provided the opportunity to determine limitations of the PET and identify upgrades for the system.

Following the initial door experiments, design work was pursued to improve the PET capabilities. The initial PET door tests were limited to a single inlet flow rate of ~ 300 gpm. Additionally, the initial piping configuration did not allow tests where the tank could be pressurized to provide additional hydrostatic head. Limitations associated with data recording were also identified in the initial tests. Modifications to the PET were performed to support variable inlet flow rates up to $\sim 4,500$ gpm by connecting to a 60-HP pump. The modifications also support completely filling the PET and then relying on the pump and pressure and air relief

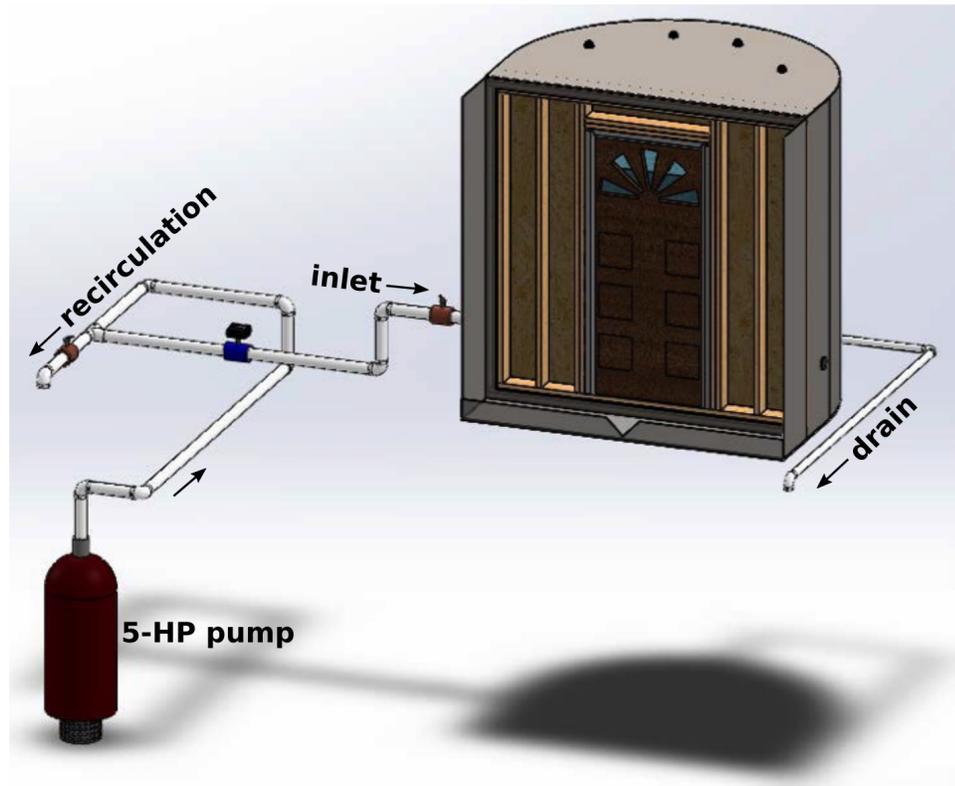


Figure 5.2: Initial PET piping configuration.

valves to provide hydrostatic head to simulate water depths up to 20 ft. Additional data recording instrumentation was also installed in the system. Figure 5.3 shows the modified PET piping configuration.

The modified piping configuration can be divided into two regions: upstream and downstream of the PET. The upstream region is the piping from the 60-HP pump to the PET. It contains two electromagnetic flow-meters, a 12 in. and an 8 in., with accompanying butterfly valves to direct the water through each flow-meter. The two different flow-meters allows for accurate measurement of a wide range of flow rates. A recirculation valve and return line are also part of the upstream region and can redirect water flow back into the reservoir when needed as a pump return line. The upstream region ends at a tee joint located at the top of the PET.

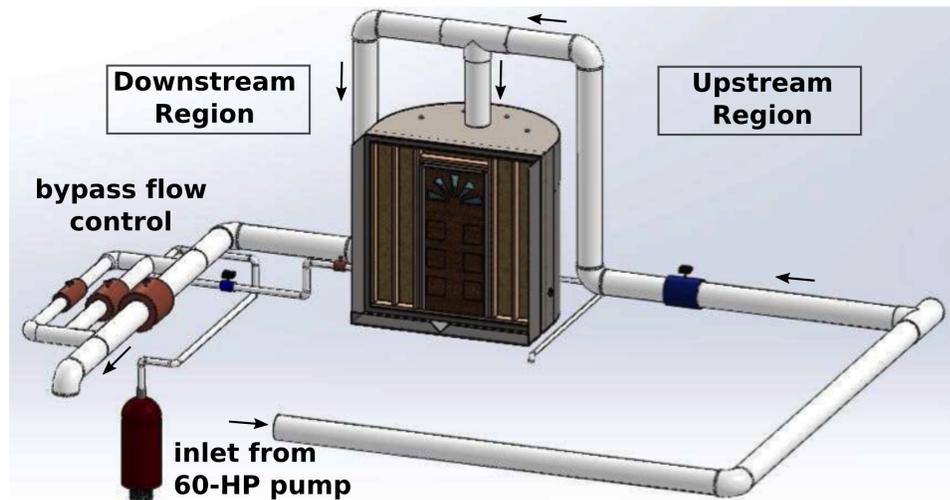


Figure 5.3: Modified PET piping configuration.

Water can then flow into the tank via a perforated center column. The circular perforations evenly distribute water towards the walls of the tank, as seen in Figure 5.4, but avoid applying additional water jet force to the installed component. Remaining water flows across the tee joint into the downstream region. This section of piping contains an additional set of 12 in. and 8 in. flow-meters and three distinct piping lines (12 in., 8 in., and 6 in.) with matching sized butterfly valves. The downstream configuration serves two functions. The first function is to adjust the water flow into the tank during filling and the second function is to adjust the water pressure and flow rate in the system once the PET is filled. After the water exits the downstream region, it returns to the reservoir. The overall PET piping and instrumentation diagram for the modified piping configuration is shown in Figure 5.5.



Figure 5.4: Perforated center column inside modified PET configuration.

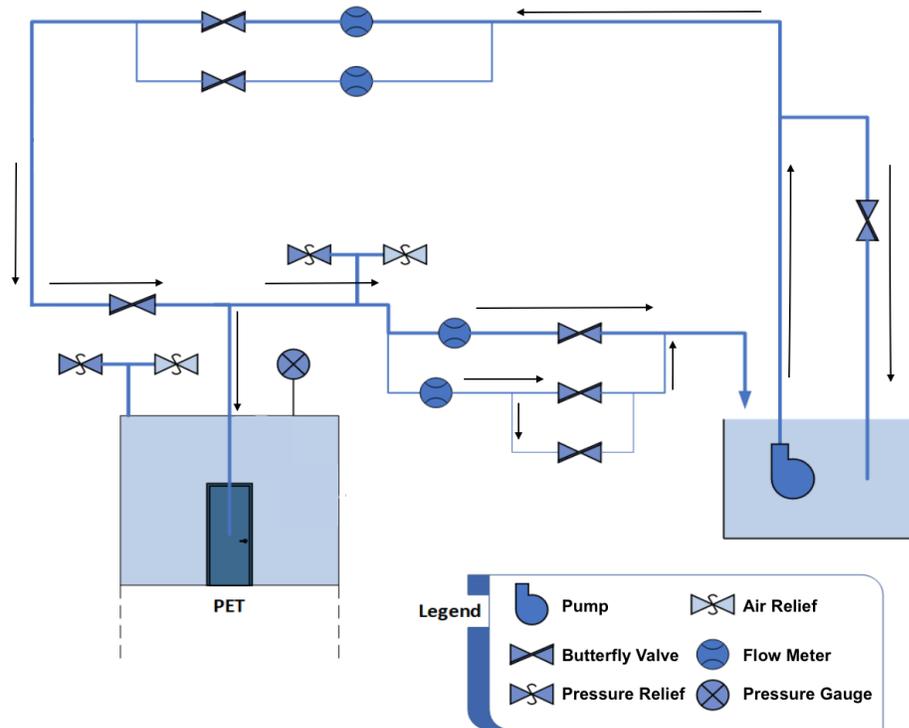


Figure 5.5: Modified PET piping and instrumentation diagram.

5.2 Measurements and Procedure

Besides the piping differences between the initial and modified PET configurations, the measurements recorded for each configuration were also different. The measurements recorded with the initial PET configuration included the flow rate

into the PET, tank water depth, and water temperature. The modified PET configuration measurements include upstream and downstream flow rates with multiple pipeline sizes, two water depths for averaging, and water temperature. The PET, regardless of the configuration, can also measure small leakage rates that do not exceed the v-notch weir barrier and pressures for simulated hydrostatic head once the PET is filled.

For both configurations, a testing procedure needed to be identified. First, each test was started with pre-set valve positions. While the initial piping configuration contained only one filling configuration, the modified PET configuration allowed for water regulation to be adjusted by valves in both the upstream and downstream piping regions. These adjustments were done to set a filling rate or reach a failure state. The measurements phase was started followed by the water supply pump activation. The test concluded when either a failure state occurred or the water leakage rate equalized or exceeded the filling rate and measurement ended. Figure 5.6 shows the PET testing procedure.

5.3 Experiments and Data

Once the configuration, measurements, and procedure were identified, a door performance experiment could be conducted. The following subsections provide an explanation of each experiment performed in the order they were performed. Therefore, the experiments start with wooden hollow core, outward swinging doors and end with steel, inward swinging doors. This experiment order was used based on the anticipated door performance. It was predicted that the outward swinging hollow core door would fail with a lower water depth and little potential damage where the inward swinging steel door would fail with a higher water depth and potentially catastrophic damage. By conducting the experiments

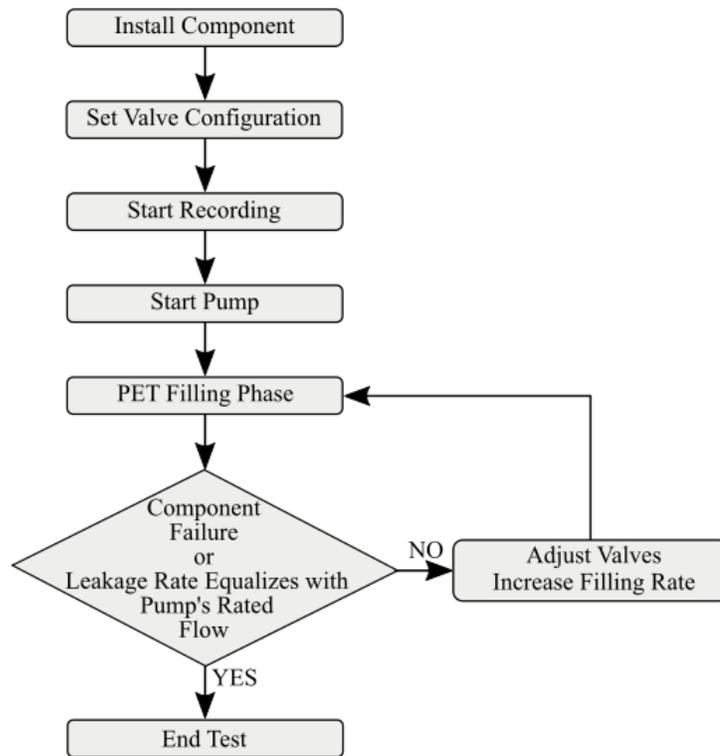


Figure 5.6: Flowchart for testing procedures.

in increasing order of anticipated water depth and potential damage, it allowed for the testing procedure and experiment setup to be perfected to ensure safety during the potentially high-water depth tests.

5.3.1 Hollow Core Door Experiments

The first components to undergo testing in the PET were wooden hollow core doors, typically installed inside residences. These doors were selected for accessibility and price to increase the number of tests that could be conducted. A wall was built to support the pre-hung doorframe using decreased stud spacing to strengthen the wall and ensure the door was the weakest component subjected to testing. A constructed wall ready for installation is shown in Figure 5.7. The

experimental approach subjected each door to a water rising scenario until catastrophic failure of the door occurred or the leakage rate equalized with the filling rate.



Figure 5.7: Hollow core door set within constructed wall for installation.

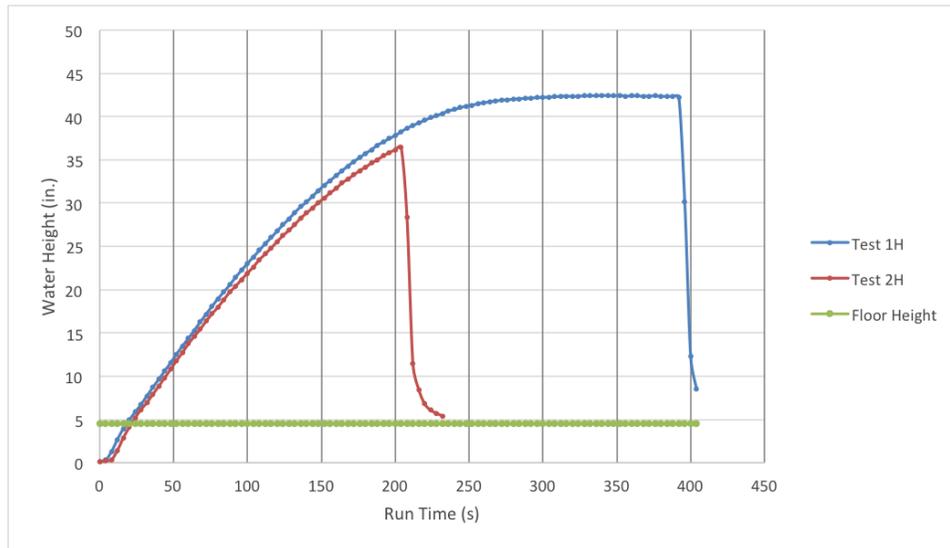
The first experiments used an outward swinging door configuration. These outward swinging door tests exhibited excessive bowing at the base of the door, which resulted in the equalization of incoming water flow and leakage occurring around the door. The bowing effect was temporarily addressed by securing a 2.4 in. plywood strip to the bottom section of the door frame, eliminating the door bowing effect and reducing leakage around the door. For subsequent tests, the orientation of the door was changed to an inward swinging arrangement. Since the rising water inside the tank presses the door against the frame doorstop, a reduc-

tion in the water leakage rate occurred. The results of the outward swinging door tests (Test 1H and Test 2H) utilizing a plywood strip are shown in Figure 5.8(a) and the results of the inward swinging door tests (Test 3H and Test 4H) are shown in Figure 5.8(b).

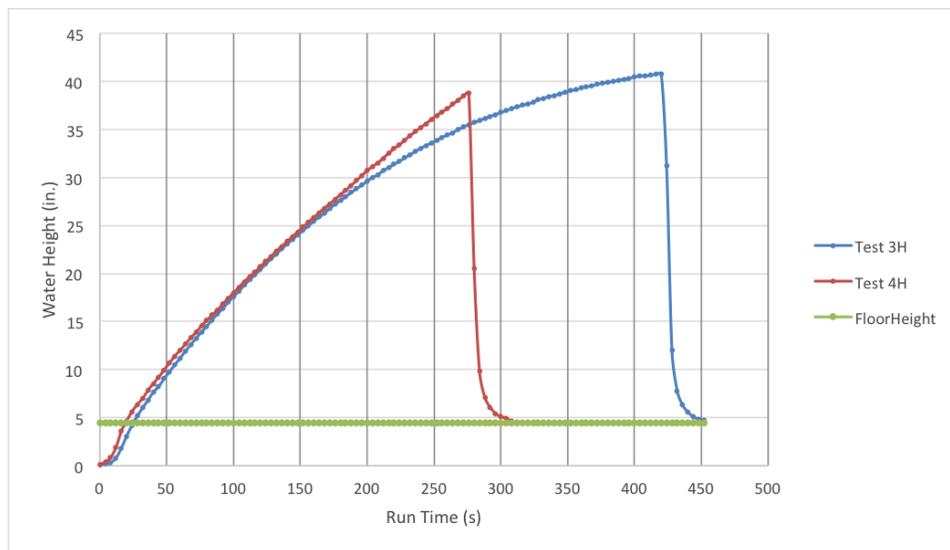
The hollow core door for Test 1H failed at a height of 42.3 in. after 372 seconds and the door for Test 2H failed at a height of 35.4 in. after 180 seconds. Both doors failed when the latch released from the strike plate and the door was forced open, allowing the water to flood from the PET. For the inward swinging door tests, the hollow core door for Test 3H failed at a height of 40.8 in. after 392 seconds and the door for Test 4H failed at a height of 38.9 in. after 256 seconds. Changing the orientation of the hollow core doors meant the doors failed catastrophically during testing, as seen in Figure 5.9. The hollow core doors broke laterally from the approximate height of the handle, with the bottom portion of the door completely disconnecting in Test 3H.

After modification of the PET, two additional inward opening hollow core door experiments were performed. These two experiments served as trials to allow acquaintance with the new equipment and capabilities and provided additional testing data that increased the range of flow rates.

The two experiments, identified as Test 5H and Test 6H, were run through the new 12 in. lines of the modified piping configuration. For Test 5H, the recirculation valve was positioned open and the valves directing the flow through the 12 in. lines were also aligned open; all other valves were closed. The only valve configuration change for Test 6H was the closure of the recirculation valve, which caused the flow through the upstream region to achieve 4,000 gpm. Both doors failed catastrophically during testing. The results of both experiments, showing



(a) Outward opening, Test1H and 2H.



(b) Inward opening, Test 3H and 4H.

Figure 5.8: Water depth vs. time plots for hollow core door tests.



Figure 5.9: Hollow core door at moment of failure in Test 3H (left) and Test 4H (right).

the water depths achieved inside the PET, are shown in Figure 5.10.

Test 5H failed at a height of 32.3 in. after 40 seconds and Test 6H failed at a height of 43.3 in. after 16 seconds. For hollow core door Test 6H, the closure of the pump recirculation valve resulted in a short experiment time due to the high flow rate and caused the door to rupture before the flow-meters reached steady-state. The importance of allowing the flow-meters to reach a steady-state before failure for data recording purposes was considered in subsequent experiments. A summary of the hollow core door results, including non-failure tests, is given in

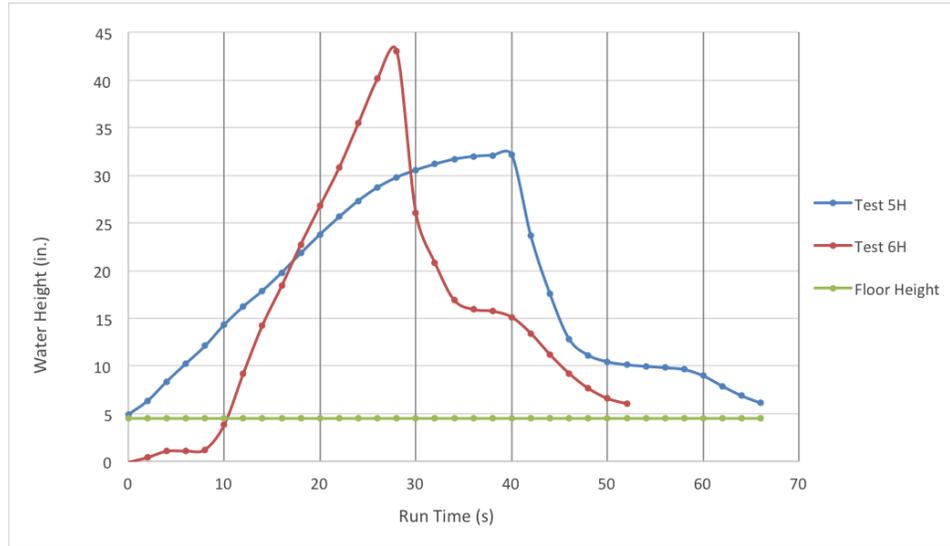


Figure 5.10: Water depth vs. time plot for inward opening hollow core door experiments, Test 5H and 6H.

Table 5.1.

Table 5.1: Summary of results for hollow core door experiments.

Test	Depth (in)	Flow Rate ($\frac{gal}{min}$)	Temp. (F)	Failure	Orientation
–	23.2	291.5	66	0	Outward
–	20.8	292.5	67	0	Outward
1H	42.3	292.5	66	1	Outward
–	21.1	297	68	0	Outward
–	24.2	294.5	67	0	Outward
2H	35.4	292.5	67	1	Outward
3H	40.8	291	68	1	Inward
4H	38.9	294	68	1	Inward
5H	32.3	1038	62	1	Inward
6H	43.3	837	63	1	Inward

Due to the design of the PET and the metal frame necessary for securely installing walls, the bottom of the doorway is raised several inches. A simple subtraction of the bottom frame height from the water depth gives the corrected failure depth of the door. Table 5.2 provides a summary of the hollow core door corrected

depth results.

Table 5.2: Summary of corrected depth results for hollow core door experiments.

Test	Depth (in)
–	18.7
–	16.3
1H	37.8
–	16.6
–	19.7
2H	31.9
3H	36.2
4H	34.3
5H	28.0
6H	39.0

5.3.2 Steel Door Experiments

Following the hollow core door experiments and modifications to the PET, water rise tests progressed to industrial steel doors. Additional doubled studs were used in the wall structure to ensure stability against greater volumes of water in the tank applying pressure to the wall. The aim of these experiments was to test the steel door to failure only and not the supporting wall structure. A complete wall installation is shown in Figure 5.11

The first three outward swinging steel door experiments were run using the following valve configuration: the recirculation valve was open; all 12 in. lines were open; the valves to the 8 in. and 6 in. lines were closed. The door was closed with the door latch, but the deadbolt was disengaged. Figure 5.12 shows the resulting water depths achieved for these three experiments. The calculated water flow rate into the PET was 1,148 gpm for Test 1S, 1,130 gpm for Test 2S, and 1,120 gpm for Test 3S.



Figure 5.11: Steel door and wall installed in the PET.

The steel door in Test 1S failed at a height of 46.1 in. In Test 2S and Test 3S, the door latch released at a similar time and water depth just above the height of the door handle at 39 in. and 37.1 in., respectively. In each of the three experiments, the door latch received enough force from the water to push the latch bolt inwards and release from the strike plate, allowing the door to open as seen in Figure 5.13.

An additional five outward swinging steel door tests were conducted with only the latch bolt engaged during testing. The 8 in. and 6 in. valves remained closed. The upstream 12 in. valve configuration for Test 4S was half-open and the valve configuration for Test 5S was three-quarters open. For the last three tests (6S, 7S, and 8S), the upstream 12 in. valve configuration was positioned one-quarter open.

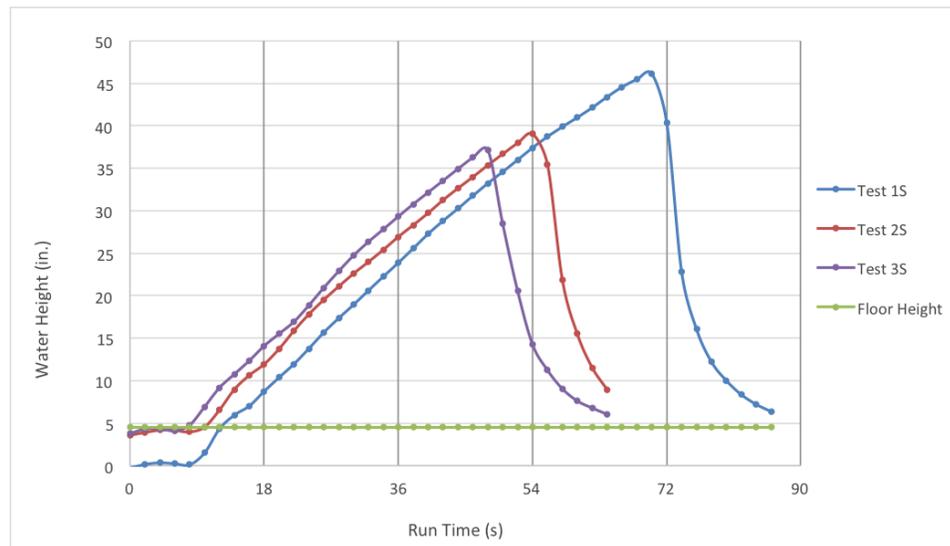


Figure 5.12: Water depth vs. time plot for outward opening steel door, Test 1S, 2S, and 3S.



Figure 5.13: Steel door at the moment of failure in Test 1S.

These valve adjustments varied the water flow entering the PET. The flow rates in were 979 gpm for Test 4S, 1,133 gpm for Test 5S, 604 gpm for Test 6S, 593 gpm for Test 7S, and 598 gpm for Test 8S. The water depths achieved before failure of Tests 4S and 5S are shown in Figure 5.14. Figure 5.15 shows the PET water depths achieved during Tests 6S through 8S.

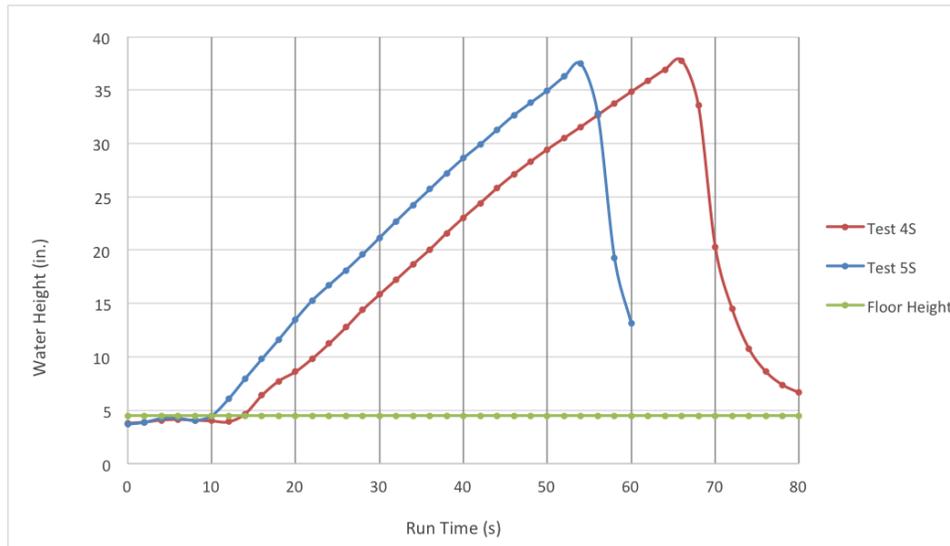


Figure 5.14: Water depth vs. time plot for outward opening steel door, Test 4S and Test 5S.

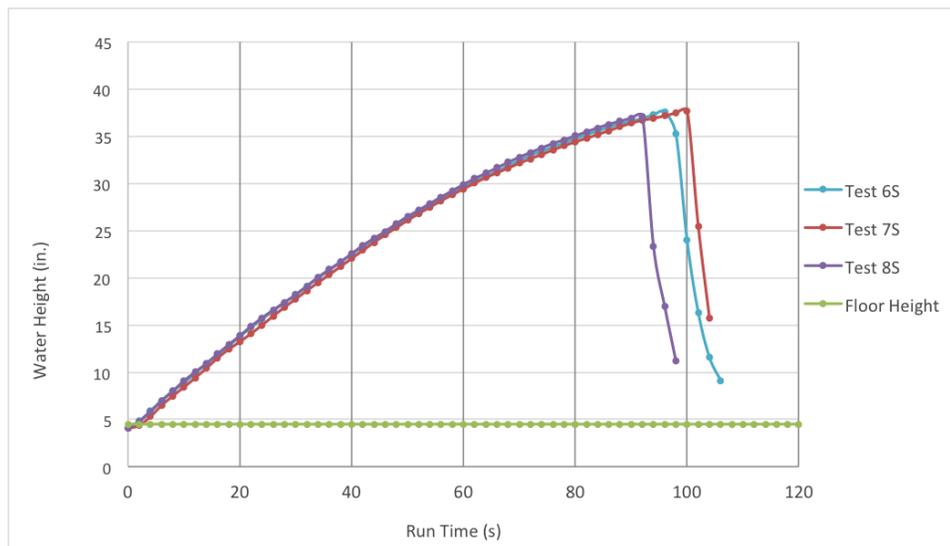


Figure 5.15: Water depth vs. time plot for outward opening steel door, Test 6S, 7S, and 8S.

The PET water depths at time of failure are similar for the last five tests; 37.8 in. for Test 4S, 37.5 in. for Test 5S, 37.6 in. for Test 6S, 37.7 in. for Test 7S, and 37.1 in. for Test 8S, regardless of the duration of experimentation. The difference in time

is simply the result of the varied flow rates used to fill the tank. By comparing results from these experiments, it is possible to conclude that outward swinging steel doors have constant water depth failure behavior when only the latch bolt is holding the door closed. The possible cause for higher water depths in the first two steel door tests is discussed later.

The next three tests (9S, 10S, and 11S) were performed with the deadbolt lock engaged. Test 9S was performed with the same constant valve configuration as Test 4S with a flow rate of 949 gpm, but unlike previous experiments, Tests 10S and 11S valve configurations were not held constant throughout the testing duration. In Test 10S, the recirculation valve and the downstream 12 in. valves were closed in steps until all water flow was directed into the PET. Test 11S opened the upstream 12 in. valve and closed the downstream 12 in. valve in steps during testing to again direct all water flow into the PET. The water depth achieved inside the PET can be seen in Figure 5.16 for Tests 9S, 10S, and 11S. Using the deadbolt produced a slight increase in the maximum depth before failure.

The step changes to the valves during Tests 10S and 11S are reflected in Figure 5.16. The decrease in water depth during Test 10S was the result of increased upstream flow increasing the water velocity in the pipe. This was an expected phenomenon due to the higher velocity water failing to separate into two streams at the tee junction. Net water flow into the PET momentarily decreased while the tank water continued to leak out around the door. PET water flow and filling was restored when the downstream section was closed.

The results of the three individual deadbolt tests demonstrated some interesting occurrences. Tests 9S and 10S experienced two failure stages, with each stage corresponding to the failure of the bolt. After the tank had filled to a sufficient

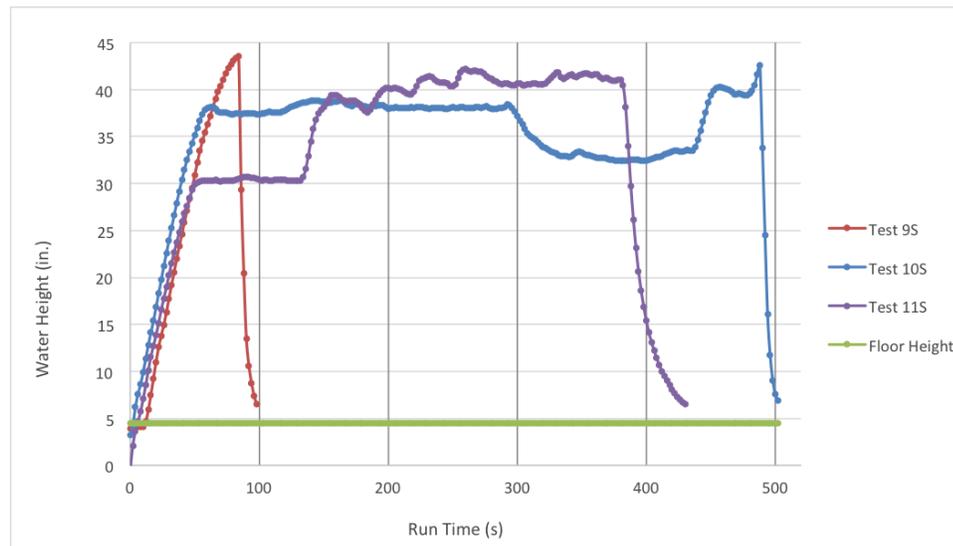


Figure 5.16: Water depth vs. time plot for outward opening steel door, Test 9S, 10S, and 11S.

depth, the latch bolt failure occurred with an accompanying audible pop. The door was then held closed only by the deadbolt until it failed from bending, as shown in Figure 5.17, allowing the door to be pushed open by the water.



Figure 5.17: Deadbolt failure from Test 10S.

While Test 11S was also performed with the door latched and deadbolt engaged, it is important to note that the deadbolt was not from the same manufac-

turer as used in the previous two experiments. It was an aftermarket deadbolt with a different internal assembly design. The original deadbolts are connected to a hardened metal rod that failed, allowing the deadbolt to bend out of the strike plate. The aftermarket deadbolt, however, connects to a metal U-shaped channel. The channel twisted marginally, but kept the deadbolt set in the strike plate.

As stated previously, the valve configuration for Test 11S was not held constant during the test. The water depth inside the PET initially stabilized at ~ 32 in. Upstream and downstream valves were adjusted and this led to the latch bolt failure observed before, increasing the leakage rate around the door while only the deadbolt held the door in place. With further valve changes over time, all water flow was eventually directed into the PET. While the deadbolt held for the entire test, the door suffered a permanent bend. The bend at the bottom of the door caused the flow into and out of the PET to equalize, and the test concluded. The final results were a failed latch bolt; a successful, but damaged deadbolt; and a bent steel door at the level of the handle, see Figure 5.18.



Figure 5.18: Test 11S steel door bending during testing (left) and post testing (right). The original position is indicated by a solid line and bent position by a dotted line.

Test 12S and Test 13S were performed without a deadbolt and new latch han-

dles. Test 12S had the same upstream valve configuration to Tests 6S, 7S, and 8S, but only the 8 in. valve was opened downstream. Test 13S was conducted through the upstream 8 in. line, with all other lines closed. These experiments served to confirm the greater depth achieved in Test 1S could be attributed to dry untested latches rather than an outlier anomaly. Test 12S and Test 13S results are shown in Figure 5.19.

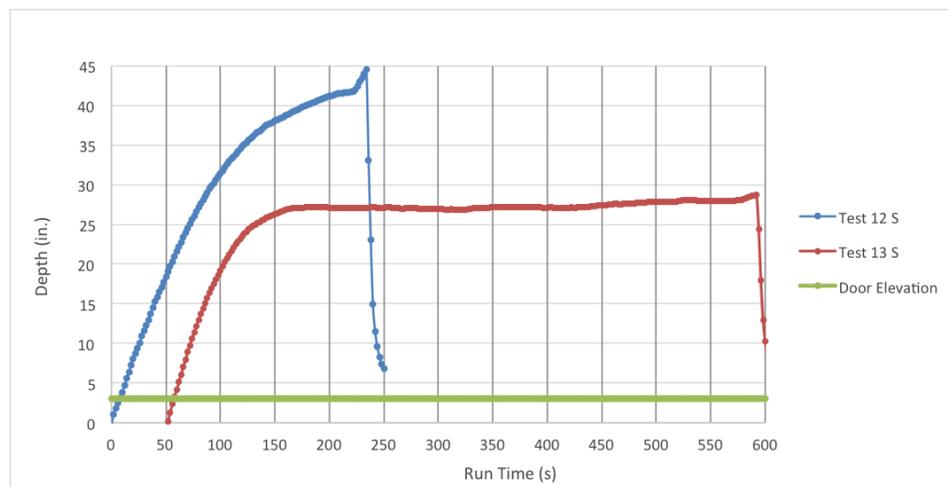


Figure 5.19: Water depth vs. time plot for outward opening steel door, Test 12S and Test 13S.

The results, shown in Figure 5.19, recorded a maximum depth of 44.5 in. for Test 12S and 41.4 in. for Test 13S; which are approximately 3.7% and 8.4% lower than Test 1S, but $\sim 15\%$ higher than depths achieved from Tests 2S through 8S. This demonstrates a slight degradation in the latch after initial testing with water, although further testing with future latches will be necessary to confirm this claim.

Having initially tested steel doors oriented to open outwards from the tank, the next test (14S) was performed with an unbolted inward opening steel door. Rising flood water in this type of experiment will push the door against the steel doorstops of the frame, preventing some bowing previously seen in experiments

and decreasing the leakage rate around the door frame, but increase the likelihood of damage to the door and its surrounding frame. Also, the force on the door latch and/or deadbolt alone will not produce a failed state. Therefore, greater water depths were expected to occur in the PET and a reinforced wall was designed and installed for testing. The test used the 12 in. lines only, with the recirculation valve positioned to half-open and the 12 in. upstream valve varied to control fill time. Figure 5.20 shows the door during testing along with resulting damage. Figure 5.21 shows the resulting water depths achieved for this experiment.



Figure 5.20: Inward swinging Test 14S at moment of failure (left) and post testing (left).

The steel door in Test 14S failed at a height of 83.5 in., when the water reached the top of the door. As expected, the change in door orientation significantly increased the water depth and damaged the door. Further tests will be necessary to fully characterize how inward opening steel doors fail. A compiled summary of the steel door results, including non-failure tests, are given in Table 5.3 for non-deadbolt doors and Table 5.4 for deadbolt doors. Depths corrected for the raised

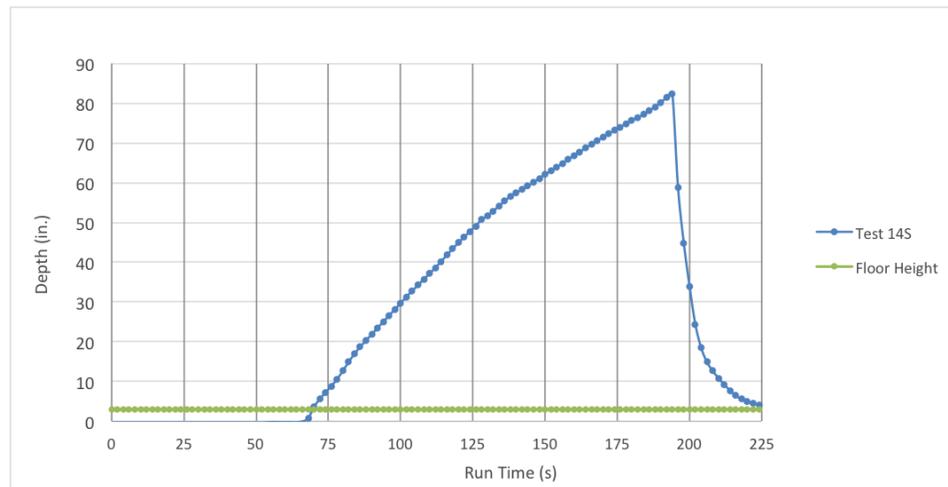


Figure 5.21: Water depth vs. time plot for inward opening steel door, Test 14S.

frame for all experiments are summarized in Table 5.5.

