© 2015 Rijan Prasad Shrestha

INVERSE UNCERTAINTY QUANTIFICATION OF INPUT MODEL PARAMETERS FOR THERMAL-HYDRAULICS SIMULATIONS USING EXPECTATION-MAXIMIZATION UNDER NON-BAYESIAN AND BAYESIAN FRAMEWORK

BY

RIJAN PRASAD SHRESTHA

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Nuclear, Plasma, and Radiological Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2015

Urbana, Illinois

Doctoral Committee:

Assistant Professor Tomasz Kozlowski, Chair
Research Scientist Dr. Brian F. Jewett
Professor Rizwan Uddin
Professor James Stubbins

ABSTRACT

Uncertainty quantification of computer simulation response requires knowledge of input model parameter uncertainty. However, like most system codes, nuclear thermal-hydraulics code TRACE does not provide any information on statistical properties of input model parameters. Moreover, the input model parameters in TRACE code are built using correlations from experiments performed under steady-state low pressure low flow conditions. Hence, they might not be accurate for use in analyses of high pressure high flow transients in a reactor core. This further highlights the need for quantification of input model parameter uncertainty.

A mathematical framework is developed where Expectation-Maximization (EM) algorithm is implemented to quantify input model parameter uncertainty using the Maximum Likelihood Estimate (MLE) and Maximum a Posteriori (MAP) estimate. The difference between experimental measurements and nominal code predictions, which are the observables, are considered scalar random variables. A dispersed normal prior is assumed on the mean and an inverse gamma prior is assumed on the variance of the observable to determine MAP estimate. A log-normal transformation is used to transform input model parameter probability distribution function to pseudo-parameter space. The theory is formulated such that the observables are expressed either as a linear or a quadratic combination of pseudo parameters for MLE. MAP estimate, on the other hand, uses a linear model. In addition, the pseudo parameters are assumed to be normally distributed. Experimental data is collected from reflooding facility that simulates post Loss of Coolant Accident (LOCA). Thermal-hydraulics system code TRACE is used for calibration purposes. Discussion of results obtained from the implementation of the developed methodology is presented. The comparisons show that calibrated code results obtained using MAP estimates show consistent improvement over those obtained from MLE. The mean and variance of the input parameters hence calculated can be used along with the underlying distribution to perform uncertainty quantification on output code responses. Moreover, MAP estimates of variance are consistently lower compared to MLE due to regularization in the form of prior knowledge, hence providing greater credibility to inverse estimation.

Acknowledgments

I would like to express to my sincere gratitude towards my advisor Professor Tomasz Kozlowski for his valuable support throughout this project. I thank him for his perpetual encouragement, patience and guidance. I would also like to thank Dr. Brian Jewett, Professor Rizwan Uddin and Professor James Stubbins for serving in my Doctoral Committee.

I would also like to thank my friends Santosh Koirala, Abhishek Jaiswal, Xu Wu and Guojun Hu for their genuine friendship and encouragement. Special thanks to Rabie Abu Saleem and Christie Marie Bermudez, you were always there willing to help me any day, any time. I would also like to acknowledge all the faculty and staff of NPRE for their continuous help and wisdom.

Finally, I would also like to thank the two most important people in my life, my parents, for their continuous support and encouragement; I would have been nowhere close to this without their love.

Table of Contents

LIST OF FIGURES....vi

CHAPTER 1 INTRODUCTION....1

1.1 Introduction....1

1.2 Problem Statement....2

1.3 Literature Review....4

1.4 Overview of Organization....10

CHAPTER 2 THEORY AND METHODOLOGY....11

2.1 Maximum-Likelihood Estimate....12

2.2 Expectation-Maximization Algorithm....13

2.3 Maximum Likelihood Estimate – Linear Model....14

2.3.1 Estimating the Covariance Matrix....17

2.3.2 Estimating the Mean Vector....18

2.4 Maximum Likelihood Estimate – Quadratic Model....21

2.5 Maximum a Posteriori Estimate....26

2.5.1 MAP Estimate of Mean Vector....31

2.5.2 Direct Estimate of Covariance Matrix....35

2.5.3 Estimation of Covariance Matrix using Conjugate Prior....37

2.5.4 Weighting Scheme and Algorithm....40

2.6 Wrapper Iteration....42

CHAPTER 3 FEBA REFLOODING FACILITY AND TRACE MODEL....45

3.1 Introduction of FEBA Test Facility....45

3.2 Experiment Description....47

3.3 Computational Model In TRACE....48

3.4 TRACE Results of the Transient....51

3.4.1 Results Comparison....52

CHAPTER 4 RESULTS....55

4.1 Maximum Likelihood Estimate of Analytical Functions....56

4.1.1 Linear Function....56

4.1.2 Non-linear Function....60
4.2 Sensitivity Analysis....64
4.3 Synthetic Reflooding Experiment....65
4.3.1 Synthetic Example I: Dispersed flow film boiling HTC and interfacial drag coefficient....66
4.3.2 Synthetic Example II: Interfacial drag coefficient and single phase vapor to liquid HTC....70
4.4 Model Uncertainty for LOCA Re-flooding....74
4.4.1 Dispersed flow film boiling HTC and departure from nucleate boiling critical heat flux....77
4.4.2 Dispersed flow film boiling HTC and interfacial drag coefficient....78
4.4.3 Interfacial drag coefficient and single phase vapor to wall HTC....79
4.4.4 Interfacial drag coefficient and departure from nucleate boiling critical heat flux.….80
4.4.5 Single phase vapor to wall HTC, dispersed flow film boiling HTC and interfacial drag coefficient....81
4.4.6 Single phase vapor to wall HTC, departure from nuclear boiling critical heat flux and dispersed flow film boiling HTC....82
4.4.7 Single phase vapor to wall HTC, departure from nuclear boiling critical heat flux and interfacial drag....83
4.4.8 Departure from nuclear boiling critical heat flux, dispersed flow film boiling HTC and interfacial drag....84
4.4.9 Single phase vapor to wall HTC, departure from nuclear boiling critical heat flux, dispersed flow film boiling HTC and interfacial drag....85
4.4.10 Error Norm Comparison....86
4.5 Validation using FEBA Test 234....89
4.6 Discussion....91
CHAPTER 5 CONCLUSION....97
APPENDIX A. EXPECTATION-MAXIMIZATION ALGORITHM....100
APPENDIX B. MINIMUM MEAN SQUARE ERROR....103
BIBLIOGRAPHY....106

LIST OF FIGURES

Figure 1.1 User effect analysis.....3
Figure 1.2 Simplified flow diagram of CIAU.....5
Figure 1.3 Propagation of input parameters in forward uncertainty quantification.....6
Figure 1.4 Example of forward propagation.....6
Figure 1.5 Elements and analysis flow of IMTHUA methodology.....9
Figure 3.1 Cross-sectional view of FEBA test facility.....46
Figure 3.2 Axial view of the simulator heater rod.....46
Figure 3.3 Transient temperature distribution.....48
Figure 3.4 Nodalization diagram in TRACE.....50
Figure 3.5 Transient clad temperature computed using TRACE.....52
Figure 3.6 Comparison between code and experimental results.....52
Figure 4.1 Comparison between base code responses and experimental responses.....58
Figure 4.2 Comparison between base code, corrected code and experimental responses.....59
Figure 4.3 Comparison of calibrated results with extreme code responses.....60
Figure 4.4 Comparison between base code responses and experimental responses.....62
Figure 4.5 Comparison of base code, corrected code and experimental responses with code …..63
Figure 4.6 Cladding temperature profile during transient (dispersed flow film boiling HTC and interfacial drag coefficient).....68
Figure 4.7 Cladding temperature profile during transient (interfacial drag coefficient and single phase vapor to liquid HTC).....71
Figure 4.8 Comparison of calibrated results (dispersed flow film boiling HTC and departure from nucleate boiling critical heat flux).....77
Figure 4.9 Comparison of calibrated results (dispersed flow film boiling HTC and interfacial drag coefficient).....78
Figure 4.10 Comparison of calibrated results (interfacial drag coefficient and single phase vapor to wall HTC).....79
Figure 4.11 Comparison of calibrated results (interfacial drag coefficient and departure from nucleate boiling critical heat flux).....80

Figure 4.12 Comparison of calibrated results (single phase vapor to wall HTC, dispersed flow film boiling HTC and interfacial drag coefficient).....81
Figure 4.13 Comparison of calibrated results (single phase vapor to wall HTC, departure from nuclear boiling critical heat flux and dispersed flow film boiling HTC).....82
Figure 4.14 Comparison of calibrated results (single phase vapor to wall HTC, departure from nuclear boiling critical heat flux and interfacial drag).....83
Figure 4.15 Comparison of calibrated results (departure from nuclear boiling critical heat flux, dispersed flow film boiling HTC and interfacial drag).....84
Figure 4.16 Comparison of calibrated results (single phase vapor to wall HTC, departure from nuclear boiling critical heat flux , dispersed flow film boiling HTC and interfacial drag).....85
Figure 4.17 Comparison of error norm (two parameters).....86
Figure 4.18 Comparison of error norm (three parameters).....87
Figure 4.19 Comparison of error norm (four parameters).....88
Figure 4.20 Comparison of calibrated results for FEBA Test 234.....90
Figure 4.21 Comparison of error norm for FEBA Test 234.....91

CHAPTER 1 INTRODUCTION

1.1 Introduction

Computer simulations are being increasingly used in scientific community in lieu of actual experiments in the design and analysis of engineering systems. Proper use of computer models decrease overhead costs and time associated with running equivalent experiments. Hence, the credibility of results from simulations are vital to scientists and engineers. However, various factors need to be considered while substituting experiments with these computer simulations. Therefore, confidence in modeling and simulation must be critically assessed which requires model *verification* and *validation (V&V)* [1]. Verification refers to the process of determining that a model implementation accurately represents the developer's conceptual description of the model and its solution. Validation refers to the process of determining the degree of accuracy of the considered model in representing real world physical phenomena. It is in the context of the latter that *uncertainty quantification (UQ)* plays a vital role. UQ aims at developing rigorous methods to properly characterize and quantify the variability and lack of knowledge of code parameters while determining the simulated *Quantity of Interest (QoI)*. It is now helpful to make a brief distinction between the concepts of *errors* and *uncertainties*.