5.4 Future Experiments

Further component experiments can be conducted in the PET. To have accurate and improved modeling, sizable data sets are required, including non-failure information. Additional door experiments with the deadbolt engaged are necessary to expand the currently limited data set. Also, having only tested one inward opening steel door to failure, additional experiments on inward opening steel doors would also be needed to generate data. Using calculated leakage rates from previous tests, a step approach will be used to adjust the incoming water flow until a failure state is reached, populating the data set. This step process could be conducted for both deadbolt and inward opening testing scenarios.

Future experimentation using the PET could also subject a wall containing feedthroughs to water incursion. Design specifications for feedthroughs used in NPPs have already been acquired. These experiments would be the first experiments to test several components at the same time. All feedthroughs would be

Table 5.3: Summary of results for steel doors experiments, non-deadbolt.

Test	Depth (in)	Flow Rate ($\frac{gal}{min}$)	Temp. (F)	Notes
1S	46.1	1148	67.4	
2S	39.0	1130	63.3	
3S	37.1	1120	63.1	
4S	37.8	979	63.0	
5S	37.5	1133	63.0	
6S	37.6	604	63.0	
7S	37.7	593	63.0	
8S	37.1	598	63.1	
12S	44.5	975	64.0	
–	25.7	248	61.6	Non-Failure
–	17.0	117	59.0	Non-Failure
–	27.4	285	59.3	Non-Failure
–	30.9	397	59.4	Non-Failure
–	32.3	484	59.6	Non-Failure
–	24.3	247	60.2	Non-Failure
–	34.8	593	60.7	Non-Failure
–	37.5	696	61.0	Non-Failure
–	38.0	734	61.2	Non-Failure
13S	41.4	1025*	61.3	
14S	83.5	1025*	61.3	Inward swinging

* Flow rate varied

Table 5.4: Summary of results for steel doors experiments, deadbolt.

Test	Depth (in)	Flow Rate ($\frac{gal}{min}$)	Temp. (F)	Notes
9S	43.6	949	63.2	
10S	42.6	2790*	63.3	
11S	42.2	3858*	63.3	Non-Failure

* Flow rate varied

packed according to the standards and regulations necessary to create a fire barrier between areas in a NPP.

Table 5.5: Summary of corrected depth results for steel door experiments.

Test	Depth (in)
1S	42.1
2S	35.0
3S	33.5
4S	33.9
5S	33.9
6S	33.5
7S	33.5
8S	33.1
9S	39.4
10S	39.4
11S	39.4
12S	41.5
–	22.7
–	14.0
–	24.4
–	27.9
–	29.3
–	21.3
–	31.8
–	34.5
–	35.0
13S	38.6
14S	80.5

Chapter 6

Developing a Model to Assess Component Fragility

Having conducted the flooding experiments and collected observational data on door failures, development proceeded on stochastic models that analyzed the fragility of components using explanatory variables. An explanatory variable is a type of independent variable that is possibly predictive of a component's fragility in a regression analysis. As part of this development process, Figure 6.1 was created as a visual roadmap to the avenues of research that were taken over the course of the work. The illustration begins with the door data. The following sections of this chapter provide details on each branch and findings for developing component fragility models dependent on explanatory variables. After checking the regression models, a model selection is made. The final avenue of the roadmap is applying the component fragility model to a flooding simulation, covered in the next chapter. Places where a branch ends are potential areas for future research and development, but fall outside of the scope of this work. Lastly, OpenBUGS scripts and summary result tables of the posterior distributions for models discussed below are provided in Appendix C.

6.1 Generalized Linear Regression Model

6.1.1 Background Analysis with Binomial Distribution

This section contains a summarized background on the binomial model development work conducted alongside research colleague Sneha Suresh [30].

Using the example Bayesian regression analysis model presented in Chapter 4, a flooding fragility model was developed using the then current data collected

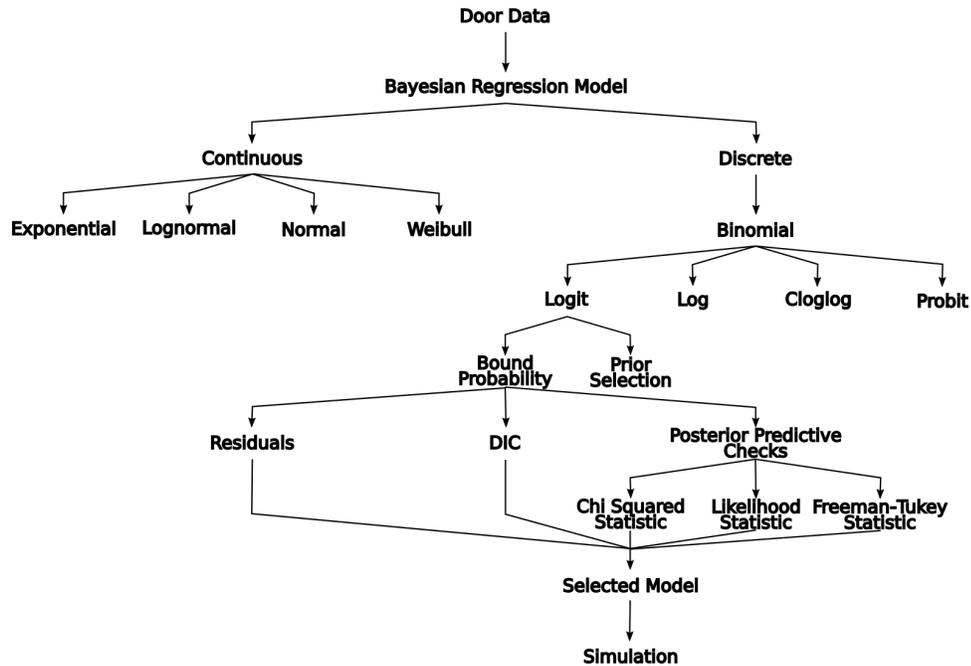


Figure 6.1: A visual roadmap to the model development process.

on hollow core door failure (Table 5.1) based on the assumption that the water's depth, flow rate, and temperature are potential variables to affect the door fragility. This initial mathematical modeling uses the discrete binomial distribution to represent failure of a door installed in the PET during a rising water flood event. This is a commonly used model for failure on demand with key parameters p , the probability of failure on demand, and trials $n = 1$ (only a single door is potentially challenged during testing). Following the preliminary example and since parameter p represents a probability, it is constrained between 0 and 1 with the logit function, defined in Equation 4.1.

The fragility regression model examined seven possibilities, with each of the variables alone driving the model to failure, a combination of two variables driving the model to failure, and a combination of all three variables driving the model to failure. These regression models are:

$$\text{logit}(p) = \text{intercept} + aD \quad (6.1)$$

$$\text{logit}(p) = \text{intercept} + bF \quad (6.2)$$

$$\text{logit}(p) = \text{intercept} + cT \quad (6.3)$$

$$\text{logit}(p) = \text{intercept} + aD + bF \quad (6.4)$$

$$\text{logit}(p) = \text{intercept} + aD + cT \quad (6.5)$$

$$\text{logit}(p) = \text{intercept} + bF + cT \quad (6.6)$$

$$\text{logit}(p) = \text{intercept} + aD + bF + cT \quad (6.7)$$

where a , b , and c are the coefficients of the covariate parameters represented as D , F , and T for depth, flow rate, and temperature respectively.

A script for OpenBUGS was written for the above seven equations. Due possibly to the small variation in the flow rates and water temperatures between tests, the outcome of failure is random with respects to these variables. Therefore, there is no predictive capability on flow rate and temperature and the model had to be reduced to the depth variable alone to run when the logit function was used. This script is shown in Table 6.1.

Table 6.1: OpenBUGS Script for binomial model using depth variable.

```

#Depth (D) Model
model {
  for(j in 1:tests) {
    failure[j] ~ dbin(p[j], num.tested)
    # Regression model
    logit(p[j]) <- int + a*depth[j]
    #failure.rep[j] ~ dbin(p[j], num.tested) # Replicate values for model validation
    #diff.obs[j] <- pow(failure[j] - num.tested*p[j], 2)/(num.tested*p[j]*(1-p[j]))
    #diff.rep[j] <- pow(failure.rep[j] - num.tested*p[j], 2)/(num.tested*p[j]*(1-p[j]))
  }
  #chisq.obs <- sum(diff.obs[])
  #chisq.rep <- sum(diff.rep[])
  #p.value <- step(chisq.rep - chisq.obs)
  # Prior distributions
  int ~ dnorm(0, 0.0001)
  a ~ dnorm(0, 0.001)
}
data
list(num.tested=1, tests=8, depth = c(23.23,20.75,42.3,21.05,24.22,35.41,40.76,38.85), failure = c(0,0,1,0,0,1,1,1))
inits
list(int=0, a=0)

```

The results, the depth coefficient and intercept posteriors summaries, are given in Table 6.2. The Bayesian p-value was unavailable, possibly due to the number of samples obtained being insufficient for the calculation.

Table 6.2: Results for Binomial model with logit function using the depth variable.

Parameter	Mean	Standard Dev.	97.5% Interval
intercept	-126.3	64.8	(-272.9, -26.74)
a (depth coeff.)	4.39	2.32	(0.91, 9.73)
Bayesian p-value	–	–	–

If the necessary coefficient posterior distributions are estimated, the failure probability for the hollow core door can be calculated. Equation 6.7, for example, would give the probability of failure for the case of the all three explanatory variables as:

$$p = \frac{1}{e^{-(intercept+aD+bF+cT)} + 1} \quad (6.8)$$

which is an "S"-shaped sigmoid function and a fragility curve could be plotted.

Link Functions

Since no conclusive results were initially obtained using the logit function, other available link functions that could provide complete results were researched. The available link functions that are supported by OpenBUGS are log, logit, cloglog, and probit [24]. They are defined as:

- $\log(p)$: natural logarithm of p
- $\text{logit}(p) = \ln\left(\frac{p}{1-p}\right)$
- $\text{cloglog}(p)$: complementary log log of p , $\ln(-\ln(1-p))$
- $\text{probit}(p)$: inverse of standard normal cdf $\phi(p)$

For the binomial distribution specifically, alternatives to logit are the complementary log-log (cloglog) and probit. Scripts using the logit function encountered errors running the combined flow rate and depth model, the combined depth and temperature model, and the model using all three parameters. After verifying that all seven models ran using the alternative link functions, cloglog was chosen for further modeling.

A means of comparing models, since the Bayesian p-value was unavailable initially, is the DIC discussed in Chapter 3. The rewritten cloglog scripts were run using 100,000 samples, skipping the first 1,000 samples, and after the first update the DIC monitor was set and a second update run (see Appendix C). According to

the summarized results, shown in Table 6.3, the best fit model is the last column with all three parameters driving the model to failure. It is also noted that the models that incorporated water depth as a variable have significantly lower DICs. Using the DIC comparison guide, the flow, temperature, and combined flow and temperature models could be ruled out while the remaining models would need validation metrics.

Table 6.3: Summary of coefficient estimates and DIC for Binomial model using cloglog function for hollow core doors.

Parameter	D	F	T	D,F	D,T	F,T	D,F,T
intercept	-129	8.69	-13.09	0.75	-4.83	29.81	3.97
^a (depth coeff.)	4.46	n/a	n/a	40.55	37.39	n/a	42.57
^b (flow rate coeff)	n/a	-0.03	n/a	-4.13	n/a	-0.23	-5.18
^c (temp. coeff.)	n/a	n/a	0.19	n/a	-16.53	0.54	3.61
DIC	0.34	12.58	12.92	0.028	0.026	11.01	0.023

6.1.2 Alternative Continuous Distribution Models

The initial mathematical modeling discussed in the background section used a binomial regression model to represent failure of a door installed in the PET during a raising water flood event. While trying to resolve some shortcomings of the OpenBUGS script use of the logit function, it was found that the binomial model was not running the complete number of samples assigned when a single component was put on demand ($n = 1$). While only one door was challenged during PET experiments, the logit script successfully ran all samples when n was any value greater than one ($n > 1$). Since the binomial model was not executing when a single door was put on demand, alternative models were investigated. The binomial

might still be a useful aleatory model when the number of failure events is a positive integer in the interval $[0, n + 1]$ or the script can be altered to accommodate the single trial case. The following alternative distributions are continuous and have one or two parameters. In some cases, either of these parameters can be turned into a regression model to represent different test environments.

Analysis with Exponential Distribution

Re-evaluating how demand was put on a door over a period of time while the water raised inside the tank, the model interpretation could take a specified time period into account. The observed data will then be in the form of door failure times in addition to depth, flow rate, and temperature. The failed hollow core door data, including time to failure, is given in Table 6.4.

Table 6.4: Data on inward swinging failures for hollow core doors.

Time (min)	Flow Rate ($\frac{gal}{min}$)	Depth (in)	Temperature (F)
3.533	290.6	40.74	69.63
2.667	294.07	38.77	69.02
0.8667	1038	32.28	62.18
0.4667	837	43.34	63.49

The simplest aleatory model for such data, when time is a random variable of interest, is an exponential distribution with unknown parameter λ . Before applying the possible influencing variables to the model, a standard exponential script was run using only the door failure times given in the first column of Table 6.4. The script for the exponential aleatory model is shown in Table 6.5 with Cramer-von Mises statistics for calculating the Bayesian p-value.

The parameter λ has a specified prior distribution that represents the possible state of knowledge about values of λ based on past or vendor provided informa-

Table 6.5: OpenBUGS script for exponential aleatory model for door failure with Jeffreys prior.

```

#Exponential Model with No Environment; P-Value

model {
  for(i in 1:N) {
    time[i] ~ dexp(lambda)
    time.rep[i] ~ dexp(lambda)
    #Rank observed times a plicate times
    time.ranked[i]<- ranked(time[,i])
    time.rep.ranked[i]<-ranked(time.rep[,i])
    #Calculate components of Cramer-von Mises statistic for observed and replicated data
    F.obs[i]<-cumulative(time[i], time.ranked[i])
    F.rep[i]<-cumulative(time.rep[i], time.rep.ranked[i])
    diff.obs[i]<-pow(F.obs[i] - (2*i-1)/(2*N),2)
    diff.rep[i]<-pow(F.rep[i] - (2*i-1)/(2*N),2)
  }
  #Calculate distribution of Cramer-von Mises statistic for observed and replicated data
  CVM.obs<-sum(diff.obs[])
  CVM.rep<-sum(diff.rep[])
  p.value<-step(CVM.rep-CVM.obs)
  # Prior distributions
  #lambda ~ dnorm(0, 0.0001)
  lambda ~ dgamma(0.0001, 0.0001) #Jeffreys prior for lambda
}}
data
list(time =c(3.533, 2.667, 0.8667, 0.4667), N=4)
inits
list(lambda=1)

```

tion. Initially a Jeffreys non-informative prior was used in the script. For exponential data, the Jeffreys prior is like a gamma distribution with both parameters equal to zero. This is entered into OpenBUGS as $dgamma(0.0001, 0.0001)$ and an initial value must be provided for λ . Table 6.6 shows the results using the Jeffreys prior distribution.

Table 6.6: Results for standard exponential script with Jeffreys prior.

Parameter	Mean	Standard Dev.	97.5% Interval
$\lambda (\frac{1}{min})$	0.53	0.27	(0.15, 1.16)
Bayesian p-value	0.63	n/a	n/a
DIC	15.13	n/a	n/a

The prior distribution was then changed in the standard model script to a diffuse normal prior, as used in the previous binomial regression model. Running the script in Table 6.5 with the normal prior distribution $\text{dnorm}(0, 0.0001)$ gives the results found in Table 6.7. Checking and comparing both the p-value and DIC, the fitting capabilities of the model are improved using the normal prior. This comparison also illustrates the possible influences prior distribution selection has on any given model. The change in λ could mean there is unexpected sensitivity to the prior selection even though they are non-informative. This would need to be addressed in a sensitivity analysis if chosen as the developed model.

Table 6.7: Result of standard exponential script with diffuse normal prior.

Parameter	Mean	Standard Dev.	97.5% Interval
$\lambda (\frac{1}{min})$	0.66	0.296	(0.21, 1.36)
Bayesian p-value	0.6	n/a	n/a
DIC	14.93	n/a	n/a

Further improvement could be made to the exponential model by including additional information, such as the environmental conditions. To determine what observable phenomena drive failure, λ is not specified by a prior distribution, but turned into its own model. The seven potential explanatory regression models are:

$$\lambda = \text{intercept} + aD \quad (6.9)$$

$$\lambda = \text{intercept} + bF \quad (6.10)$$

$$\lambda = \text{intercept} + cT \quad (6.11)$$

$$\lambda = \text{intercept} + aD + bF \quad (6.12)$$

$$\lambda = \text{intercept} + aD + cT \quad (6.13)$$

$$\lambda = \text{intercept} + bF + cT \quad (6.14)$$

$$\lambda = \text{intercept} + aD + bF + cT \quad (6.15)$$

The OpenBUGS script for all possible regression equations is shown in Table 6.8 and includes the Bayesian model check, p-value. Diffuse normal priors were used for the coefficient prior distributions. It should be noted that while initial values were provided, the chain still contained uninitialized variables, and OpenBUGS was allowed to generate these values.

The model was analyzed with 1,000 burn-in iterations, followed by 10,000,000 iterations for parameter estimation. The code was run with the same number of samples as outlined with the binomial model background. The results for each coefficient, in addition to p-value and DIC, are summarized in Table 6.9 with full summaries found in Appendix C.

The Bayesian p-value for the depth model (0.47) indicates a better fit than the standard exponential model (0.6) and is close to the ideal 0.5 value. Taking into consideration the environment at failure improved the model without over-fitting the observed data. However, with calculated Bayesian p-values of 0.86, 0.34, 0.04, 0.37, 4.02E-5, and 8.54E-5 for the other respective six model equations far enough from the ideal value, it is worth looking at different aleatory models. Changing the

Table 6.8: OpenBUGS script for exponential aleatory model with Bayesian p-value calculation.

```

#Hollow Core Data
#Exponential Model Using Flow Rate (F), Depth (D), and Temperature (T); P-Value

model {
  for(i in 1:N) {
    time[i] ~ dexp(lambda[i])
    time.rep[i] ~ dexp(lambda[i])
    #lambda[i] <- int + a*D[i]
    lambda[i] <- int + b*F[i]
    #lambda[i] <- int + c*T[i]
    #lambda[i] <- int + a*D[i] +b*F[i]
    #lambda[i] <- int + a*D[i] +c*T[i]
    #lambda[i] <- int + b*F[i] +c*T[i]
    #lambda[i] <- int + a*D[i] +b*F[i] +c*T[i]
    #Rank observed times a plicate times
    time.ranked[i]<- ranked(time[],i)
    time.rep.ranked[i]<-ranked(time.rep[],i)
    #Calculate components of Cramer-von Mises stastic for observed and replicated data
    F.obs[i]<-cumulative(time[i], time.ranked[i])
    F.rep[i]<-cumulative(time.rep[i], time.rep.ranked[i])
    diff.obs[i]<-pow(F.obs[i] - (2*i-1)/(2*N),2)
    diff.rep[i]<-pow(F.rep[i] - (2*i-1)/(2*N),2)
  }
  #Calculate distribution of Cramer-von Mises statistic for observed and replicated data
  CVM.obs<-sum(diff.obs[])
  CVM.rep<-sum(diff.rep[])
  p.value<-step(CVM.rep-CVM.obs)
  # Prior distributions
  int ~ dnorm(0, 0.0001)
  a ~ dnorm(0, 0.0001)
  b ~ dnorm(0, 0.0001)
  c ~ dnorm(0, 0.0001)
}
data
list(time =c(3.533, 2.667, 0.8667, 0.4667), N=4,
F =c(290.6, 294.07, 411.25, 472.89), D =c(40.74, 38.77, 32.28, 43.34), T =c(69.63, 69.02, 62.18, 63.49))
inits
list(int=1,a=1,b=1,c=1)

```

model could give better predictive ability and there are several alternatives to the exponential model.

Analysis with Lognormal Distribution

A lognormal aleatory model has two unknown parameters, commonly denoted by μ and σ . The shape of the distribution is fixed with the median determined by μ

Table 6.9: Summary of coefficient estimates, p-value, and DIC for exponential model for hollow core doors.

Parameter	D	F	T	D,F	D,T	F,T	D,F,T
intercept	2.49	-42.92	3.55	-16.44	17.09	-6.16	-12.32
^a (depth coeff.)	-0.04	n/a	n/a	-0.74	0.21	n/a	-0.99
^b (flow rate coeff.)	n/a	0.15	n/a	0.16	n/a	0.49	0.29
^c (temp. coeff.)	n/a	n/a	-0.04	n/a	-0.36	-1.97	-0.44
p-value	0.47	0.86	0.34	0.04	0.37	4.02E-5	8.54E-5
DIC	15.61	41.02	13.12	56.35	13.89	213.8	115

and the other parameter, σ , determines the spread of the distribution. This model could possibly be improved by including the additional testing environment information. The parameter μ will capture these influences on the results.

The example OpenBUGS script shown in Table 6.10 implements a lognormal aleatory model, $time[i] \sim \text{dlnorm}(\mu[i], \tau)$, with diffuse normal priors on the parameters of the μ regression model. Note that τ is equal to $\frac{1}{\text{variance}}$ or $\frac{1}{\sigma^2}$. The model was analyzed with 1,000 burn-in iterations, followed by 10,000,000 iterations. The posterior distributions of the parameters are summarized, along with Bayesian p-value and DIC, in Table 6.11.

From the lognormal consideration, all of the regression models perform reasonably well. The combined flow rate and water temperature model under-predicted by approximately the same margin that flow rate over-predicted. The flow rate model has the significantly smaller DIC in this consideration, 1.69 versus 10.79 and greater, so it would be selected under this comparison criterion.

Table 6.10: OpenBUGS script for lognormal model.

```

#Hollow Core Data
#Lognormal Model Using Flow Rate (F), Depth (D), and Temperature (T); P-Value
model {
  for(i in 1:N) {
    time[i] ~ dlnorm(mu[i], tau)
    time.rep[i] ~ dlnorm(mu[i], tau)
    #mu[i] <- int + a*D[i]
    #mu[i] <- int + b*F[i]
    mu[i] <- int + c*T[i]
    #mu[i] <- int + a*D[i] + b*F[i]
    #mu[i] <- int + a*D[i] + c*T[i]
    #mu[i] <- int + b*F[i] + c*T[i]
    #mu[i] <- int + a*D[i] + b*F[i] + c*T[i]
    #Rank observed times a plicate times
    time.ranked[i] <- ranked(time[],i)
    time.rep.ranked[i] <- ranked(time.rep[],i)
    #Calculate components of Cramer-von Mises statistic for observed and replicated data
    F.obs[i] <- cumulative(time[i], time.ranked[i])
    F.rep[i] <- cumulative(time.rep[i], time.rep.ranked[i])
    diff.obs[i] <- pow(F.obs[i] - (2*i-1)/(2*N),2)
    diff.rep[i] <- pow(F.rep[i] - (2*i-1)/(2*N),2)
  }
  #Calculate distribution of Cramer-von Mises statistic for observed and replicated data
  CVM.obs <- sum(diff.obs[])
  CVM.rep <- sum(diff.rep[])
  p.value <- step(CVM.rep-CVM.obs)
  tau <- pow(sigma, -2)
  sigma ~ dunif(0,10)
  # Prior distributions
  int ~ dnorm(0, 0.0001)
  a ~ dnorm(0, 0.0001)
  b ~ dnorm(0, 0.0001)
  c ~ dnorm(0, 0.0001)
}
data
list(time =c(3.533, 2.667, 0.8667, 0.4667), N=4,
F =c(290.6, 294.07, 411.25, 472.89), D =c(40.74, 38.77, 32.28, 43.34), T =c(69.63, 69.02, 62.18, 63.49))
inits
list(int=1,a=1,b=1,c=1)

```

Analysis with Normal Distribution

Like the lognormal distribution in the previous section, the normal distribution is characterized by the same two parameters, μ and σ . Therefore, the OpenBUGS script will be similar to Table 6.10 with the influencing variables applied to μ , but the aleatory model is input as $time[i] \sim dnorm(\mu[i], \tau)$. Table 6.12 summarizes the

Table 6.11: Summary of coefficient estimates, p-value, and DIC for lognormal model for hollow core doors.

Parameter	D	F	T	D,F	D,T	F,T	D,F,T
intercept	0.67	4.22	-15.33	4.16	-12.3	0.85	10.81
^a (depth coeff.)	-8.57E-3	n/a	n/a	2.28E-3	-0.08	n/a	0.04
^b (flow rate coeff)	n/a	-0.01	n/a	-0.01	n/a	-8.73E-3	-0.01
^c (temp. coeff.)	n/a	n/a	0.24	n/a	0.24	0.04	-0.10
p-value	0.33	0.55	0.45	0.42	0.41	0.46	0.44
DIC	20.25	1.69	13.43	12.37	12.6	10.79	11.92

results from running the normal distribution script.

Table 6.12: Summary of coefficient estimates, p-value, and DIC for normal model for hollow core doors.

Parameter	D	F	T	D,F	D,T	F,T	D,F,T
intercept	0.34	7.46	-20.98	4.43	-20.77	-2.10	5.73
^a (depth coeff.)	0.04	n/a	n/a	0.08	-0.08	n/a	0.07
^b (flow rate coeff)	n/a	-0.02	n/a	-0.02	n/a	-0.01	-0.016
^c (temp. coeff.)	n/a	n/a	0.35	n/a	0.39	0.12	-0.012
p-value	0.33	0.45	0.41	0.41	0.41	0.45	0.42
DIC	20.76	11.11	11.83	16.12	15.73	14.08	14.22

The Bayesian p-values are all reasonable choices, with all models under-predicting the failure time. From this consideration, the flow rate model and the combined flow rate and water temperature models are closest to 0.5. Next looking at the calculated DIC for comparison, the flow rate model also has the smaller DIC, 11.11 vs 14.08, so it would be selected under this criterion.

Analysis with Weibull Distribution

The Weibull distribution is another alternative aleatory model for random durations, with two unknown parameters denoted as β and α . Unlike the lognormal or normal distribution, where the shape of the distribution is fixed, β is a shape parameter. If β equals 1, the Weibull distribution reduces to the exponential distribution. The second parameter, α , is a scale parameter. OpenBUGS uses a slightly different parameterization with the scale parameter as $\lambda = \alpha^{-\beta}$.

Allowing either parameter to be turned into a model and applying the influencing environment data could potentially improve upon this model. Therefore, two scripts for the Weibull distribution were created which alternated between one parameter capturing the influence of testing information and the other parameter being given a prior distribution and monitored by OpenBUGS. The example script shown in Table 6.13 sets β as its own model, $\beta[i]$, and a prior to λ . Note that to run the code, the prior for the monitored Weibull parameter was set to a Jeffreys non-informative prior. The results of Table 6.13 script are given in Table 6.14, and the results from the alternate script that switched the roles of β and λ are given in Table 6.15.

For both Weibull considerations, the water depth model would be selected, with the environment impact on λ returning closer p-value of 0.46 to the ideal target. It is also of interest that the β value of this model is 1.03, within rounding of 1 where the Weibull reduces to the exponential distribution. Depth was also the best-fit model for the exponential distribution.

Table 6.13: OpenBUGS script of a Weibull distribution with influence on the beta parameter.

```

#Hollow Core Door Data
#Weibull Model (Beta) Using Flow Rate (F), Depth (D), and Temperature (T); P-Value
model {
  for(i in 1:N) {
    time[i] ~ dweib(beta[i], lambda)
    time.rep[i] ~ dweib(beta[i], lambda)
    #beta[i] <- int + a*D[i]
    beta[i] <- int + b*F[i]
    #beta[i] <- int + c*T[i]
    #beta[i] <- int + a*D[i] + b*F[i]
    #beta[i] <- int + a*D[i] + c*T[i]
    #beta[i] <- int + b*F[i] + c*T[i]
    #beta[i] <- int + a*D[i] + b*F[i] + c*T[i]
    #Rank observed times and duplicate times
    time.ranked[i] <- ranked(time[],i)
    time.rep.ranked[i] <- ranked(time.rep[],i)
    #Calculate components of Cramer-von Mises statistic for observed and replicated data
    F.obs[i] <- cumulative(time[i], time.ranked[i])
    F.rep[i] <- cumulative(time.rep[i], time.rep.ranked[i])
    diff.obs[i] <- pow(F.obs[i] - (2*i-1)/(2*N),2)
    diff.rep[i] <- pow(F.rep[i] - (2*i-1)/(2*N),2)
  }
  #Calculate distribution of Cramer-von Mises statistic for observed and replicated data
  CVM.obs <- sum(diff.obs[])
  CVM.rep <- sum(diff.rep[])
  p.value <- step(CVM.rep-CVM.obs)
  # Prior distributions
  lambda ~ dgamma(0.0001, 0.0001)
  int ~ dnorm(0, 0.0001)
  a ~ dnorm(0, 0.0001)
  b ~ dnorm(0, 0.0001)
  c ~ dnorm(0, 0.0001)
}
data
list(time =c(3.533, 2.667, 0.8667, 0.4667), N=4,
F =c(290.6, 294.07, 411.25, 472.89), D =c(40.74, 38.77, 32.28, 43.34), T =c(69.63, 69.02, 62.18, 63.49))
inits
list(lambda=1,int=1,a=1,b=1,c=1)

```

Summary of Alternative Models

The following Bayesian p-values were closest to the ideal value, usually under-predicting the failure time, for each distribution: exponential, 0.47; lognormal, 0.55; normal, 0.45; Weibull(β), 0.34; Weibull(λ), 0.46. These are all reasonable choices that divide between including water depth information (exponential,

Table 6.14: Summary of coefficient estimates, p-value, and DIC for Weibull model with environment impact on beta parameter for hollow core door.

Parameter	D	F	T	D,F	D,T	F,T	D,F,T
intercept	10.24	-79.1	-6.38	-68.59	-1.13	-94.76	-102.1
^a (depth coeff.)	-0.21	n/a	n/a	-0.44	-0.096	n/a	-0.84
^b (flow rate coeff)	n/a	0.28	n/a	0.3	n/a	0.26	0.295
^c (temp. coeff.)	n/a	n/a	0.13	n/a	0.11	0.31	0.74
$\lambda (\frac{1}{min})$	0.24	0.57	0.22	0.40	0.22	0.61	0.32
p-value	0.34	0.29	0.38	0.18	0.27	0.28	0.15
DIC	15.43	63.56	16.09	64.05	15.73	49.37	49.38

Table 6.15: Summary of coefficient estimates, p-value, and DIC for Weibull model with environment impact on lambda parameter for hollow core door.

Parameter	D	F	T	D,F	D,T	F,T	D,F,T
intercept	1.986	-44.93	2.07	-52.66	2.78	-47.73	-4.26
^a (depth coeff.)	-0.03	n/a	n/a	-0.79	-0.02	n/a	-1.44
^b (flow rate coeff)	n/a	0.15	n/a	0.29	n/a	0.39	0.38
^c (temp. coeff.)	n/a	n/a	-0.02	n/a	-0.01	-0.94	-0.697
β	1.03	2.38	1.34	1.82	1.155	2.22	1.67
p-value	0.46	0.98	0.36	2.36E-4	0.35	8.98E-3	5.03E-6
DIC	17.27	36.67	13.89	104.0	13.71	123.6	166.6

Weibull) and water flow rate information (lognormal, normal) in the modeling. The flow rate model returned the significantly smallest DIC criterion for all models with the lognormal distribution, justifying the additional complexity of distributions with more than one parameter and the inclusion of additional experimental information. Factoring in the environment at failure has also shown to enhance a model's reliability predictions.

Using a continuous aleatory model is a fertile area for future dissertation research. If work is continued looking into continuous distribution models as an alternative model development path, a next step would be to include censored data. A data point is a censored observation when the exact value is not known, but some knowledge is known about possible values (e.g. lies above a point c or within a specific interval). The most common application is in survival analysis. The non-failure data collected during experiments should be included in the continuous distribution models as censored data. This would mean a larger data set which is more ideal for model fitting.

For the next step of model development, working with several models highlighted the importance of prior distribution selection and its importance in developing a binomial regression model. Priors distributions play an important role in Bayesian analysis and could have an impact on a model's ability to run without error. Taking this knowledge, the next phase of development returned to the binomial distribution.

6.1.3 Returning to the Binomial Distribution

The Special Case of $n=1$

The initial shortcomings of using the binomial distribution was its apparent inability to consistently run all possible logistic regression models for a single component ($n = 1$). While changing the link function provided a workaround, the cause of run error remained unknown. Work on alternative model distributions, presented in the Section 6.1.2, highlighted the possible importance of prior distribution selection to this problem and a possible solution approach.

While the logit function should transform the parameter p onto an appropriate

scale, in this case probability of failure is between 0 and 1, in practice this was not always occurring. Periodically, illogical or extreme values are being selected by the sampler from the prior distribution. This can cause errors such as numerical overflow or, within the logistic regression, results in negative parameter values which cannot be log transformed. The improper value prompted a binomial calculation that OpenBUGS is unable to perform, causing the run to crash. One possible solution would be priors that are only distributed or truncated over positive values.

To test the influence positive priors have on the model execution, a simple binomial model with only failure data and a water depth regression model were run using gamma distributions ($d\text{gamma}(\alpha, \beta)$), uniform distributions ($d\text{unif}(a, b)$), and beta distributions ($d\text{beta}(\alpha, \beta)$) with varying parameter values for each distribution (see Appendix C). At this time both hollow door and steel door data was available for these investigatory trials. A flat distribution ($d\text{flat}()$), a non-proper distribution available in OpenBUGS, was also tested, but was unable to produce a complete run. From the options above, the uniform distribution was selected for further testing. It allows precise cutoff options for comparing different spans of values and gave reasonable p-values for the regression models.

The uniform distribution looked at the positive diffuse priors such as $d\text{unif}(0, 1000)$ and smaller more informed uniform priors ($d\text{unif}(-10, 10)$ and $d\text{unif}(-50, 50)$). These slightly informed priors with steel door data failed to produce results with the three explanatory variables case, depth and temperature case, and for $d\text{unif}(-50, 50)$ failed to run the flow regression model. The above tests used the same prior distribution for all unknowns. A few mixed cases were run where the intercept prior was assigned either $d\text{unif}(-10, 10)$ or $d\text{unif}(-50, 50)$ and the covariate coefficients were assigned the positive diffuse $d\text{unif}(0, 1000)$ prior

distribution. Both mixed prior tests worked using the steel door data, but only the first successfully ran using the hollow core door data.

While using the different uniform priors produced complete batches of runs with calculated p-values, comparing the results revealed an introduced sensitivity in the model to the priors. Tables 6.16 and 6.17 present the depth coefficient posterior results from several prior tests using the two data sets. The change in posterior mean shows that the prior is taking preference in the analysis over the experimental data. This is not the desired outcome when using non-informative priors. Since no prior knowledge regarding a door's performance in a flooding scenario is being taken into account, data should take the preference.

Table 6.16: Comparison of depth coefficient posterior results for several prior cases using steel door data.

Prior	Mean	Standard Dev.	97.5% Interval
dunif(0, 1000)	0.013	0.0095	(4.79E-4, 0.035)
dunif(-10,10) intercept dunif(0, 1000) coeff.	0.23	0.05	(0.11, 0.3)
dunif(-50,50) intercept dunif(0, 1000) coeff.	0.91	0.29	(0.32, 1.35)

Table 6.17: Comparison of depth coefficient posterior results for several prior cases using hollow core door data.

Prior	Mean	Standard Dev.	97.5% Interval
dunif(0, 1000)	0.028	0.02	(0.001, 0.07)
dunif(-10,10) intercept dunif(0, 1000) coeff.	0.28	0.07	(0.12, 0.41)

Continued research into prior selection as a solution for the $n = 1$ case could produce working results with additional development given to slightly informed priors, along with improved initial values. This development path could provide

future dissertation work. It should also be noted that subtle differences in programs could resolve some of these problems. Not all available programs, for instance, use the same sampling approach. A similar model setup in R or JAGS could run without additional considerations for the case of $n = 1$.

A more robust path was sought that could use non-informative priors and easily accommodate data sets from various component experiments without excessive change. Rather than adjusting priors, the found solution focused on the parameter that was failing to meet specifications. The binomial probability of failure, p , needs to take on values between 0 and 1 in order for OpenBUGS to perform the calculation, as referenced earlier. This requirement can be achieved by restricting p using a couple of built in scalar functions, `max` and `min`. They are defined and operate as follows:

- `max(e1, e2)` $e1$ if $e1 > e2$; $e2$ otherwise,
- `min(e1, e2)` $e1$ if $e1 < e2$; $e2$ otherwise.

For the probability of failure to be properly scaled, the following criteria need to hold true:

- return 1 if probability of failure is greater than 1,
- return 0 if probability of failure is less than 0,
- otherwise p .

The quantity `p.bound[i] ← max(0, min(1, p[i]))` performs all three listed criteria. Inserting `p.bound` into the the model script restricts the probability to lie between 0 and 1 and prevents OpenBUGS from crashing [31]. A logistic link function can now be used when $n = 1$ for all regression models.

Centering and Predictive Significance

At the beginning of model development, the water temperature data was included as an explanatory variable that was expected to be eliminated as part of the Bayesian analysis. Contrary to this assumption, some models explicitly using temperature or a combination including temperature data have some of the smallest DIC amongst the compared regression models (see Table 6.9 and Table 6.15). To address the possibility of temperature as a failure influence, centering was used on the covariates.

Interpreting coefficients in models with interactions can be simplified by subtracting the mean, $\bar{x} = N^{-1} \sum x_i$, of each input variable x_i [20]. For example, the temperature T in Equation 6.3 would be subtracted by \bar{T} and the following logistic regression would be fit:

$$\text{logit}(p) = \text{intercept} + c(T - \bar{T}), \quad (6.16)$$

where the data is now centered at zero.

Centering also typically increases the speed of convergence. MCMC convergence issues can occur when the covariates are used directly. High posterior correlation between the intercept and coefficients cause the Gibbs sampling algorithm to run slowly. Centering the predictor variable essentially relocates the y-axis to the average $x = \bar{x}$, reducing the correlation and decreasing the number of iterations to converge [31].

First using the hollow core door data, the single explanatory variable models (D, F, and T) were run as a standard logit regression. The results are given in Table 6.18. The predictor variables were then centered and the models run again

for comparison, with results given in Table 6.19.

Table 6.18: Coefficient results for standard logit regression models using single variables for hollow core doors.

Model	Mean	Standard Dev.	97.5% Interval
Depth	4.61	2.38	(0.97 , 10.02)
Flow Rate	0.16	0.13	(0.01 , 0.48)
Temperature	-0.44	0.47	(-1.47 , 0.37)

Table 6.19: Coefficient results for centered logit regression models using single variables for hollow core doors.

Model	Mean	Standard Dev.	97.5% Interval
Depth	87.23	59.37	(9.81 , 229.1)
Flow Rate	0.199	0.17	(0.008 , 0.64)
Temperature	-0.51	0.52	(-1.68 , 0.35)

The main effects of using explanatory variables are now interpretable based on comparison to the mean of the data. While the coefficients of the flow rate and temperature variables stayed small and relatively the same, the posterior mean of the depth coefficient increased from 4.61 to 87.23. For hollow core doors, the predictive difference is clearly higher than the comparison of flow rate or temperature.

The DIC for both sets of results are shown in Table 6.20. While the DIC for flow rate and temperature models slightly increased by 2.9% and 1.1% respectively, the DIC for the depth model decreased by 13.1%. Again, centering has increased the predictive difference of water depth over flow rate and temperature.

Overall, the results of single variable hollow core door regression models indicate water depth as the leading indicator of component failure. Also addressed in these results is that water temperature is not important to the failing of a door. It could possibly aid in improving a model with its inclusion along with another

Table 6.20: DIC results for single variable regression models using hollow core door data.

Model	Stand. DIC	Centered DIC
Depth	0.50	0.066
Flow Rate	14.48	14.9
Temperature	16.81	17.0

explanatory variable, but it would not be recommended for predictive analysis on its own.

Looking at the steel door data, using the same single explanatory variable models, leads to a different discovery. Table 6.21 gives the results for the standard models, and Table 6.22 gives the results when centering is applied. Again, depth's predictive difference is greater than flow rate, but the highest is temperature. Additionally, temperature has the smallest DIC between the three models (Table 6.23).

Table 6.21: Coefficient results for standard logit regression models using single variables for steel doors.

Model	Mean	Standard Dev.	97.5% Interval
Depth	1.66	0.91	(0.42 , 3.89)
Flow Rate	0.013	0.006	(0.004 , 0.028)
Temperature	2.56	0.88	(1.10 , 4.51)

Table 6.22: Coefficient results for centered logit regression models using single variables for steel doors.

Model	Mean	Standard Dev.	97.5% Interval
Depth	2.05	1.26	(0.46 , 5.21)
Flow Rate	0.013	0.006	(0.005 , 0.028)
Temperature	7.85	4.69	(2.04 , 19.74)

To understand why temperature appears to be the leading indicator of failure, the steel door data, along with its collection process, must be examined. Of the

Table 6.23: DIC results for single variable regression models using steel door data.

Model	Stand. DIC	Centered DIC
Depth	13.86	14.39
Flow Rate	16.0	15.98
Temperature	8.294	8.98

nineteen test results recorded in Table 5.3, the first nine tests all resulted in door failures. These nine tests were conducted exclusively during the spring. The remainder of the tests, nine non-failure and one failure, were conducted in a single day during the winter when the reservoir water was cooler. The results could mean that warmer water temperatures cause steel doors to fail in flooding events, implying a correlation of variables observed together. It is noted, however, that correlation does not necessarily mean causation. The relationship could have alternative explanations, such as a third-cause fallacy, where a spurious correlation is mistaken for causation. A spurious correlation is a relationship in which events or variables are associated, but not causally related, due to the presence of a third factor [32]. Seasonal weather changing the interior temperature of the lab is a hidden third factor. Therefore, steel door flooding failure and water temperature may be correlated with each other only because they are correlated with the weather when testing was conducted. By conducting all non-failure tests in the cooler winter conditions and majority of failures in the warmer spring, an unintentional bias was introduced into the temperature data. This bias, that temperature impacts failure, becomes apparent when looking at the centering comparison.

There is another means of verifying the introduced bias in temperature by looking at the residuals. As explained in Chapter 3, Pearson residuals should take on values between -2.0 and 2.0. Any data point with values outside this range repre-

sent an outlier. If there is a bias introduced from when the tests were conducted, the last data point, a failure during winter testing, should be considered an outlier. Figure 6.2 shows the residual box plot for the temperature regression model. Note that the last data point has an outlier residual value of 3.53 ± 6.037 , confirming the bias.

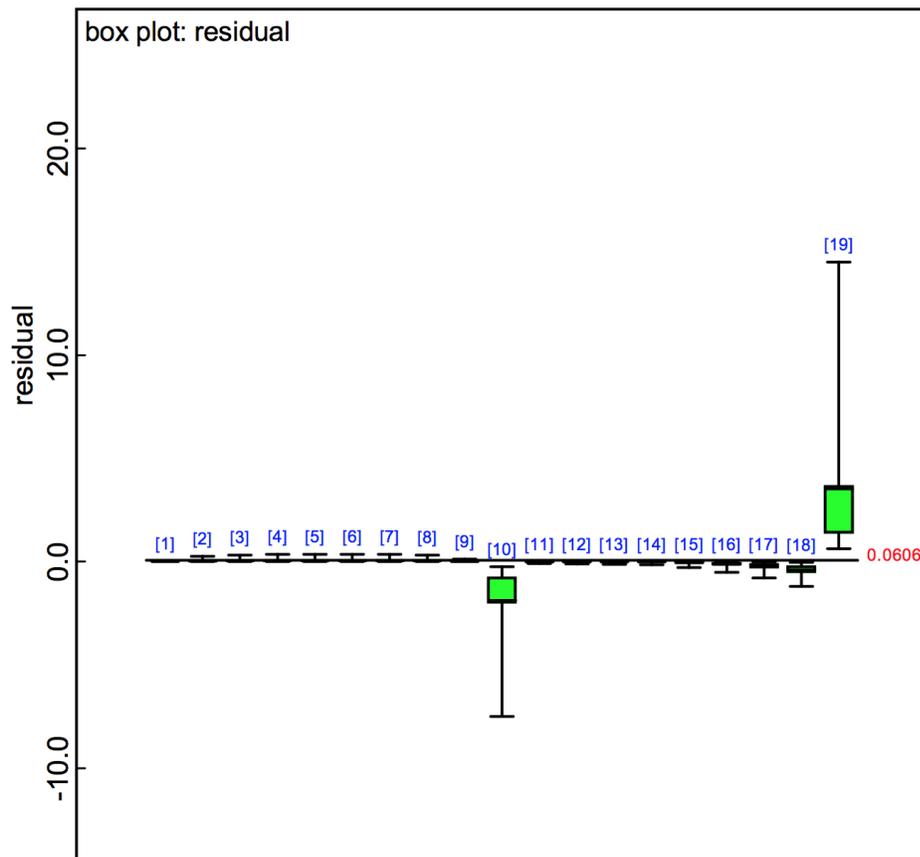


Figure 6.2: A box plot of the residuals for the temperature regression model using steel door data.

Since the steel temperature data is biased, it will be dropped from consideration as an explanatory variable for now. In experiments, spurious correlation can be identified by controlling and extensively testing the relationship between dependent and independent variables. For component flooding experiments, steps

could be taken to control the temperature of the reservoir water. If future testing corrects for this bias, temperature data could again be considered as part of the Bayesian analysis for steel doors. Of the remaining depth and flow rate data, centering simplified interpreting coefficients and again indicated depth as a significant indicator of failure.

Bayesian P-value Numerical Error

With the logit error resolved, there was one piece of model script that still occasionally caused runs to crash that needed addressing. The χ^2 discrepancy function used to calculate the Bayesian p-value would periodically have to be commented out, effectively removing it from the script, to allow the coefficient's posteriors to be estimated. There was no notable pattern for which regression model would run with the p-value calculation, so the execution of the discrepancy function was looked at closer to resolve the error.

For the binomial distribution, the χ^2 discrepancy function is expressed as:

$$T(y, \theta) = \sum_i \frac{(y_i - np_i)^2}{np_i(1 - p_i)}, \quad (6.17)$$

where p and n are the binomial parameters for probability of failure and number of trials, and y is the number of failures in n trials. One approach to find a numerical error is to input extreme values into the equation and see how it behaves. In this case, the extremes for probability of failure are $p=0$ or 1 . First looking at $p=0$, the denominator is multiplied by p and would become zero. For $p=1$, the term $(1 - p_i) \rightarrow 0$ and again the denominator would become zero. In both cases the discrepancy function is attempting to divide by zero.

To avoid numerical errors if $p=0$ or 1 , a small $\varepsilon = 0.00001$ can be added to the

denominator. This prevents a division by zero without affecting the outcome and allows a complete run with a p-value calculation. The χ^2 discrepancy function is now given as:

$$T(y, \theta) = \sum_i \frac{(y_i - np_i)^2}{np_i(1 - p_i) + \varepsilon}. \quad (6.18)$$

In addition to the χ^2 , two additional discrepancy functions were researched for p-value calculations. The likelihood ratio statistic and the Freeman-Tukey statistic [26], listed in Table 3.1, for the binomial distribution are as follows:

$$\text{Likelihood ratio : } T(y, \theta) = 2 \sum_i y_i \log \left(\frac{y_i + \varepsilon}{np_i + \varepsilon} \right), \quad (6.19)$$

$$\text{Freeman-Tukey : } T(y, \theta) = \sum_i (\sqrt{y_i} - \sqrt{np_i})^2. \quad (6.20)$$

Note the ε has been included on the likelihood ratio statistic to avoid any numerical errors. There are now three robust metrics for comparison when checking models.

Introducing Variability in the Model

Development of the logistic regression models so far has been interpreting directly the failure response given some predictor(s) data. It is also possible to interpret indirectly by incorporating an additional random variability. These models assume that besides the observed variables, there could be an unobserved variable or random effects. Therefore, the probability of the binomial distribution is allowed to adjust by some small amount, λ_i , for each observation.

Using the steel door observed data, a script was written where logistic regression equations contain a random or latent effect. Table 6.24 shows the model por-

tion of the script. In the case of the depth model, previously given by Equation 6.1, it would now be defined as follows:

$$\text{logit}(p) = \text{intercept} + aD + \lambda_i, \quad (6.21)$$

with $\lambda_i \sim N(0, \sigma^2)$ and unknown variance. A prior distribution is specified for σ . By allowing the probability to vary on an observation by observation bases, more variability should be accounted for.

The resulting p-values and DIC for the depth, flow rate, and combined regression models are given in Table 6.25. The larger p-values (all greater than 0.95) strongly suggest lack of fit. The regression models without variability are favorable over the inclusion of unobserved effects for their better fit. Regression coefficients are the main interest, especially for later simulation, over the random effects.

Table 6.25: P-values and DIC of logistic regression model with an added latent variable using steel door data.

Model	χ^2	Likelihood Ratio	Freeman-Tukey	DIC
Depth	0.97	0.97	0.97	0.41
Flow rate	0.99	0.99	0.99	0.13
Depth, Flow rate	0.99	0.99	0.99	0.08

6.2 Verification Case

As part of the model development process, several adjustments were applied to the OpenBUGS script to increase the robustness. These changes included placing a bound restriction on probability and adding a small ϵ to p-value calculations so numerical errors are avoided. To assure that these adjustments are not affecting the results of the model, a verification case is needed. The NASA O-ring example presented in Chapter 4 can be used to verify the developed model, as well as the

Table 6.24: Script showing model portion of logistic regression model with an added latent variable.

```

model{
  for(i in 1:tests){
    failure[i] ~ dbin(p.bound[i], numtested)
    p.bound[i] <- max(0, min(1, p[i]))

    #Regression Model
    #logit(p[i]) <- int + flow*WFlow[i] + lambda[i]
    #logit(p[i]) <- int + depth*WDepth[i] + lambda[i]
    #logit(p[i]) <- int + temp*WTemp[i] + lambda[i]
    logit(p[i]) <- int + flow*WFlow[i] + depth*WDepth[i] + lambda[i]

    lambda[i] ~ dnorm(0, inv.omega.lambda.squared)

    failure.rep[i] ~ dbin(p.bound[i], numtested)

    #Fit Assessment: Pearson Residuals Posterior Predictive check (Bayesian P-Value)
    residual[i] <- (failure[i] - (numtested*p.bound[i]))/sqrt(numtested*p.bound[i]*(1-p.bound[i]) + 0.00001)
    residual.rep[i] <- (failure.rep[i] - (numtested*p.bound[i]))/sqrt(numtested*p.bound[i]*(1-p.bound[i]) + 0.00001)
    sq[i] <- pow(residual[i], 2)
    sq.rep[i] <- pow(residual.rep[i], 2)

    #Fit Assessment: Likelihood Statistic Posterior Predictive check (Bayesian P-Value)
    like.obs[i] <- failure[i]*log((failure[i] + 0.00001)/(numtested*p.bound[i] + 0.00001))
    like.rep[i] <- failure.rep[i]*log((failure.rep[i] + 0.00001)/(numtested*p.bound[i] + 0.00001))

    #Fit Assessment: Freeman-Tukey Statistic Posterior Predictive check (Bayesian P-Value)
    diff.obs[i] <- pow(sqrt(failure[i]) - sqrt(numtested*p.bound[i]), 2)
    diff.rep[i] <- pow(sqrt(failure.rep[i]) - sqrt(numtested*p.bound[i]), 2)
  }

  chisq.obs <- sum(sq[])
  chisq.rep <- sum(sq.rep[])
  p.chisq <- step(freeman.rep - freeman.obs)

  likelihood.obs <- sum(like.obs[])
  likelihood.rep <- sum(like.rep[])
  p.likelihood <- step(likelihood.rep - likelihood.obs)

  freeman.obs <- sum(diff.obs[])
  freeman.rep <- sum(diff.rep[])
  p.freeman <- step(chisq.rep - chisq.obs)

  #Prior Distributions
  int ~ dnorm(0, .000001)
  depth ~ dnorm(0, .000001)
  flow ~ dnorm(0, .000001)
  temp ~ dnorm(0, .000001)

  inv.omega.lambda.squared <- 1/pow(omega.lambda, 2)
  omega.lambda ~ dunif(0, 1000)
}

```

model checking and comparison metrics.

Reference [19] recommends the temperature regression model for predictive analyses. Using this model for verification, the OpenBUGS script for this model is shown in Table 6.26. One thousand burn-in iterations were used, followed by 100,000 iterations to estimate the parameters. Table 6.27 shows the posterior mean, standard deviation, and 95% credible interval for each parameter in the logistic regression. Table 6.28 shows the same summary provided from Reference [19]. The posterior distributions for a and b are approximately the same in both analyses. The bound restriction does not appear to affect the posterior results.

Table 6.26: Portion of developed OpenBUGS script for logistic regression on temperature using primary O-ring distress data.

```

model{
  for(i in 1:tests){
    failure[i] ~ dbin(p.bound[i], numtested)
    p.bound[i] <- max(0, min(1, p[i]))

    #Regression Model
    logit(p[i]) <- a + b*temp[i]

    failure.rep[i] ~ dbin(p.bound[i], numtested)

    #Fit Assessment: Pearson Residuals Posterior Predictive check (Bayesian P-Value)
    residual[i] <- (failure[i] - (numtested*p.bound[i]))/sqrt(numtested*p.bound[i]*(1-p.bound[i]) + 0.00001)
    residual.rep[i] <- (failure.rep[i] - (numtested*p.bound[i]))/sqrt(numtested*p.bound[i]*(1-p.bound[i]) + 0.00001)
    sq[i] <- pow(residual[i], 2)
    sq.rep[i] <- pow(residual.rep[i], 2)

    #Fit Assessment: Likelihood Statistic Posterior Predictive check (Bayesian P-Value)
    like.obs[i] <- failure[i]*log((failure[i] + 0.00001)/(numtested*p.bound[i] + 0.00001))
    like.rep[i] <- failure.rep[i]*log((failure.rep[i] + 0.00001)/(numtested*p.bound[i] + 0.00001))

    #Fit Assessment: Freeman-Tukey Statistic Posterior Predictive check (Bayesian P-Value)
    diff.obs[i] <- pow(sqrt(failure[i]) - sqrt(numtested*p.bound[i]), 2)
    diff.rep[i] <- pow(sqrt(failure.rep[i]) - sqrt(numtested*p.bound[i]), 2)
    .....
    prop[i] <- failure[i]/numtested
    Ds[i] <- 2*numtested*(prop[i]*log((prop[i]+0.00001)/(p.bound[i]+0.00001))
      + (1-prop[i])*log((1-prop[i]+0.00001)/((1-p.bound[i]+0.00001))))
  }

  chisq.obs <- sum(sq[])
  chisq.rep <- sum(sq.rep[])
  p.chisq <- step(chisq.rep - chisq.obs)

  likelihood.obs <- sum(like.obs[])
  likelihood.rep <- sum(like.rep[])
  p.likelihood <- step(likelihood.rep - likelihood.obs)

  freeman.obs <- sum(diff.obs[])
  freeman.rep <- sum(diff.rep[])
  p.freeman <- step(freeman.rep - freeman.obs)

  dev.sat <- sum(Ds[])

  #Prior Distributions
  a ~ dnorm(0, 0.000001)
  b ~ dnorm(0, 0.000001)
  c ~ dnorm(0, 0.000001)
}

```

Table 6.27: Summary posterior estimates of logistic regression on temperature using developed model.

Parameter	Mean	Standard Dev.	95% Interval
a (intercept)	5.215	3.18	(-0.01 , 10.45)
b (temp. coeff.)	-0.12	0.049	(-0.20 , -0.04)

Table 6.28: Summary posterior estimates of logistic regression parameters, temperature included as explanatory variable [19].

Parameter	Mean	Standard Dev.	95% Interval
a (intercept)	5.225	3.16	(-1.00 , 11.48)
b (temp. coeff.)	-0.12	0.049	(-0.22 , -0.025)

The original analysis used the Bayesian χ^2 statistic to calculate the p-value and DIC for model comparison (Table 6.29). The developed model analysis calculates two additional p-values using the likelihood ratio and Freeman-Tukey statistics. Table 6.30 shows the p-value and DIC from the verification analysis of the temperature regression model. The χ^2 p-value and DIC are within rounding in both analyses, verifying the small ϵ does not affect results. The likelihood ratio and Freeman-Tukey statistics calculate a larger p-value, closer to the ideal value. These p-values still suggests that the temperature model performs adequately.

Table 6.29: Model-checking results for logistic regression of primary O-ring distress [19].

Explanatory Variables	DIC	χ^2 p-value
Temperature	35.75	0.21

Table 6.30: Model-checking results for logistic regression on temperature using developed model.

Explanatory Variables	DIC	χ^2 p-value	Likelihood ratio p-value	Freeman-Tukey p-value
Temperature	35.7	0.22	0.42	0.56

Finally, making use of the residual information, a box plot of the standard Pearson residuals is shown in Figure 6.3. The 21st data point collected on flight 61-A has a value outside the $[-2.0, 2.0]$ interval and therefore represents an outlier in the temperature data. The saturated deviance has posterior mean 20.16 (95% interval 18.19 to 24.26), and comparison with the sample size of 23 suggests that this model fits adequately.

The results from the model verification show the developed model performs as expected with the O-ring example data. The added robustness measures do not affect the posterior results or create sensitivities. Also, the additional quantitative measures of predictive model validity metrics provide further insight into the O-ring data and model fit. The developed model is ready for final analysis of the experimentally collected door data.

6.3 Model Results

The Bayesian model developed for the assessment of component fragility uses the discrete binomial distribution with a logistic regression model for probability of failure, p . It uses diffuse normal prior distributions ($\text{dnorm}(0, 0.000001)$) and has a restrictive bound on probability to allow complete performance of the OpenBUGS script. Model checking makes use of posterior predictive checks to calculate three types of Bayesian p-values and a saturated deviance to assess adequacy of the

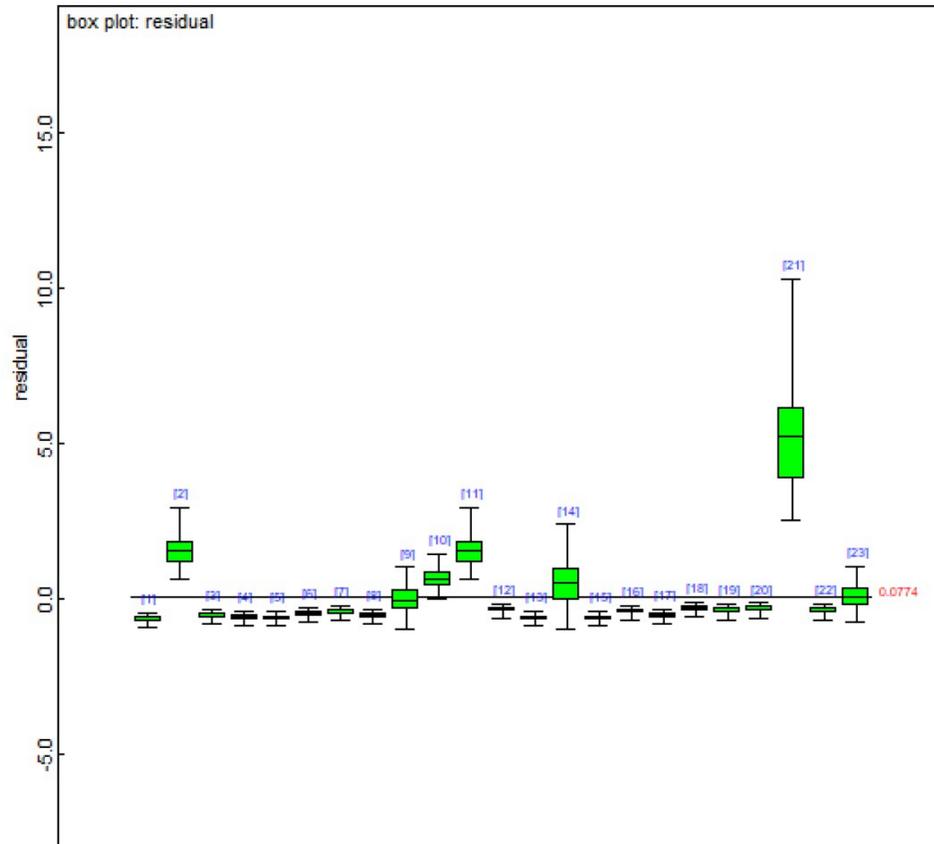


Figure 6.3: A box plot of the residuals for the temperature regression model for primary O-ring distress.

model fit. Comparison between regression models is based on the DIC.

Since each of the possible logistic regression models must estimate more than one parameter, it is recommended that at least two chains are run in OpenBUGS. Running multiple parallel chains that start at dispersed initial values around the estimate of the posterior distribution allows diagnosis of convergence [19] [24]. Any values prior to convergence should not be included to estimate a parameter and burn-in set accordingly.

A quantitative convergence check built into OpenBUGS is the Brooks-Gelman-Rubin (BGR) diagnostic. It assesses convergence by comparing variance within-

and between-chains. A normalized ratio, R , of these estimates tends towards 1 as convergence is approached. For practical purposes, the ratio should be less than approximately 1.2 to signify convergence [19].

The BGR diagnostic gives a plot showing the R ratio as red, the within-chain estimates as blue, and the between estimates as green. R should start out greater than 1 if the initial values are suitably dispersed and converge towards 1, indicated on the plot by a dashed black line. The green and blue plotted estimates should also converge to stability. A table listing the plot values can also be obtained from the plot properties [24].

BGR convergence plots for each regression model are provided in Appendix D. To ensure proper convergence of all parameter estimates, ten thousand burn-in iterations were used for the hollow core door results, except for the combined flow rate and temperature model that required 20,000 burn-in iterations. Five thousand burn-in iterations were used for all steel door results. The number of sample iterations generated in updates after burn-in is specified below with each result.

6.3.1 Hollow Core Doors

When performing the model fit using the hollow core door data, the burn-in iterations were followed by 110,000 iterations to estimate the parameters, except for the combined flow rate and temperature regression model which used 120,000 iterations. A second update followed after the DIC calculation was set. The OpenBUGS script with regression models, prior distributions, and dispersed initial values is shown in Table 6.31. A specific regression model is run by uncommenting the script line and selecting its corresponding initial values.

Tables 6.32, 6.33, and 6.34 show the posterior mean, standard deviation, and 95% credible interval for each parameter for the single explanatory variable re-

Table 6.31: Model portion of final developed OpenBUGS script for logistic regression using hollow core door data.

```

#Bound Binomial Model using Logit Regression: Final
#Hollow Core Door Data

model{

  for(i in 1:tests){
    failure[i] ~ dbin(p.bound[i], numtested)
    p.bound[i] <- max(0, min(1, p[i]))

    #Regression Model
    logit(p[i]) <- int + depth*WDepth[i]
    #logit(p[i]) <- int + flow*WFlow[i]
    #logit(p[i]) <- int + temp*WTemp[i]
    #logit(p[i]) <- int + depth*WDepth[i] + flow*WFlow[i]
    #logit(p[i]) <- int + depth*WDepth[i] + temp*WTemp[i]
    #logit(p[i]) <- int + flow*WFlow[i] + temp*WTemp[i]
    #logit(p[i]) <- int + depth*WDepth[i] + flow*WFlow[i] + temp*WTemp[i]
  }

  #Prior Distributions
  int ~ dnorm(0, 0.000001)
  depth ~ dnorm(0, 0.000001)
  flow ~ dnorm(0, 0.000001)
  temp ~ dnorm(0, 0.000001)
}

inits
#Depth
list(int= -599, depth= 69, flow= 0, temp= 0, failure.rep= c(0,0,0,0,0,0,0,0,0))
list(int= -1905, depth= 21, flow= 0, temp= 0, failure.rep= c(0,0,0,0,0,0,0,0,0))
#Flow
list(int= -8, depth= 0, flow= 0.4, temp= 0, failure.rep= c(0,0,0,0,0,0,0,0,0))
list(int= -116, depth= 0, flow= 0, temp= 0, failure.rep= c(0,0,0,0,0,0,0,0,0))
#Temp
list(int= 92, depth= 0, flow= 0, temp= 0, failure.rep= c(0,0,0,0,0,0,0,0,0))
list(int= 0, depth= 0, flow= 0, temp= -1, failure.rep= c(0,0,0,0,0,0,0,0,0))
#Depth and Flow
list(int= -916, depth= 64, flow= 4, temp= 0, failure.rep= c(0,0,0,0,0,0,0,0,0))
list(int= -2260, depth= 16, flow= 0, temp= 0, failure.rep= c(0,0,0,0,0,0,0,0,0))
#Depth and Temp
list(int= 1006, depth= 1744, flow= 0, temp= -236, failure.rep= c(0,0,0,0,0,0,0,0,0))
list(int= -996, depth= 548, flow= 0, temp= -772, failure.rep= c(0,0,0,0,0,0,0,0,0))
#Flow and Temp
list(int= -36, depth= 0, flow= 0.4, temp= 2, failure.rep= c(0,0,0,0,0,0,0,0,0))
list(int= -220, depth= 0, flow= 0, temp= 0, failure.rep= c(0,0,0,0,0,0,0,0,0))
#Depth and Flow and Temp
list(int= 1004, depth= 1672, flow= 240, temp= -522, failure.rep= c(0,0,0,0,0,0,0,0,0))
list(int= -1022, depth= 540, flow= 18, temp= -1586, failure.rep= c(0,0,0,0,0,0,0,0,0))

```

gression models in the following order: depth, flow rate, and temperature. The next cases considered the combination of two explanatory variables in the regression model. Table 6.35 shows the summary posterior estimates with depth and flow rate as explanatory variables, Table 6.36 shows the summary posterior estimates with depth and temperature as explanatory variables, and Table 6.37 shows the summary posterior estimates with flow rate and temperature as explanatory variables. The final case considered is the most complex regression model, which includes all three variables as indicators of failure. Table 6.38 shows the summary posterior estimates using the three observables. Included in each of these tables is also the saturated deviance for checking the model fit. The remaining model checking and comparison results for hollow core door failure are given in Table 6.39.

Table 6.32: Summary posterior estimates of logistic regression parameters for hollow core doors, water depth included as explanatory variable.

Parameter	Mean	Standard Dev.	95% Interval
intercept	-1252.0	654.5	(-2444.0, -322.9)
depth coeff.	45.23	24.01	(11.53, 89.3)
sat. deviance	0.03	0.29	(0.0, 0.01)

Table 6.33: Summary posterior estimates of logistic regression parameters for hollow core doors, water flow rate included as explanatory variable.

Parameter	Mean	Standard Dev.	95% Interval
intercept	-63.1	55.28	(-173.2, -4.33)
flow rate coeff.	0.22	0.19	(0.01, 0.59)
sat. deviance	13.83	2.32	(11.41, 18.45)

Table 6.34: Summary posterior estimates of logistic regression parameters for hollow core doors, water temperature included as explanatory variable.

Parameter	Mean	Standard Dev.	95% Interval
intercept	34.2	34.58	(-15.36, 96.77)
temp. coeff.	-0.51	0.52	(-1.44, 0.24)
sat. deviance	15.02	2.25	(12.87, 19.52)

Table 6.35: Summary posterior estimates of logistic regression parameters for hollow core doors, water depth and flow rate included as explanatory variable.

Parameter	Mean	Standard Dev.	95% Interval
intercept	-1589.0	670.6	(-2790.0, -584.8)
depth coeff.	39.61	23.41	(8.82, 82.97)
flow rate coeff.	1.59	1.61	(-0.21, 4.82)
sat. deviance	0.04	0.41	(0.0, 0.01)

Table 6.36: Summary posterior estimates of logistic regression parameters for hollow core doors, water depth and temperature included as explanatory variable.

Parameter	Mean	Standard Dev.	95% Interval
intercept	5.05	999.2	(-1634.0, 1653.0)
depth coeff.	1142.0	596.7	(295.5, 2241.0)
temp. coeff.	-501.8	266.4	(-989.9, -128.3)
sat. deviance	9.25E-4	0.05	(0.0, 0.0)

Table 6.37: Summary posterior estimates of logistic regression parameters for hollow core doors, water flow rate and temperature included as explanatory variable.

Parameter	Mean	Standard Dev.	95% Interval
intercept	-124.6	91.96	(-286.6, 8.38)
flow rate coeff.	0.21	0.19	(0.02, 0.58)
temp. coeff.	0.93	1.09	(-0.74, 2.81)
sat. deviance	14.32	2.86	(11.05, 19.91)

Table 6.38: Summary posterior estimates of logistic regression parameters for hollow core doors; water depth, flow rate, and temperature included as explanatory variable.

Parameter	Mean	Standard Dev.	95% Interval
intercept	-8.73	991.5	(-1633.0, 1627.0)
depth coeff.	1118.0	567.9	(335.9, 2168.0)
flow rate coeff.	129.1	109.4	(2.37, 341.9)
temp. coeff.	-1061.0	528.6	(-2046.0, -346.8)
sat. deviance	0.001	0.07	(0.0, 0.0)

Table 6.39: Model checking and comparison results for logistic regression for hollow core doors using developed model.

Explanatory Variables	DIC	χ^2 p-value	Likelihood ratio p-value	Freeman-Tukey p-value
Depth	0.06	1.0	1.0	1.0
Flow Rate	15.08	0.08	0.38	0.24
Temp.	17.07	0.18	0.36	0.29
Depth and Flow Rate	0.06	0.99	0.99	1.0
Depth and Temp.	0.002	1.0	1.0	1.0
Flow Rate and Temp.	16.6	0.09	0.29	0.17
Depth, Flow Rate, Temp.	0.002	1.0	1.0	1.0

The saturated deviance for the flow rate model, the temperature model, and the combined flow rate and temperature model compared with the sample size of 10 suggests that these three models fit adequately. The saturated deviance for models using depth as an explanatory variable indicate that the models' fit fails to represent the data. In contrast to the saturated deviance, the models that incorporated water depth as an explanatory variable have significantly lower DICs. Using the DIC comparison guide, the flow, temperature, and combined flow and temperature models, which have similar DICs, could be ruled out by the significant difference.

Next comparing the Bayesian p-values, the regression model with only flow rate as an explanatory variable has the largest Bayesian p-value closest to 0.5 using the likelihood ratio (0.38). It also has nearly the same average p-value as the model with only temperature. Again, the p-values for models using depth as an explanatory variable indicate poor predictive capabilities.

Recommending a model for predictive analyses is not a straight forward decision. As covered in the discussion on centering and predictive significance (see Section 6.1.3), the water depth is a leading indicator of component failure over flow rate and temperature. The significantly smaller DIC for models including depth verifies the importance of water depth. However, model checking metrics would recommend the model with only flow rate as a variable. It has the slightly larger p-value closest to the ideal and from among the non-depth models it also has the slightly smaller DIC. Looking back at the experiments conducted on hollow core doors, the majority (8/10) of tests were conducted using the initial PET configuration limited to a single inlet flow rate. Additionally, no tests were specifically conducted where the flow rate was less than the leakage rate to collect non-failure data. Therefore, it is recommended to perform analyses with the flow rate regression model until additional data may become available that's collected over a larger range of flow rate, including non-failure tests.

6.3.2 Steel Doors

For the case of steel doors, the burn-in iterations were followed by 105,000 iterations to estimate the parameters. A second update followed after the DIC calculation was set. The OpenBUGS script with regression models, prior distributions, and dispersed initial values is shown in Table 6.40. A specific model is run by uncommenting the script line and selecting its corresponding initial values.

The third case considered is the more complex regression model, which includes both depth and flow rate as explanatory variables. Table 6.43 shows the summary posterior estimates for the two variable case. Included in each of these tables is also the saturated deviance for checking the model fit. The remaining model checking and comparison results for hollow core door failure are given in Table 6.44.

Table 6.41: Summary posterior estimates of logistic regression parameters for steel doors, water depth included as explanatory variable.

Parameter	Mean	Standard Dev.	95% Interval
intercept	-75.68	46.99	(-167.0, -20.48)
depth coeff.	2.05	1.26	(0.57, 4.5)
sat. deviance	12.88	2.27	(10.65, 17.43)

Table 6.42: Summary posterior estimates of logistic regression parameters for steel doors, water flow rate included as explanatory variable.

Parameter	Mean	Standard Dev.	95% Interval
intercept	-8.51	4.05	(-16.27, -3.39)
flow rate coeff.	0.01	0.006	(0.005, 0.03)
sat. deviance	14.29	2.37	(12.07, 18.92)

Table 6.43: Summary posterior estimates of logistic regression parameters for steel doors, water depth and flow rate included as explanatory variables.

Parameter	Mean	Standard Dev.	95% Interval
intercept	-72.5	50.5	(-170.8, -13.02)
depth coeff.	1.83	1.36	(0.19, 4.46)
flow rate coeff.	0.007	0.006	(-0.002, 0.02)
sat. deviance	13.31	2.82	(10.1, 18.82)

Table 6.44: Model checking and comparison results for logistic regression for steel doors using developed model.

Explanatory Variables	DIC	χ^2 p-value	Likelihood ratio p-value	Freeman-Tukey p-value
Depth	14.42	0.19	0.38	0.33
Flow Rate	16.01	0.26	0.36	0.23
Depth and Flow Rate	15.66	0.14	0.29	0.21

The saturated deviance for all three models compared with the sample size of 19 suggests that all three models fit adequately. The DIC is nearly the same for all three models, the smallest belonging to the depth model by a non-significant amount. The model with only depth as an explanatory variable has the largest Bayesian p-value using the likelihood ratio (0.38). It also has the slightly larger average p-value than the regression model with only flow rate and the combined model with both variables. Given the results, the model with only depth is recommended for predictive analyses. This selected steel door fragility regression model will also be used in Chapter 7 to demonstrate an application.

With depth selected as the explanatory variable regression model, the parameters in Table 6.41 are used with the fragility model to calculate the failure probability for a steel door as a function of water depth. The probability p is given by:

$$p = \frac{1}{e^{-(-75.68+2.05x)} + 1} \quad (6.22)$$

where x is the given water depth. Figure 6.4 shows the plot of failure probability versus water depth with 95% credible intervals. It should be noted that the mean, shown in red, is close to the bound at low probabilities. This is due to a couple of non-failure tests reaching water depths greater than some observed failure depths,

bringing the mean near the credible interval at low fragility probabilities. This fragility curve and probability of failure equation can now be applied to practical applications, such as simulating a door failure during a flooding event.

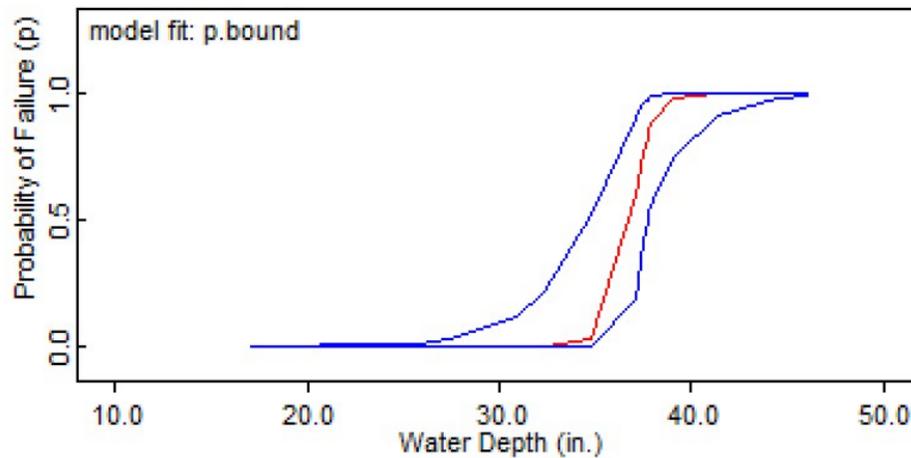


Figure 6.4: Fragility curve showing probability of failure versus water depth plotted in OpenBUGS. Blue curves represent the 95% credible intervals.