The American Institute of Aeronautics and Astronautics (AIAA) defines *errors* as recognizable deficiencies of the models or the algorithms implemented [2]. Thus, it is associated with the limitations of numerical implementation like presence of round-off errors, incorrect numerical convergence, unidentified bugs in the code, etc. On the other hand, *uncertainty* is the measure of precision during any measurement. *Uncertainties* are associated with implementation of physical models and specifications of input parameters to the code. Thus, one may view errors as dealing with mathematics and numerical solution to the code, and uncertainties as dealing with the physics and input parameters to

the code. Furthermore, uncertainty can be classified into two categories, (i) *aleatoric uncertainty* and (ii) *epistemic uncertainty*. Aleatoric uncertainty, also known as stochastic uncertainty, is the physical variability of the system and is characterized using probabilistic approaches. Epistemic uncertainty, also called reducible uncertainty, is deficiency caused due to lack of knowledge about a system. It can arise from assumptions or model simplifications. Epistemic uncertainty, unlike aleatory uncertainty, can not be characterized by probabilistic approaches since an inherent lack of knowledge makes it difficult to make assumptions and hence infer any reasonable information.

1.2 Problem Statement

U.S. NRC (Nuclear Regulatory Committee) has advocated the use of Best Estimate Plus Uncertainty (BEPU) approach for thermal-hydraulics (TH) transient calculations over conservative approach. Conservative approach calculates the output response of a code using extreme (unfavorable) values of input parameters. For example, this approach consistently over-estimates the cladding temperature and hence under-predicts the time to failure. The reason behind this approach is to model the physical phenomena such that it always predicts the worst case scenario [3]. In contrast, best-estimate codes try to predict the outcome as realistically as possible. TRACE (TRAC/RELAP Advanced Computational Engine) and RELAP5 (Reactor Excursion and Leak Analysis Program) are examples of best-estimate thermal-hydraulics codes. BEPU approach facilitates more precise calculation of safety margins and helps in regulating licensing and operation of nuclear reactors [4]. In this approach, statistical properties of the code parameters need to be accurately quantified in order to achieve the goals set by the *BEPU* approach. Figure 1.1 illustrates the need for uncertainty quantification in a TH calculation.

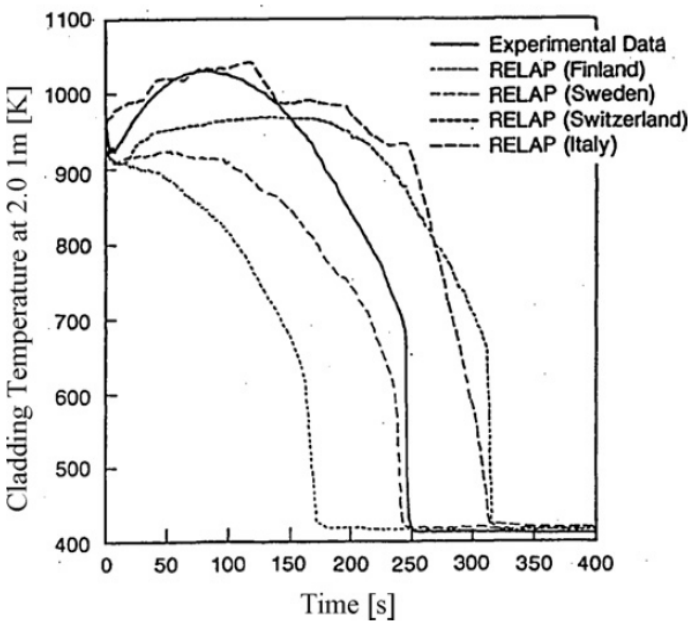

Figure 1.1 User effect analysis [5]

Figure 1.1 shows one of the problems scientists face while using a complicated legacy TH code. The plot shows Loss of Coolant Accident (LOCA) cladding temperature behavior for a given experimental facility as obtained by various participating institutions. The participants were not allowed to communicate with each other. With numerous code parameters and multitude of options for physical models for each parameter to choose from, the participants predictably made different choices – hence a wide variety of temperature prediction. More importantly, not a single participant modeled the response satisfactorily close to the experimental data.

With the advent of enhanced multi-physics, multi-scale modeling and simulation (M&S) capabilities, there has been a strong focus on reducing the uncertainties in empirically developed models [6]. More recently, U.S. Department of Energy (DOE) Nuclear Energy Advanced Modeling and Simulation (NEAMS) program has initiated an approach that would transfer contemporary advances in computational sciences and computer technologies to future programs [7][8]. One of the goals of this program is to incorporate

any incremental improvement on knowledge-base of TH code parameters. Though work has been done to this end, it is not as well-developed and comprehensive as the methodologies developed for *forward uncertainty quantification.* Forward UQ is defined as the quantification of uncertainties in system code responses obtained by propagating uncertain input parameters.

A lot of scientific literature merely supposes arbitrarily assigned probability distribution and bounds on TH code parameters to facilitate *forward quantification.* Historically, *expert judgment* has been used to specify the probability distribution and bounds on code parameters. The main drawbacks of such approach are connected with (a) the need for engineering judgment for limiting (in a given case) the number of the uncertain input parameters, and (b) the need for engineering judgment for fixing the range of variation and the probability density function (p.d.f.) for each uncertain input parameter [9]. This thesis aims to replace such *ad-hoc expert judgment* on the statistical properties of TH input code parameters with systematic and rigorously derived methodology.

1.3 Literature Review

Uncertainty quantification can be broadly classified into two categories – *forward uncertainty quantification* and *inverse uncertainty quantification.* As described earlier, the former quantifies uncertainties in system code outputs obtained by propagating uncertainties in input parameters. Thus, it requires knowledge of input (uncertain) parameter space and its variability. There has been a number of studies on *forward uncertainty quantification* and various analytical techniques have been developed. In *forward propagation* the idea is to observe the behavior of the outputs as a function of code (uncertain) parameters. Some examples of this approach implemented by commercial vendors in nuclear industry are GRS [9], CSAU [10], AREVA [11] and ASTRUM (Automated Statistical Treatment of Uncertainty Method) by Westinghouse [12][13]. *Code with Internal Assessment of Uncertainty* (CIAU) methodology was

proposed by D'Auria et. al. [14]. The main idea of CIAU is the identification and the characterization of current plant state and the association of uncertainty to each state. Output quantity (QoI) and time uncertainties are combined for each plant state under this approach. A simplified flowchart for CIAU is shown in Figure 1.2. Furthermore, Jaeger et. al. coupled TRACE with uncertainty and sensitivity toolkit called SUSA under the premises of BEPU for analysis of Supercritical Water Reactors [15]. *Forward uncertainty quantification* enables the end-users to evaluate reliability (equivalently, performance) of the outputs and helps in optimizing the process under consideration. Figure 1.3 shows a schematic of how the statistical knowledge of code input (uncertain) parameters could be propagated to deduce statistical inference about output response.