Prior Distribution Sensitivity Analysis

Before applying the chosen depth fragility model to a simulation, a sensitivity analysis to alternative prior distributions should be made. There is no such thing as the true prior. If a specific choice of non-informative priors is influential, it could suggest that a conclusion cannot be drawn from the data alone and more informative priors based on background should be considered. At a minimum, a weakly informative prior would need to be determined.

For the sensitivity analysis of the non-informative priors, two alternatives are considered: a flat distribution and a uniform distribution. A flat distribution, `dflat()`, is an improper uniform distribution that extends over the whole real number line available in OpenBUGS. The proper uniform distribution, how-

ever, can have explicit limits and should have large variance, for this case $\text{dunif}(-5000, 5000)$, to be non-informative.

A summary of the sensitivity analysis on the depth coefficient posterior is given in Table 6.45 and illustrated in Figure 6.5. The coefficient's posterior distribution mean is unchanged by the alternative priors. Therefore, the posterior distribution of the depth coefficient does not have unintended sensitivities to the non-informative prior distribution.

Table 6.45: Summary posterior estimates of depth coefficient; normal, flat, and uniform distributions used as non-informative priors.

Prior	Mean	Standard Dev.	95% Interval
Normal	2.05	1.26	(0.57, 4.5)
Flat	2.05	1.27	(0.57, 4.5)
Uniform	2.05	1.27	(0.57, 4.5)

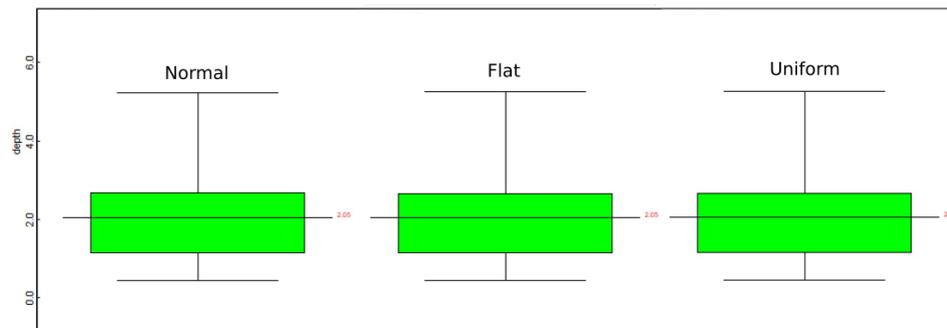


Figure 6.5: Box plot of the depth coefficient estimates using normal, flat, and uniform priors.

Chapter 7

Linking a Component Fragility Model to Simulation

7.1 Smooth Particle Hydrodynamics

Smooth Particle Hydrodynamics (SPH) is a method for modeling 3D fluid flow and rigid body interactions. The particle-based method obtains approximate numerical solutions to the equations of fluid dynamics, the Navier Stokes equations. SPH was originally developed for solving astrophysical problem [33] [34], and later its range of application extended to be used in fluid dynamics [35].

Two approaches are generally used for solving the Navier Stokes equations: a grid based Eulerian approach or a particle based Lagrangian approach. Traditional computational fluid dynamics (CFD) requires an underlying stationary grid when solving fluid equations of motion to determine properties at a fixed grid location. SPH uses the mesh-free Lagrangian method to simulate flow. The Lagrangian method tracks a particle as it moves through the simulation [36].

SPH works by representing the fluid particles, where the physical properties and equations are based on the continuum equations of fluid dynamics. Physical quantities are estimated by interpolation, meaning the property of one particle will depend on the surrounding particles. The integral representation of a field variable A at location r is defined as:

$$A_i(\mathbf{r}) = \int A(\mathbf{r}')W(\mathbf{r} - \mathbf{r}', h)d\mathbf{r}', \quad (7.1)$$

where i denotes the particle of interest, W is the smoothing kernel, and h is the smoothing length. In order to solve Equation 7.1, the integral can be rewritten into numerical form as:

$$A_i(\mathbf{r}) = \sum_j A_j \frac{m_j}{\rho_j} W(\mathbf{r} - \mathbf{r}_j, h), \quad (7.2)$$

where j denotes the surrounding particles, m is the mass of the particle, and ρ is the density of the particle. Also, the gradient and Laplacian of the summation interpolate as follows:

$$\nabla A_i(\mathbf{r}) = \sum_j A_j \frac{m_j}{\rho_j} \nabla W(\mathbf{r} - \mathbf{r}_j, h) \quad (7.3)$$

$$\nabla^2 A_i(\mathbf{r}) = \sum_j A_j \frac{m_j}{\rho_j} \nabla^2 W(\mathbf{r} - \mathbf{r}_j, h) \quad (7.4)$$

assuming the smoothing kernel(W) is first and second order differentiable [37].

The SPH approximation is illustrated in Figure 7.1. The smoothing kernels are weighting functions that determine how much a neighboring particle affects the particle of interest. Closer neighbors will have a larger influence than those farther away. The smoothing length specifies a radius from the particle of interest. Any surrounding particles in the radius will influence the particle of interest. Increasing the smoothing length increases the number of influencing particles.

The smoothing kernel and smoothing length are incorporated into the equations of motion. When considering an incompressible flow of a Newtonian fluid, the momentum conservation equation is given as:

$$\frac{d\mathbf{v}_i}{dt} = - \sum_j m_j \left(\frac{P_j}{\rho_j^2} + \frac{P_i}{\rho_i^2} + \Pi_{ij} \right) \nabla_i W_{ij} + \mathbf{g}, \quad (7.5)$$

where v is the velocity vector, P is the pressure, Π_{ij} represents the viscosity term, and g is gravity. It calculates the change of momentum of the particle of interest,

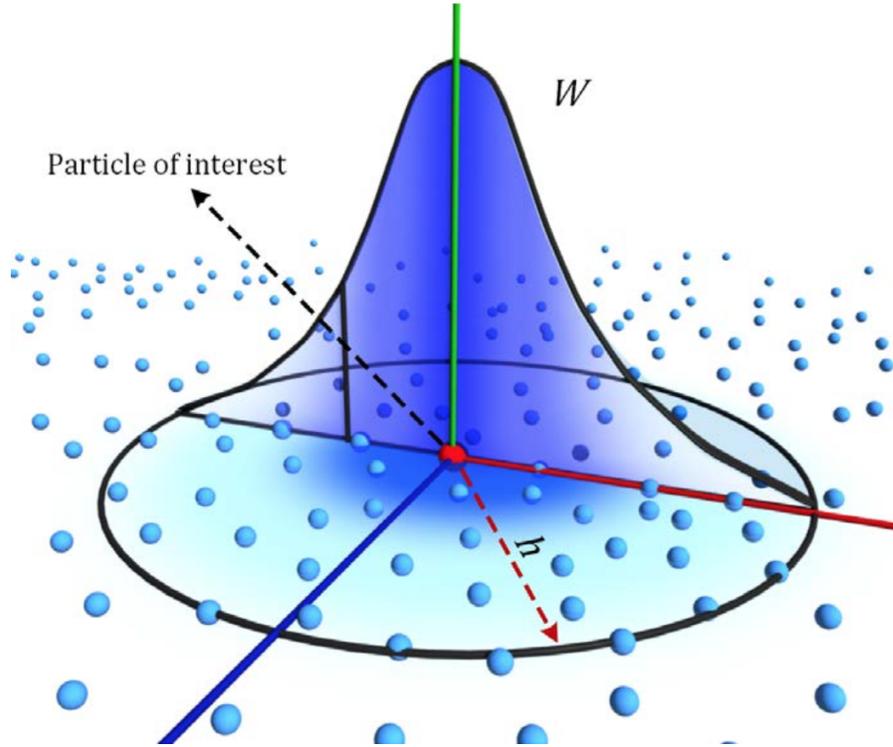


Figure 7.1: Illustration of SPH approximation for the particle of interest using smoothing kernel and smoothing length [38].

accounting for pressure, viscosity, and other external forces.

Fluid density is also an important field variable of SPH simulation, since pressure force arises as a result of changes in the fluid density. The mass continuity equation is as follows:

$$\frac{d\rho_i}{dt} = \sum_j m_j (\mathbf{v}_i - \mathbf{v}_j) \cdot \nabla W_{ij}, \quad (7.6)$$

which contains the density rate of change based on the relative motion of the particles [37].

Finally, compressibility is important when dealing with fluids. There are multiple methods that account for compressibility. The Implicit Incompressible SPH

(IISPH) is a technique that uses an SPH approximation of the continuity equation to obtain a discretized form of pressure Poisson equation:

$$\nabla^2 \rho_i(t) = \frac{\rho_0 - \rho_i^*}{\Delta t^2}, \quad (7.7)$$

where ρ^* are intermediate densities. IISPH takes into account the pressure force and is computed based on particle velocities instead of position. Once the Poisson equation is iterated until relative density is below a specified threshold, new positions and velocities are computed [35].

Neutrino

Neutrino is a general purpose simulation and visualization environment developed by Neutrino Dynamic Initiative as a proprietary code available to universities for research [38]. The Neutrino fluid solver is based on IISPH. It also factors in accurate boundary handling and adaptive time stepping to increase accuracy and calculation speed [39].

Neutrino has a graphical user interface (GUI) for the creation of 3D models. These models can include: rigid bodies, such as boxes, cuboids, and planes; particle emitters; and particle killers to name a couple. Neutrino also provides a variety of tools to measure parameters during simulation. This includes measurement fields that give the average fluid height in a specified area/volume.

An additional feature of Neutrino are custom Python dynamic expression scripts. A Python script can be loaded in for a specific component, such as a rigid body, and executed during model simulation. This means a Python dynamic expression script can be utilized to couple Neutrino with a component fragility model and dictate how a rigid body responds to the fluid flow interaction.

To demonstrate the ability to link component fragility modeling to simulation, Neutrino VC 141 was used. This version was released on 22 February 2019.

7.2 Simulation Build

To show how the fragility model could be coupled to a simulation, a basic Neutrino model was created. A simplified PET environment was chosen that consists of a room that water floods, a wall with a door, and a channel to direct water away. Figure 7.2 shows the setup of the model.

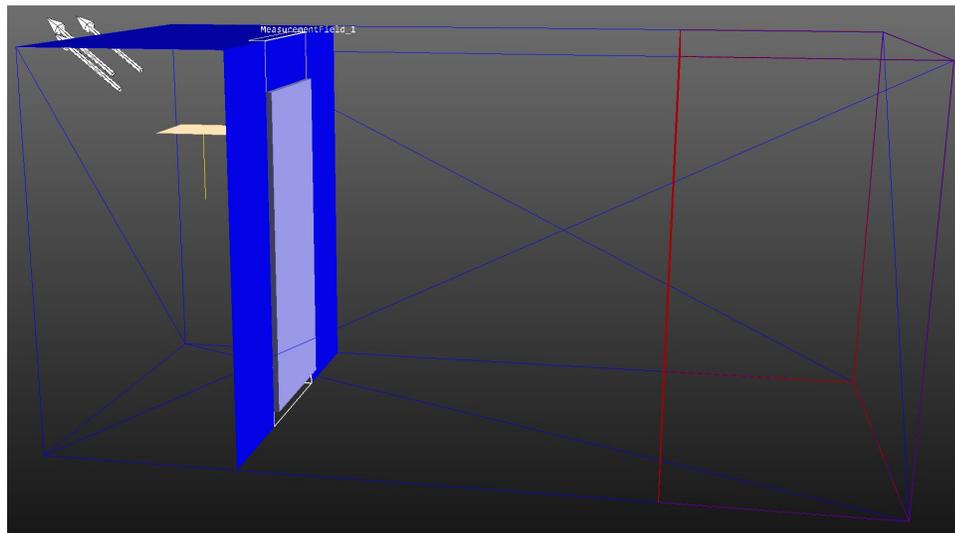


Figure 7.2: Simplified PET setup for simulation.

A rigid box is partitioned into a room and channel by the wall constructed with rigid planes. Another rigid plane is placed over the room to prevent sprayed particles from leaving the box. The door is modeled by a static rigid cuboid. A gap was left under the door to simulate the leakage observed during experiments. A flow particle emitter inside the room creates the fluid particles. It was placed off-center from the door to prevent inaccurate fluid height measurements. A measurement field is placed over the door to read the average fluid height. The end of

the channel is an extent particle killer. The particle killer decreases the run time by removing particles from the system that no longer matter.

For this setup, the fluid particles fall from the emitter at a specified flow rate and begin to fill the room. Some particles will flow under the door gap and down the channel. When the particles reach the end of the channel they are killed, much like the water flowing through floor grating back to the reservoir. As the fluid rises in the room, the door fails at a water depth prescribed by the fragility model.

7.3 Sampling the Fragility Curve

To determine a water height at which a door would fail and then have a Neutrino simulation respond accordingly, two Python scripts were developed. The first is used prior to simulation to sample the fragility curve and write a failure depth to file. Neutrino loads in the second Python script to read the failure depth file and set conditions for when the door will fail during simulation. Together the two scripts link the fragility model and the simulation model. Figure 7.3 depicts the flow of the fragility/simulation coupling described below.

For the first script, referred to as the fragility script, the probability equation for p is defined and the intercept and depth coefficient posterior means are set according to Equation 6.22. The total height of the space being flooded, a room 94.5 inches tall, is divided into evenly spaced intervals using the `linspace` function. In this case, each increment increases by a tenth of an inch. Inputting the string of possible depth values into the probability of failure equation calculates a fragility at each depth increment. Plotting the results produces the fragility curve, as seen in Figure 7.4.

Next, iterating through each depth interval, the corresponding fragility probability is input into a binomial distribution and a single sample is taken. If the

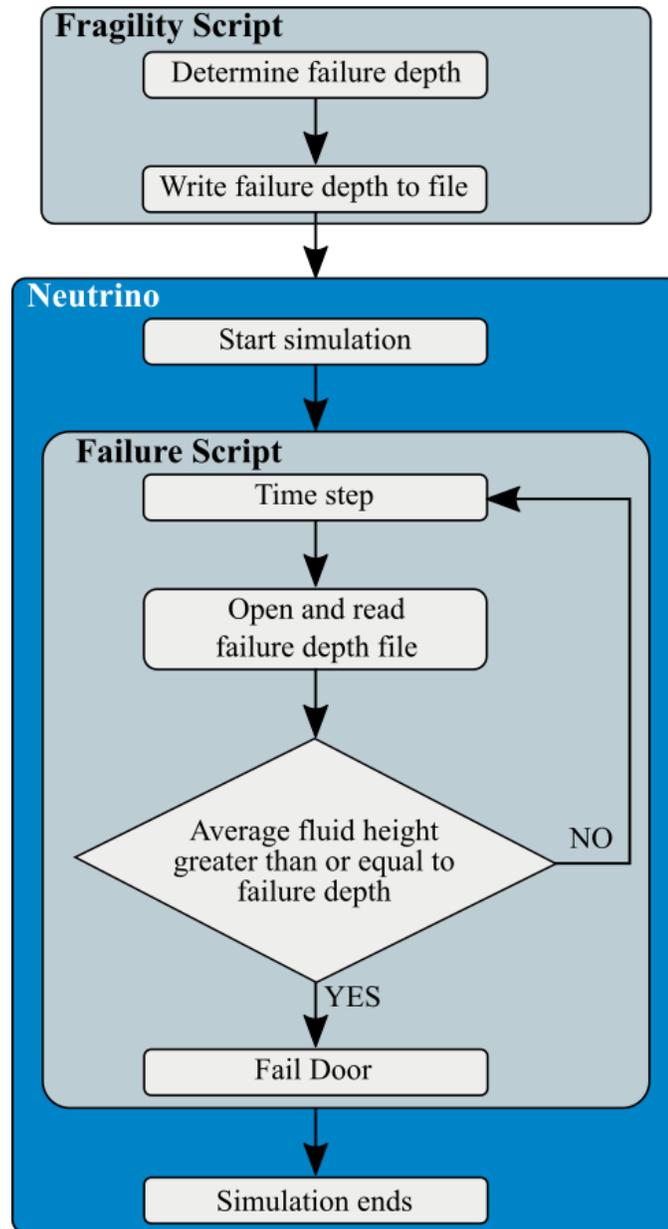


Figure 7.3: Fragility model and Neutrino coupling flow chart.

binomial returns a 0 the iteration continues, but if it returns a 1 the door has failed. Once the door fails, a failure depth is set at the current iteration depth and sampling ends. The determined failure depth is converted from inches to meters to conform to Neutrino's default units. Lastly, the fragility script writes the failure

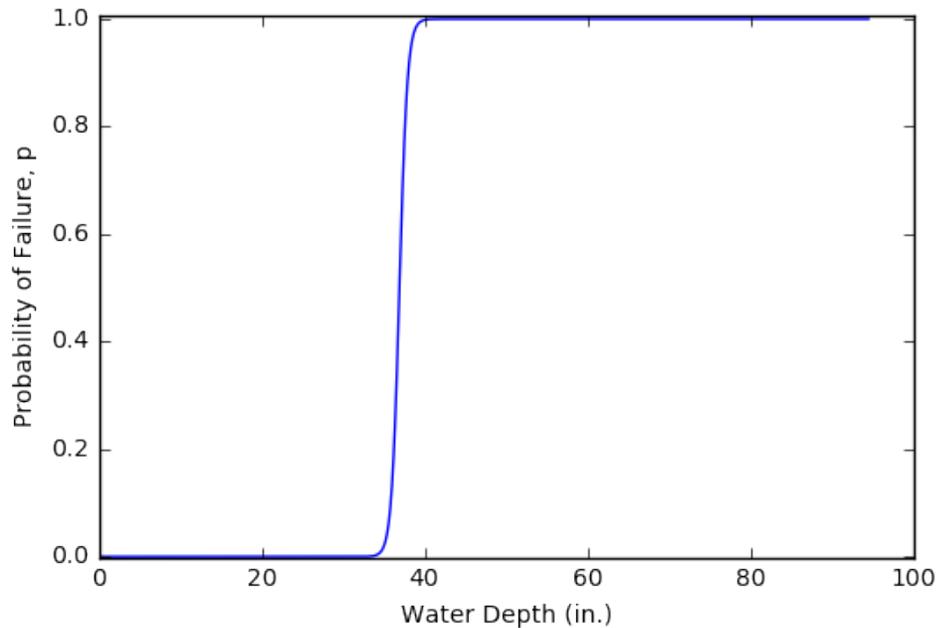


Figure 7.4: Fragility model curve for steel doors plotted in Python.

depth to an output file that can be called upon by the second Python script. Table 7.1 shows the full Python fragility script. Running the fragility script a sufficiently large number of times develops a distribution of failure depth results with variance dependent on the fragility model uncertainty.

The second script, referred to as the failure script, is loaded into Neutrino as a Python dynamic expression script for the rigid cuboid door. At each time step of the simulation, Neutrino reads the failure depth file and compares the value to the current average fluid height at that frame of the simulation. If the average fluid height is greater than or equal to the calculated failure depth, the door is removed from the scene and the fluid is released from the room. This represents a door failure in the simulated model. Table 7.2 shows the Python script for executing the door failure.

As stated in Section 6.1.2, analysis with continuous aleatory models is a valu-

Table 7.1: Python fragility script for determining the failure depth.

```

import numpy as np
import math

water_depth = np.linspace(0.0, 94.5, num=946)

def prob( x ):
    regress=intercept+depth_coeff*x
    p=1/(1+math.exp(-(regress)))
    return p

intercept= -75.68
depth_coeff= 2.05

fragility = [prob(d) for d in water_depth]

n=1
for p, d in zip(fragility, water_depth):
    failure = np.random.binomial(n,p,1)
    if failure == 1:
        depth_failure = d
        break

depth_meter = round(depth_failure * 0.0254,3)

#Write failure depth to file
outPath = '/Users/Alison/depth_meter.csv'
with open(outPath, 'w') as outputFile:
    outputFile.write(str(depth_meter))

```

able area for future research. To generate a flowchart similar to Figure 7.3 for use with continuous distributions, the fragility script would require modification. This would necessitate further investigation and research, including thought into generalizing the approach for use with different NPP configurations.

7.4 Simulation Results

For the resulting simulation, the default fluid properties provided by Neutrino for water were used. A user is allowed to change parameters, such as viscosity or temperature. Such adjustments were not necessary for coupling demonstration. The flow rate on the particle emitter was set to 0.32 cubic meters per second (~ 5000

Table 7.2: Python failure script for setting door failure when corresponding depth is reached.

```
import math
import csv

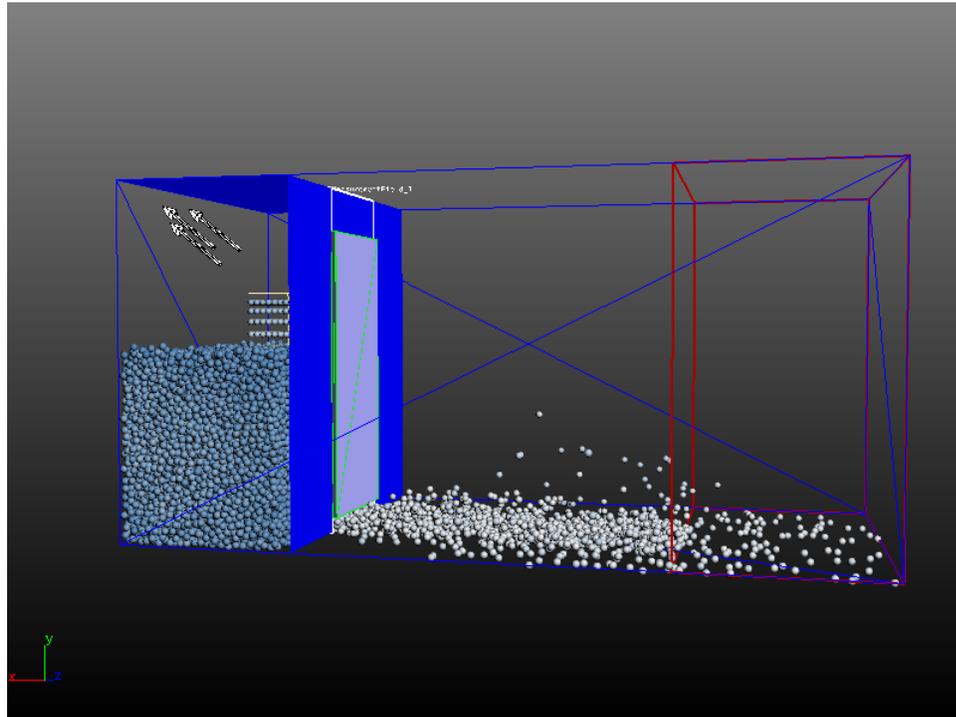
Path = 'E:\AlisonWells\depth_meter.csv'
inputFile = open(Path, 'r')
inputReader = csv.reader(inputFile, delimiter=',')
data = list(inputReader)
failure_depth = []
current_depth = []
inputFile.close()

value1 = float(data[0][0])
failure_depth.append(value1)
current_depth.append(MeasurementField_1.avgFluidHeight)
print(failure_depth, current_depth)

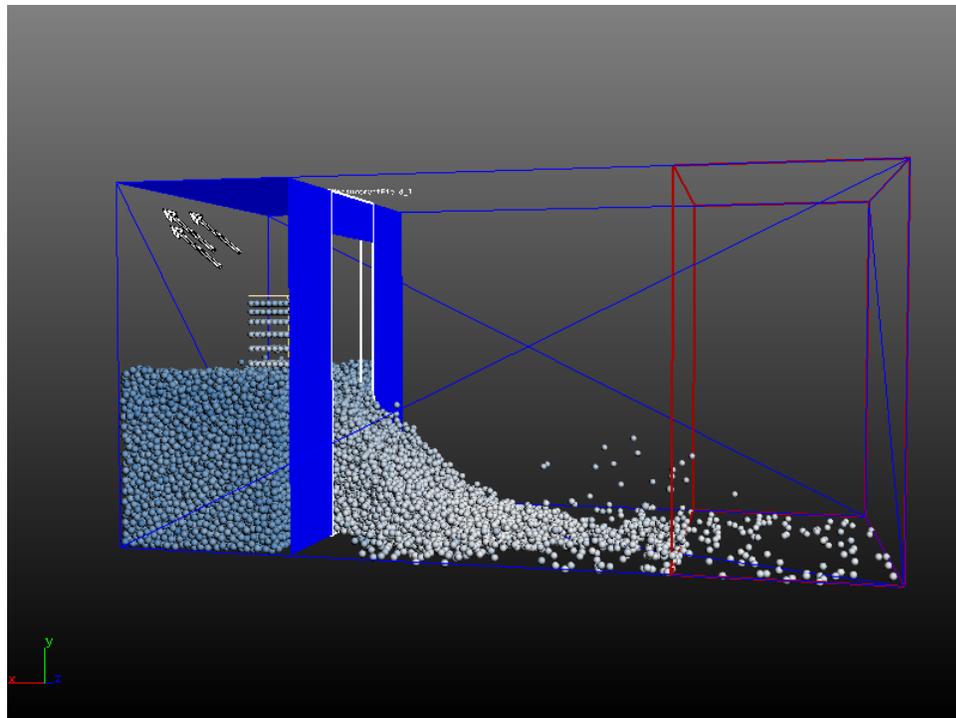
if current_depth >= failure_depth:
    Scene.deleteObject("Rigid Cuboid", "Door", True)
```

gal/min). This higher flow rate was chosen to reduce the length of the simulation and guarantee the room fill rate was greater than leakage under the door. Another variable that can be changed is the particle spacing size. Setting a large particle spacing would prevent particles leakage under the door, while very small particle spacing increases the simulation runtime. For this simulation, the particle spacing was set to 0.05 m.

The run was simulated for 2000 frames, where each frame represents 0.02 seconds of actual time. The total run time is 40 seconds. The fragility script calculated a failure depth of 0.919 meters (36.2 in.) and Neutrino compared this value against the average fluid height obtained from the measurement field. The door failed and was removed from simulation at time step 1429. Figure 7.5(a) shows the simulation one frame before failure and Figure 7.5(b) 20 frames after failure. The resulting fluid depths achieved in the room during the simulation run are shown in Figure 7.6.



(a) Simulation one frame before door failure.



(b) Simulation 20 frames after door failure.

Figure 7.5: Water depth vs. time plots for hollow core door tests.

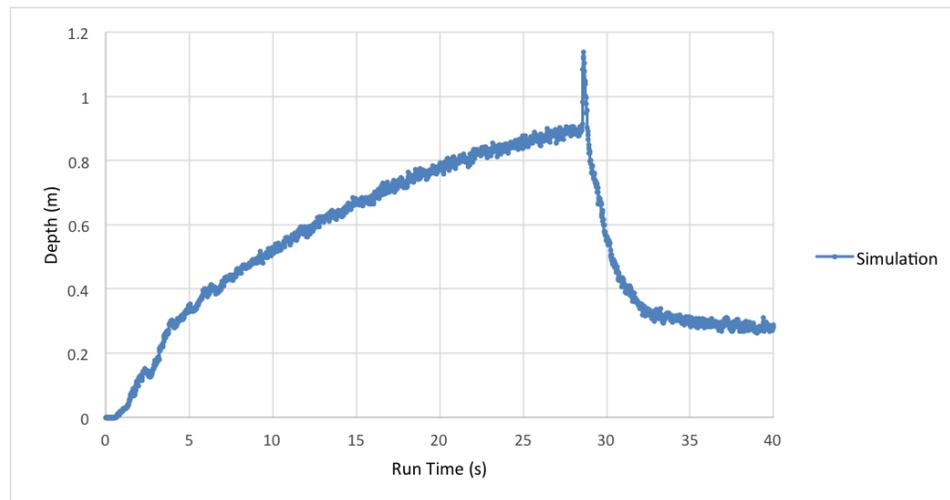


Figure 7.6: Fluid height versus time frame for SPH simulation.

The resulting simulation plot replicates the shape and behavior of the door failure plots presented in Chapter 5. The inclusion of the experiment based component fragility in flooding simulation improves the simulation, although more work needs to be done with SPH and specifically Neutrino to increase the ability of SPH to model flooding scenarios realistically. This simulation result is only presented as a demonstration of how component fragility modeling can be linked the 3D flooding simulation.

Chapter 8

Conclusions

The response of components to flooding conditions can have a considerable impact on the stable operation of NPPs. This work set out to establish a protocol for assessing NPP component fragility in flooding events. This first required conducting full-scale experiments to provide the necessary data on component performance when in contact with floodwater. These experiments focused on two types of non-watertight doors: hollow core and steel.

A Bayesian analysis featuring experiment-driven regression modeling was chosen to assess if and what observable phenomena drive failure. The availability of modern software tools like OpenBUGS, which utilizes MCMC methods, makes these types of calculations straightforward and tractable. For the case of doors, the water depth, flow rate, and temperature were included as possible indicators of failure.

Background analysis using a discrete distribution model found that models incorporating water depth as a primary explanatory variable had a better fit. However, initial shortcomings from using the binomial distribution meant alternative continuous aleatory models were also considered for assessing component fragility. The concluding developed Bayesian model uses the binomial distribution with a logistic regression model for probability of failure and uses non-informative priors so experimental data takes preference in the analysis. Added robustness measures were implemented for when demand was placed on a single component and to prevent illogical results from the discrepancy functions when calculating p-values. A verification of the developed model was performed using the preliminary example data.

Analysis of the experimental door data further established how component tests should be conducted. Hollow core doors were initially tested to gain experience and developed testing methodology for future experiments. The final analysis highlighted that observables, such as flow rate, should be tested over a large range of values. This includes specifically carrying out tests to obtain non-failure results. The quality of data affects the ability of the regression to adequately represent a components fragility. Additionally, analyzing the steel door data revealed that biases can be introduced during testing. Careful consideration should be given to all observables before experiments are performed and what additional tests need to be conducted to prevent an observable bias during fragility modeling.

Finally, an application of the component fragility model was demonstrated to show how improvements might be made to 3D flooding simulations. For a steel door, the depth regression fragility model was coupled to the SPH simulation code Neutrino via Python dynamic expression scripts. A simplified version of the PET was built inside the simulation environment and door failure was determined by sampling the fragility curve. When the average fluid height reached the predetermined failure depth, the door is removed from the simulation. No longer a barrier against the water particles, the remaining water flooded out. The resulting simulations replicate the observed door failures in experiments.

The work presented is a significant step towards better characterization and improved understanding of the reliability of NPP components in flood events. Component flooding experiments should continue to be conducted and analyzed using Bayesian regression methods established in this work. By analyzing the probabilistic risk associated with component flooding, data informed fragility modeling can be used to improve simulation and be integrated into comprehensive NPP risk

models to enhance the overall risk understanding.

Appendix A

Nuclear Power Plant Flooding Events

To more clearly establish the need for CFEL and to provide a working level knowledge of the types of potential NPP flooding events and the components that merit selection for testing, an open literature review of US NPP flooding events was performed. Brief summaries of eighteen NPP flooding or near flooding events is provided below to help establish the framework of potential scenarios and to help identify high priority components to be tested. The listing is not considered all inclusive, rather the listing is intended to demonstrate the broad scope of actual nuclear power plant flooding events and the identification of configurations that could be vulnerable to flooding events.

A.1 Indian Point Energy Center, Unit 2 (New York) [40]

On October 24, 1980 workers entered the containment building to repair a nuclear instrument. It was discovered that several inches of water, amounting to over 10,000 gallons, had accumulated on the containment floor. This accumulation flooded the reactor vessel pit and submerge the lower nine feet of the reactor vessel while the reactor was at operating temperature. This put thermal stress on the vessel and increased its chance of rupturing.

The flooded condition resulted from the following combination of conditions. There were numerous leaks from containment piping and fan coolers, which had a history of leaking. Both containment sump pumps were inoperable-one due to blown fuses and the other due to binding of its float switch. There was no high-level water alarm and the range of the sump level indicating light failed to indicate the overflowing sump level. The moisture level indicators for the containment did

not reveal the flood, because they were calibrated and/or ranged to sense high humidity caused by hot water or steam and were insensitive to the moisture levels resulting from small cold water leaks. The rising water was eventually detected when it submerged a power range nuclear detector, causing it to short out and workers entered containment to repair it.

A.2 Sequoyah Nuclear Plant, Unit 1 (Tennessee) [41]

On February 11, 1981, the Unit 1 reactor was in cold shutdown. An auxiliary unit worker, who did not have adequate training or orientation at the particular duty station involved, misunderstood a verbal instruction and opened a single valve in the residual heat removal system. A rapid primary system depressurization to the atmospheric pressure occurred. A total of about 110,000 gallons of water sprayed into the containment from the primary system and from the refueling water storage tank. The open valve was not detected for at least 35 minutes, because several alarm lights were already on as part of the shutdown cooling.

A.3 Cooper Nuclear Station (Nebraska) [42]

On April 4, 1984, construction workers used a bulldozer to do some landscaping on the plant's grounds. The worker sheared off a fire hydrant from the fire protection system. The flow from the broken hydrant caused the pressure in the fire header piping to drop. When the hydrant was isolated and the system re-pressurized, a water hammer forced the standby gas treatment (SBGT) system fire suppression deluge valve open and released water into the area. The water pouring in flooded the SBGT's charcoal filters. Both trains of the SBGT system were rendered inoperable.

A.4 Connecticut Yankee Nuclear Power Plant (Connecticut) [43]

On August 21, 1984, workers prepared to remove equipment from the reactor vessel during a refueling outage, and filled the reactor cavity with water. The plant experienced a failure of the refueling cavity water pneumatic seal when it was forced out of the normal position by static water pressure. The cavity water level decreased to the level of the reactor vessel flange in approximately 20 minutes. This flooded the containment with about 200,000 gallons of water.

If reactor fuel had been in transfer at the time, it could have been partially or completely uncovered. This could have led to possible high radiation levels, fuel cladding failure, and release of radioactivity. Additionally, if the fuel transfer tube had been open, the spent fuel pool could have drained to a level that would have uncovered the top of the fuel assemblies stored in the pool.

A.5 LaSalle County Nuclear Generating Station (Illinois) [44]

On May 31, 1985 a 108-inch circulating water pump expansion joint failed due to water hammer in the Lake Screen pump house. It resulted in a 2000 GPM flood until the water level reached approximately 15 feet, matching the lakes level.

A.6 Edwin I. Hatch Nuclear Power Plant, Unit 1 (Georgia) [45]

On December 21, 1985, the Unit 1 reactor was shut down for a refueling outage with a residual heat removal (RHR) loop suction valve also disassembled for repairs. A maintenance isolation switch, located between the RHR valve and the torus, was shut. The maintenance isolation valve is air-operated and opens on loss of power to its solenoid valve.

Power to the solenoid valve was lost when the station conducted a planned

loss-of-offsite-power test. An emergency core cooling system pump room at the plant was flooded to a level of 14 feet when the maintenance isolation valve lost power and opened. The water level in the pump room stopped rising when the depth equalized with the torus water level. Both RHR pumps and one core spray pump were disabled when they became submerged.

A.7 River Bend Nuclear Generating Station (Louisiana) [46]

On April 19, 1989 a freeze seal was used to allow repair on two manual isolation valves that connected to a safety-related auxiliary building cooler. The freeze failed when the flow of nitrogen through a specialized blanket, placed around the 6 inch service waterline, stopped. The standard precaution of moderating the frozen seal's temperature has not been followed by workers. The failure resulted in approximately 15,000 gallons of service water being discharged through the disassembled isolation valve, covering about 4 inches of the auxiliary building's upper floor.

A portion of the water seeped through holes in the floor and into nonsafety-related cabinets containing disconnect links and a transformer. Several circuits shorted and caused a fire that damaged one cabinet and its components. A supply breaker opened and de-energized the cabinet and two others, causing the loss of the operating residual heat removal (RHR) system, spent fuel cooling, and lighting in the auxiliary building, the control building, and the reactor building. The operators isolated the service water system in 15 minutes and restarted the RHR system in 17 minutes.

A.8 Perry Nuclear Power Plant (Ohio) [47]

On December 22, 1991, a 36 inch fiberglass auxiliary circulating water line failed. The failure occurred in a non-isolable section located in the yard area just before the line enters the heater bay building. The piping originated in the pump House located on the shores of Lake Erie. An estimated 2.9 million gallons of water was pumped out of the lake before workers turned off the pumps.

A small percentage of this water enter the auxiliary building, the heater bay, the service pump house, and the emergency service water pump house. The water flowed through two electrical manholes that had previously leaked during rain. These manholes were not designed to be leak tight against standing water. Conduits within formed a direct path into the buildings. Specifically, water entered the emergency water pump house through a series of conduits and ran down a cable into a compartment of a motor control center. A short circuit then occurred in a space heater transformer.

A.9 Sequoyah Nuclear Generating Station (Tennessee) [48]

On July 11, 1994, heavy rainfall (1 inch in 15 minutes) caused the storm drainage system to backup and flood the turbine building railroad bay (TBRB), cable tunnels, electrical manholes, and other areas on site. Water intruded into the 6.9 kV unit boards to approximately 2 inches deep and poured through the conduits, dousing the distribution boards and the distribution panels. The electrical equipment, however, remained functional.

A similar flooding event occurred on June 30, 1999 when the storm drainage system was unable to accommodate the combined demands of a heavy rainfall (0.67 inches in 15 minutes) and the estimated 800 GPM discharge flow from a tem-

porary plant modification. Both units' main generator bus duct coolers' were discharged to the storm drain at the entrance of the TBRB. Inadequate performance of the storm drain system caused water to backup and again flood the TBRB. Water, about 1 inch deep at the base of the unit boards, came to within 3 to 4 inches of contacting energized control circuitry and came within approximately 5 inches of contacting the board insulators. Equipment remained functional and no trip, transient, or engineered safety feature actuation occurred. Both units operated at 100% power throughout the event.

The flooding condition was determined to be a result of improper surface grading, ground water in-leakage, and failure to perform preventive maintenance on the storm drain system.

A.10 Columbia Generating Station (Washington) [49]

On June 17, 1998, WNP-2 was in cold shutdown and preparations were underway for plant startup. This event was initiated by workers performing cutting and grinding activities in the diesel generator building. The fire detectors, sensing smoke from the maintenance activities, activated a fire protection station. Three main fire pumps automatically started, but no water sprayed from the fire sprinklers. The nozzles were still blocked by metal caps that normally melt during an actual fire.

A resulting water hammer caused a 12 inch, cast-iron, fire protection isolation valve to rupture in the fire protection system riser in the northeast stairwell of the reactor building. Water from the stairwell entered a residual heat removal (RHR) pump room through a watertight door that had not been adequately secured and began flooding the room. A reactor drains system valve failed to close as designed and allowed water to flow into the low-pressure core spray (LPCS) pump room.

The flood water completely submerged a RHR pump and motor and the Division II keepfill pump also located in the room. Water in the LPCS pump room rose to a level just below the pump motor and also completely submerged the minimum flow valve and the Division I keepfill pump. Approximately 163,000 gallons of water were introduced to these three areas.

A.11 Susquehanna Steam Electric Station (Pennsylvania) [50] [51]

On July 16, 2010, operators had to manually scram the Unit 1 reactor due to a non-isolable circulating water leak in the main condenser bay. Two large gasketed manway hatches were dislodged and extruded from the manway joint when a system pressure spike was experienced. The control room received a maintenance condenser bay flooding alarm and responding workers observed water in the lower level, near the vicinity of the main condenser waterbox. After manually down-powering and unsuccessful attempts to isolate the leak, Unit 1 was scrammed in order to secure the circulating water pumps, isolate the circulating water system, and isolate the main condenser. After approximately $3\frac{1}{2}$ hours, the circulating water motor-operated valves were manually shut, isolating the leak. Approximately 12 feet of water entered the main condenser bay, corresponding to about 950,000 gallons of river water.

A.12 Calvert Cliffs Nuclear Power Plant (Maryland) [52] [53]

On January 21, 2014, a dual reactor trip from 100 percent power occurred at the plant after a loss of power to a 13 kV service bus. Ice and snow from a winter storm had caused a ventilation filter on the bus to push in and allow snow and water to contact with breakers inside. Loss of the service bus resulted in a loss of the Unit 2 motor generator sets for the control element drive mechanisms. This likely cause

the automatic reactor trip on Unit 2. It also resulted in the loss of power to the Unit 2 circulating water pumps, and the main condenser was unavailable.

Additionally, due to the bus loss, power was lost to one safety-related 4kV bus on both units. One emergency diesel generator on each unit started as expected to supply power to its respective 4kV bus until offsite power was restored. The loss of the service bus also caused an unexpected automatic reactor trip of Unit 1. The preliminary cause of the Unit 1 trip was a malfunction of the digital turbine control system during the electrical transient following the loss of the service bus and the Unit 2 plant trip. The plant operators brought both units to a stable hot shutdown condition in accordance with plant operating procedures with no other complications.

A.13 Indian Point Energy Center, Unit 3 (New York) [54]

On May 9, 2015, a main transformer exploded and caught on fire. The heat from the transformer fire caused activation of the fire protection systems, and fire brigades were brought in. When the majority of the fire was out, the brigade leader ordered the fire deluge valves to be closed in order to apply foam to the fire. The responding worker reported water on the floor of the switchgear room and took steps to stop the flooding.

The valves that opened to spray water onto the transformer failed to operate properly and discharged water to the floor at a rate of 50 GPM for approximately 30 minutes. The floor drains were restricted and could only pass a maximum of 25 GPM. The water flooded the deluge valve room to a height of 4 to 6 inches and flowed underneath the door to the switchgear room, where it pooled to approximately 1 inch in depth. Later inspection found that if the floor flooded to a depth of about 5 inches, the water intrusion into the switchgear room could have chal-

lenged the reliability of the safety-related electrical equipment required to respond to a reactor transient and would have resulted in a station blackout.

A.14 LaSalle County Nuclear Generating Station (Illinois) [55]

On March 11, 1997 an unusual event was declared. It was determined that the lake level was above the maximum level given in the safety analysis report and used in flooding potential analysis.

Contributing factors included the lake blowdown valve was partially closed and made unavailable due to reliability concerns. Two of the three lake make-up pumps were in operation and the lake level began to steadily increase. Additionally, heavy rain and melting snow added about 8 inches to the lake level.

As cautious measures, workers monitored lower elevations of the plant, and increase the flood wall height with sand bags. The lake blowdown valve was repaired and lake levels begin to decrease and return to normal operating levels, terminating the unusual event. Operating procedures were revised after this event.

A.15 Oconee Nuclear Station (South Carolina) [56]

On August 13, 2003, workers removed a 6" x 10" access cover from a flood barrier to route temporary power cables into the standby shutdown facility (SSF) for an SSF outage. The bottom edge of the opening created was located 4.6 feet off the ground. The plant's safety studies reported site flooding capabilities of 4.71 feet. During an external flooding event, the breached flood protection barrier could have provided a flowpath for water to enter the SSF, and render the equipment inoperable.

On June 2, 2005, NRC inspectors notify the licensee of the condition and issue and Problem Investigation Program report (PIP). Corrective action was not taken

and in August the licensee was issued a further PIP. The opening was sealed on August 3, 2005.

A.16 Vermont Yankee Nuclear Power Plant (Vermont) [57]

On May 24, 2012, workers discovered the potential for water intrusion into the vital switchgear rooms via an underground spare conduit, a metal tube encasing electrical cables. The flood seal used in this spare conduit was not adequate to prevent dislodgment. The flood seal could allow water flow into the interior flood design controls for the switchgear rooms. Were flooding of the switchgear rooms to occur, the operability of switchgear providing electrical power to engineered safety feature systems and emergency core cooling systems could be affected. This could threaten the capability to shut down the reactor and maintain it in a safe shutdown condition.

A.17 Three Mile Island Nuclear Generating Station (Pennsylvania) [58]

On August 10, 2012, a Unit 1 flood inspection walkdown found that conduits carrying cabling from yard electrical vaults through the air intake tunnel to the auxiliary building did not contain internal seals for flood protection. The conduit seals were never installed during the plant's construction. The unsealed conduits could have potentially provide a leak path during flood conditions to the auxiliary building, where flood waters could impact the decay heat removal function.

A.18 Millstone Nuclear Power Stations, Unit 2 (Connecticut) [59]

On October 15, 2012 during a beyond design basis flooding walkdown, it was discovered that a total of 20 four-inch and two-inch diameter electrical conduits were lacking flood seals as specified in the plants original design. These unsealed

electrical conduits connected the service water pump room of the intake structure and the turbine building. Since these conduits were unsealed at both ends, this condition could have resulted in flooding of the turbine building such that it would have rendered all the auxiliary feed water pumps inoperable. Upon discovering this condition, the openings on both ends of the conduits were sealed to restore the design basis for flood protection.

Appendix B

Basic User Guide to OpenBUGS

OpenBUGS is the open source version of the BUGS software for implementing Markov chain Monte Carlo (MCMC) sampling. The freely available software can be found at www.openbugs.info.

The following provides basic guidance for running OpenBUGS scripts and is not intended to be a complete operating guide. The OpenBUGS user manual [24] comes with the program, accessible from the Help menu, and offers further information and tutorials. Additional resources are also available to provide introduction to this program and examples [20] [19] [31].

B.1 OpenBUGS Script

An OpenBUGS script can be divided into three sections: model, data, and initial values. A sample script is provided in Table B.1. The model includes the likelihood function, prior distributions, and any derived quantities, such as residuals or Bayesian p-values. The data and initial values can be listed within the script or loaded from separate text files. Additionally, initial values can also be generated by OpenBUGS rather than loading in provided values.

B.2 Running OpenBUGS models

To begin, make a new document (File→New) and type in the BUGS model code or open a document with an already written code (File→Open). Figure B.1 shows a script document open in the OpenBUGS screen.

The first step is checking the syntax of the model specification. Open Specification Tool from the Model menu. This may appear as Specification... on the

Table B.1: An example OpenBUGS script for a Binomial model.

```
#Binomial Model

model{

  for(i in 1:10){
    Y[i] ~ dbin(p, 1)
  }

  #Prior Distributions
  p ~ dnorm(0, .0001)
}

data
list(Y= c(0,0,1,0,0,1,1,1,1,1))

inits
list(p= 0)
```

drop-down menu. A window like the one shown in Figure B.2 will appear. Highlight the word "model" in the BUGS script by double-clicking the word. Click on "check model" in the Specification Tool screen. Status messages are displayed at the bottom left of the OpenBUGS screen. If there are no errors, the message "model is syntactically correct" will appear.

Now any provided data is loaded. Highlight the word "list" in the data section of the script and click "load data" on the Specification Tool. The status message will display "data loaded."

Next the BUGS model needs to be compiled, which means the program constructs an internal representation and works out sampling methods to be used for each stochastic node. First, input the number of parallel chains to run in the "num of chains" box. The default value is 1. Click "compile" and the status message will

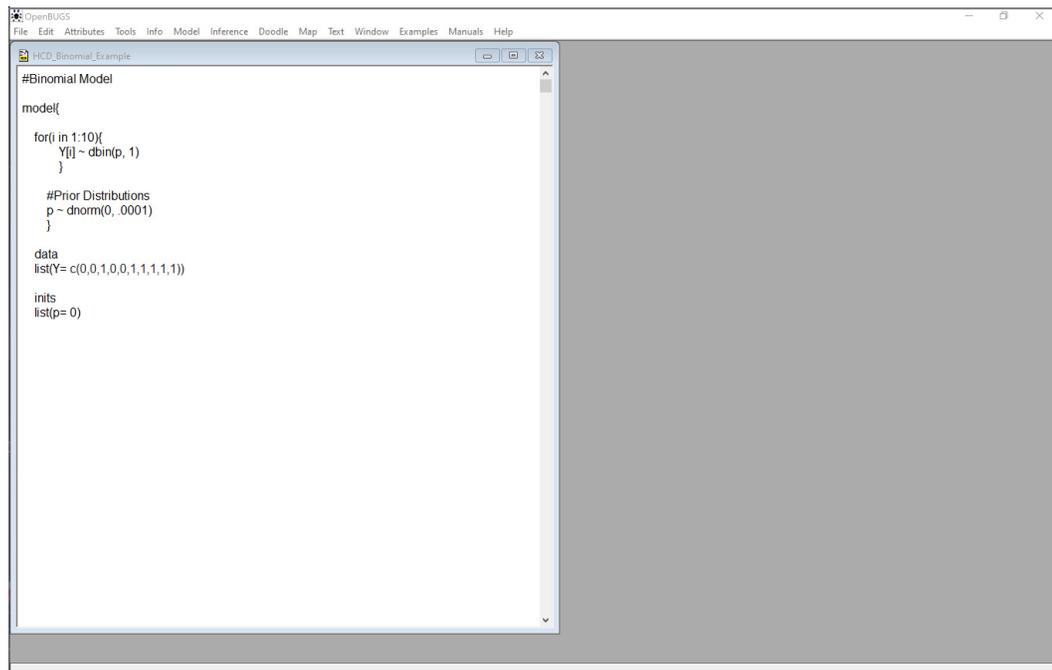


Figure B.1: OpenBUGS screen with an open script.

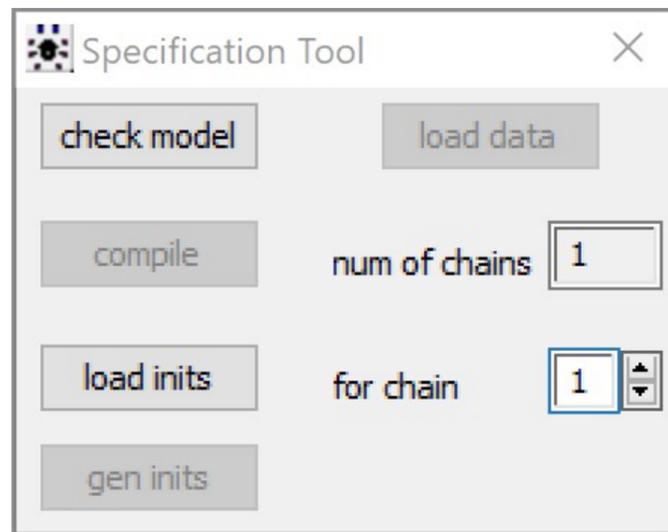


Figure B.2: The Specification Tool.

now report "model compiled."

The simulation is started at set values for unknown quantities by loading initial values. Highlight the word "list" in the initial value section of the script, then click

the "load inits" button on the Specification Tool. The status message will display "initial values generated, model initialized" if only one chain is run. If more than one chain is being run, advance the "for chain" number by clicking the up arrow and repeat loading initial values for each chain until the model initialized message appears. As noted before, OpenBUGS can generate the initial values by clicking the "gen inits" button instead.

The next step is to select the nodes for the program to monitor. Open Samples from the Inference menu and the Sample Monitor Tool (Figure B.3) will appear. Type the name of the node in the "node" box that will be monitored or type * to monitor all available nodes. In order to discard samples for burn-in, change the value in the "beg" box to the iteration at which monitoring starts. Click the "set" button to save the setting. Enter any additional nodes to monitor in the "node" box, clicking "set" after each one.

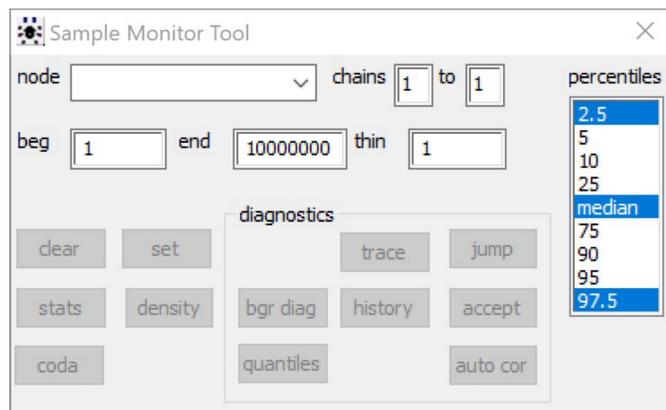


Figure B.3: The Sample Monitor Tool.

Now open Update from the Model menu. The Update Tool will appear, as shown in Figure B.4. Enter the number of updates to preform and click "update." The status message will read "model is updating." The iteration box in the Update Tool will display the iterations according to the increment specified by the refresh

box. Once the update has completed, the status message will display "...updates took X s."

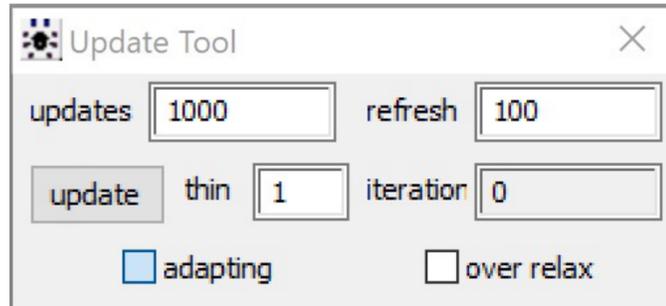


Figure B.4: The Update Tool.

Back in the Sample Monitor Tool, type the name of the node, select the node from the drop-down list of monitored nodes, or type * to select all nodes monitored. Click "stats" to see summary statistics such as posterior mean, standard deviation, and selected percentiles. Click "density" to display a graph of the empirical distributions. These summary statistics and plots can be copied and pasted into other programs.

Appendix C

OpenBugs Scripts and Results

C.1 Scripts and Statistical Summaries from Section 6.1.1

C.1.1 Binomial depth model with logit regression and hollow core door data.

```
#Depth (D) Model
model {
  for(i in 1:tests) {
    failure[i] ~ dbin(p[i], num.tested)
    # Regression model
    logit(p[i]) <- int + a*depth[i]
    #failure.rep[i] ~ dbin(p[i], num.tested) # Replicate values for model validation
    #diff.obs[i] <- pow(failure[i] - num.tested*p[i], 2)/(num.tested*p[i]*(1-p[i]))
    #diff.rep[i] <- pow(failure.rep[i] - num.tested*p[i], 2)/(num.tested*p[i]*(1-p[i]))
  }
  #chisq.obs <- sum(diff.obs[])
  #chisq.rep <- sum(diff.rep[])
  #p.value <- step(chisq.rep - chisq.obs)
  # Prior distributions
  int ~ dnorm(0, 0.0001)
  a ~ dnorm(0, 0.001)
}
data
list(num.tested=1, tests=8, depth = c(23.23,20.75,42.3,21.05,24.22,35.41,40.76,38.85), failure = c(0,0,1,0,0,1,1,1))
inits
list(int=0, a=0)
```

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
a	4.386	2.317	0.05865	0.9092	4.055	9.729	1001	99000
int	-126.3	64.8	1.639	-272.9	-118.0	-26.74	1001	99000

C.1.2 Binomial model using cloglog and hollow core door data.

```

#Hollow Core Door Data
#Binomial Model using Cloglog Link Function
model {
  for(i in 1:tests) {
    failure[i] ~ dbin(p[i], num.tested)
    # Regression model
    #cloglog(p[i]) <- int + a*depth[i]
    cloglog(p[i]) <- int + b*flow[i]
    #cloglog(p[i]) <- int + c*temp[i]
    #cloglog(p[i]) <- int + a*depth[i]+ b*flow[i]
    #cloglog(p[i]) <- int + a*depth[i]+ c*temp[i]
    #cloglog(p[i]) <- int + b*flow[i]+ c*temp[i]
    #cloglog(p[i]) <- int + a*depth[i]+ b*flow[i]+ c*temp[i]

    #failure.rep[i] ~ dbin(p[i], num.tested) # Replicate values for model validation
    #diff.obs[i] <- pow(failure[i] - num.tested*p[i], 2)/(num.tested*p[i]*(1-p[i]))
    #diff.rep[i] <- pow(failure.rep[i] - num.tested*p[i], 2)/(num.tested*p[i]*(1-p[i]))
  }
  #chisq.obs <- sum(diff.obs[])
  #chisq.rep <- sum(diff.rep[])
  #p.value <- step(chisq.rep - chisq.obs)
  # Prior distributions
  int ~ dnorm(0, 0.0001)
  a ~ dnorm(0, 0.001)
  b ~ dnorm(0, 0.001)
  c ~ dnorm(0, 0.001)
}
data
  list(num.tested=1, tests=8,
       depth = c(23.23,20.75,42.3,21.05,24.22,35.41,40.76,38.85),
       flow = c(291.5,292.5,292.5,297,294.5,292.5,291,294) ,
       temp = c(65.98,67.04,66.02,67.67,66.6,66.87,68.33,68.14),
       failure = c(0,0,1,0,0,1,1,1))
inits
list(int=0, a=0, b=0, c=0)

```

Depth Model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
a	4.46	2.282	0.08684	0.6702	4.222	9.512	1001	199000
int	-129.0	63.93	2.451	-268.1	-122.9	-20.87	1001	199000

	Dbar	Dhat	DIC	pD
failure	0.1719	8.935E-9	0.3439	0.1719
total	0.1719	8.935E-9	0.3439	0.1719

Flow Rate Model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
b	-0.03135	0.06679	0.003161	-0.1372	-0.02193	0.08251	1001	199000
int	8.692	19.6	0.9275	-24.83	5.954	39.67	1001	199000

	Dbar	Dhat	DIC	pD
failure	11.58	10.59	12.58	0.9954
total	11.58	10.59	12.58	0.9954

Temperature Model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
c	0.1879	0.1919	0.009072	-0.09623	0.1397	0.6931	1001	199000
int	-13.09	12.88	0.609	-47.06	-9.926	5.974	1001	199000

	Dbar	Dhat	DIC	pD
failure	11.94	10.97	12.92	0.975
total	11.94	10.97	12.92	0.975

Depth and Flow Rate Model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
a	40.55	19.53	0.7316	8.344	38.1	85.4	1001	199000
b	-4.125	2.058	0.07648	-8.935	-3.829	-0.836	1001	199000
int	0.7463	99.37	1.353	-192.3	-0.03126	195.7	1001	199000

	Dbar	Dhat	DIC	pD
failure	0.01401	-1.776E-15	0.02802	0.01401
total	0.01401	-1.776E-15	0.02802	0.01401

Depth and Temperature Model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
a	37.39	18.42	0.7135	7.163	35.63	78.46	1001	199000
c	-16.53	8.432	0.326	-35.33	-15.7	-1.679	1001	199000
int	-4.827	98.47	1.487	-197.6	-6.129	189.0	1001	199000

	Dbar	Dhat	DIC	pD
failure	0.01308	-1.776E-15	0.02615	0.01308
total	0.01308	-1.776E-15	0.02615	0.01308

Flow Rate and Temperature Model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
b	-0.226	0.0757	0.003584	-0.374	-0.2167	-0.06405	1001	199000
c	0.5369	0.4016	0.01901	-0.09096	0.4379	1.249	1001	199000
int	29.81	12.89	0.6085	9.719	33.0	50.49	1001	199000

	Dbar	Dhat	DIC	pD
failure	9.748	8.486	11.01	1.263
total	9.748	8.486	11.01	1.263

Depth, Flow Rate, and Temperature Model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
a	42.57	19.52	0.7677	12.09	40.31	87.04	1001	199000
b	-5.175	7.52	0.3522	-20.39	-4.645	6.861	1001	199000
c	3.613	31.61	1.478	-51.08	2.732	63.67	1001	199000
int	3.966	98.49	1.215	-190.5	4.611	197.6	1001	199000

	Dbar	Dhat	DIC	pD
failure	0.01133	-1.776E-15	0.02266	0.01133
total	0.01133	-1.776E-15	0.02266	0.01133

C.2 Scripts and Statistical Summaries from Section 6.1.2

C.2.1 Simple exponential model with hollow core door failure data only.

```

#Hollow Core Door Data
#Exponential Model with No Enviroment; P-Value

model {
  for(i in 1:N) {
    time[i] ~ dexp(lambda)
    time.rep[i] ~ dexp(lambda)
    #Rank observed times a plicate times
    time.ranked[i]<- ranked(time[],i)
    time.rep.ranked[i]<-ranked(time.rep[],i)
    #Calculate components of Cramer-von Mises statistic for observed and replicated data
    F.obs[i]<-cumulative(time[i], time.ranked[i])
    F.rep[i]<-cumulative(time.rep[i], time.rep.ranked[i])
    diff.obs[i]<-pow(F.obs[i] - (2*i-1)/(2*N),2)
    diff.rep[i]<-pow(F.rep[i] - (2*i-1)/(2*N),2)
  }
  #Calculate distribution of Cramer-von Mises statistic for observed and replicated data
  CVM.obs<-sum(diff.obs[])
  CVM.rep<-sum(diff.rep[])
  p.value<-step(CVM.rep-CVM.obs)
  # Prior distributions
  #lambda ~ dnorm(0, 0.0001)
  lambda ~ dgamma(0.0001, 0.0001) #Jeffreys prior for lambda
}
data
list(time =c(3.533, 2.667, 0.8667, 0.4667), N=4)
inits
list(lambda=1)

```

Using Jeffreys Prior:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
lambda	0.5309	0.2655	6.142E-4	0.1452	0.4866	1.164	1001	199000
p.value	0.6309	0.4826	0.001062	0.0	1.0	1.0	1001	199000

	Dbar	Dhat	DIC	pD
time	14.1	13.06	15.13	1.033
total	14.1	13.06	15.13	1.033

Using Normal Prior:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
lambda	0.6617	0.2958	0.001484	0.213	0.618	1.36	1001	199000
p.value	0.6008	0.4897	0.001686	0.0	1.0	1.0	1001	199000

	Dbar	Dhat	DIC	pD
time	14.11	13.28	14.93	0.8274
total	14.11	13.28	14.93	0.8274

C.2.2 Exponential model using hollow core door data.

```

#Hollow Core Data
#Exponential Model Using Flow Rate (F), Depth (D), and Temperature (T); P-Value

model {
  for(i in 1:N) {
    time[i] ~ dexp(lambda[i])
    time.rep[i] ~ dexp(lambda[i])
    #lambda[i] <- int + a*D[i]
    lambda[i] <- int + b*F[i]
    #lambda[i] <- int + c*T[i]
    #lambda[i] <- int + a*D[i] +b*F[i]
    #lambda[i] <- int + a*D[i] +c*T[i]
    #lambda[i] <- int + b*F[i] +c*T[i]
    #lambda[i] <- int + a*D[i] +b*F[i] +c*T[i]
    #Rank observed times a plicate times
    time.ranked[i]<- ranked(time[i],i)
    time.rep.ranked[i]<-ranked(time.rep[i],i)
    #Calculate components of Cramer-von Mises statistic for observed and replicated data
    F.obs[i]<-cumulative(time[i], time.ranked[i])
    F.rep[i]<-cumulative(time.rep[i], time.rep.ranked[i])
    diff.obs[i]<-pow(F.obs[i] - (2*i-1)/(2*N),2)
    diff.rep[i]<-pow(F.rep[i] - (2*i-1)/(2*N),2)
  }
  #Calculate distribution of Cramer-von Mises statistic for observed and replicated data
  CVM.obs<-sum(diff.obs[])
  CVM.rep<-sum(diff.rep[])
  p.value<-step(CVM.rep-CVM.obs)
  # Prior distributions
  int ~ dnorm(0, 0.0001)
  a ~ dnorm(0, 0.0001)
  b ~ dnorm(0, 0.0001)
  c ~ dnorm(0, 0.0001)
}
data
list(time =c(3.533, 2.667, 0.8667, 0.4667), N=4,
F =c(290.6, 294.07, 411.25, 472.89), D =c(40.74, 38.77, 32.28, 43.34), T =c(69.63, 69.02, 62.18, 63.49))
inits
list(int=1,a=1,b=1,c=1)

```

Depth Model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
a	-0.0402	0.1121	0.005292	-0.213	-0.04579	0.255	1001	199000
int	2.49	4.302	0.203	-7.997	2.462	9.608	1001	199000
p.value	0.4689	0.499	0.008561	0.0	0.0	1.0	1001	199000

	Dbar	Dhat	DIC	pD
time	14.56	13.51	15.61	1.05
total	14.56	13.51	15.61	1.05

Flow Rate Model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
b	0.1488	0.05106	0.002419	0.08296	0.145	0.2488	1001	199000
int	-42.92	14.85	0.7039	-71.94	-41.72	-23.83	1001	199000
p.value	0.8624	0.3445	0.004637	0.0	1.0	1.0	1001	199000

	Dbar	Dhat	DIC	pD
time	40.33	39.65	41.02	0.6838
total	40.33	39.65	41.02	0.6838

Temperature Model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
c	-0.03892	0.1607	0.007615	-0.1926	-0.1059	0.3821	1001	199000
int	3.553	10.47	0.496	-23.47	7.837	13.88	1001	199000
p.value	0.3392	0.4734	0.007256	0.0	0.0	1.0	1001	199000

	Dbar	Dhat	DIC	pD
time	12.28	11.44	13.12	0.841
total	12.28	11.44	13.12	0.841

Depth and Flow Rate Model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
a	-0.7446	0.2207	0.01045	-1.135	-0.8148	-0.3435	1001	199000
b	0.1619	0.05149	0.00244	0.09493	0.1509	0.2663	1001	199000
int	-16.44	8.28	0.3923	-32.0	-12.87	-6.736	1001	199000
p.value	0.04328	0.2035	0.004797	0.0	0.0	1.0	1001	199000

	Dbar	Dhat	DIC	pD
time	55.72	55.09	56.35	0.6314
total	55.72	55.09	56.35	0.6314

Depth and Temperature Model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
a	0.2098	0.112	0.005286	0.03587	0.1957	0.486	1001	199000
c	-0.3576	0.1236	0.005854	-0.5455	-0.3947	-0.1366	1001	199000
int	17.09	9.305	0.4408	-2.42	19.07	31.13	1001	199000
p.value	0.3741	0.4839	0.00662	0.0	0.0	1.0	1001	199000

	Dbar	Dhat	DIC	pD
time	12.96	12.03	13.89	0.9296
total	12.96	12.03	13.89	0.9296

Flow Rate and Temperature Model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
b	0.494	0.03732	0.001768	0.4414	0.4923	0.572	1001	199000
c	-1.969	0.134	0.006343	-2.138	-2.018	-1.735	1001	199000
int	-6.161	6.361	0.3013	-21.51	-5.551	4.559	1001	199000
p.value	4.02E-5	0.00634	1.41E-5	0.0	0.0	0.0	1001	199000

	Dbar	Dhat	DIC	pD
time	213.3	212.7	213.8	0.5229
total	213.3	212.7	213.8	0.5229

Depth, Flow Rate, and Temperature Model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
a	-0.9899	0.269	0.01274	-1.524	-1.061	-0.4955	1001	199000
b	0.2885	0.04951	0.002346	0.2055	0.2857	0.3749	1001	199000
c	-0.4438	0.08632	0.004084	-0.5563	-0.4626	-0.2419	1001	199000
int	-12.32	5.86	0.2773	-23.18	-13.2	-2.352	1001	199000
p.value	8.543E-5	0.009242	2.032E-5	0.0	0.0	0.0	1001	199000

	Dbar	Dhat	DIC	pD
time	114.4	113.8	115.0	0.59
total	114.4	113.8	115.0	0.59

C.2.3 Lognormal model using hollow core door data.

```

#Hollow Core Data
#Lognormal Model Using Flow Rate (F), Depth (D), and Temperature (T); P-Value
model {
  for(i in 1:N) {
    time[i] ~ dlnorm(mu[i], tau)
    time.rep[i] ~ dlnorm(mu[i], tau)
    #mu[i] <- int + a*D[i]
    #mu[i] <- int + b*F[i]
    mu[i] <- int + c*T[i]
    #mu[i] <- int + a*D[i] +b*F[i]
    #mu[i] <- int + a*D[i] +c*T[i]
    #mu[i] <- int + b*F[i] +c*T[i]
    #mu[i] <- int + a*D[i] +b*F[i] +c*T[i]
    #Rank observed times a plicate times
    time.ranked[i]<- ranked(time[],i)
    time.rep.ranked[i]<-ranked(time.rep[],i)
    #Calculate components of Cramer-von Mises statistic for observed and replicated data
    F.obs[i]<-cumulative(time[i], time.ranked[i])
    F.rep[i]<-cumulative(time.rep[i], time.rep.ranked[i])
    diff.obs[i]<-pow(F.obs[i] - (2*i-1)/(2*N),2)
    diff.rep[i]<-pow(F.rep[i] - (2*i-1)/(2*N),2)
  }
  #Calculate distribution of Cramer-von Mises statistic for observed and replicated data
  CVM.obs<-sum(diff.obs[])
  CVM.rep<-sum(diff.rep[])
  p.value<-step(CVM.rep-CVM.obs)
  tau <- pow(sigma, -2)
  sigma ~ dunif(0,10)
  # Prior distributions
  int ~ dnorm(0, 0.0001)
  a ~ dnorm(0, 0.0001)
  b ~ dnorm(0, 0.0001)
  c ~ dnorm(0, 0.0001)
}
data
list(time =c(3.533, 2.667, 0.8667, 0.4667), N=4,
F =c(290.6, 294.07, 411.25, 472.89), D =c(40.74, 38.77, 32.28, 43.34), T =c(69.63, 69.02, 62.18, 63.49))
inits
list(int=1,a=1,b=1,c=1)

```

Depth Model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
a	-0.008572	0.4	0.01012	-0.8835	-0.008084	0.8351	1001	199000
int	0.6652	15.59	0.3945	-32.25	0.6508	34.81	1001	199000
p.value	0.3276	0.4694	0.003545	0.0	0.0	1.0	1001	199000

	Dbar	Dhat	DIC	pD
time	19.29	18.34	20.25	0.957
total	19.29	18.34	20.25	0.957

Flow Rate Model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
b	-0.01059	0.007668	1.184E-4	-0.02237	-0.01051	2.214E-4	1001	199000
int	4.222	2.874	0.04426	0.1531	4.192	8.622	1001	199000
p.value	0.5501	0.4975	0.002613	0.0	1.0	1.0	1001	199000

	Dbar	Dhat	DIC	pD
time	3.617	5.544	1.69	-1.927
total	3.617	5.544	1.69	-1.927

Temperature Model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
c	0.237	0.2681	0.0102	-0.3508	0.2361	0.8091	1001	199000
int	-15.33	17.73	0.6743	-53.04	-15.26	23.49	1001	199000
p.value	0.4505	0.4975	0.005198	0.0	0.0	1.0	1001	199000

	Dbar	Dhat	DIC	pD
time	13.56	13.69	13.43	-0.1301
total	13.56	13.69	13.43	-0.1301

Depth and Flow Rate Model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
a	0.002279	0.3875	0.01026	-0.9108	0.004171	0.8555	1001	199000
b	-0.01066	0.02151	2.965E-4	-0.06055	-0.01052	0.03744	1001	199000
int	4.159	16.65	0.4542	-32.07	4.005	44.2	1001	199000
p.value	0.4236	0.4941	0.004258	0.0	0.0	1.0	1001	199000

	Dbar	Dhat	DIC	pD
time	14.45	16.54	12.37	-2.087
total	14.45	16.54	12.37	-2.087

Depth and Temperature Model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
a	-0.07574	0.4561	0.01289	-1.1	-0.08159	0.9578	1001	199000
c	0.2356	0.5059	0.01964	-1.044	0.2662	1.373	1001	199000
int	-12.3	30.04	1.156	-79.24	-14.11	62.62	1001	199000
p.value	0.406	0.4911	0.00726	0.0	0.0	1.0	1001	199000

	Dbar	Dhat	DIC	pD
time	14.55	16.5	12.6	-1.953
total	14.55	16.5	12.6	-1.953

Flow Rate and Temperature Model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
b	-0.00873	0.02464	8.761E-4	-0.06197	-0.00936	0.04797	1001	199000
c	0.04058	0.511	0.02216	-1.062	0.02838	1.229	1001	199000
int	0.8547	40.85	1.779	-94.56	1.822	88.03	1001	199000
p.value	0.462	0.4986	0.004086	0.0	0.0	1.0	1001	199000

	Dbar	Dhat	DIC	pD
time	12.73	14.67	10.79	-1.939
total	12.73	14.67	10.79	-1.939

Depth, Flow Rate, and Temperature Model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
a	0.04089	0.6594	0.02611	-1.349	0.07743	1.409	1001	199000
b	-0.01414	0.0528	0.002277	-0.1201	-0.017	0.09396	1001	199000
c	-0.104	1.284	0.05925	-2.723	-0.183	2.536	1001	199000
int	10.81	84.35	3.879	-165.2	15.36	177.5	1001	199000
p.value	0.436	0.4959	0.006942	0.0	0.0	1.0	1001	199000

	Dbar	Dhat	DIC	pD
time	14.6	17.28	11.92	-2.678
total	14.6	17.28	11.92	-2.678

C.2.4 Normal model using hollow core door data.

```

#Hollow Core Data
#Normal Model Using Flow Rate (F), Depth (D), and Temperature (T); P-Value
model {
  for(i in 1:N) {
    time[i] ~ dnorm(mu[i], tau)
    time.rep[i] ~ dnorm(mu[i], tau)
    mu[i] <- int + a*D[i]
    #mu[i] <- int + b*F[i]
    #mu[i] <- int + c*T[i]
    #mu[i] <- int + a*D[i] +b*F[i]
    #mu[i] <- int + a*D[i] +c*T[i]
    #mu[i] <- int + b*F[i] +c*T[i]
    #mu[i] <- int + a*D[i] +b*F[i] +c*T[i]
    #Rank observed times a plicate times
    time.ranked[i]<- ranked(time[],i)
    time.rep.ranked[i]<-ranked(time.rep[],i)
    #Calculate components of Cramer-von Mises statistic for observed and replicated data
    F.obs[i]<-cumulative(time[i], time.ranked[i])
    F.rep[i]<-cumulative(time.rep[i], time.rep.ranked[i])
    diff.obs[i]<-pow(F.obs[i] - (2*i-1)/(2*N),2)
    diff.rep[i]<-pow(F.rep[i] - (2*i-1)/(2*N),2)
  }
  #Calculate distribution of Cramer-von Mises statistic for observed and replicated data
  CVM.obs<-sum(diff.obs[])
  CVM.rep<-sum(diff.rep[])
  p.value<-step(CVM.rep-CVM.obs)
  tau <- pow(sigma, -2)
  sigma ~ dunif(0,10)
  # Prior distributions
  int ~ dnorm(0, 0.0001)
  a ~ dnorm(0, 0.0001)
  b ~ dnorm(0, 0.0001)
  c ~ dnorm(0, 0.0001)
}
data
list(time =c(3.533, 2.667, 0.8667, 0.4667), N=4,
      F =c(290.6, 294.07, 411.25, 472.89), D =c(40.74, 38.77, 32.28, 43.34), T =c(69.63, 69.02, 62.18, 63.49))
inits
list(int=1,a=1,b=1,c=1)

```

Depth Model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
a	0.03984	0.4826	0.01286	-1.007	0.04594	1.057	1001	199000
int	0.3359	18.82	0.5012	-39.29	0.09661	41.02	1001	199000
p.value	0.3345	0.4718	0.002969	0.0	0.0	1.0	1001	199000

	Dbar	Dhat	DIC	pD
time	19.37	17.99	20.76	1.384
total	19.37	17.99	20.76	1.384

Flow Rate Model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
b	-0.01521	0.01442	2.069E-4	-0.04394	-0.01542	0.016	1001	199000
int	7.464	5.406	0.07755	-4.243	7.546	18.22	1001	199000
p.value	0.4464	0.4971	0.00234	0.0	0.0	1.0	1001	199000