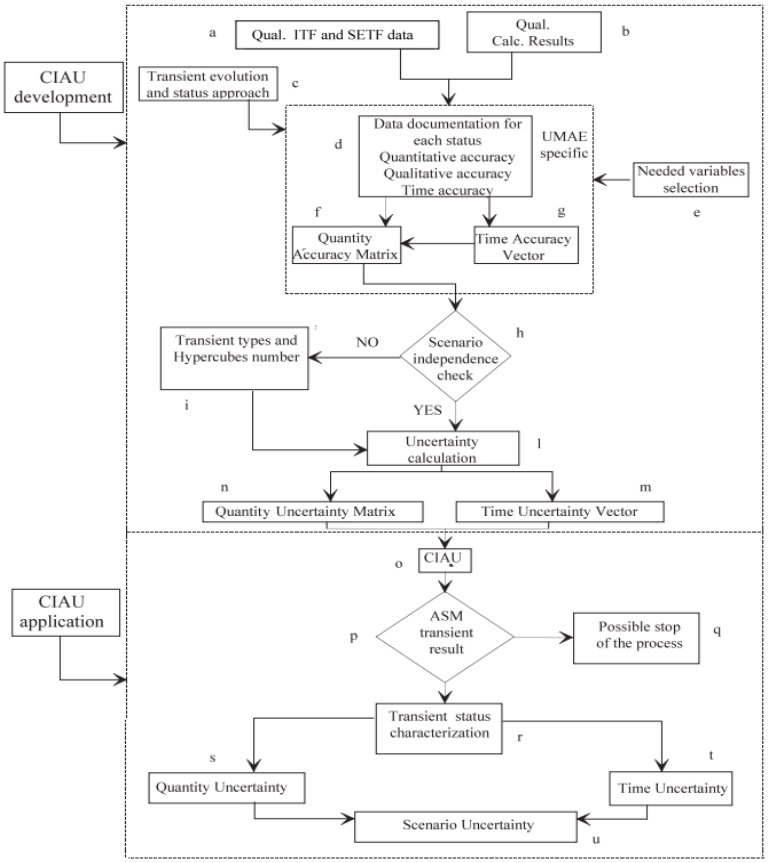

Figure 1.2 Simplified flow diagram of CIAU [14]



Figure 1.3 Propagation of input parameters in forward uncertainty quantification [16][17]

Analysis of *forward uncertainty quantification* could be carried out using probabilistic sampling methodologies, like Monte-Carlo (MC) sampling or Latin Hypercube Sampling (LHS). With the availability of adequate computing resources Full Factorial Numerical Integration (FFNI) [18] or Dimension Reduction (DR) [19] techniques could also be implemented. Other simpler non-probabilistic approaches for *forward uncertainty quantification* include interval analysis [20] and evidence theory [21][22][23]. Figure 1.4 shows an example of application of *forward* propagation on a transient TH problem for predicting the clad temperature profile.

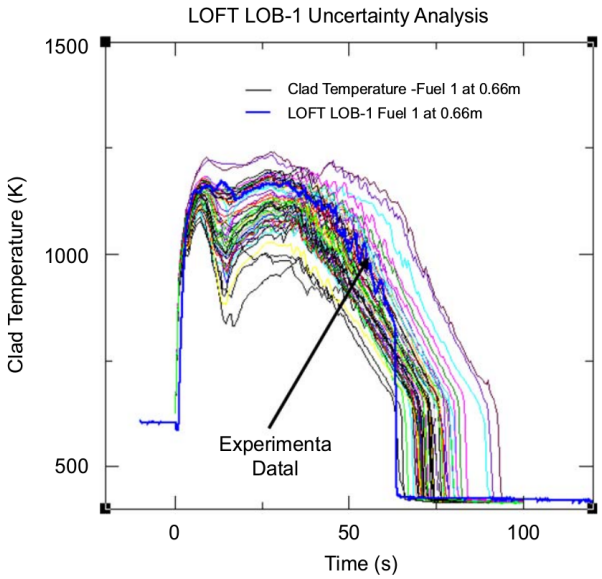

Figure 1.4 Example of forward propagation [24]

An important step before *forward propagation* is to determine the values and ranges of the parameters that characterize the system. This is called the *inverse uncertainty quantification.* The *inverse* problem seeks to estimate uncertain input parameters given relevant experimental measurements and code simulation results. It aims to quantify the discrepancies between experiment and simulation. *Inverse uncertainty quantification* is sometime referred to as *parameter calibration* or *parameter estimation*. In short, the *inverse problem* seeks to determine an unknown code parameter vector $\underline{\theta}$ that yields code output responses equal to the observed experimental data $\underline{Y}$. In other way, it can be viewed as a deterministic non-linear optimization problem that aims at minimizing the norm between the observed (experimental) and predicted (calculated) responses. However, deterministic optimization problem fails to capture the uncertainty associated with its estimates. Since the goal of the inverse problem is to characterize the uncertainty of the input parameters, merely determining point estimates of the best-fit input parameters is not sufficient. Rather, a statistical description of the code model parameters that are consistent with the observed data is sought in this thesis.

As discussed in Section 1.2 current practice is based on ad-hoc assignment of statistical parameters and probability distribution function (p.d.f.) on code inputs. These are based on users' expert-judgment instead of a rigorous mathematical derivation. For example, based on *expert-judgment*, an interval of 20% on either side of default input model parameter value may be assumed along with its p.d.f. to quantify the uncertainty range for that model parameter. However reasonable as it may look, quantifying the input uncertainty using *expert-judgment* is unscientific and lacks mathematical rigor. Hence, it should be avoided if possible. This thesis provides both *frequentist* and *Bayesian* framework to quantify the uncertainty in the code (physical model) parameters of a thermal-hydraulics code. Each framework is based on the application of Expectation-Maximization (EM) algorithm on a set of experimental observations and thermal-hydraulics (in this thesis, TRACE) simulation results. Once inverse quantification is completed, forward uncertainty quantification can be carried out by running a large number of code simulations to generate confidence intervals on the output.

Dempster, Laird and Rubin developed the EM algorithm to find the *Maximum Likelihood Estimate* (MLE) of parameters in the context of incomplete data models [25][26]. EM algorithm has been widely used to determine MLE of parameters in Gaussian Mixture Model (GMM) and Hidden Markov Model (HMM) [27][28]. Under the Bayesian approach, *Maximum a Posteriori* (MAP) estimate of the code's input (physical model) parameters are sought. Modification of the EM algorithm to find the posterior mode of parameters in GMM has been reported in literature for a wide range of fields [29][30][31] [32].

Physical models in the thermal-hydraulics system code such as TRACE were developed using experimental data from Separate Effect Tests (SET). The codes using these models were validated using Integral Test Facilities (ITF). However, SET experiments can not be isolated to a single physical phenomena [33]. Even SET experiments believed to isolate certain phenomenon are partly Integral Effect (IE) in nature and a direct approach to estimate individual model parameters does not yield optimum result. In this thesis, EM algorithm is used in the context of an Integral Effect Test (IET) to approximate the *most likely* (non-Bayesian) and *posterior* (Bayesian) mode of model parameters.

de Crecy et al. implemented EM algorithm to estimate the MLE of model parameters under the assumption of a linear model [34][35]. MLE is convenient to use, however it does not account for any prior information on the parameters. Moreover, MLE when solved using EM algorithm may fail as a result of singularities and degeneracies. Other inverse methodologies and hybrid approaches include *Integrated Methodology on Thermal-hydraulics Uncertainty Assessment* (IMTHUA) developed by Mosleh et. al. [24]. The Bayesian methodology developed by Mosleh et al. demonstrates a mechanism of updating the results of uncertainty quantification of TH system codes when additional data are available from tests and experiments. This updating technique thus developed enable to refine and adjust the uncertainty distributions by propagating inputs and sub-model uncertainty contributors to TH code predictions. However, this methodology is essentially a forward treatment of TH code uncertainty and assumes availability of

adequate knowledge about the input parameter space. The schematic for IMTHUA is shown in Figure 1.5. The phase for uncertainty quantification of the input parameter is given in block B in Figure 1.5, which is suggested to be derived from engineering judgment or availability of knowledge.

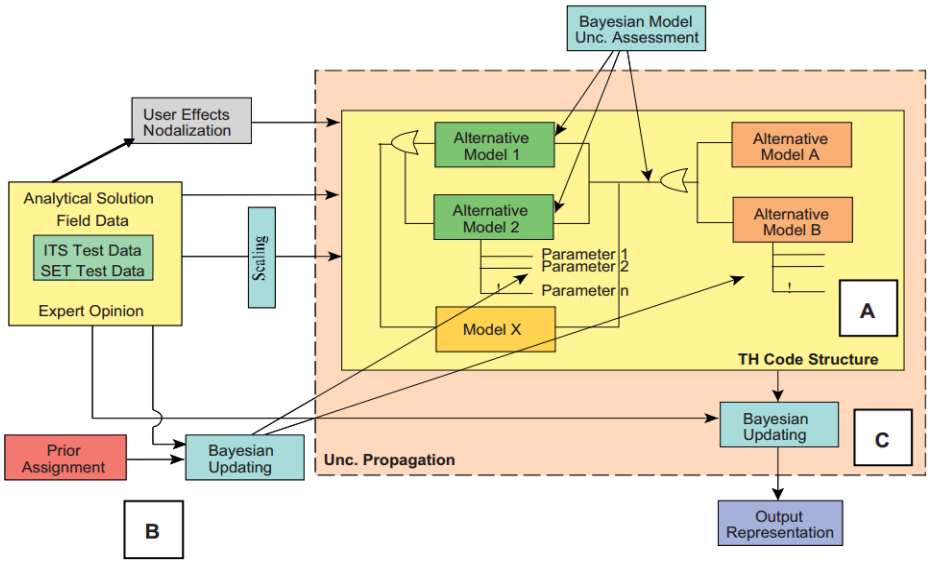

Figure 1.5 Elements and analysis flow of IMTHUA methodology [24]

Cacuci et. al. formulated mathematical methodology for *Predictive Modeling of Coupled Multi-Physics Systems* (PMCMPS) that utilizes maximum entropy principle to construct optimal approximation of unknown a priori distribution in coupled systems [36][37]. Earlier, Cacuci and Ionescu-Bujor developed rigorous methodology for computing best-estimate predictive results using experimental information in conjunction with models of stationary and time-dependent systems [38]. When only first and second moments of model parameters and experimental responses are a priori available, as is most often the case in practice, Cacuci et al. demonstrated that implementation of the maximum entropy principle of statistical mechanics in conjunction with information theory could be used to construct a Gaussian prior distribution that takes all of the available information into account while minimizing the introduction of spurious information. Advanced calibration

and validation approach that employs data integration from multiple sources and facilitates simultaneous calibration of related parameters has been proposed more recently by Bui, Williams and Dinh [39]. The technical approach for model calibration and validation as suggested by Bui, Williams and Dinh involves generating simulation data needed to construct model surrogate using DAKOTA [40], then constructing model surrogate using a combination of Principal Component Analysis (PCA) [41], process convolution techniques [42] and Gaussian processes (GPs). Finally, they recommended Bayesian calibration using Markov Chain Monte Carlo (MCMC). A comprehensive review and comparison of different TH uncertainty quantification methodologies are presented in [43][44].

1.4 Overview of Organization

The organization of this thesis is as follows. Chapter 2 presents the theoretical framework developed. It shows complete derivation to quantify the statistical parameters of the TRACE code input parameters using Expectation-Maximization algorithm under *non-Bayesian* (linear and quadratic) and *Bayesian* framework. Chapter 3 describes the reflooding experiment that serves as a benchmark to calibrate the code input parameters. This chapter presents computational TRACE model of the experimental facility and TRACE code prediction for the reflood phase of PWR Large Break LOCA (LB-LOCA) experiment. Chapter 4 presents examples of implementation of the methodology developed in Chapter 2. This Chapter includes (a) calibrating parameters of analytical functions, (b) calibrating TRACE code coefficients using manufactured (known) TRACE-generated data and (c) calibrating parameters using actual experimental data from Chapter 3. More specifically, calibration of important input (physical models) TRACE code coefficients are carried out and their statistical parameters are quantified for different cases. Validation of the methodology is presented at the end of the Chapter. Chapter 5 summarizes the research and discusses shortcomings, and recommends possible extension of this research to improve accuracy and fidelity.

CHAPTER 2 THEORY AND METHODOLOGY

This Chapter presents a detailed description and derivation of the methodology used to quantify the uncertainties in code model parameters. We proceed by providing a few definitions used for subsequent derivations along with general introduction to relevant theories. Comprehensive mathematical derivation specific to the context of the problem is provided after that.

In mathematical modeling, hypothesis about the physical processes are given by parametric families of probability distributions called models. Once a model is specified, tests for goodness of fit are carried out to determine the set of parameters that best fits the observed data. This is called *parameter estimation*. Least Square Estimation (LSE) and Maximum-Likelihood Estimation (MLE) are two popular means of parameter estimation. LSE is a good tool that provides quantitative measure to summarize observed data but it could not be used for hypothesis-testing and establishing confidence intervals. In essence, LSE returns a single value (or vector in case of multi-dimension) for the parameter estimate. MLE, on the other hand, is a formulation based on probabilistic assessment of parametric models and observed data and hence facilitates quantification of precision – variance. MLE provides *sufficiency:* complete information about the parameter of interest, *consistency:* true parameter value that generated the data, and *efficiency:* lowest possible variance [45]. In this Chapter, comprehensive derivation to estimate MLE are carried out under two assumptions (i) linear and (ii) quadratic. Moreover, in some circumstances prior knowledge may be available about Quantity of Interest (QoI). This knowledge can come either from past experience, published literature, or simply intuition. Formulation based on MLE does not account such prior information. Moreover, MLE may fail sometime as a result of singularities and degeneracies. Formulation based on Maximum a Posteriori (MAP) estimates of the parameters, on the other hand, addresses both these issues by assigning appropriate priors to *QoIs* and hence avoids singularities during computation.

2.1 Maximum-Likelihood Estimate

Let $f(X|\underline{\theta})$ denote the probability density function that represents the probability of observing a random variable X given a set of parameters $\underline{\theta}$. If the observations $x_1, x_2,... x_n$ of random variable $X_1, X_2, \ldots X_n$ are assumed to be identical and independently distributed (i.i.d.) then the joint probability distribution of observing the data-set is given by the product of probability of observing each realization. Then,

$$f(x_{1,}x_2,...x_n|\underline{\theta})=f(x_1|\underline{\theta}).f(x_2|\underline{\theta})...f(x_n|\underline{\theta})=\prod_{i=1}^{n} f(x_i|\underline{\theta}) \tag{2.1}$$

Eq. (2.1) can be interpreted as the *likelihood function*. This can be viewed as the inverse problem where we reverse the roles and treat the parameter $\underline{\theta}$ as the unknown in the availability of the dataset $x_1, x_2,... x_n$. Consequently, we write the following.

$$L(\underline{\theta}|\underline{x})=\prod_{i=1}^{n} f(x_i|\underline{\theta}) \tag{2.2}$$

where L is the likelihood function dependent on $\underline{\theta}$. The principle of *maximum likelihood estimation* developed by Fisher states that the parameters of the probability density should be selected such that the distribution makes the observed dataset most likely [25]. This is done by maximizing the likelihood function over the parameter space. The parameters thus obtained are called the *maximum likelihood estimates*. In practice, it is more convenient to maximize log of the likelihood. Since log is a monotonically increasing function, maximizing the likelihood function or the log-likelihood function yield the same estimate [46]. Eq. (2.3) below is known as the likelihood equation.

$$\frac{\partial}{\partial \theta_i}\ln L(\underline{\theta}|\underline{x})=0 \tag{2.3}$$

for $i = 1,2 \ldots I$ where I is the number of unknown parameters. Substituting Eq. (2.2) in Eq. (2.3) we obtain the following.

$$\frac{\partial}{\partial \theta_i} \sum_{n=1}^{N} \ln f(x_n|\underline{\theta}) = 0 \tag{2.4}$$

The solutions to Eq. (2.4) are the MLE of the parameters. This concept will be used throughout the derivation in the following sub-sections.

2.2 Expectation-Maximization Algorithm

Expectation-Maximization (EM) algorithm is an iterative method to find the maximum likelihood estimate of the parameters in a statistical model [26][47][48]. It is generally used when the likelihood equation can not be solved directly for the MLE parameters (e.g. non-linear in parameters $\underline{\theta}$) or is too complicated to solve analytically. The general principle of the EM algorithm is to start with a guess for I parameters where the number of total unknowns (including hidden variables) is greater than I. The algorithm then finds estimates of hidden variables under previously assumed parameter values. After that, it iterates between an Expectation (E) step and Maximization (M) step until convergence. For illustration, we take a simple case of bivariate normal distribution with mean vector $\underline{\mu}$ and covariance matrix $\underline{\underline{\Sigma}}$, which are yet to be determined. First, an initial guess for $\underline{\underline{\Sigma}}$ is provided. In the E-step, expected log-likelihood function is formulated using the current estimate of $\underline{\underline{\Sigma}}$. In the M-step, the log likelihood function, which is a function of $\underline{\mu}$, is maximized to find the optimum mean vector components. The updated $\underline{\mu}$ is then used to generate better approximation to $\underline{\underline{\Sigma}}$ for next (E) iteration, and the process is repeated. A detailed review of EM algorithm is presented in Appendix A. Desirable properties of EM algorithm include numerical stability, reliable convergence, easy implementation and a low cost per iteration. However, the convergence of EM algorithm is slower than the quadratic convergence obtained via Newton-type non-linear solvers [49]. Dempster et al. showed a linear rate of convergence for EM algorithm. This is somewhat balanced by

lower cost per iteration in cases with no missing data. The EM algorithm is a very generic algorithm and can be adjusted according to user needs. Details of the implementation specific to our problem are presented in the sub-sections below.

2.3 Maximum Likelihood Estimate – Linear Model

Method to estimate MLE of model parameters under linear model assumption was originally developed by de Crecy [34]. Given a set of experimental observations, for example temperature and pressure from a thermal-hydraulics experiment, denoted by $Y_{exp,1}$, $Y_{exp,2}$ … $Y_{exp,N}$ and corresponding set of code results (code output) given by $Y_{code,1}$, $Y_{code,2}$ …. $Y_{code,N}$, we wish to quantify the uncertainty in the input (model parameters) of the code. Special attention is given to the parameters present in the physical models, rather than the parameters related to nodalization, geometry or initial and boundary conditions. Let N be the number of experimental responses considered. Then, for each $Y_{exp,n}$ we have a corresponding $Y_{code,n}$, for $n=1,2,...N$. Let I model parameters (e.g. wall-liquid heat transfer coefficient, interfacial friction coefficient, etc.) be considered. The parameters are assumed to be statistically uncorrelated. Modification of these code parameters is performed via physical multipliers p_i. Multipliers are real numbers $p>0$ that are multiplied to the code model parameters (e.g. wall-liquid heat transfer coefficient, interfacial friction coefficient, etc.) with the aim of making the code-calculated results best-fit the observed experimental results. The physical multiplier p_i for each model parameter is related to the pseudo parameter α_i as shown in Eq. (2.5):

$$p_i = e^{\alpha_i} \tag{2.5}$$

for $i= 1,2...,I$

where p is the actual multiplier of the model parameter and α is the pseudo parameter. The desired property of Eq. (2.5) is such that it is continuous, differentiable at least once and monotonically increasing. The base (unmodified) code results are obtained with $p=1$,

which corresponds to $\alpha = 0$ (also called nominal case). A linear relationship (or any other relationship that satisfies above requirements) could also be chosen between α and p in place of exponential transformation. The exponential transformation was chosen to avoid negative values for the physical multiplier p.