	Dbar	Dhat	DIC	pD
time	11.2	11.28	11.11	-0.08326
total	11.2	11.28	11.11	-0.08326

Temperature Model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
c	0.3461	0.3062	0.01207	-0.3439	0.3645	0.9033	1001	199000
int	-20.98	20.25	0.7984	-57.88	-22.2	24.69	1001	199000
p.value	0.4083	0.4915	0.00522	0.0	0.0	1.0	1001	199000

	Dbar	Dhat	DIC	pD
time	11.72	11.61	11.83	0.1122
total	11.72	11.61	11.83	0.1122

Depth and Flow Rate Model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
a	0.0836	0.4879	0.01272	-0.9648	0.06517	1.232	1001	199000
b	-0.01577	0.02533	3.513E-4	-0.07305	-0.01563	0.0412	1001	199000
int	4.426	20.62	0.5436	-43.75	5.096	48.77	1001	199000
p.value	0.4082	0.4915	0.003661	0.0	0.0	1.0	1001	199000

	Dbar	Dhat	DIC	pD
time	16.15	16.18	16.12	-0.03206
total	16.15	16.18	16.12	-0.03206

Depth and Temperature Model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
a	-0.08156	0.4861	0.01341	-1.284	-0.06938	0.9681	1001	199000
c	0.3908	0.5368	0.02048	-0.8905	0.3975	1.574	1001	199000
int	-20.77	32.15	1.178	-92.61	-21.63	57.47	1001	199000
p.value	0.4064	0.4912	0.005824	0.0	0.0	1.0	1001	199000

	Dbar	Dhat	DIC	pD
time	15.75	15.77	15.73	-0.0203
total	15.75	15.77	15.73	-0.0203

Flow Rate and Temperature Model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
b	-0.01062	0.02965	0.001061	-0.07879	-0.009506	0.05346	1001	199000
c	0.1193	0.5994	0.02605	-1.245	0.1543	1.308	1001	199000
int	-2.102	48.1	2.101	-96.63	-4.933	107.7	1001	199000
p.value	0.4502	0.4975	0.003091	0.0	0.0	1.0	1001	199000

	Dbar	Dhat	DIC	pD
time	14.2	14.32	14.08	-0.1212
total	14.2	14.32	14.08	-0.1212

Depth, Flow Rate, and Temperature Model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
a	0.07232	0.709	0.02689	-1.358	0.05116	1.475	1001	199000
b	-0.01581	0.06043	0.002617	-0.1634	-0.01195	0.09731	1001	199000
c	-0.01284	1.45	0.06689	-3.79	0.1153	2.615	1001	199000
int	5.729	97.62	4.489	-171.1	-2.026	260.7	1001	199000
p.value	0.4206	0.4937	0.004882	0.0	0.0	1.0	1001	199000

	Dbar	Dhat	DIC	pD
time	15.23	16.24	14.22	-1.007
total	15.23	16.24	14.22	-1.007

C.2.5 Weibull model with influence on beta using hollow core door data.

```

#Hollow Core Door Data
#Weibull Model (Beta) Using Flow Rate (F), Depth (D), and Temperature (T); P-Value
model {
  for(i in 1:N) {
    time[i] ~ dweib(beta[i], lambda)
    time.rep[i] ~ dweib(beta[i], lambda)
    #beta[i] <- int + a*D[i]
    beta[i] <- int + b*F[i]
    #beta[i] <- int + c*T[i]
    #beta[i] <- int + a*D[i] +b*F[i]
    #beta[i] <- int + a*D[i] +c*T[i]
    #beta[i] <- int + b*F[i] +c*T[i]
    #beta[i] <- int + a*D[i] +b*F[i] +c*T[i]
    #Rank observed times and duplicate times
    time.ranked[i]<- ranked(time[],i)
    time.rep.ranked[i]<-ranked(time.rep[],i)
    #Calculate components of Cramer-von Mises statistic for observed and replicated data
    F.obs[i]<-cumulative(time[i], time.ranked[i])
    F.rep[i]<-cumulative(time.rep[i], time.rep.ranked[i])
    diff.obs[i]<-pow(F.obs[i] - (2*i-1)/(2*N),2)
    diff.rep[i]<-pow(F.rep[i] - (2*i-1)/(2*N),2)
  }
  #Calculate distribution of Cramer-von Mises statistic for observed and replicated data
  CVM.obs<-sum(diff.obs[])
  CVM.rep<-sum(diff.rep[])
  p.value<-step(CVM.rep-CVM.obs)
  # Prior distributions
  lambda ~ dgamma(0.0001, 0.0001)
  int ~ dnorm(0, 0.0001)
  a ~ dnorm(0, 0.0001)
  b ~ dnorm(0, 0.0001)
  c ~ dnorm(0, 0.0001)
}
data
list(time =c(3.533, 2.667, 0.8667, 0.4667), N=4,
F =c(290.6, 294.07, 411.25, 472.89), D =c(40.74, 38.77, 32.28, 43.34), T =c(69.63, 69.02, 62.18, 63.49))
inits
list(lambda=1,int=1,a=1,b=1,c=1)

```

Depth Model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
a	-0.206	0.1656	0.007807	-0.4687	-0.2116	0.1399	1001	199000
int	10.24	6.722	0.3168	-3.801	10.32	21.22	1001	199000
lambda	0.2376	0.2	0.003956	0.01953	0.1829	0.7657	1001	199000
p.value	0.3378	0.4729	0.005645	0.0	0.0	1.0	1001	199000

	Dbar	Dhat	DIC	pD
time	13.57	11.72	15.43	1.859
total	13.57	11.72	15.43	1.859

Flow Rate Model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
b	0.275	0.08184	0.003878	0.1774	0.2466	0.4142	1001	199000
int	-79.1	23.73	1.124	-119.5	-71.03	-50.78	1001	199000
lambda	0.5652	0.4136	0.008364	0.06954	0.4663	1.625	1001	199000
p.value	0.2897	0.4536	0.002917	0.0	0.0	1.0	1001	199000

	Dbar	Dhat	DIC	pD
time	61.95	60.34	63.56	1.613
total	61.95	60.34	63.56	1.613

Temperature Model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
c	0.1252	0.1874	0.008872	-0.1123	0.08358	0.603	1001	199000
int	-6.383	12.07	0.5714	-36.74	-3.886	9.124	1001	199000
lambda	0.2211	0.2287	0.006317	0.003269	0.1534	0.8348	1001	199000
p.value	0.378	0.4849	0.008676	0.0	0.0	1.0	1001	199000

	Dbar	Dhat	DIC	pD
time	14.3	12.52	16.09	1.786
total	14.3	12.52	16.09	1.786

Depth and Flow Rate Model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
a	-0.4428	0.5908	0.028	-1.59	-0.218	0.1908	1001	199000
b	0.3009	0.1201	0.005694	0.1669	0.2472	0.5524	1001	199000
int	-68.59	11.94	0.565	-95.34	-65.53	-52.34	1001	199000
lambda	0.4022	0.451	0.01447	0.006719	0.253	1.609	1001	199000
p.value	0.1832	0.3869	0.006551	0.0	0.0	1.0	1001	199000

	Dbar	Dhat	DIC	pD
time	62.28	60.52	64.05	1.768
total	62.28	60.52	64.05	1.768

Depth and Temperature Model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
a	-0.09592	0.1285	0.006023	-0.3447	-0.106	0.2037	1001	199000
c	0.1069	0.2434	0.01153	-0.2659	0.02917	0.6407	1001	199000
int	-1.134	15.02	0.7114	-35.33	5.972	15.87	1001	199000
lambda	0.2213	0.2555	0.007551	0.003024	0.1361	0.9131	1001	199000
p.value	0.2728	0.4454	0.007379	0.0	0.0	1.0	1001	199000

	Dbar	Dhat	DIC	pD
time	13.67	11.61	15.73	2.059
total	13.67	11.61	15.73	2.059

Flow Rate and Temperature Model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
b	0.2552	0.1039	0.004923	0.0812	0.2384	0.4464	1001	199000
c	0.3085	0.311	0.01474	-0.3406	0.4605	0.6457	1001	199000
int	-94.76	12.15	0.5747	-109.3	-97.67	-66.02	1001	199000
lambda	0.6067	0.4687	0.008827	0.06998	0.4842	1.826	1001	199000
p.value	0.2808	0.4494	0.002575	0.0	0.0	1.0	1001	199000

	Dbar	Dhat	DIC	pD
time	47.35	45.33	49.37	2.023
total	47.35	45.33	49.37	2.023

Depth, Flow Rate, and Temperature Model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
a	-0.8401	0.5524	0.02617	-1.997	-0.6524	-0.2301	1001	199000
b	0.2953	0.1322	0.006262	0.0917	0.2848	0.5213	1001	199000
c	0.7369	0.2667	0.01263	0.1727	0.8104	1.134	1001	199000
int	-102.1	10.46	0.4947	-119.5	-102.3	-83.0	1001	199000
lambda	0.3199	0.3697	0.01174	0.003922	0.203	1.333	1001	199000
p.value	0.1503	0.3573	0.005703	0.0	0.0	1.0	1001	199000

	Dbar	Dhat	DIC	pD
time	47.0	44.61	49.38	2.384
total	47.0	44.61	49.38	2.384

C.2.6 Weibull model with influence on lambda using hollow core door data.

```

#Hollow Core Door Data
#Weibull Model (Lambda) Using Flow Rate (F), Depth (D), and Temperature (T); P-Value
model {
  for(i in 1:N) {
    time[i] ~ dweib(beta, lambda[i])
    time.rep[i] ~ dweib(beta, lambda[i])
    lambda[i] <- int + a*D[i]
    #lambda[i] <- int + b*F[i]
    #lambda[i] <- int + c*T[i]
    #lambda[i] <- int + a*D[i] +b*F[i]
    #lambda[i] <- int + a*D[i] +c*T[i]
    #lambda[i] <- int + b*F[i] +c*T[i]
    #lambda[i] <- int + a*D[i] +b*F[i] +c*T[i]
    #Rank observed times and duplicate times
    time.ranked[i]<- ranked(time[],i)
    time.rep.ranked[i]<-ranked(time.rep[],i)
    #Calculate components of Cramer-von Mises statistic for observed and replicated data
    F.obs[i]<-cumulative(time[i], time.ranked[i])
    F.rep[i]<-cumulative(time.rep[i], time.rep.ranked[i])
    diff.obs[i]<-pow(F.obs[i] - (2*i-1)/(2*N),2)
    diff.rep[i]<-pow(F.rep[i] - (2*i-1)/(2*N),2)
  }
  #Calculate distribution of Cramer-von Mises statistic for observed and replicated data
  CVM.obs<-sum(diff.obs[])
  CVM.rep<-sum(diff.rep[])
  p.value<-step(CVM.rep-CVM.obs)
  # Prior distributions
  beta ~ dgamma(0.0001, 0.0001)
  int ~ dnorm(0, 0.0001)
  a ~ dnorm(0, 0.0001)
  b ~ dnorm(0, 0.0001)
  c ~ dnorm(0, 0.0001)
}
data
list(time =c(3.533, 2.667, 0.8667, 0.4667), N=4,
      F =c(290.6, 294.07, 411.25, 472.89), D =c(40.74, 38.77, 32.28, 43.34), T =c(69.63, 69.02, 62.18, 63.49))
inits
list(beta=1,int=1,a=1,b=1,c=1)

```

Depth Model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
a	-0.03097	0.06433	0.002928	-0.1774	-0.02648	0.08876	1001	199000
beta	1.034	0.4195	0.005967	0.3544	0.9874	1.976	1001	199000
int	1.986	2.605	0.1187	-2.406	1.685	8.204	1001	199000
p.value	0.4579	0.4982	0.005322	0.0	0.0	1.0	1001	199000

	Dbar	Dhat	DIC	pD
time	15.37	13.46	17.27	1.906
total	15.37	13.46	17.27	1.906

Flow Rate Model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
b	0.1549	0.003813	1.803E-4	0.1455	0.1559	0.1602	1001	199000
beta	2.384	0.3304	0.003064	1.727	2.388	3.017	1001	199000
int	-44.93	1.109	0.0524	-46.47	-45.23	-42.24	1001	199000
p.value	0.9767	0.151	3.447E-4	1.0	1.0	1.0	1001	199000

	Dbar	Dhat	DIC	pD
time	35.28	33.88	36.67	1.392
total	35.28	33.88	36.67	1.392

Temperature Model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
beta	1.344	0.7023	0.02129	0.3514	1.214	3.059	1001	199000
c	-0.0212	0.08003	0.003779	-0.1629	-0.02285	0.2098	1001	199000
int	2.07	5.164	0.2439	-12.45	2.05	11.45	1001	199000
p.value	0.3605	0.4801	0.007654	0.0	0.0	1.0	1001	199000

	Dbar	Dhat	DIC	pD
time	12.85	11.81	13.89	1.039
total	12.85	11.81	13.89	1.039

Depth and Flow Rate Model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
a	-0.7899	0.09154	0.004336	-0.9684	-0.7812	-0.6638	1001	199000
b	0.2924	0.008621	4.081E-4	0.2777	0.2905	0.3137	1001	199000
beta	1.82	0.2049	0.001867	1.416	1.821	2.22	1001	199000
int	-52.66	2.281	0.1079	-56.17	-53.1	-47.85	1001	199000
p.value	2.362E-4	0.01537	3.412E-5	0.0	0.0	0.0	1001	199000

	Dbar	Dhat	DIC	pD
time	102.6	101.1	104.0	1.474
total	102.6	101.1	104.0	1.474

Depth and Temperature Model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
a	-0.02435	0.1045	0.004916	-0.1972	-0.01825	0.1538	1001	199000
beta	1.155	0.6476	0.02267	0.2269	1.065	2.653	1001	199000
c	-0.01198	0.1683	0.00797	-0.1913	-0.04161	0.3941	1001	199000
int	2.784	10.65	0.5041	-18.58	2.344	17.19	1001	199000
p.value	0.3527	0.4778	0.009334	0.0	0.0	1.0	1001	199000

	Dbar	Dhat	DIC	pD
time	11.99	10.27	13.71	1.718
total	11.99	10.27	13.71	1.718

Flow Rate and Temperature Model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
b	0.3895	0.005789	2.741E-4	0.3782	0.3888	0.3997	1001	199000
beta	2.22	0.1939	0.001058	1.836	2.221	2.593	1001	199000
c	-0.9389	0.02521	0.001193	-0.9832	-0.9364	-0.9018	1001	199000
int	-47.73	1.23	0.05817	-49.96	-47.81	-45.77	1001	199000
p.value	0.00898	0.09434	4.859E-4	0.0	0.0	0.0	1001	199000

	Dbar	Dhat	DIC	pD
time	122.1	120.6	123.6	1.504
total	122.1	120.6	123.6	1.504

Depth, Flow Rate, and Temperature Model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
a	-1.444	0.09774	0.004629	-1.614	-1.401	-1.307	1001	199000
b	0.3845	0.01573	7.454E-4	0.3617	0.3813	0.422	1001	199000
beta	1.666	0.1657	9.45E-4	1.339	1.667	1.987	1001	199000
c	-0.6967	0.0378	0.001787	-0.7541	-0.704	-0.6166	1001	199000
int	-4.26	1.365	0.06418	-7.759	-4.154	-1.737	1001	199000
p.value	5.025E-6	0.002242	5.026E-6	0.0	0.0	0.0	1001	199000

	Dbar	Dhat	DIC	pD
time	165.0	163.5	166.6	1.523
total	165.0	163.5	166.6	1.523

C.3 Scripts and Statistical Summaries from Section 6.1.3

C.3.1 Testing gamma, uniform, and beta prior distributions using a binomial depth regression model and steel door data.

dgamma(0.0001, 0.0001):

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
depth	0.0	0.0	2.242E-13	0.0	0.0	0.0	1001	199000
int	0.0	0.0	2.242E-13	0.0	0.0	0.0	1001	199000
p.value	1.0	0.0	2.242E-13	1.0	1.0	1.0	1001	199000

dunif(0,100000):

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
depth	0.01249	0.009497	3.066E-5	4.809E-4	0.01053	0.0352	1001	199000
int	0.3	0.2574	9.065E-4	0.009223	0.2324	0.9526	1001	199000
p.value	0.2235	0.4166	8.769E-4	0.0	0.0	1.0	1001	199000

dbeta(0.0001, 0.0001):

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
depth	0.0	0.0	2.242E-13	0.0	0.0	0.0	1001	199000
int	0.0	0.0	2.242E-13	0.0	0.0	0.0	1001	199000
p.value	1.0	0.0	2.242E-13	1.0	1.0	1.0	1001	199000

dbeta(0.5, 0.5):

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
depth	0.0	0.0	2.242E-13	0.0	0.0	0.0	1001	199000
int	0.0	0.0	2.242E-13	0.0	0.0	0.0	1001	199000
p.value	1.0	0.0	2.242E-13	1.0	1.0	1.0	1001	199000

dbeta(1, 1):

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
depth	0.01265	0.009558	6.398E-5	5.204E-4	0.01073	0.03578	1001	199000
int	0.2815	0.2246	0.001648	0.009427	0.2259	0.8334	1001	199000
p.value	0.2296	0.4206	0.001553	0.0	0.0	1.0	1001	199000

C.3.2 Testing gamma, uniform, and beta prior distributions using a binomial depth regression model and hollow core door data.

dgamma(0.0001, 0.0001):

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
depth	0.0	0.0	2.242E-13	0.0	0.0	0.0	1001	199000
int	0.0	0.0	2.242E-13	0.0	0.0	0.0	1001	199000
p.value	1.0	0.0	2.242E-13	1.0	1.0	1.0	1001	199000

dunif(0,100000):

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
depth	0.02749	0.01987	6.347E-5	0.001278	0.02388	0.07505	1001	199000
int	0.4786	0.4166	0.001381	0.01465	0.3671	1.535	1001	199000
p.value	0.3019	0.4591	0.001003	0.0	0.0	1.0	1001	199000

dbeta(0.0001, 0.0001):

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
depth	0.0	0.0	2.242E-13	0.0	0.0	0.0	1001	199000
int	0.0	0.0	2.242E-13	0.0	0.0	0.0	1001	199000
p.value	1.0	0.0	2.242E-13	1.0	1.0	1.0	1001	199000

dbeta(0.5, 0.5):

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
depth	0.0	0.0	2.242E-13	0.0	0.0	0.0	1001	199000
int	0.0	0.0	2.242E-13	0.0	0.0	0.0	1001	199000
p.value	1.0	0.0	2.242E-13	1.0	1.0	1.0	1001	199000

dbeta(1, 1):

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
depth	0.02876	0.01983	1.369E-4	0.001583	0.0254	0.07537	1001	199000
int	0.3636	0.2649	0.001715	0.01266	0.3122	0.9319	1001	199000
p.value	0.332	0.4709	0.001596	0.0	0.0	1.0	1001	199000

C.3.3 Summaries from testing uniform prior distributions on binomial depth regression model using steel door data.

dunif(0, 1000) for all priors:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
depth	0.01251	0.009528	3.068E-5	4.786E-4	0.01053	0.03545	1001	199000
int	0.2986	0.2559	9.174E-4	0.009092	0.2312	0.9467	1001	199000
p.value	0.2248	0.4175	9.977E-4	0.0	0.0	1.0	1001	199000

dunif(-10, 10) for intercept prior, dunif(0, 1000) for covariate priors:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
depth	0.2295	0.04986	6.169E-4	0.1089	0.2396	0.3006	1001	199000
int	-7.882	1.724	0.02162	-9.933	-8.305	-3.602	1001	199000
p.value	0.7874	0.4092	0.001037	0.0	1.0	1.0	1001	199000

dunif(-50, 50) for intercept prior, dunif(0, 1000) for covariate priors:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
depth	0.907	0.2879	0.0104	0.3231	0.932	1.346	1001	199000
int	-32.92	10.67	0.3856	-49.21	-33.84	-11.34	1001	199000
p.value	0.3989	0.4897	0.005096	0.0	0.0	1.0	1001	199000

C.3.4 Summaries from testing uniform prior distributions on binomial depth regression model using hollow core door data.

dunif(0, 1000) for all priors:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
depth	0.02751	0.01982	6.105E-5	0.001296	0.02395	0.0747	1001	199000
int	0.4786	0.4174	0.001405	0.01418	0.3672	1.545	1001	199000
p.value	0.3009	0.4587	9.796E-4	0.0	0.0	1.0	1001	199000

dunif(-10, 10) for intercept prior, dunif(0, 1000) for covariate priors:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
depth	0.2835	0.07121	5.387E-4	0.1239	0.291	0.4051	1001	199000
int	-7.753	1.828	0.01454	-9.926	-8.201	-3.246	1001	199000
p.value	0.901	0.2987	7.709E-4	0.0	1.0	1.0	1001	199000

C.3.5 Summaries for single covariate regression models comparing the use of centering on hollow core door data.

Standard depth regression model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
depth	4.606	2.381	0.01857	0.9739	4.291	10.02	1001	199000
int	-127.7	65.07	0.5108	-274.7	-119.6	-27.49	1001	199000

	Dbar	Dhat	DIC	pD
failure	0.2511	1.75E-7	0.5022	0.2511
total	0.2511	1.75E-7	0.5022	0.2511

Centered depth regression model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
depth	87.23	59.37	0.4243	9.812	75.25	229.1	1001	199000
int	72.49	59.03	0.4213	-2.344	60.2	214.0	1001	199000

	Dbar	Dhat	DIC	pD
failure	0.03285	-2.22E-15	0.06571	0.03285
total	0.03285	-2.22E-15	0.06571	0.03285

Standard flow rate regression model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
flow	0.1991	0.1722	0.00135	0.007635	0.1522	0.6393	1001	199000
int	25.67	22.21	0.1744	0.9523	19.62	82.38	1001	199000

	Dbar	Dhat	DIC	pD
failure	13.67	12.44	14.9	1.231
total	13.67	12.44	14.9	1.231

Centered flow rate regression model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
flow	0.1991	0.1722	0.00135	0.007635	0.1522	0.6393	1001	199000
int	25.67	22.21	0.1744	0.9523	19.62	82.38	1001	199000

	Dbar	Dhat	DIC	pD
failure	13.67	12.44	14.9	1.231
total	13.67	12.44	14.9	1.231

Standard temperature regression model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
int	0.6507	0.7828	0.004614	-0.8068	0.6191	2.303	1001	199000
temp	-0.5122	0.5175	0.003288	-1.676	-0.4568	0.3538	1001	199000

	Dbar	Dhat	DIC	pD
failure	14.98	12.97	17.0	2.015
total	14.98	12.97	17.0	2.015

Centered temperature regression model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
int	0.6507	0.7828	0.004614	-0.8068	0.6191	2.303	1001	199000
temp	-0.5122	0.5175	0.003288	-1.676	-0.4568	0.3538	1001	199000

	Dbar	Dhat	DIC	pD
failure	14.98	12.97	17.0	2.015
total	14.98	12.97	17.0	2.015

C.3.6 Summaries for single covariate regression models comparing the use of centering on steel door data.

Standard depth regression model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
depth	1.657	0.9125	0.006358	0.4177	1.476	3.885	1001	199000
int	-60.94	34.1	0.2369	-144.4	-54.14	-14.82	1001	199000

	Dbar	Dhat	DIC	pD
failure	12.38	10.9	13.86	1.481
total	12.38	10.9	13.86	1.481

Centered depth regression model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
depth	2.047	1.257	0.008997	0.4562	1.763	5.208	1001	199000
int	-4.021	3.198	0.02268	-11.98	-3.334	0.1763	1001	199000

	Dbar	Dhat	DIC	pD
failure	12.88	11.36	14.39	1.511
total	12.88	11.36	14.39	1.511

Standard flow rate regression model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
flow	0.01299	0.006127	6.345E-5	0.004426	0.01181	0.02808	1001	199000
int	-8.355	3.924	0.04127	-17.94	-7.665	-2.681	1001	199000

	Dbar	Dhat	DIC	pD
failure	14.27	12.54	16.0	1.73
total	14.27	12.54	16.0	1.73

Centered flow rate regression model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
flow	0.01307	0.006054	3.838E-5	0.004516	0.01196	0.02784	1001	199000
int	0.6088	0.9083	0.00618	-1.049	0.5575	2.537	1001	199000

	Dbar	Dhat	DIC	pD
failure	14.27	12.55	15.98	1.719
total	14.27	12.55	15.98	1.719

Standard temperature regression model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
int	-158.0	54.0	0.3416	-277.7	-152.4	-68.07	1001	199000
temp	2.562	0.8762	0.00555	1.104	2.47	4.506	1001	199000

	Dbar	Dhat	DIC	pD
failure	6.772	5.25	8.294	1.522
total	6.772	5.25	8.294	1.522

Centered temperature regression model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
int	2.954	2.777	0.01896	-1.155	2.471	9.665	1001	199000
temp	7.853	4.692	0.03381	2.041	6.741	19.74	1001	199000

	Dbar	Dhat	DIC	pD
failure	7.533	6.09	8.976	1.443
total	7.533	6.09	8.976	1.443

C.3.7 Calculated residuals of temperature regression model using steel door data.

residual[1]	4.653E-4	0.005662	1.887E-5	0.0	3.511E-14	0.002992	1001	199000
residual[2]	0.03484	0.07225	3.301E-4	4.985E-12	0.003612	0.2561	1001	199000
residual[3]	0.04727	0.08944	4.184E-4	1.474E-10	0.007712	0.3187	1001	199000
residual[4]	0.0554	0.1001	4.739E-4	8.003E-10	0.01085	0.359	1001	199000
residual[5]	0.0554	0.1001	4.739E-4	8.003E-10	0.01085	0.359	1001	199000
residual[6]	0.0554	0.1001	4.739E-4	8.003E-10	0.01085	0.359	1001	199000
residual[7]	0.0554	0.1001	4.739E-4	8.003E-10	0.01085	0.359	1001	199000
residual[8]	0.04727	0.08944	4.184E-4	1.474E-10	0.007712	0.3187	1001	199000
residual[9]	0.01333	0.03759	1.59E-4	0.0	7.874E-5	0.1202	1001	199000
residual[10]	-1.881	3.563	0.02364	-7.49	-1.199	-0.2488	1001	199000
residual[11]	-0.00841	0.02757	1.145E-4	-0.07842	-3.417E-5	-1.029E-16	1001	199000
residual[12]	-0.01227	0.03483	1.496E-4	-0.1073	-2.157E-4	-1.601E-14	1001	199000
residual[13]	-0.014	0.03781	1.645E-4	-0.1191	-3.91E-4	-9.058E-14	1001	199000
residual[14]	-0.0184	0.04485	2.003E-4	-0.1467	-0.001204	-2.743E-12	1001	199000
residual[15]	-0.0461	0.07958	3.907E-4	-0.2801	-0.01363	-6.752E-8	1001	199000
residual[16]	-0.1176	0.1406	7.608E-4	-0.5077	-0.06693	-2.548E-4	1001	199000
residual[17]	-0.2348	0.2128	0.001226	-0.7875	-0.1759	-0.009082	1001	199000
residual[18]	-0.4075	0.3111	0.001858	-1.185	-0.3372	-0.03789	1001	199000
residual[19]	3.526	6.037	0.04379	0.6243	2.141	14.5	1001	199000

C.3.8 Logistic regression model with an added latent variable using steel door data.

Depth model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
depth	44.33	19.12	0.4271	11.51	42.83	85.89	1001	199000
deviance	0.1847	0.7545	0.0103	-4.219E-15	3.638E-7	2.225	1001	199000
int	-1537.0	671.2	13.96	-2990.0	-1489.0	-370.4	1001	199000
omega.lambda	286.5	230.0	7.142	34.73	210.6	890.5	1001	199000
p.chisq	0.9721	0.1648	0.001694	0.0	1.0	1.0	1001	199000
p.freeman	0.9727	0.1628	0.001585	0.0	1.0	1.0	1001	199000
p.likelihood	0.9746	0.1574	0.001182	0.0	1.0	1.0	1001	199000

	Dbar	Dhat	DIC	pD
failure	0.2049	-4.219E-15	0.4099	0.2049
total	0.2049	-4.219E-15	0.4099	0.2049

Flow rate model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
deviance	0.06877	0.4716	0.005519	-4.219E-15	-4.219E-15	0.687	1001	199000
flow	2.188	0.9619	0.02096	0.6061	2.104	4.303	1001	199000
int	-1346.0	602.4	12.24	-2664.0	-1295.0	-350.5	1001	199000
omega.lambda	539.9	238.1	5.773	127.9	528.2	968.4	1001	199000
p.chisq	0.9898	0.1003	6.694E-4	1.0	1.0	1.0	1001	199000
p.freeman	0.9897	0.1011	6.992E-4	1.0	1.0	1.0	1001	199000
p.likelihood	0.99	0.09933	6.932E-4	1.0	1.0	1.0	1001	199000

	Dbar	Dhat	DIC	pD
failure	0.06466	-4.219E-15	0.1293	0.06466
total	0.06466	-4.219E-15	0.1293	0.06466

Depth and Flow Rate model:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
depth	-87.45	70.31	1.427	-240.1	-81.09	29.49	1001	199000
deviance	0.06281	0.5599	0.0166	-4.219E-15	-4.219E-15	0.3845	1001	199000
flow	5.887	3.308	0.06902	0.773	5.408	13.61	1001	199000
int	-561.1	941.5	13.94	-2415.0	-552.6	1291.0	1001	199000
omega.lambda	729.2	208.8	4.917	242.2	775.5	990.8	1001	199000
p.chisq	0.992	0.08928	0.001084	1.0	1.0	1.0	1001	199000
p.freeman	0.9921	0.08881	0.001097	1.0	1.0	1.0	1001	199000
p.likelihood	0.9927	0.08485	9.798E-4	1.0	1.0	1.0	1001	199000

	Dbar	Dhat	DIC	pD
failure	0.03968	-4.219E-15	0.07935	0.03968
total	0.03968	-4.219E-15	0.07935	0.03968

C.4 Scripts and Statistical Summaries from Section 6.2

C.4.1 Summaries for O-ring verification case using the developed assessment model.

	mean	sd	MC_error	val5.0pc	median	val95.0pc	start	sample
a	5.215	3.182	0.01427	-0.006098	5.218	10.45	1001	398000
b	-0.1191	0.04904	2.193E-4	-0.2011	-0.1184	-0.04001	1001	398000
dev.sat	20.16	2.079	0.008956	18.19	19.53	24.26	1001	398000
p.chisq	0.2155	0.4111	9.254E-4	0.0	0.0	1.0	1001	398000
p.freeman	0.556	0.4969	0.001071	0.0	1.0	1.0	1001	398000
p.likelihood	0.4242	0.4942	0.001166	0.0	0.0	1.0	1001	398000
residual[1]	-0.664	0.1241	4.424E-4	-0.8767	-0.6592	-0.469	1001	398000
residual[2]	1.561	0.5909	0.002321	0.7644	1.476	2.635	1001	398000
residual[3]	-0.5582	0.1177	4.328E-4	-0.7593	-0.5542	-0.3727	1001	398000
residual[4]	-0.5911	0.1186	4.297E-4	-0.7937	-0.5868	-0.4042	1001	398000
residual[5]	-0.6263	0.1206	4.322E-4	-0.8325	-0.6217	-0.4365	1001	398000
residual[6]	-0.4717	0.1191	4.576E-4	-0.6779	-0.4656	-0.2868	1001	398000
residual[7]	-0.4465	0.1202	4.673E-4	-0.6559	-0.4393	-0.2619	1001	398000
residual[8]	-0.5275	0.1177	4.394E-4	-0.7289	-0.523	-0.3423	1001	398000
residual[9]	-0.03125	0.4963	0.002087	-0.7961	-0.05558	0.8145	1001	398000
residual[10]	0.6447	0.3797	0.001422	0.07588	0.615	1.308	1001	398000
residual[11]	1.561	0.5909	0.002321	0.7644	1.476	2.635	1001	398000
residual[12]	-0.342	0.125	5.07E-4	-0.5691	-0.3279	-0.1635	1001	398000
residual[13]	-0.6263	0.1206	4.322E-4	-0.8325	-0.6217	-0.4365	1001	398000
residual[14]	0.5028	0.8597	0.003783	-0.7628	0.4352	1.99	1001	398000
residual[15]	-0.6263	0.1206	4.322E-4	-0.8325	-0.6217	-0.4365	1001	398000
residual[16]	-0.4006	0.1224	4.856E-4	-0.6177	-0.3911	-0.2175	1001	398000
residual[17]	-0.5275	0.1177	4.394E-4	-0.7289	-0.523	-0.3423	1001	398000
residual[18]	-0.2934	0.1266	5.204E-4	-0.5285	-0.2748	-0.1223	1001	398000
residual[19]	-0.3798	0.1234	4.936E-4	-0.6004	-0.3688	-0.1979	1001	398000
residual[20]	-0.3248	0.1257	5.123E-4	-0.5549	-0.3092	-0.1485	1001	398000
residual[21]	5.263	2.042	0.00864	2.826	4.853	9.049	1001	398000
residual[22]	-0.3798	0.1234	4.936E-4	-0.6004	-0.3688	-0.1979	1001	398000
residual[23]	0.08065	0.4643	0.001925	-0.6303	0.05411	0.8764	1001	398000

	Dbar	Dhat	DIC	pD
failure	33.71	31.72	35.7	1.987
total	33.71	31.72	35.7	1.987

C.5 Scripts and Statistical Summaries from Section 6.3

C.5.1 Summaries for hollow core door cases using the developed assessment model.