For each experimental observation $Y_{exp,n}$, $n = 1, 2, \ldots N$, it can be hypothesized that there exists some pseudo parameter vector $\underline{\alpha}_n$ such that the following holds:

$$dY_n^* = Y_{exp,n} - Y_{code,n}^* = 0$$

where $Y_{code,n}^*$ is calculated at $\underline{\alpha}_n$. Now, taking into account $Y_{exp,n}$ and the sensitivity of the code with respect to the pseudo parameters α_i, i.e. the sensitivity matrix $\frac{\partial Y_{code,n}}{\partial \alpha_i}|_{\alpha_i=0}$, we aim to determine the mean of these pseudo parameter vectors, i.e. $E[\underline{\alpha}_n]$. For simplicity, we will assume that the experimental observations have no error and are the true values of the measurements carried out. Furthermore, each element in the sensitivity matrix *(N by I)* is obtained as shown in Eq. (2.6):

$$\frac{\partial Y_{code,n}}{\partial \alpha_i}|_{\alpha_i=0} = \left(\frac{\partial Y_{code,n}}{\partial p_i}|_{p_i=1}\right) \cdot \left(\frac{\partial p_i}{\partial \alpha_i}|_{\alpha_i=0}\right) \tag{2.6}$$

For each observation, we have one random vector $\underline{\alpha}_n$ *(n = 1, 2, … N).* We assume these random vectors $\underline{\alpha}_n$ are *independently and identically distributed (i.i.d.)* random vectors with mean $\underline{b}$ and covariance $\underline{\underline{\Sigma}}$. Furthermore, we make the following assumptions.

1. The random vectors $\underline{\alpha}_n$ follow multivariate normal distribution.
2. A linear relationship exists between the code response $Y_{n,code}$ and pseudo parameter vector $\underline{\alpha}_n$.

Then, we have the following:

$$Y_{\exp,n} - Y_{code,n}|_{\underline{\alpha}=0} = \left(\frac{\partial Y_n}{\partial \underline{\alpha}}|_{\underline{\alpha}=0}\right)^T d\,\underline{\alpha} = \left(\frac{\partial Y_n}{\partial \underline{\alpha}}|_{\underline{\alpha}=0}\right)^T \underline{\alpha}_n \tag{2.7}$$

Furthermore, if there exists pseudo parameter vector $\underline{\alpha}_n$ such that

$$dY_n^* = 0$$

then we have the following:

$$Y_{\exp,n} - Y_{code,n}|_{\underline{\alpha}=\underline{\alpha}_n} = Y_{\exp,n} - Y_{code,n}|_{\underline{\alpha}=0} - \left(\frac{\partial Y_n}{\partial \underline{\alpha}}|_{\underline{\alpha}=0}\right)^T d\,\underline{\alpha} = Y_{\exp,n} - Y_{code,n}|_{\underline{\alpha}=0} - \left(\frac{\partial Y_n}{\partial \underline{\alpha}}|_{\underline{\alpha}=0}\right)^T \underline{\alpha}_n$$

$$0 = dY_n - \left(\frac{\partial Y_n}{\partial \underline{\alpha}}|_{\underline{\alpha}=0}\right)^T \underline{\alpha}_n \tag{2.8}$$

where $dY_n = Y_{\exp,n} - Y_{code,n}|_{\underline{\alpha}=0}$

Eq. (2.8) can be thought to have the form $Y = \underline{A}^T \underline{X}$ where both $\underline{A}$ and $\underline{X}$ are column vectors of the same order (equal to the number of code parameters I considered). Eq. (2.8) is a scalar equation with I unknowns; equivalently Y and $\underline{A}^T$ are completely known. An estimator to $\underline{X}$ can be defined whenever $\underline{A}$ and Y are given. The estimator of $\underline{X}$ that we seek should be as close as possible to the real value of $\underline{X}$. In Estimation Theory, the *a posteriori* mean of $\underline{X}$ minimizes the *Mean Square Error* and could be used as an appropriate estimator. Appendix B provides the theory and detail derivation of an appropriate estimator, also called *Minimum Mean Square Error (MMSE)* estimator, using *Gauss-Markov Theorem*. Then, the estimator of each $\underline{\alpha}_n$ is given by Eq. (2.9), where the covariance matrix from the previous iteration is used [50][51].

$$\hat{\underline{\alpha}}_n - \bar{\underline{\alpha}} = {}^{m-1}\underline{\underline{\Sigma}} \left(\frac{\partial Y_n}{\partial \underline{\alpha}} \right) \frac{\left(Y_{\exp,n} - Y_{code,n}|_{\underline{\alpha}=0} - \left(\frac{\partial Y_n}{\partial \underline{\alpha}}|_{\underline{\alpha}=0} \right)^T \underline{b}^m \right)}{\frac{\partial Y_n}{\partial \underline{\alpha}}^T {}^{m-1}\underline{\underline{\Sigma}} \frac{\partial Y_n}{\partial \underline{\alpha}}} \tag{2.9}$$

where $\hat{\underline{\alpha}}_n$ is the estimator of $\underline{\alpha}_n$. We can see from Eq. (2.9) that this vector is weighted based on the previous estimate of covariance matrix $\underline{\underline{\Sigma}}$, the superscript represents the index of iteration. The error covariance matrix of $\underline{\alpha}_n$ is then given as follows.

$$\underline{\underline{\Sigma}}_n = {}^{m-1}\underline{\underline{\Sigma}} - \frac{{}^{m-1}\underline{\underline{\Sigma}} \frac{\partial Y_n}{\partial \underline{\alpha}} \frac{\partial Y_n}{\partial \underline{\alpha}}^T {}^{m-1}\underline{\underline{\Sigma}}}{\frac{\partial Y_n}{\partial \underline{\alpha}}^T {}^{m-1}\underline{\underline{\Sigma}} \frac{\partial Y_n}{\partial \underline{\alpha}}} \tag{2.10}$$

2.3.1 Estimating the Covariance Matrix

Then to find ${}^m\underline{\underline{\Sigma}}$ we substitute the expected value of $(\underline{\alpha}_n - \bar{\underline{\alpha}})(\underline{\alpha}_n - \bar{\underline{\alpha}})^T$ in the following equation

$${}^m\underline{\underline{\Sigma}} = \left(\frac{1}{N} \right) \sum_{n=1}^{N} (\underline{\alpha}_n - \bar{\underline{\alpha}})(\underline{\alpha}_n - \bar{\underline{\alpha}})^T \tag{2.11}$$

where the over-bar represents averaged quantity.

Taking expected values of $(\underline{\alpha}_n - \bar{\underline{\alpha}})(\underline{\alpha}_n - \bar{\underline{\alpha}})^T$ we get Eq. (2.12).

$$E[(\underline{\alpha}_n - \bar{\underline{\alpha}})(\underline{\alpha}_n - \bar{\underline{\alpha}})^T] = E[\underline{\alpha}_n - \bar{\underline{\alpha}}] E[\underline{\alpha}_n - \bar{\underline{\alpha}}]^T + \underline{\underline{\Sigma}}_n \approx (\hat{\underline{\alpha}}_n - \underline{b})(\hat{\underline{\alpha}}_n - \underline{b})^T + \underline{\underline{\Sigma}}_n \tag{2.12}$$

Note, that we are substituting the estimators we calculated, not the exact expected values.

Substituting Eq. (2.12) into Eq. (2.11) we arrive at the updated estimate of ${}^{m}\underline{\underline{\Sigma}}$ as follows:

$$ {}^{m}\underline{\underline{\Sigma}} = \frac{1}{N}\sum_{n=1}^{N}\left((\hat{\underline{\alpha}}_n - \underline{b})(\hat{\underline{\alpha}}_n - \underline{b})^T + \underline{\underline{\Sigma}}_n \right) \tag{2.13} $$

Substituting Eq. (2.9) and Eq. (2.10) in Eq. (2.13) we obtain the following:

$$ {}^{m}\underline{\underline{\Sigma}} = \frac{1}{N}\sum_{n=1}^{N} \left(\left({}^{m-1}\underline{\underline{\Sigma}} \left(\frac{\partial Y_n}{\partial \underline{\alpha}} \right) \frac{\left(Y_{\exp,n} - Y_{code,n} - \frac{\partial Y_n}{\partial \underline{\alpha}}^T \underline{b}^m \right)}{\left(\frac{\partial Y_n}{\partial \underline{\alpha}}^T {}^{m-1}\underline{\underline{\Sigma}} \frac{\partial Y_n}{\partial \underline{\alpha}} \right)} \right) \left({}^{m-1}\underline{\underline{\Sigma}} \left(\frac{\partial Y_n}{\partial \underline{\alpha}} \right) \frac{\left(Y_{\exp,n} - Y_{code,n} - \frac{\partial Y_n}{\partial \underline{\alpha}}^T \underline{b}^m \right)}{\left(\frac{\partial Y_n}{\partial \underline{\alpha}}^T {}^{m-1}\underline{\underline{\Sigma}} \frac{\partial Y_n}{\partial \underline{\alpha}} \right)} \right)^T + \left({}^{m-1}\underline{\underline{\Sigma}} - \frac{{}^{m-1}\underline{\underline{\Sigma}} \frac{\partial Y_n}{\partial \underline{\alpha}} \frac{\partial Y_n}{\partial \underline{\alpha}}^T {}^{m-1}\underline{\underline{\Sigma}}}{\frac{\partial Y_n}{\partial \underline{\alpha}}^T {}^{m-1}\underline{\underline{\Sigma}} \frac{\partial Y_n}{\partial \underline{\alpha}}} \right) \right) $$

which after simplifying and rearranging terms becomes :

$$ {}^{m}\underline{\underline{\Sigma}} = {}^{m-1}\underline{\underline{\Sigma}} + \frac{1}{N}\sum_{n=1}^{N} \frac{{}^{m-1}\underline{\underline{\Sigma}} \frac{\partial Y_n}{\partial \underline{\alpha}} \frac{\partial Y_n}{\partial \underline{\alpha}}^T {}^{m-1}\underline{\underline{\Sigma}}}{\frac{\partial Y_n}{\partial \underline{\alpha}}^T {}^{m-1}\underline{\underline{\Sigma}} \frac{\partial Y_n}{\partial \underline{\alpha}}} \left(\frac{\left(Y_{\exp,n} - Y_{code,n} - \frac{\partial Y_n}{\partial \underline{\alpha}}^T b^m \right)^2}{\frac{\partial Y_n}{\partial \underline{\alpha}}^T {}^{m-1}\underline{\underline{\Sigma}} \frac{\partial Y_n}{\partial \underline{\alpha}}} - 1 \right) \tag{2.14} $$

2.3.2 Estimating the Mean Vector

Once we have an updated estimate of ${}^{m}\underline{\underline{\Sigma}}$, we can formulate the problem to express dY_n as *independent but non-identically* distributed scalar random normal variables with the following statistical parameters: mean $\frac{\partial Y_n}{\partial \underline{\alpha}}^T \underline{b}$ and variance $\frac{\partial Y_n}{\partial \underline{\alpha}}^T {}^{m}\underline{\underline{\Sigma}} \frac{\partial Y_n}{\partial \underline{\alpha}}$, where we need to estimate $\underline{b}$.

For a univariate normal distribution, the probability density function is given as follows.

$$f(x_k) = \frac{1}{\sqrt{2\pi}\sigma_k} e^{-\frac{1}{2}\left(\frac{x_k - \mu_k}{\sigma_k}\right)^2} \tag{2.15}$$

where x_k is the random vector variable, σ_k is the standard deviation and μ_k is the mean value. We substitute x_k as $Y_{exp,n} - Y_{code,n,}$ σ_k as $\frac{\partial Y_n}{\partial \underline{\alpha}}^T {}^m\underline{\underline{\Sigma}} \frac{\partial Y_n}{\partial \underline{\alpha}}$ and mean as $\frac{\partial Y_n}{\partial \underline{\alpha}}^T \underline{b}$.

So for each observation we have the PDF as follows.

$$f_n = \frac{1}{\sqrt{2\pi\left(\frac{\partial Y_n}{\partial \underline{\alpha}}^T {}^m\underline{\underline{\Sigma}} \frac{\partial Y_n}{\partial \underline{\alpha}}\right)}} . e^{-\left(\frac{\left(Y_{exp,n} - Y_{code,n} - \frac{\partial Y_n}{\partial \underline{\alpha}}^T \underline{b}\right)^2}{2.\frac{\partial Y_n}{\partial \underline{\alpha}}^T {}^m\underline{\underline{\Sigma}} \frac{\partial Y_n}{\partial \underline{\alpha}}}\right)}$$

The likelihood is then defined as:

$$L(\underline{b}) = \prod_{n=1}^{N} f_n$$

$$L(\underline{b}) = \prod_{n=1}^{N} \left(\frac{1}{\sqrt{2\pi\left(\frac{\partial Y_n}{\partial \underline{\alpha}}^T {}^m\underline{\underline{\Sigma}} \frac{\partial Y_n}{\partial \underline{\alpha}}\right)}} . e^{-\left(\frac{\left(Y_{exp,n} - Y_{code,n} - \frac{\partial Y_n}{\partial \underline{\alpha}}^T \underline{b}\right)^2}{2.\frac{\partial Y_n}{\partial \underline{\alpha}}^T {}^m\underline{\underline{\Sigma}} \frac{\partial Y_n}{\partial \underline{\alpha}}}\right)} \right) \tag{2.16}$$

It is convenient to deal with the log of likelihood since taking the logarithm of Eq. (2.16) cancels the exponential. By doing so, we obtain:

$$\log\left[L(\underline{b})\right]=\log\left[\prod_{n=1}^{N}\left(\frac{1}{\sqrt{2\pi\left(\frac{\partial Y_n}{\partial \underline{\alpha}}^T \underline{\underline{\Sigma}} \frac{\partial Y_n}{\partial \underline{\alpha}}\right)}}.e^{-\left(\frac{\left(Y_{\exp,n}-Y_{code,n}-\frac{\partial Y_n}{\partial \underline{\alpha}}^T \underline{b}\right)^2}{2.\frac{\partial Y_n}{\partial \underline{\alpha}}^T \underline{\underline{\Sigma}} \frac{\partial Y_n}{\partial \underline{\alpha}}}\right)}\right)\right]$$

$$\log\left[L(\underline{b})\right]=\log\left[\prod_{n=1}^{N}\left(\frac{1}{\sqrt{2\pi\left(\frac{\partial Y_n}{\partial \underline{\alpha}}^T \underline{\underline{\Sigma}} \frac{\partial Y_n}{\partial \underline{\alpha}}\right)}}\right)\right]+\log\left[\prod_{n=1}^{N} e^{-\left(\frac{\left(Y_{\exp,n}-Y_{code,n}-\frac{\partial Y_n}{\partial \underline{\alpha}}^T \underline{b}\right)^2}{2.\frac{\partial Y_n}{\partial \underline{\alpha}}^T \underline{\underline{\Sigma}} \frac{\partial Y_n}{\partial \underline{\alpha}}}\right)}\right]$$

$$\log\left[L(\underline{b})\right]=\log\left[\prod_{n=1}^{N}(1)\right]-\log\left[\prod_{n=1}^{N}\left(\sqrt{2\pi\left(\frac{\partial Y_n}{\partial \underline{\alpha}}^T \underline{\underline{\Sigma}} \frac{\partial Y_n}{\partial \underline{\alpha}}\right)}\right)\right]+\log\left[\prod_{n=1}^{N} e^{-\left(\frac{\left(Y_{\exp,n}-Y_{code,n}-\frac{\partial Y_n}{\partial \underline{\alpha}}^T \underline{b}\right)^2}{2.\frac{\partial Y_n}{\partial \underline{\alpha}}^T \underline{\underline{\Sigma}} \frac{\partial Y_n}{\partial \underline{\alpha}}}\right)}\right]$$

$$\log\left[L(\underline{b})\right]=\frac{N}{2}\log\left[2\pi\right]-\frac{1}{2}\sum_{n=1}^{N}\log\left[\frac{\partial Y_n}{\partial \underline{\alpha}}^T \underline{\underline{\Sigma}} \frac{\partial Y_n}{\partial \underline{\alpha}}\right]-\frac{1}{2}\sum_{n=1}^{N}\left(\frac{\left(Y_{\exp,n}-Y_{code,n}-\frac{\partial Y_n}{\partial \underline{\alpha}}^T \underline{b}\right)^2}{\frac{\partial Y_n}{\partial \underline{\alpha}}^T \underline{\underline{\Sigma}} \frac{\partial Y_n}{\partial \underline{\alpha}}}\right) \tag{2.17}$$

where $\log\left[a.b\right]=\log\left[a\right]+\log\left[b\right]$ was used repeatedly in the above equations.

Taking the derivatives with respect to the components of $\underline{b}$ and setting the likelihood equal to zero, we obtain the components of the mean vector. Setting $\frac{\partial \log\left[L(\underline{b})\right]}{\partial b_i}=0$ (refer to Section 2.1 for MLE estimation) for each *i = 1,2,...I*, we find the optimum values of the mean vector that maximizes the log of likelihood.