Depth model:

	mean	sd	MC_error	val2.5pc	val5.0pc	median	val95.0pc	val97.5pc	start	sample
depth	45.23	24.01	0.1366	8.145	11.53	42.21	89.3	99.39	10001	420000
dev.sat	0.02897	0.2935	7.21E-4	0.0	0.0	0.0	0.006472	0.1216	10001	420000
int	-1252.0	654.5	3.745	-2716.0	-2444.0	-1175.0	-322.9	-227.4	10001	420000
p.bound[1]	9.767E-5	0.005036	1.747E-5	0.0	0.0	0.0	6.059E-16	2.756E-11	10001	420000
p.bound[2]	1.327E-5	0.00188	6.872E-6	0.0	0.0	0.0	0.0	0.0	10001	420000
p.bound[3]	1.0	1.495E-4	3.378E-7	1.0	1.0	1.0	1.0	1.0	10001	420000
p.bound[4]	1.508E-5	0.001983	7.346E-6	0.0	0.0	0.0	0.0	0.0	10001	420000
p.bound[5]	0.004974	0.0496	1.259E-4	0.0	0.0	0.0	4.089E-5	0.006355	10001	420000
p.bound[6]	1.0	0.001808	5.097E-6	1.0	1.0	1.0	1.0	1.0	10001	420000
p.bound[7]	1.0	2.148E-4	5.008E-7	1.0	1.0	1.0	1.0	1.0	10001	420000
p.bound[8]	1.0	3.613E-4	9.119E-7	1.0	1.0	1.0	1.0	1.0	10001	420000
p.bound[9]	0.9963	0.04245	9.984E-5	0.9987	1.0	1.0	1.0	1.0	10001	420000
p.bound[10]	1.0	1.187E-4	2.659E-7	1.0	1.0	1.0	1.0	1.0	10001	420000
p.chisq	0.9957	0.06511	1.451E-4	1.0	1.0	1.0	1.0	1.0	10001	420000
p.freeman	0.9953	0.06824	1.514E-4	1.0	1.0	1.0	1.0	1.0	10001	420000
p.likelihood	0.9962	0.06128	1.223E-4	1.0	1.0	1.0	1.0	1.0	10001	420000

	Dbar	Dhat	DIC	pD
failure	0.02906	-2.22E-15	0.05812	0.02906
total	0.02906	-2.22E-15	0.05812	0.02906

Flow Rate model:

	mean	sd	MC_error	val2.5pc	val5.0pc	median	val95.0pc	val97.5pc	start	sample
dev.sat	13.83	2.316	0.01441	11.3	11.41	13.19	18.45	19.93	10001	420000
flow	0.2152	0.1886	0.0013	0.007943	0.0148	0.1642	0.5908	0.7042	10001	420000
int	-63.1	55.28	0.3813	-206.5	-173.2	-48.17	-4.334	-2.279	10001	420000
p.bound[1]	0.4241	0.1751	7.311E-4	0.1153	0.1496	0.4164	0.7249	0.7735	10001	420000
p.bound[2]	0.4695	0.1704	6.279E-4	0.1555	0.1941	0.4668	0.7547	0.7992	10001	420000
p.bound[3]	0.4695	0.1704	6.279E-4	0.1555	0.1941	0.4668	0.7547	0.7992	10001	420000
p.bound[4]	0.6598	0.1889	9.096E-4	0.2645	0.322	0.6767	0.9381	0.9612	10001	420000
p.bound[5]	0.5627	0.1742	6.809E-4	0.2175	0.2648	0.5693	0.8383	0.8744	10001	420000
p.bound[6]	0.4695	0.1704	6.279E-4	0.1555	0.1941	0.4668	0.7547	0.7992	10001	420000
p.bound[7]	0.4028	0.1783	7.912E-4	0.09491	0.1274	0.3932	0.7125	0.7629	10001	420000
p.bound[8]	0.5401	0.1717	6.36E-4	0.2041	0.2495	0.5444	0.8148	0.8528	10001	420000
p.bound[9]	0.9974	0.03073	7.826E-5	0.9973	1.0	1.0	1.0	1.0	10001	420000
p.bound[10]	0.9964	0.03336	9.034E-5	0.9872	0.9997	1.0	1.0	1.0	10001	420000
p.chisq	0.08255	0.2752	5.985E-4	0.0	0.0	0.0	1.0	1.0	10001	420000
p.freeman	0.2427	0.4287	0.001123	0.0	0.0	0.0	1.0	1.0	10001	420000
p.likelihood	0.3784	0.485	0.001285	0.0	0.0	0.0	1.0	1.0	10001	420000

	Dbar	Dhat	DIC	pD
failure	13.82	12.57	15.08	1.256
total	13.82	12.57	15.08	1.256

Temperature model:

	mean	sd	MC_error	val2.5pc	val5.0pc	median	val95.0pc	val97.5pc	start	sample
dev.sat	15.02	2.25	0.01041	12.81	12.87	14.34	19.52	21.05	10001	420000
int	34.2	34.58	0.1736	-23.54	-15.36	30.48	96.77	113.2	10001	420000
p.bound[1]	0.662	0.1602	6.134E-4	0.3261	0.3783	0.6742	0.9039	0.9312	10001	420000
p.bound[2]	0.5535	0.1671	6.069E-4	0.2222	0.2682	0.5591	0.819	0.8553	10001	420000
p.bound[3]	0.6585	0.1598	6.09E-4	0.324	0.3758	0.6703	0.9005	0.9282	10001	420000
p.bound[4]	0.4843	0.1901	7.592E-4	0.1334	0.1742	0.4836	0.799	0.8427	10001	420000
p.bound[5]	0.6016	0.1592	5.702E-4	0.2764	0.3253	0.6099	0.8492	0.8818	10001	420000
p.bound[6]	0.5724	0.1631	5.84E-4	0.2442	0.2916	0.579	0.8291	0.8639	10001	420000
p.bound[7]	0.4203	0.2175	9.343E-4	0.06194	0.09194	0.4056	0.7996	0.8507	10001	420000
p.bound[8]	0.4375	0.2099	8.876E-4	0.07901	0.1122	0.4274	0.7981	0.8473	10001	420000
p.bound[9]	0.8206	0.2184	9.157E-4	0.2199	0.3254	0.9162	0.9993	0.9998	10001	420000
p.bound[10]	0.7937	0.1995	8.395E-4	0.2876	0.3761	0.8591	0.9955	0.9982	10001	420000
p.chisq	0.1809	0.3849	8.648E-4	0.0	0.0	0.0	1.0	1.0	10001	420000
p.freeman	0.2927	0.455	0.00114	0.0	0.0	0.0	1.0	1.0	10001	420000
p.likelihood	0.355	0.4785	0.001188	0.0	0.0	0.0	1.0	1.0	10001	420000
temp	-0.5065	0.5182	0.002601	-1.688	-1.442	-0.4519	0.2384	0.363	10001	420000

	Dbar	Dhat	DIC	pD
failure	15.01	12.96	17.07	2.055
total	15.01	12.96	17.07	2.055

Depth and Flow Rate model:

	mean	sd	MC_error	val2.5pc	val5.0pc	median	val95.0pc	val97.5pc	start	sample
depth	39.61	23.41	0.4147	6.698	8.82	35.6	82.97	94.66	10001	420000
dev.sat	0.0362	0.41	0.007254	0.0	0.0	0.0	0.008782	0.1333	10001	420000
flow	1.59	1.61	0.02838	-0.3915	-0.2099	1.18	4.816	5.66	10001	420000
int	-1589.0	670.6	13.45	-3023.0	-2790.0	-1535.0	-584.8	-482.1	10001	420000
p.bound[1]	8.899E-6	8.051E-4	3.116E-6	0.0	0.0	0.0	4.283E-17	4.464E-13	10001	420000
p.bound[2]	3.835E-7	6.464E-5	1.851E-7	0.0	0.0	0.0	0.0	0.0	10001	420000
p.bound[3]	1.0	1.292E-5	4.922E-8	1.0	1.0	1.0	1.0	1.0	10001	420000
p.bound[4]	1.65E-4	0.006684	7.282E-5	0.0	0.0	0.0	0.0	3.613E-16	10001	420000
p.bound[5]	0.005523	0.05552	6.999E-4	0.0	0.0	0.0	3.056E-5	0.008386	10001	420000
p.bound[6]	0.9968	0.03827	3.19E-4	0.9996	1.0	1.0	1.0	1.0	10001	420000
p.bound[7]	1.0	0.001043	2.147E-6	1.0	1.0	1.0	1.0	1.0	10001	420000
p.bound[8]	1.0	8.17E-5	4.044E-7	1.0	1.0	1.0	1.0	1.0	10001	420000
p.bound[9]	0.9994	0.01873	9.885E-5	1.0	1.0	1.0	1.0	1.0	10001	420000
p.bound[10]	1.0	0.0	1.543E-13	1.0	1.0	1.0	1.0	1.0	10001	420000
p.chisq	0.9949	0.07098	7.265E-4	1.0	1.0	1.0	1.0	1.0	10001	420000
p.freeman	0.9949	0.07113	7.006E-4	1.0	1.0	1.0	1.0	1.0	10001	420000
p.likelihood	0.9963	0.06068	3.381E-4	1.0	1.0	1.0	1.0	1.0	10001	420000

	Dbar	Dhat	DIC	pD
failure	0.02854	-2.22E-15	0.05709	0.02854
total	0.02854	-2.22E-15	0.05709	0.02854

Depth and Temperature model:

	mean	sd	MC_error	val2.5pc	val5.0pc	median	val95.0pc	val97.5pc	start	sample
depth	1142.0	596.7	3.933	207.4	295.5	1071.0	2231.0	2478.0	10001	420000
dev.sat	9.249E-4	0.04891	1.069E-4	0.0	0.0	0.0	0.0	0.0	10001	420000
int	5.053	999.2	5.967	-1951.0	-1634.0	2.896	1653.0	1971.0	10001	420000
p.bound[1]	2.254E-6	8.123E-4	2.166E-6	0.0	0.0	0.0	0.0	0.0	10001	420000
p.bound[2]	2.786E-19	1.0E-10	1.543E-13	0.0	0.0	0.0	0.0	0.0	10001	420000
p.bound[3]	1.0	0.0	1.543E-13	1.0	1.0	1.0	1.0	1.0	10001	420000
p.bound[4]	1.575E-19	1.0E-10	1.543E-13	0.0	0.0	0.0	0.0	0.0	10001	420000
p.bound[5]	1.27E-4	0.007797	1.767E-5	0.0	0.0	0.0	0.0	0.0	10001	420000
p.bound[6]	1.0	3.799E-4	5.996E-7	1.0	1.0	1.0	1.0	1.0	10001	420000
p.bound[7]	1.0	0.0	1.543E-13	1.0	1.0	1.0	1.0	1.0	10001	420000
p.bound[8]	1.0	0.0	1.543E-13	1.0	1.0	1.0	1.0	1.0	10001	420000
p.bound[9]	0.9998	0.009074	2.115E-5	1.0	1.0	1.0	1.0	1.0	10001	420000
p.bound[10]	1.0	0.0	1.543E-13	1.0	1.0	1.0	1.0	1.0	10001	420000
p.chisq	0.9999	0.01215	2.472E-5	1.0	1.0	1.0	1.0	1.0	10001	420000
p.freeman	0.9999	0.01175	2.287E-5	1.0	1.0	1.0	1.0	1.0	10001	420000
p.likelihood	0.9998	0.01327	2.452E-5	1.0	1.0	1.0	1.0	1.0	10001	420000
temp	-501.8	266.4	1.719	-1102.0	-989.9	-468.2	-128.3	-90.08	10001	420000

	Dbar	Dhat	DIC	pD
failure	0.001057	-2.22E-15	0.002115	0.001057
total	0.001057	-2.22E-15	0.002115	0.001057

Flow Rate and Temperature model:

	mean	sd	MC_error	val2.5pc	val5.0pc	median	val95.0pc	val97.5pc	start	sample
dev.sat	14.32	2.864	0.04103	10.84	11.05	13.66	19.91	21.65	20001	440000
flow	0.2117	0.1853	0.003107	0.01283	0.01988	0.1603	0.5814	0.7007	20001	440000
int	-124.6	91.96	1.498	-329.5	-286.6	-116.6	8.383	32.1	20001	440000
p.bound[1]	0.2749	0.2193	0.002788	0.009963	0.01861	0.2205	0.7181	0.7917	20001	440000
p.bound[2]	0.4655	0.1849	0.001024	0.1281	0.1677	0.4618	0.7749	0.8224	20001	440000
p.bound[3]	0.3099	0.2258	0.002796	0.01493	0.02644	0.265	0.7465	0.8135	20001	440000
p.bound[4]	0.7391	0.1918	0.002204	0.2908	0.3638	0.7807	0.9756	0.9864	20001	440000
p.bound[5]	0.4713	0.2082	0.001939	0.09256	0.1311	0.471	0.8129	0.8567	20001	440000
p.bound[6]	0.4331	0.186	0.001212	0.1022	0.138	0.4277	0.7479	0.7971	20001	440000
p.bound[7]	0.6159	0.2669	0.003408	0.08379	0.1336	0.6553	0.973	0.9863	20001	440000
p.bound[8]	0.699	0.2256	0.00276	0.1868	0.2603	0.7474	0.9764	0.9873	20001	440000
p.bound[9]	0.9975	0.02834	1.389E-4	0.9963	1.0	1.0	1.0	1.0	20001	440000
p.bound[10]	0.9963	0.03195	1.775E-4	0.9842	0.9994	1.0	1.0	1.0	20001	440000
p.chisq	0.09106	0.2877	0.001429	0.0	0.0	0.0	1.0	1.0	20001	440000
p.freeman	0.1707	0.3763	0.002166	0.0	0.0	0.0	1.0	1.0	20001	440000
p.likelihood	0.2853	0.4515	0.002387	0.0	0.0	0.0	1.0	1.0	20001	440000
temp	0.9329	1.089	0.01652	-1.045	-0.74	0.8719	2.812	3.286	20001	440000

	Dbar	Dhat	DIC	pD
failure	14.29	11.97	16.6	2.312
total	14.29	11.97	16.6	2.312

Depth, Flow Rate, and Temperature model:

	mean	sd	MC_error	val2.5pc	val5.0pc	median	val95.0pc	val97.5pc	start	sample
depth	1118.0	567.9	4.89	258.4	335.9	1039.0	2168.0	2415.0	10001	420000
dev.sat	0.001085	0.0652	1.626E-4	0.0	0.0	0.0	0.0	0.0	10001	420000
flow	129.1	109.4	0.9017	-2.922	2.369	103.8	341.9	395.7	10001	420000
int	-8.726	991.5	7.364	-1937.0	-1633.0	-13.88	1627.0	1938.0	10001	420000
p.bound[1]	1.751E-6	4.011E-4	1.748E-6	0.0	0.0	0.0	0.0	0.0	10001	420000
p.bound[2]	0.0	0.0	1.543E-13	0.0	0.0	0.0	0.0	0.0	10001	420000
p.bound[3]	1.0	0.0	1.543E-13	1.0	1.0	1.0	1.0	1.0	10001	420000
p.bound[4]	0.0	0.0	1.543E-13	0.0	0.0	0.0	0.0	0.0	10001	420000
p.bound[5]	1.637E-4	0.009607	2.836E-5	0.0	0.0	0.0	0.0	0.0	10001	420000
p.bound[6]	0.9999	0.007338	1.567E-5	1.0	1.0	1.0	1.0	1.0	10001	420000
p.bound[7]	1.0	0.003074	5.973E-6	1.0	1.0	1.0	1.0	1.0	10001	420000
p.bound[8]	1.0	0.0	1.543E-13	1.0	1.0	1.0	1.0	1.0	10001	420000
p.bound[9]	1.0	2.367E-4	6.137E-7	1.0	1.0	1.0	1.0	1.0	10001	420000
p.bound[10]	1.0	0.0	1.543E-13	1.0	1.0	1.0	1.0	1.0	10001	420000
p.chisq	0.9999	0.01225	2.825E-5	1.0	1.0	1.0	1.0	1.0	10001	420000
p.freeman	0.9998	0.01263	3.014E-5	1.0	1.0	1.0	1.0	1.0	10001	420000
p.likelihood	0.9999	0.01123	2.102E-5	1.0	1.0	1.0	1.0	1.0	10001	420000
temp	-1061.0	528.6	4.652	-2293.0	-2046.0	-981.0	-346.8	-271.6	10001	420000

	Dbar	Dhat	DIC	pD
failure	8.954E-4	-2.22E-15	0.001791	8.954E-4
total	8.954E-4	-2.22E-15	0.001791	8.954E-4

C.5.2 Summaries for steel door cases using the developed assessment model.

Depth model:

	mean	sd	MC_error	val2.5pc	val5.0pc	median	val95.0pc	val97.5pc	start	sample
depth	2.051	1.257	0.008842	0.454	0.5713	1.769	4.495	5.216	5001	410000
dev.sat	12.88	2.274	0.01393	10.59	10.65	12.21	17.43	18.94	5001	410000
int	-75.68	46.99	0.3304	-194.1	-167.0	-65.1	-20.48	-16.15	5001	410000
p.bound[1]	0.9989	0.008775	2.901E-5	0.9906	0.9969	1.0	1.0	1.0	5001	410000
p.bound[2]	0.9527	0.06868	3.269E-4	0.7508	0.8076	0.9806	0.9998	0.9999	5001	410000
p.bound[3]	0.5863	0.1932	7.852E-4	0.1862	0.2423	0.6022	0.8769	0.9091	5001	410000
p.bound[4]	0.8259	0.1224	5.317E-4	0.5279	0.585	0.8524	0.9752	0.9844	5001	410000
p.bound[5]	0.7426	0.1455	5.158E-4	0.4136	0.4705	0.7635	0.9422	0.9597	5001	410000
p.bound[6]	0.7741	0.1367	5.198E-4	0.4571	0.5147	0.797	0.9553	0.9696	5001	410000
p.bound[7]	0.8019	0.1291	5.28E-4	0.4949	0.5526	0.8264	0.9662	0.9778	5001	410000
p.bound[8]	0.5863	0.1932	7.852E-4	0.1862	0.2423	0.6022	0.8769	0.9091	5001	410000
p.bound[9]	0.998	0.01201	4.207E-5	0.9802	0.9921	1.0	1.0	1.0	5001	410000
p.bound[10]	0.001599	0.01311	4.429E-5	0.0	0.0	2.918E-9	0.003224	0.01206	5001	410000
p.bound[11]	1.988E-4	0.00455	1.262E-5	0.0	0.0	6.038E-16	2.167E-5	2.265E-4	5001	410000
p.bound[12]	0.002777	0.018	6.33E-5	0.0	1.084E-19	5.916E-8	0.00858	0.02592	5001	410000
p.bound[13]	0.01107	0.04097	1.604E-4	4.919E-15	5.503E-13	2.862E-5	0.06292	0.1228	5001	410000
p.bound[14]	0.02189	0.06174	2.573E-4	7.276E-12	2.935E-10	3.407E-4	0.1338	0.2169	5001	410000
p.bound[15]	0.001061	0.01042	3.412E-5	0.0	0.0	2.462E-10	0.001442	0.006407	5001	410000
p.bound[16]	0.09811	0.1454	6.99E-4	3.023E-6	2.031E-5	0.02732	0.4305	0.523	5001	410000
p.bound[17]	0.7426	0.1455	5.158E-4	0.4136	0.4705	0.7635	0.9422	0.9597	5001	410000
p.bound[18]	0.8645	0.1108	5.187E-4	0.5826	0.6414	0.8936	0.9875	0.9928	5001	410000
p.bound[19]	0.9913	0.0272	1.096E-4	0.9173	0.9523	0.9997	1.0	1.0	5001	410000
p.chisq	0.1929	0.3946	0.001226	0.0	0.0	0.0	1.0	1.0	5001	410000
p.freeman	0.3346	0.4719	0.001277	0.0	0.0	0.0	1.0	1.0	5001	410000
p.likelihood	0.3825	0.486	0.001284	0.0	0.0	0.0	1.0	1.0	5001	410000

	Dbar	Dhat	DIC	pD
failure	12.89	11.36	14.42	1.528
total	12.89	11.36	14.42	1.528

Flow Rate model:

	mean	sd	MC_error	val2.5pc	val5.0pc	median	val95.0pc	val97.5pc	start	sample
dev.sat	14.29	2.371	0.04147	12.0	12.07	13.56	18.92	20.5	5001	410000
flow	0.01324	0.00633	1.237E-4	0.004552	0.005388	0.01209	0.02528	0.02886	5001	410000
int	-8.513	4.045	0.07928	-18.37	-16.27	-7.811	-3.387	-2.781	5001	410000
p.bound[1]	0.9844	0.03563	4.511E-4	0.8813	0.9218	0.9976	1.0	1.0	5001	410000
p.bound[2]	0.9826	0.03801	4.851E-4	0.8706	0.9139	0.997	1.0	1.0	5001	410000
p.bound[3]	0.9815	0.03942	5.052E-4	0.8645	0.9088	0.9966	1.0	1.0	5001	410000
p.bound[4]	0.9542	0.06816	9.094E-4	0.7567	0.8107	0.9817	0.9999	1.0	5001	410000
p.bound[5]	0.9829	0.0376	4.792E-4	0.8724	0.9152	0.9971	1.0	1.0	5001	410000
p.bound[6]	0.3908	0.1698	0.001643	0.09577	0.1285	0.3821	0.687	0.7435	5001	410000
p.bound[7]	0.3616	0.1668	0.001721	0.08027	0.1102	0.35	0.6578	0.7126	5001	410000
p.bound[8]	0.3748	0.1682	0.001686	0.08718	0.118	0.3642	0.671	0.7263	5001	410000
p.bound[9]	0.9529	0.06928	9.247E-4	0.7524	0.8063	0.9809	0.9999	1.0	5001	410000
p.bound[10]	0.02787	0.05128	7.892E-4	1.266E-5	4.478E-5	0.007991	0.124	0.1773	5001	410000
p.bound[11]	0.01265	0.03306	4.864E-4	3.043E-7	1.696E-6	0.001677	0.06083	0.0987	5001	410000
p.bound[12]	0.03554	0.05866	9.1E-4	3.481E-5	1.15E-4	0.01242	0.1508	0.2058	5001	410000
p.bound[13]	0.07872	0.08996	0.0014	7.302E-4	0.001729	0.0467	0.2658	0.3294	5001	410000
p.bound[14]	0.1546	0.1244	0.001836	0.007278	0.01352	0.1226	0.4018	0.463	5001	410000
p.bound[15]	0.02769	0.0511	7.861E-4	1.23E-5	4.363E-5	0.007895	0.1234	0.1765	5001	410000
p.bound[16]	0.3616	0.1668	0.001721	0.08027	0.1102	0.35	0.6578	0.7126	5001	410000
p.bound[17]	0.6429	0.1771	0.001774	0.2763	0.3301	0.656	0.9104	0.9387	5001	410000
p.bound[18]	0.7272	0.1669	0.001886	0.3608	0.4197	0.747	0.9575	0.974	5001	410000
p.bound[19]	0.9663	0.05666	7.497E-4	0.7983	0.8507	0.9894	1.0	1.0	5001	410000
p.chisq	0.2614	0.4394	0.003948	0.0	0.0	0.0	1.0	1.0	5001	410000
p.freeman	0.2253	0.4178	0.003569	0.0	0.0	0.0	1.0	1.0	5001	410000
p.likelihood	0.3591	0.4798	0.003713	0.0	0.0	0.0	1.0	1.0	5001	410000

	Dbar	Dhat	DIC	pD
failure	14.29	12.57	16.01	1.724
total	14.29	12.57	16.01	1.724

Depth and Flow Rate model:

	mean	sd	MC_error	val2.5pc	val5.0pc	median	val95.0pc	val97.5pc	start	sample
depth	1.833	1.362	0.01331	0.02651	0.1877	1.547	4.46	5.231	5001	410000
dev.sat	13.31	2.829	0.02513	9.925	10.1	12.64	18.82	20.49	5001	410000
flow	0.006935	0.006407	5.789E-5	-0.002997	-0.001705	0.005981	0.01896	0.02237	5001	410000
int	-72.5	50.5	0.4962	-200.0	-170.8	-61.18	-13.02	-8.288	5001	410000
p.bound[1]	0.9989	0.008774	3.181E-5	0.9916	0.9973	1.0	1.0	1.0	5001	410000
p.bound[2]	0.9821	0.04543	2.582E-4	0.8526	0.9064	0.9981	1.0	1.0	5001	410000
p.bound[3]	0.8399	0.2094	0.001363	0.2544	0.3619	0.9367	0.9999	1.0	5001	410000
p.bound[4]	0.9205	0.1015	5.766E-4	0.6297	0.7045	0.9603	0.9995	0.9998	5001	410000
p.bound[5]	0.9062	0.1437	8.68E-4	0.4678	0.5858	0.9708	0.9999	1.0	5001	410000
p.bound[6]	0.6138	0.2208	0.001414	0.1632	0.2191	0.6362	0.9327	0.9568	5001	410000
p.bound[7]	0.6302	0.2273	0.001552	0.1575	0.2157	0.6597	0.9467	0.9668	5001	410000
p.bound[8]	0.4327	0.23	0.001544	0.05646	0.08549	0.418	0.8312	0.8817	5001	410000
p.bound[9]	0.9966	0.01963	7.657E-5	0.9674	0.9872	1.0	1.0	1.0	5001	410000
p.bound[10]	0.001902	0.01385	5.106E-5	0.0	0.0	2.119E-9	0.004682	0.01656	5001	410000
p.bound[11]	0.001235	0.01762	5.719E-5	0.0	0.0	1.402E-15	2.243E-4	0.002252	5001	410000
p.bound[12]	0.00272	0.01657	6.494E-5	0.0	0.0	3.664E-8	0.009362	0.02735	5001	410000
p.bound[13]	0.009087	0.03429	1.635E-4	3.057E-16	6.079E-14	1.635E-5	0.05186	0.1015	5001	410000
p.bound[14]	0.02084	0.05926	3.119E-4	9.122E-13	5.726E-11	2.505E-4	0.13	0.2102	5001	410000
p.bound[15]	0.001847	0.01504	5.355E-5	0.0	0.0	2.462E-10	0.003298	0.01353	5001	410000
p.bound[16]	0.09375	0.1447	9.385E-4	9.469E-7	7.915E-6	0.02253	0.4259	0.524	5001	410000
p.bound[17]	0.7053	0.176	7.135E-4	0.3086	0.373	0.7331	0.9437	0.9615	5001	410000
p.bound[18]	0.8536	0.1311	7.371E-4	0.5081	0.5844	0.8927	0.9897	0.9941	5001	410000
p.bound[19]	0.9946	0.02018	9.412E-5	0.946	0.9718	0.9999	1.0	1.0	5001	410000
p.chisq	0.1417	0.3488	0.001286	0.0	0.0	0.0	1.0	1.0	5001	410000
p.freeman	0.2076	0.4056	0.001466	0.0	0.0	0.0	1.0	1.0	5001	410000
p.likelihood	0.2933	0.4553	0.001609	0.0	0.0	0.0	1.0	1.0	5001	410000

	Dbar	Dhat	DIC	pD
failure	13.35	11.05	15.66	2.304
total	13.35	11.05	15.66	2.304

C.5.3 Summaries for steel door depth regression model sensitivity analysis on prior distributions.**Flat prior distribution (*dflat()*):**

	mean	sd	MC_error	val2.5pc	val5.0pc	median	val95.0pc	val97.5pc	start	sample
depth2	2.047	1.267	0.009356	0.4545	0.5681	1.763	4.506	5.244	5001	410000

Uniform prior distribution (*dunif*(−5000,5000)):

	mean	sd	MC_error	val2.5pc	val5.0pc	median	val95.0pc	val97.5pc	start	sample
depth2	2.047	1.267	0.009356	0.4545	0.5681	1.763	4.506	5.244	5001	410000

Appendix D

OpenBUGS Convergence BGR

D.1 BGR convergence plots for hollow core door results

Depth regression model

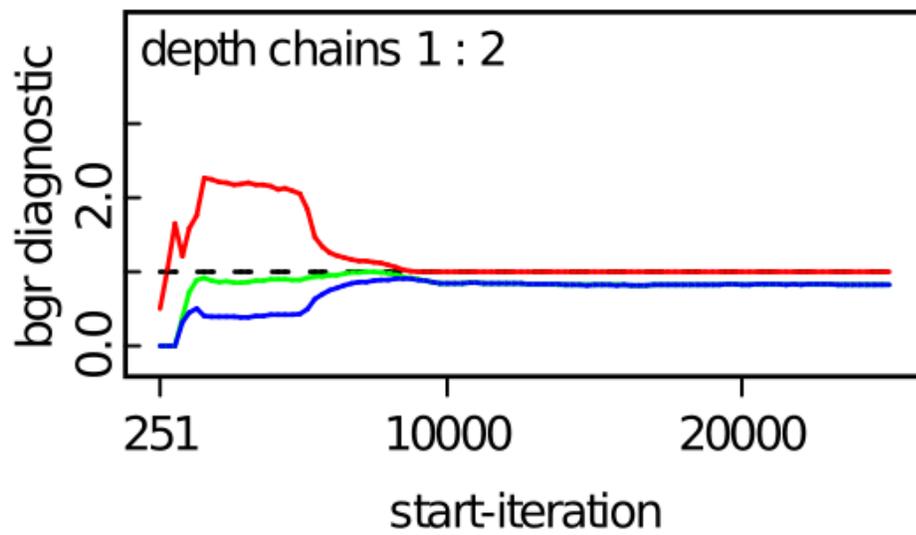


Figure D.1: Depth coefficient convergence plot (HCD-D Model).

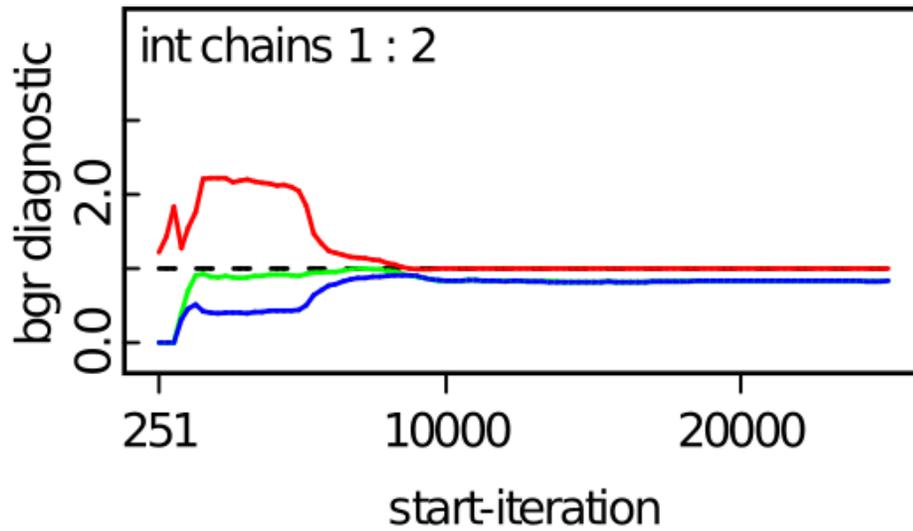


Figure D.2: Intercept convergence plot (HCD-D Model).

Flow rate regression model

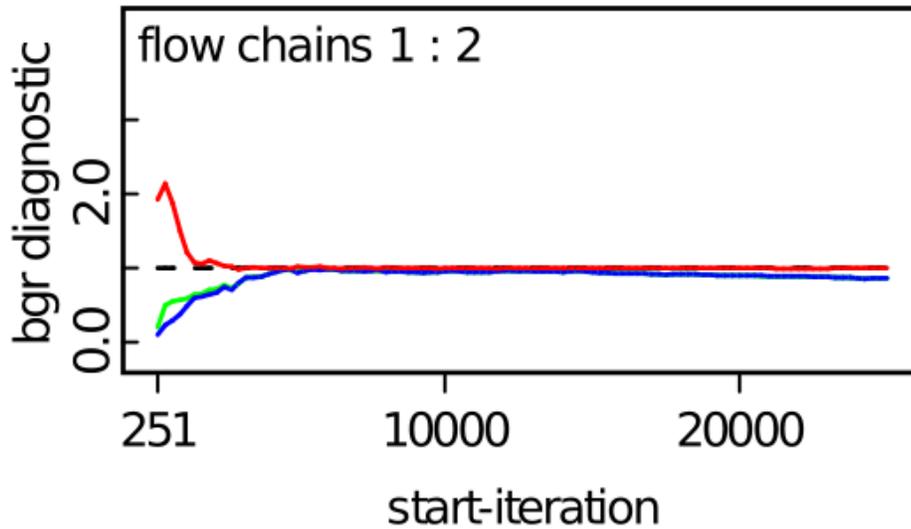


Figure D.3: Flow rate coefficient convergence plot (HCD-F Model).

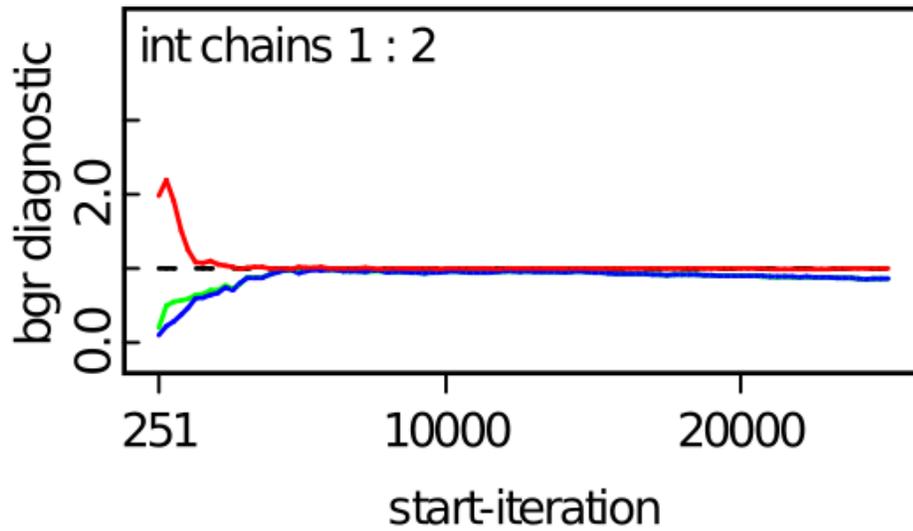


Figure D.4: Intercept convergence plot (HCD-F Model).

Temperature regression model

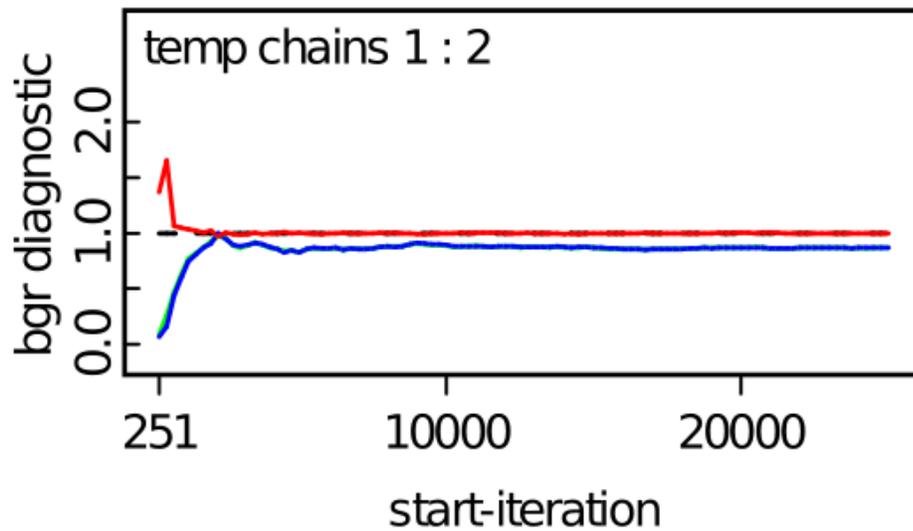


Figure D.5: Temperature coefficient convergence plot (HCD-T Model).

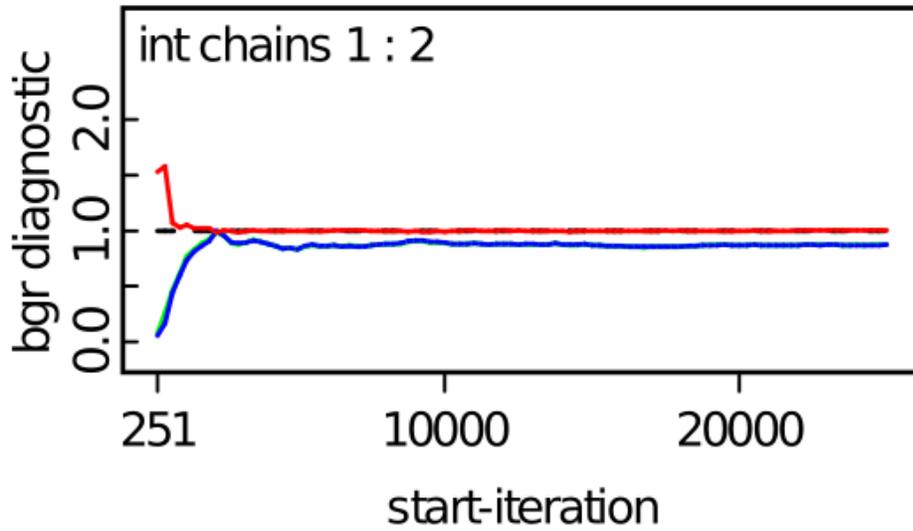


Figure D.6: Intercept convergence plot (HCD-T Model).

Depth and Flow rate regression model

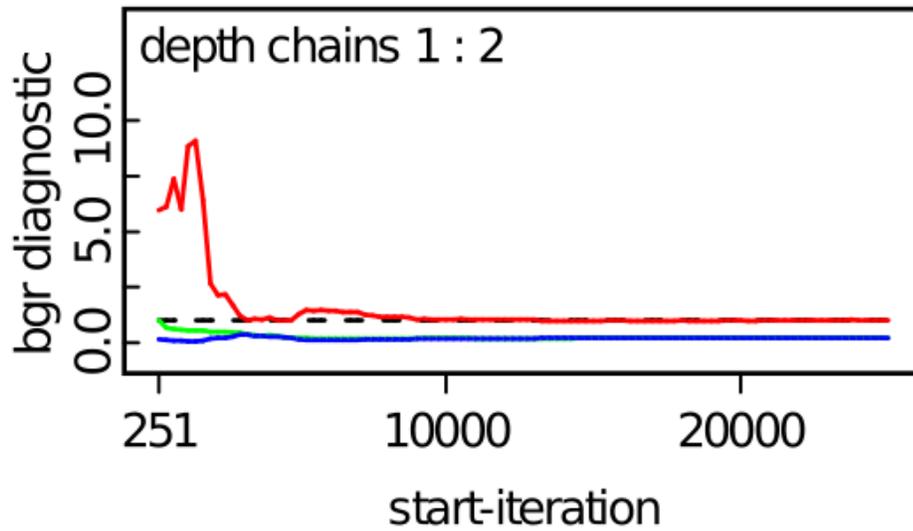


Figure D.7: Depth coefficient convergence plot (HCD-DF Model).

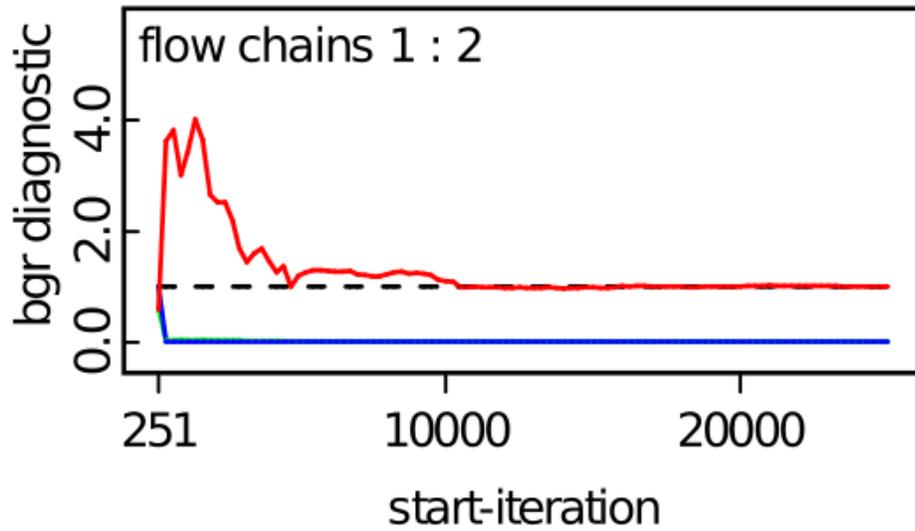


Figure D.8: Flow rate coefficient convergence plot (HCD-DF Model).

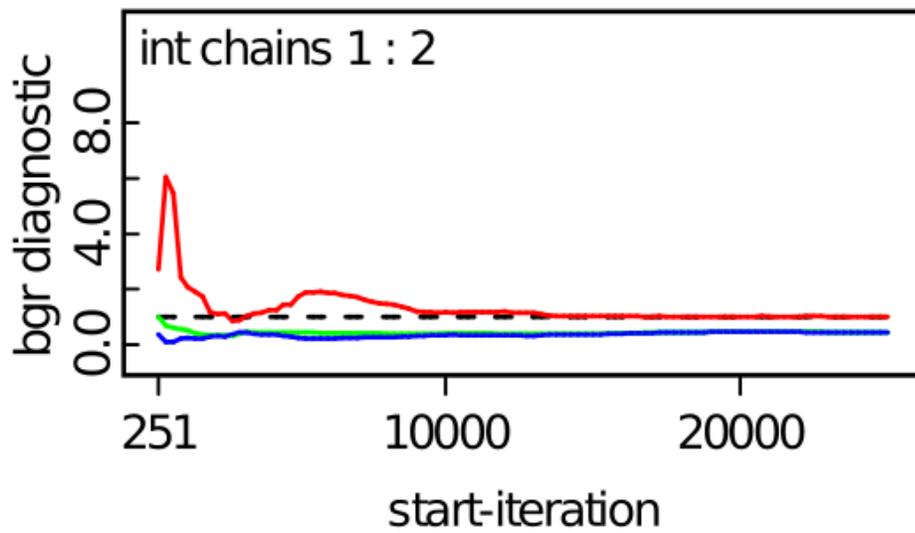


Figure D.9: Intercept convergence plot (HCD-DF Model).