Ultimately, we arrive at an *I by I* system of linear equations with the coefficient matrix

dependent on the most recent values of the covariance matrix. After simplifying, we obtain the following:

$$\begin{pmatrix} \sum_{n=1}^{N} \frac{\left(\frac{\partial Y_n}{\partial \alpha_1}\right)^2}{\frac{\partial Y_n}{\partial \underline{\alpha}}^T \underline{\underline{\Sigma}}^m \frac{\partial Y_n}{\partial \underline{\alpha}}} & \cdots & \sum_{n=1}^{N} \frac{\frac{\partial Y_n}{\partial \alpha_1}\frac{\partial Y_n}{\partial \alpha_I}}{\frac{\partial Y_n}{\partial \underline{\alpha}}^T \underline{\underline{\Sigma}}^{(m)} \frac{\partial Y_n}{\partial \underline{\alpha}}} \\ \vdots & \ddots & \vdots \\ \sum_{n=1}^{N} \frac{\frac{\partial Y_n}{\partial \alpha_I}\frac{\partial Y_n}{\partial \alpha_1}}{\frac{\partial Y_n}{\partial \underline{\alpha}}^T \underline{\underline{\Sigma}}^{(m)} \frac{\partial Y_n}{\partial \underline{\alpha}}} & \cdots & \sum_{n=1}^{N} \frac{\left(\frac{\partial Y_n}{\partial \alpha_I}\right)^2}{\frac{\partial Y_n}{\partial \underline{\alpha}}^T \underline{\underline{\Sigma}}^m \frac{\partial Y_n}{\partial \underline{\alpha}}} \end{pmatrix} \begin{pmatrix} b_1 \\ \vdots \\ b_I \end{pmatrix} = \begin{pmatrix} \sum_{n=1}^{N} \frac{\frac{\partial Y_n}{\partial a_1}\left(Y_n^{\exp} - Y_n^{code}\right)}{\frac{\partial Y_n}{\partial \underline{\alpha}}^T \underline{\underline{\Sigma}}^{(m)} \frac{\partial Y_n}{\partial \underline{\alpha}}} \\ \vdots \\ \sum_{n=1}^{N} \frac{\frac{\partial Y_n}{\partial a_I}\left(Y_n^{\exp} - Y_n^{code}\right)}{\frac{\partial Y_n}{\partial \underline{\alpha}}^T \underline{\underline{\Sigma}}^{(m)} \frac{\partial Y_n}{\partial \underline{\alpha}}} \end{pmatrix} \tag{2.18}$$

This gives us the current estimation of the mean vector. This value is then substituted in Eq. (2.14) to get updated values on ${}^{m}\underline{\underline{\Sigma}}$. This process is repeated until convergence.

2.4 Maximum Likelihood Estimate – Quadratic Model

For a quadratic model we take additional (quadratic) terms in the Taylor series expansion of each dY_n. Then, with these assumptions

1. The random vectors $\underline{\alpha}_n$ follow multivariate normal distribution.
2. A quadratic relationship exists between the code response $Y_{n,code}$ and pseudo parameter vector $\underline{\alpha}_n$.

we obtain the following (compare with Eq. (2.7)):

$$\begin{aligned} Y_{\exp,n} = Y_{code,n}\big|_{\underline{\alpha}=\underline{\alpha}_n} = Y_{code,n}\big|_{\underline{\alpha}=0} + \sum_{i=1}^{I} \frac{\partial Y_n}{\partial \alpha_i}\left(\alpha_i - 0\right) + \\ \frac{1}{2!}\sum_{i=1}^{I}\sum_{j=1}^{I} \frac{\partial^2 Y_n}{\partial \alpha_i \partial \alpha_j}\left(\alpha_i - 0\right)\left(\alpha_j - 0\right) \end{aligned} \tag{2.19}$$

$$Y_{\exp,n} = Y_{code,n}|_{\underline{\alpha}=\underline{\alpha}_n} = Y_{code,n}|_{\underline{\alpha}=0} + \sum_{i=1}^{I} \frac{\partial Y_n}{\partial \alpha_i}\alpha_i + \frac{1}{2!}\sum_{i=1}^{I}\sum_{j=1}^{I} \frac{\partial^2 Y_n}{\partial \alpha_i \partial \alpha_j}\alpha_i \alpha_j \qquad (2.20)$$

where the Taylor series is taken around the nominal value of *0* as described earlier. In the case of two parameters, this can be compactly written as follows

$$Y_{\exp,n} = Y_{code,n}|_{\underline{\alpha}=0} + \left(\frac{\partial Y_n}{\partial \underline{\alpha}}\right)^T \underline{\alpha} + \frac{1}{2}\begin{pmatrix} \frac{\partial^2 Y_n}{\partial \alpha_1^2} \\ \frac{\partial^2 Y_n}{\partial \alpha_2^2} \end{pmatrix}^T \begin{pmatrix} \alpha_1^2 \\ \alpha_2^2 \end{pmatrix} + \alpha_1 \alpha_2 \frac{\partial}{\partial \alpha_1}\left(\frac{\partial Y_n}{\partial \alpha_2}\right) \qquad (2.21)$$

Furthermore, if there exists pseudo parameter vector $\underline{\alpha}_n$ such that

$$0 = dY_n^*$$

then it can be expressed as:

$$Y_{\exp,n} - Y_{code,n}|_{\underline{\alpha}=\underline{\alpha}_n} = Y_{\exp,n} - Y_{code,n}|_{\underline{\alpha}=0} - \left(\frac{\partial Y_n}{\partial \underline{\alpha}}\right)^T \underline{\alpha} - \frac{1}{2}\begin{pmatrix} \frac{\partial^2 Y_n}{\partial \alpha_1^2} \\ \frac{\partial^2 Y_n}{\partial \alpha_2^2} \end{pmatrix}^T \begin{pmatrix} \alpha_1^2 \\ \alpha_2^2 \end{pmatrix} - \alpha_1 \alpha_2 \frac{\partial}{\partial \alpha_1}\left(\frac{\partial Y_n}{\partial \alpha_2}\right)$$

$$dY_n - \left(\frac{\partial Y_n}{\partial \underline{\alpha}}|_{\underline{\alpha}=0}\right)^T \underline{\alpha}_n - \frac{1}{2}\begin{pmatrix} \frac{\partial^2 Y_n}{\partial \alpha_1^2} \\ \frac{\partial^2 Y_n}{\partial \alpha_2^2} \end{pmatrix}^T \begin{pmatrix} \alpha_1^2 \\ \alpha_2^2 \end{pmatrix} - \alpha_1 \alpha_2 \frac{\partial}{\partial \alpha_1}\left(\frac{\partial Y_n}{\partial \alpha_2}\right) = 0$$

where $dY_n = Y_{\exp,n} - Y_{code,n}|_{\underline{\alpha}=0}$

We can now formulate the problem to express dY_n as *independent but non-identically* distributed scalar random normal variables. The mean and variance for *each* random variable dY_n are expressed in terms of the mean and covariance matrix of the random vectors of pseudo-parameters. Since E[dY_n^*]=0 and $E[\underline{\alpha}_n] = \underline{b}$, we have the following (*specific to the case of two parameters*):

$$\mu_n = E[dY_n] = \frac{\partial Y_n}{\partial \underline{\alpha}}^T \underline{b} + \frac{1}{2}\frac{\partial^2 Y_n}{\partial \alpha_1^2}\left(\sigma_1^2 + b_1^2\right) + \frac{1}{2}\frac{\partial^2 Y_n}{\partial \alpha_2^2}\left(\sigma_2^2 + b_2^2\right) + \frac{\partial}{\partial \alpha_1}\left(\frac{\partial Y_n}{\partial \alpha_2}\right) E[\alpha_1] E[\alpha_2] \tag{2.22}$$

where σ_i^2 and b_i are the variance and the mean of the ith component of $\underline{\alpha}_n$, respectively. It is helpful to differentiate between σ_i in Eq. (2.22) and σ_n in Eq. (2.23). The former is the standard deviation of pseudo parameters under consideration, the latter is the standard deviation in each observation dY_n. Furthermore, using the definition of non-central moments, the variance of the scalar random variable dY_n can be written as:

$$\begin{aligned}\sigma_n^2 = var[dY_n] = \frac{\partial Y_n}{\partial \underline{\alpha}}^T \underline{\underline{\Sigma}} \frac{\partial Y_n}{\partial \underline{\alpha}} + \\ \left(\frac{1}{2}\frac{\partial^2 Y_n}{\partial \alpha_1^2}\right)^2 \left[E[\alpha_1]^4 + 6E[\alpha_1]^2 var(\alpha_1) + 3var(\alpha_1)^2 - \left(var(\alpha_1) + E[\alpha_1]^2\right)^2\right] + \\ \left(\frac{1}{2}\frac{\partial^2 Y_n}{\partial \alpha_2^2}\right)^2 \left[E[\alpha_2]^4 + 6E[\alpha_2]^2 var(\alpha_2) + 3var(\alpha_2)^2 - \left(var(\alpha_2) + E[\alpha_2]^2\right)^2\right] + \\ \left(\frac{\partial}{\partial \alpha_1}\frac{\partial Y_n}{\partial \alpha_2}\right)^2 \left[(var[\alpha_1])(var[\alpha_2]) + E[\alpha_1]^2 var[\alpha_2] + E[\alpha_2]^2 var[\alpha_1]\right]\end{aligned} \tag{2.23}$$

Since it is assumed that the random vectors $\underline{\alpha}_n$ follow multivariate normal distribution, dY_n is *approximated* as univariate normal distribution. Then,

$dY_n \sim N\left(E[dY_n], var[dY_n]\right)$. The likelihood function L is given as follows (compare with Eq. (2.16)).

$$L = \left[\prod_{n=1}^{N} \left(\frac{1}{\sqrt{2\pi\, var[dY_n]}} . e^{-\left(\frac{\left(Y_{exp,n} - Y_{code,n} - E[dY_n]\right)^2}{2.\, var[dY_n]} \right)} \right) \right]$$