Depth and Temperature regression model

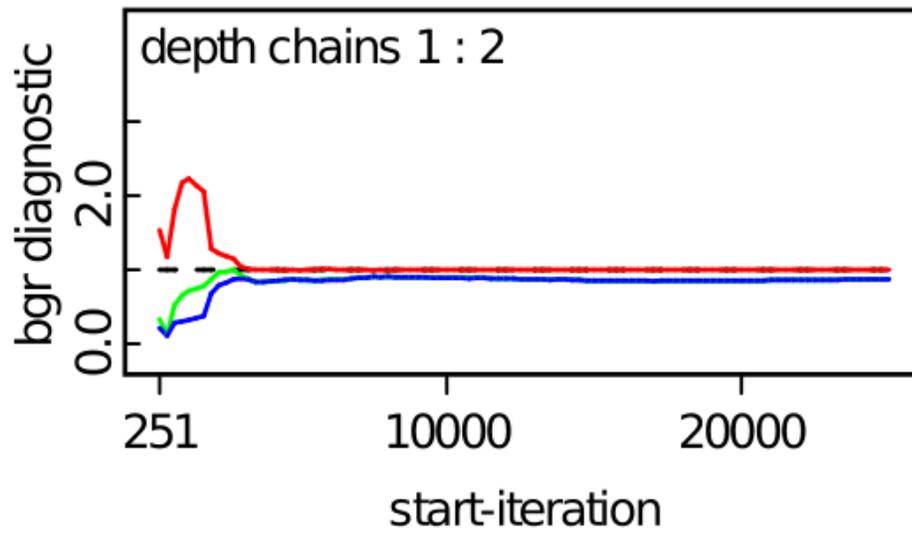


Figure D.10: Depth coefficient convergence plot (HCD-DT Model).

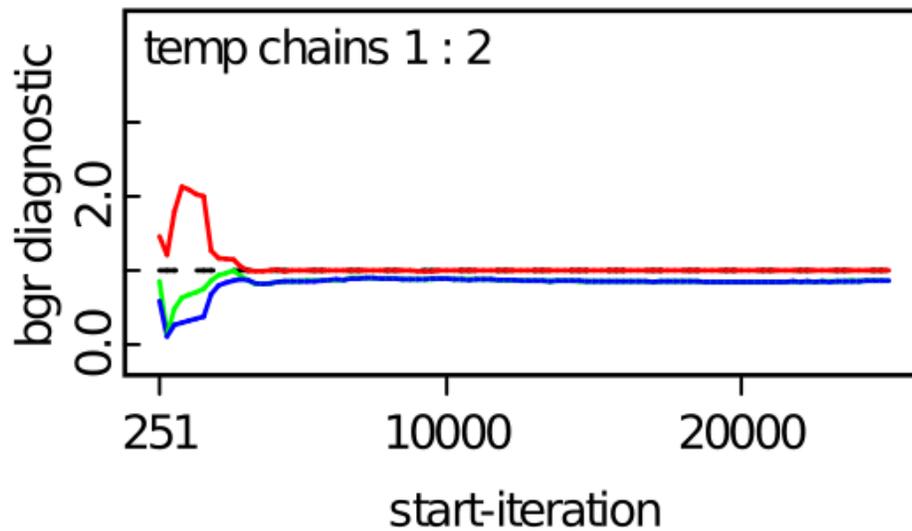


Figure D.11: Temperature coefficient convergence plot (HCD-DT Model).

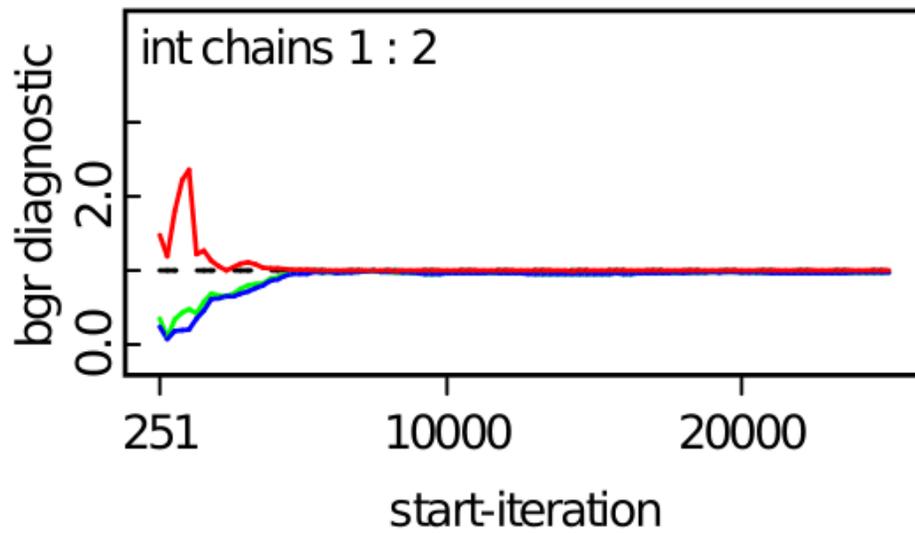


Figure D.12: Intercept convergence plot (HCD-DT Model).

Flow rate and Temperature regression model

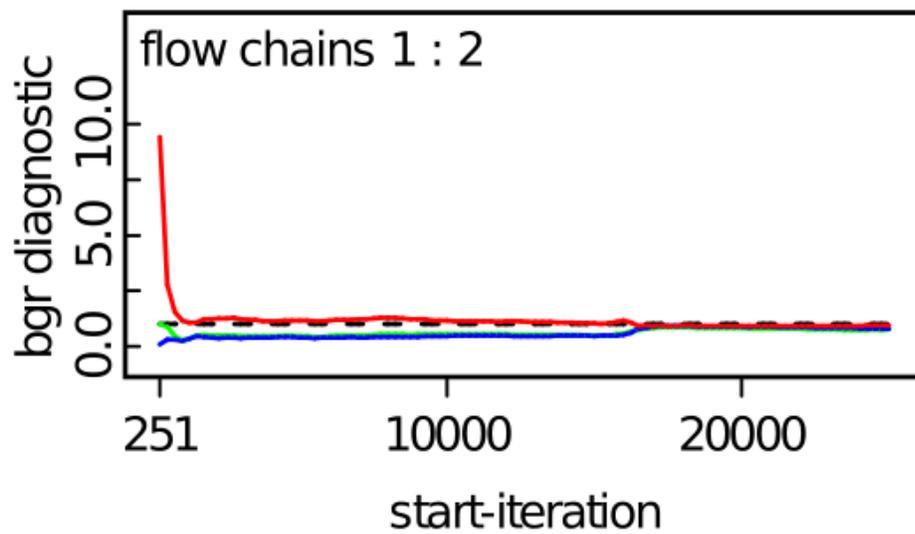


Figure D.13: Flow rate coefficient convergence plot (HCD-FT Model).

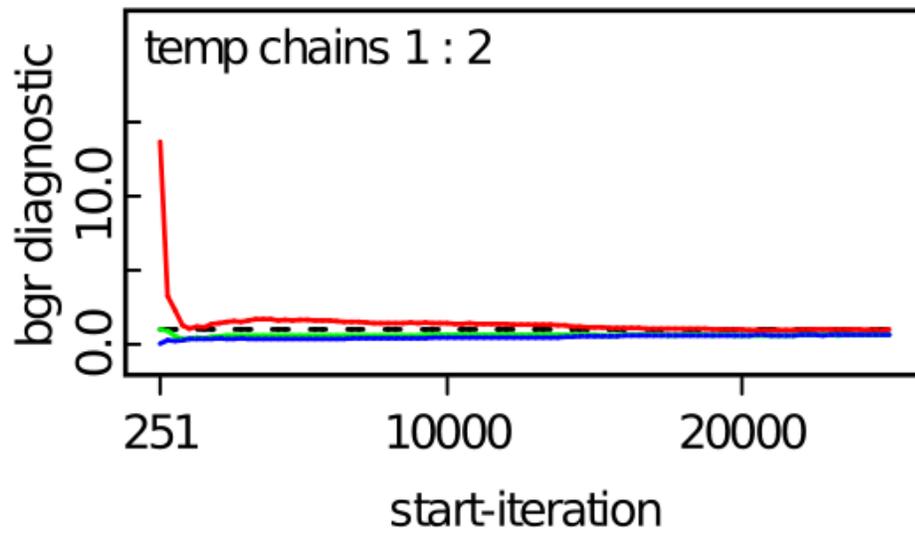


Figure D.14: Temperature coefficient convergence plot (HCD-FT Model).

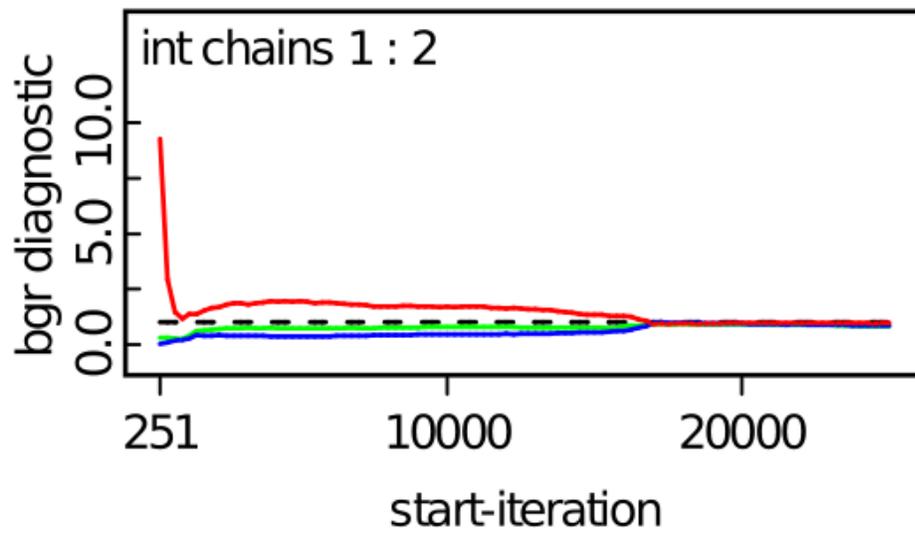


Figure D.15: Intercept convergence plot (HCD-FT Model).

Depth, Flow rate, and Temperature regression model

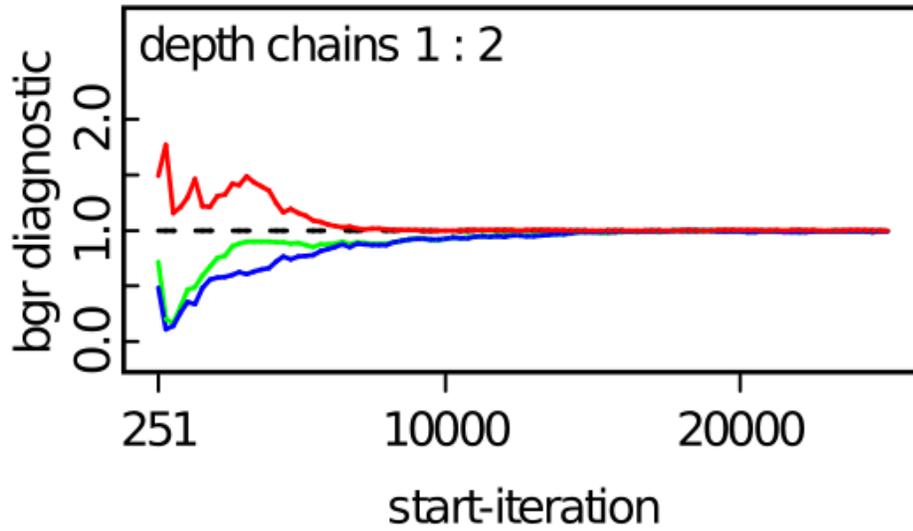


Figure D.16: Depth coefficient convergence plot (HCD-DFT Model).

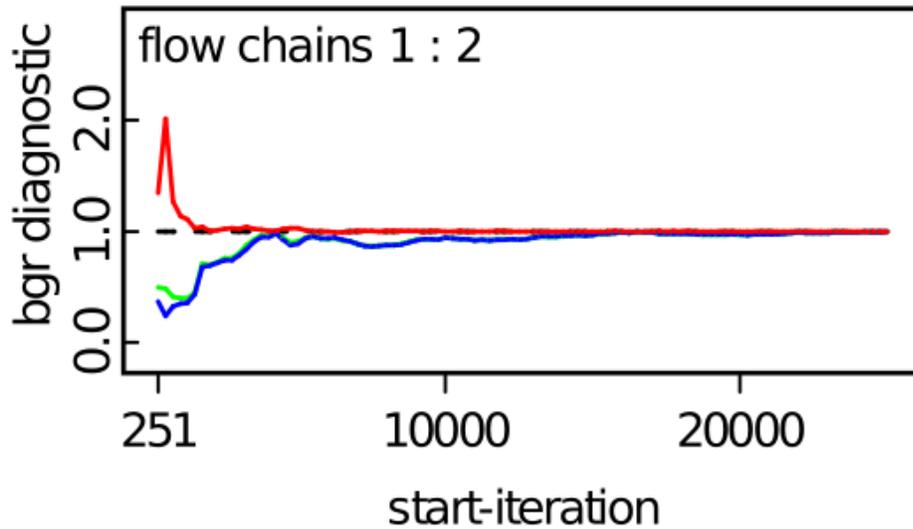


Figure D.17: Flow rate coefficient convergence plot (HCD-DFT Model).

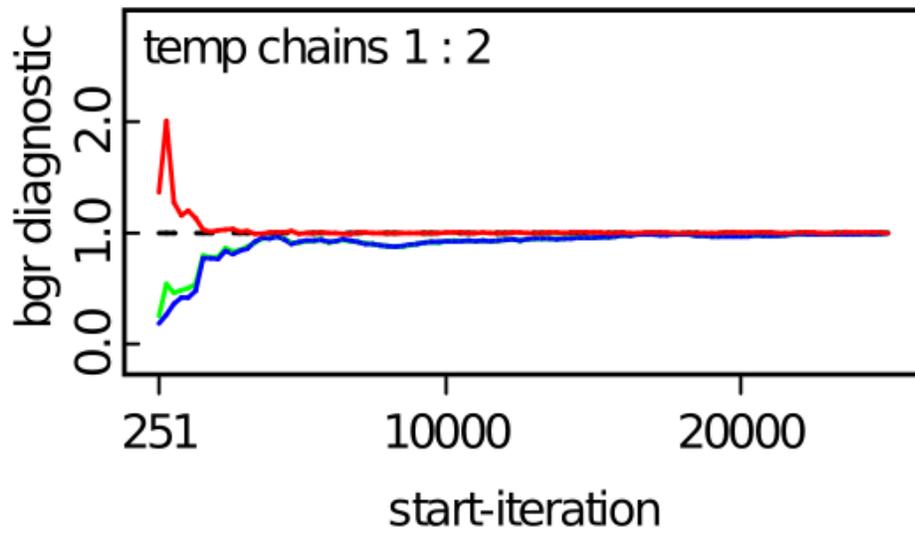


Figure D.18: Temperature coefficient convergence plot (HCD-DFT Model).

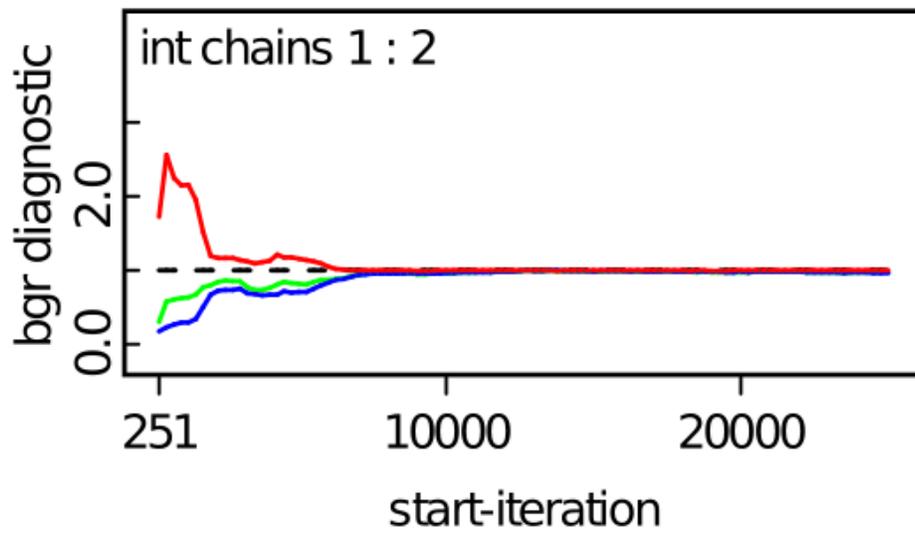


Figure D.19: Intercept convergence plot (HCD-DFT Model).

D.2 BGR convergence plots for steel door results

Depth regression model

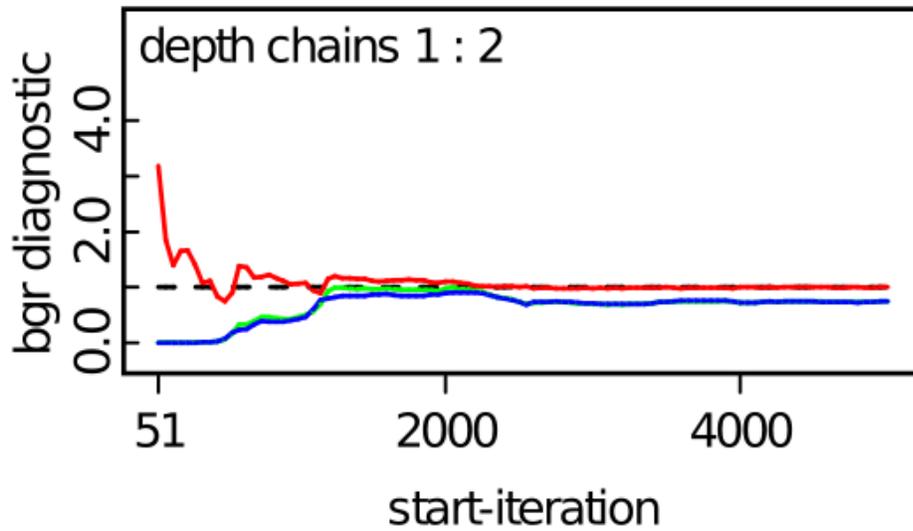


Figure D.20: Depth coefficient convergence plot (SD-D Model).

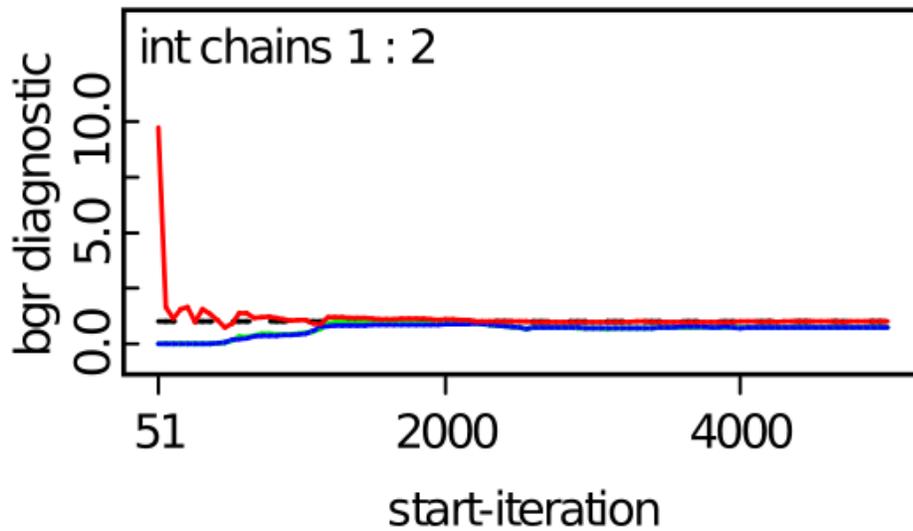


Figure D.21: Intercept convergence plot (SD-D Model).

Flow rate regression model

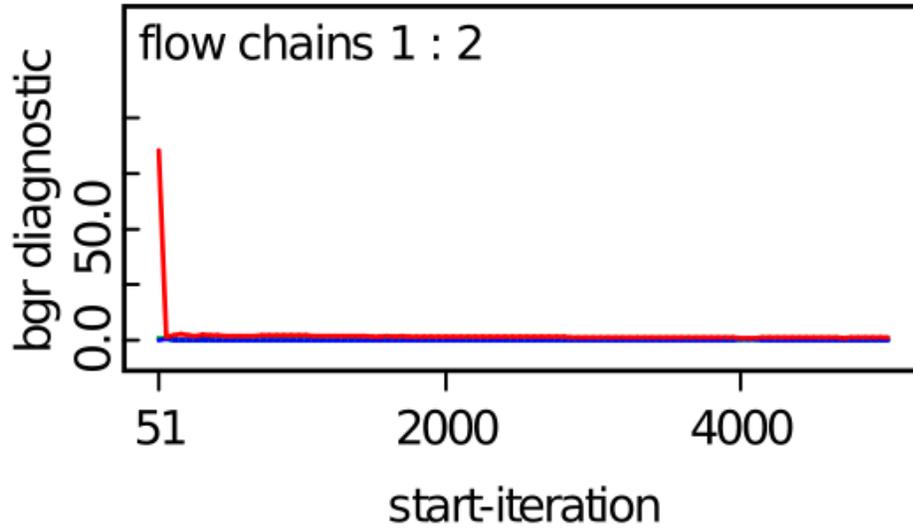


Figure D.22: Flow rate coefficient convergence plot (SD-F Model).

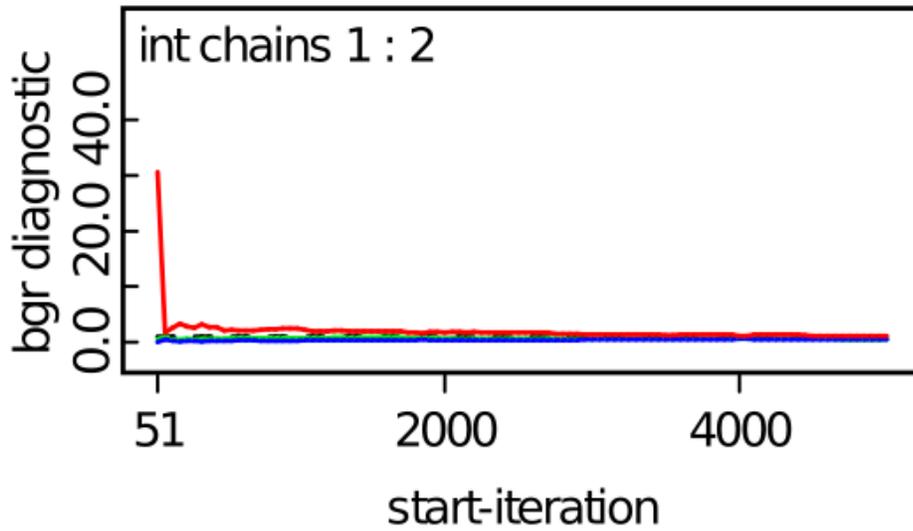


Figure D.23: Intercept convergence plot (SD-F Model).

Depth and Flow rate regression model

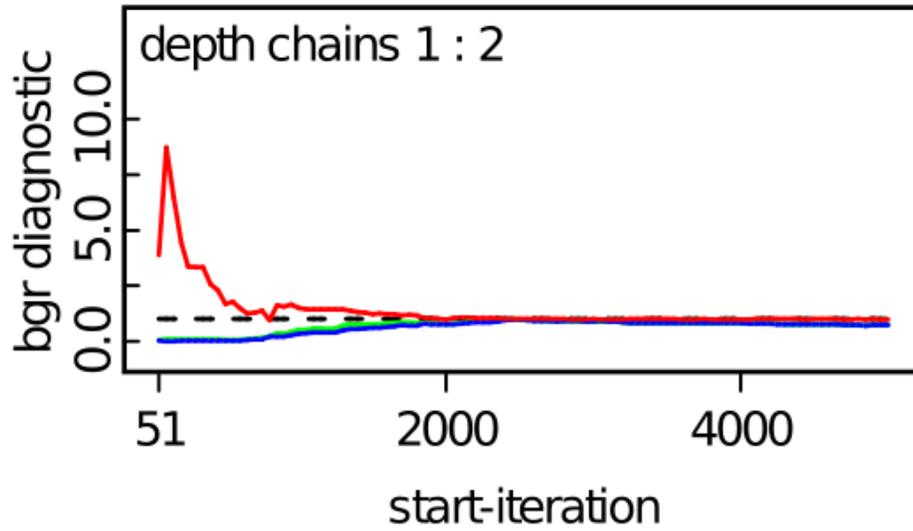


Figure D.24: Depth coefficient convergence plot (SD-DF Model).

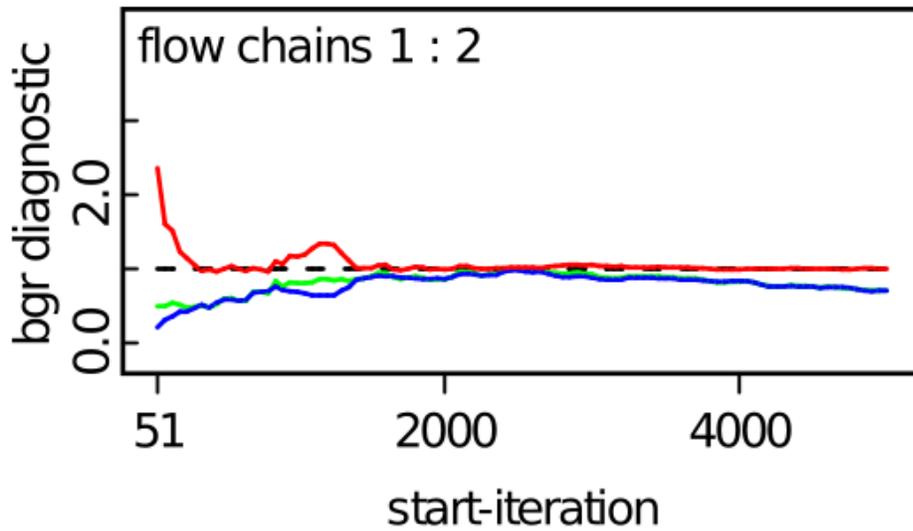


Figure D.25: Flow rate coefficient convergence plot (SD-DF Model).

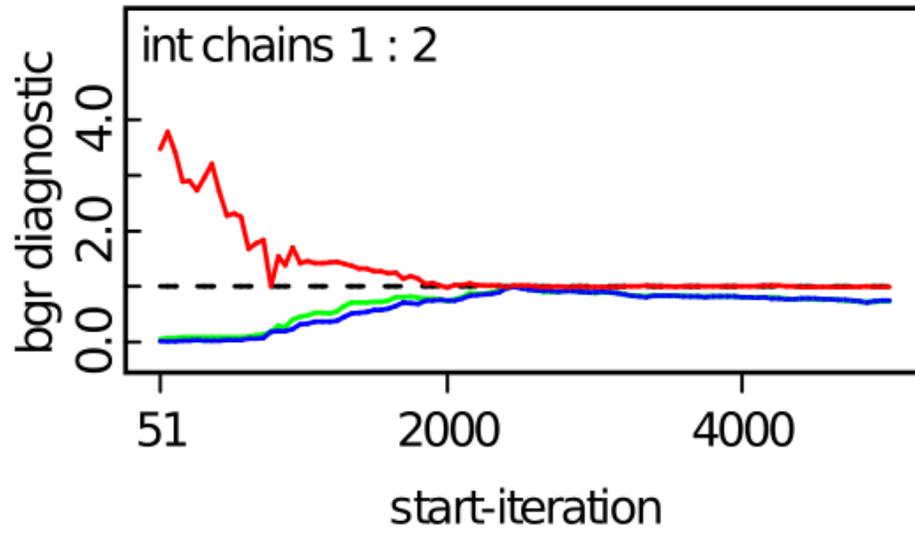


Figure D.26: Intercept convergence plot (SD-DF Model).

Bibliography

- [1] World Nuclear Association. Nuclear Power in the World Today, September 2019.
- [2] Water & Sustainability (Volume 3): U.S. Water Consumption for Power Production-The Next Half Century. Technical Report 1006786, Electric Power Research Institute, March 2002.
- [3] GL-88-20. Individual Plant Examination for Severe Accident Vulnerabilities - 10 CFR 50.54(f). Technical report, U.S. Nuclear Regulatory Commission, November 1988.
- [4] Design Bases for Protection Against Natural Phenomena. *10 CFR Appendix A to Part 50—General Design Criteria for Nuclear Power Plants*, August 2017.
- [5] Ben Chie Yen. Flood hazards for nuclear power plants. *Nuclear Engineering and Design*, 110(2):213–219, December 1988.
- [6] Harald Thielen. The Fukushima Daiichi nuclear accident—an overview. *Health Physics*, 103(2):169–174, August 2012.
- [7] Joon-Eon Yang. Fukushima Dai-Ichi Accident: Lessons Learned And Future Actions From The Risk Perspectives. *Nuclear Engineering and Technology*, 46(1):27–38, February 2014.
- [8] Mikel Kevin Fernandez-Cosials, Pilar Barreira, Gonzalo Jimenez, Cesar Queral. External flooding event analysis in a PWR-W with MAAP5. *Annals of Nuclear Energy*, 76:226–236, February 2015.

- [9] C. Rosales-Cooper, M. Johnson, E. Leeds, G. Tracy, J. Wiggins, B. Dean, V. McCree, C. Pederson, M. Dapas, K.S. West, S. Flanders, D. Skeen, R. Hannah, K. Foggie. Reflections on Fukushima: NRC Senior Leadership Visit to Japan. Technical Report NUREG/KM-0008, U.S. Nuclear Regulatory Commission, December 2014.
- [10] U.S. Nuclear Regulatory Commission. Event Notification Report for June 6, 2011: Event Number 46929, June 2011.
- [11] Standard & Poor's Rating Services. Omaha Public Power District, Nebraska; CP; Retail Electric. Technical Report SP-Report-9/30/14, McGraw Hill Financial, September 2014.
- [12] Mason Willrich. *Preview Modernizing America's Electricity Infrastructure*, chapter Chapter 5: Publicly Owned Utilities. MIT Press, 2017.
- [13] World-nuclear-news.org. Final shutdown for Fort Calhoun, October 2016.
- [14] Standard for level 1/large early release frequency probabilistic risk assessment for nuclear power plant applications. Technical Report ASME/ANS RA-Sb-2013, American Soc. Mechanical Engineers and American Nuclear Soc. (ASME/ANS), 2011.
- [15] Prabir C. Basu, M.K. Ravindra, Yoshinori Mihara. Component fragility for use in PSA of nuclear power plant. *Nuclear Engineering and Design*, 323:209–227, 2017.
- [16] David Kamerman. The Use of Flooding Fragility Curves in Nuclear Power Plant Risk Analysis. Master's thesis, Idaho State University, April 2016.

- [17] George Box, George Tiao. *Bayesian Inference in Statistical Analysis*. John Wiley & Sons Wiley, 1992.
- [18] Thomas Bayes. An essay towards solving a problem in the doctrine of chances. By the late Rev. Mr. Bayes, F. R. S. communicated by Mr. Price, in a letter to John Canton, A. M. F. R. S. *Philosophical Transactions of the Royal Society of London*, 53:370–418, January 1763.
- [19] Dana Kelly, Curtis Smith. *Bayesian Inference for Probabilistic Risk Assessment: A Practitioner's Guidebook*. Springer, 2011.
- [20] David Lunn, Christopher Jackson, Nicky Best, Andrew Thomas, David Spiegelhalter. *The BUGS Book: A Practical Introduction to Bayesian Analysis*. CRC Press, 2013.
- [21] Harry F. Martz, Ray A. Waller. *Bayesian Reliability Analysis*. John Wiley & Sons, 1982.
- [22] A. Gelman, J.B. Carlin, H.S. Stern, D.B. Dunson, A. Vehtari, D.B. Rubin. *Bayesian Data Analysis*. Texts in Statistical Science. CRC Press, third edition, 2014.
- [23] S. M. Lynch. *Introduction to Applied Bayesian Statistics and Estimation for Social Scientists*. Springer, New York, 2007.
- [24] OpenBUGS. *OpenBUGS User Manual*, version 3.2.3 edition, March 2014.
- [25] George Box. Science and statistics. *Journal of the American Statistical Association*, 71(356):791–799, December 1976.

- [26] Paul B. Conn, Devin S. Johnson, Perry J. Williams, Sharon R. Melin, Mevin B. Hooten. A guide to Bayesian model checking for ecologists. *Ecological Monographs*, 88(4):526–542, 2018.
- [27] Roger Levy. Probabilistic Models in the Study of Language. Chapter 8: Hierarchical Models, November 2012.
- [28] Report To the President : Actions to Implement the Recommendations of the Presidential Commission on the Space Shuttle Challenger Accident. Print, National Aeronautics and Space Administration, June 1986.
- [29] Bishwo Bhandari. Full Scale Door Testing Under Flooding Conditions to Develop Testing Protocol. Master's thesis, Idaho State University, September 2016.
- [30] Sneha Suresh. Development of an Interior Component Flooding Fragility Model and Design of the Component Flooding Evaluation Laboratory Safety Circuit. Master's thesis, Idaho State University, April 2017.
- [31] Andrew Gelman, Jennifer Hill. *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge University Press, 2007.
- [32] William C. Burns. *Spurious Correlations*, 1997.
- [33] R. A. Gingold, J.J. Monaghan. Smooth Particle Hydrodynamics: Theory and Applications to Non-Spherical Stars. *Monthly Notices of the Royal Astronomical Society*, 181:375–389, 1977.
- [34] L. B. Lucy. A numerical approach to the testing of the fission hypothesis. *Astronomical Journal*, 82:1013–1024, 1977.

- [35] R. Sampath, N. Montanari, N. Akinci, S. Prescott and C. Smith. Large-scale solitary wave simulation with implicit incompressible SPH. *Journal of Ocean Engineering and Marine Energy*, 2(3):313–329, 2016.
- [36] Nadir Akinci. *Interface Handling in Smoothed Particle Hydrodynamics*. PhD thesis, Albert Ludwig University of Freiburg, Freiburg im Breisgau, Baden-Württemberg, Germany, May 2014.
- [37] J. J. Monaghan. Smoothed Particle Hydrodynamics. *Annual Review of Astronomy and Astrophysics*, 30:534–574, 1992.
- [38] Neutrino Dynamic Initiative. *Neutrino Documentation*, release 1.0 edition, March 2018.
- [39] N. Akinci, M. Ihmsen, G. Akinci, B. Solenthaler and M. Teschner. Versatile Rigid-Fluid Coupling for Incompressible SPH. *ACM Transactions on Graphics (Proc. SIGGRAPH 2012)*, 31(4), 2012.
- [40] Prevention of Damage Due to Water Leakage Inside Containment (October 17, 1980 Indian Point 2 Event). Technical Report BL-80-24, United States Nuclear Regulatory Commission, November 1980.
- [41] Inadvertent Containment Spray Due to Personnel Error. Technical Report IN 81-10, United States Nuclear Regulatory Commission, March 1981.
- [42] Actuation of Fire Suppression System Causing Inoperability of Safety-Related Ventilation Equipment. Technical Report IN 87-14, United States Nuclear Regulatory Commission, March 1987.
- [43] Potential for Loss of Water from the Refueling Cavity. Technical Report IN 84-93, United States Nuclear Regulatory Commission, December 1984.

- [44] Kewaunee, Flooding Significance Determination Process Risk Assessment Report. Technical Report ML053180483, United States Nuclear Regulatory Commission, November 2005.
- [45] Inadequate or Inadvertent Blocking of Valve Movement. Technical Report IN 87-38, United States Nuclear Regulatory Commission, August 1987.
- [46] Potential Problems with the Use of Freeze Seals. Technical Report IN 91-41, United States Nuclear Regulatory Commission, June 1991.
- [47] Water Leakage from Yard Area Through Conduits Into Buildings. Technical Report IN 92-69, United States Nuclear Regulatory Commission, September 1992.
- [48] NRC Inspection Report 50-327/99-04, 50-328/99-04. Technical report, United States Nuclear Regulatory Commission, August 1999.
- [49] Fire Protection System Design Deficiencies and Common-Mode Flooding of Emergency Core Cooling System Rooms at Washington Nuclear Project Unit 2. Technical Report IN 98-31, United States Nuclear Regulatory Commission, August 1998.
- [50] Preliminary Notification of Event or Unusual Occurrence PNO-I-10-004. Technical Report ML102020350, United States Nuclear Regulatory Commission, July 2010.
- [51] Preliminary Notification of Event or Unusual Occurrence PNO-I-10-004A. Technical Report ML102160675, United States Nuclear Regulatory Commission, August 2010.

- [52] Calvert Cliffs Nuclear Power Plant Dual Unit Trip Following Loss of the '21' 13kV Bus. Technical Report PNO-I-14-001, United States Nuclear Regulatory Commission, January 2014.
- [53] Licensee Event Report 2014-001, Revision 00. Technical Report ML14080A308, United States Nuclear Regulatory Commission, March 2014.
- [54] Special Inspection Report 05000286/2015010. Technical Report ML15204A499, United States Nuclear Regulatory Commission, July 2015.
- [55] Inadequate Understanding of Lake Design Basis Results in Lake Level Higher Than Permitted by UFSAR. Technical Report LER 97-009 3731997009R00, United States Nuclear Regulatory Commission, April 1997.
- [56] Integrated Inspection Report 05000269/2006002, 05000270/2006002, 05000287/2006002. Technical Report ML061180451, United States Nuclear Regulatory Commission, April 2006.
- [57] Potential to Flood Switchgear Rooms Due to Missing Conduit Flood Seal. Technical Report ML12262A354, United States Nuclear Regulatory Commission, September 2012.
- [58] Missing Seals in Air Intake Tunnel Conduits. Technical Report ML12284A516, United States Nuclear Regulatory Commission, October 2012.
- [59] Potential for a Loss of Safety Function Due to Postulated Flood Conditions. Technical Report ML12363A104, United States Nuclear Regulatory Commission, December 2012.