Furthermore, taking the log of the likelihood above we get the following

$$\log(L) = \sum_{n=1}^{N} \log\left(\frac{1}{\sqrt{2\pi\,\sigma_n^2}} \right) - \sum_{n=1}^{N} \left(\frac{\left(Y_{exp,n} - Y_{code,n} - \mu_n\right)^2}{2.\,\sigma_n^2} \right)$$

$$\log(L) = \sum_{n=1}^{N} -\frac{1}{2}\log(2\pi\,\sigma_n^2) - \frac{1}{2}\sum_{n=1}^{N} \left(\frac{\left(Y_{exp,n} - Y_{code,n} - \mu_n\right)^2}{\sigma_n^2} \right)$$

$$\log(L) = \sum_{n=1}^{N} -\frac{1}{2}\left(\log(2\pi) + \log\left(\sigma_n^2\right)\right) - \frac{1}{2}\sum_{n=1}^{N} \left(\frac{\left(Y_{exp,n} - Y_{code,n} - \mu_n\right)^2}{\sigma_n^2} \right)$$

Taking $\frac{\partial \log(L)}{\partial b_i} = 0$ we solve for the mean vector $\underline{b}$. Specifically for the case of two variables we have the following for b_1 and b_2, respectively.

$$\frac{\partial \log(L)}{\partial b_1} =$$

$$\sum_{n=1}^{N}\left(-\frac{1}{2}\left(\frac{1}{\sigma_n^2}\right)\left[\left(\frac{1}{2}\frac{\partial^2 Y_n}{\partial \alpha_1^2}\right)^2\left(4b_1^3+12\sigma_1^2 b_1-\left(\sigma_1^2+b_1^2\right)2b_1\right)+\left(\frac{\partial^2 Y_n}{\partial \alpha_1 \partial \alpha_2}\right)^2 2\sigma_2^2 b_1\right]\right)-$$

$$\sum_{n=1}^{N}\frac{1}{2}\left(\frac{\left\{\begin{array}{l} 2\sigma_n^2\left(Y_{\exp,n}-Y_{code,n}-\mu_n\right)(-1)\left(\dfrac{\partial Y_n}{\partial \alpha_1}+\dfrac{1}{2}\dfrac{\partial^2 Y_n}{\partial \alpha_1^2}.2b_1+\left(\dfrac{\partial^2 Y_n}{\partial \alpha_1 \partial \alpha_2}\right).b_2\right)- \\ \left(Y_{\exp,n}-Y_{code,n}-\mu_n\right)^2\left[\begin{array}{l}\left(\dfrac{1}{2}\dfrac{\partial^2 Y_n}{\partial \alpha_1^2}\right)^2\left(4b_1^3+12\sigma_1^2 b_1-\left(\sigma_1^2+b_1^2\right)2b_1\right)+ \\ \left(\dfrac{\partial^2 Y_n}{\partial \alpha_1 \partial \alpha_2}\right)^2 2\sigma_2^2 b_1\end{array}\right]\end{array}\right\}}{\sigma_n^4}\right) \tag{2.24}$$

$$\frac{\partial \log(L)}{\partial b_2} =$$

$$\sum_{n=1}^{N}\left(-\frac{1}{2}\left(\frac{1}{\sigma_n^2}\right)\left[\left(\frac{1}{2}\frac{\partial^2 Y_n}{\partial \alpha_2^2}\right)^2\left(4b_2^3+12\sigma_2^2 b_2-\left(\sigma_2^2+b_2^2\right)2b_2\right)+\left(\frac{\partial^2 Y_n}{\partial \alpha_1 \partial \alpha_2}\right)^2 2\sigma_1^2 b_2\right]\right)-$$

$$\sum_{n=1}^{N}\frac{1}{2}\left(\frac{\left\{\begin{array}{l} 2\sigma_n^2\left(Y_{\exp,n}-Y_{code,n}-\mu_n\right)(-1)\left(\dfrac{\partial Y_n}{\partial \alpha_2}+\dfrac{1}{2}\dfrac{\partial^2 Y_n}{\partial \alpha_2^2}.2b_2+\left(\dfrac{\partial^2 Y_n}{\partial \alpha_1 \partial \alpha_2}\right).b_1\right)- \\ \left(Y_{\exp,n}-Y_{code,n}-\mu_n\right)^2\left[\begin{array}{l}\left(\dfrac{1}{2}\dfrac{\partial^2 Y_n}{\partial \alpha_2^2}\right)^2\left(4b_2^3+12\sigma_2^2 b_2-\left(\sigma_2^2+b_2^2\right)2b_2\right)+ \\ \left(\dfrac{\partial^2 Y_n}{\partial \alpha_1 \partial \alpha_2}\right)^2 2\sigma_1^2 b_2\end{array}\right]\end{array}\right\}}{\sigma_n^4}\right) \tag{2.25}$$

The set of equations arising from the maximization of the log-likelihood are a set of non-linear equations. In the example above, we have $n = 2$. This set of non-linear equations can be solved using initial guess of $b_i = 0$.

Furthermore, the mixed derivative is approximated by the following.

$$\frac{\partial^2 f(a,b)}{\partial x \partial y} \approx \frac{f(a+h,b+h)-f(a+h,b-h)-f(a-h,b+h)+f(a-h,b-h)}{4h^2}$$

Note that the covariance matrix $\underline{\underline{\Sigma}}$ of *i.i.d.* random vectors $\underline{\alpha}_n$ has not been explicitly prescribed yet. Like in Section 2.3 we make use of Expectation-Maximization (EM) algorithm and start with an initial guess of $\underline{\underline{\Sigma}}$ such that the only unknowns in the likelihood equation above are the individual entries of the mean vector i.e. b_i for $I = 1, 2...I$. Then, maximizing the log of the likelihood with respect to the entries of the mean of pseudo-parameter vector (i.e. b_i) and solving the resulting system of non-linear equations, we find the mean vector for the current iteration. Once the updated mean vector is found, the log-likelihood can again be maximized with respect to the diagonal entries (variances) of the covariance matrix $\underline{\underline{\Sigma}}$ (*the pseudo parameters are not correlated*). Then, the resulting system of non-linear equations can be solved to find the variance of $\underline{\alpha}_n$. A simpler way to update the covariance matrix is to take the linear terms in Eq. (2.21) and use the linear *minimum mean square error* estimator on each observation for $n = 1,2...N$. This truncated method yields Eq. (2.14) of Section 2.3. The iteration is performed until convergence is achieved which can be defined such that the maximum difference between two subsequent iterations of $\underline{b}$ is below some pre-set threshold value.

2.5 Maximum a Posteriori Estimate

As described in the preceding Sections, the *likelihood* of observing $x_1, x_2,... x_n$ of random variable $X_1, X_2, \ldots X_n$ is given by the following.

$$f(x_1, x_2, \ldots x_n|\underline{\theta}) = f(x_1|\underline{\theta}).\, f(x_2|\underline{\theta}) \ldots f(x_n|\underline{\theta}) = \prod_{i=1}^{n} f(x_i|\underline{\theta})$$

$$L(\underline{\theta}|\underline{x}) = \prod_{i=1}^{n} f(x_i|\underline{\theta})$$

where L is the likelihood function dependent on $\underline{\theta}$. If a prior distribution of the parameters $\underline{\theta}$ exist, then it could be assigned. If "g" represents a prior distribution of the parameters then, using Bayes' Theorem, the posterior distribution "f" of the parameters $\underline{\theta}$ is:

$$f(\underline{\theta}|\underline{x}) = \frac{L(\underline{\theta}|\underline{x}).\, g(\underline{\theta})}{\int_{\underline{\Theta}} L(\underline{\theta}'|\underline{x}).\, g(\underline{\theta}')\, d\underline{\theta}'} \tag{2.26}$$

The denominator in the above equation is called the *partition function* and does not depend on the parameters $\underline{\theta}$. Hence, it does not affect the maximization of the posterior distribution. The Maximum a Posteriori (MAP) estimate of the parameters is found by maximizing the posterior density as follows.

$$\hat{\underline{\theta}}_{MAP} = \underset{\underline{\theta}}{arg\,max} \left(f(\underline{\theta}|\underline{x}) \right) \tag{2.27}$$

Furthermore, maximizing any function is equivalent to maximizing its log since log is a monotonically increasing function. Therefore, maximizing the posterior distribution f with respect to the parameters $\underline{\theta}$ is equivalent to the following:

$$\hat{\underline{\theta}}_{MAP} = \underset{\underline{\theta}}{arg\,max} \left(L(\underline{\theta}|\underline{x}).\, g(\underline{\theta}) \right) = \underset{\underline{\theta}}{arg\,max} \left(\log(L) + \log(g) \right)$$

Similar to Section 2.3, we make the following assumptions.

1. The random vectors $\underline{\alpha}_n$ follow multivariate normal distribution.

2. A linear relationship exists between the code response $Y_{n,code}$ and pseudo parameter vector $\underline{\alpha}_n$.

Then we have the following.

$$Y_{\exp,n} - Y_{code,n}|_{\underline{\alpha}=0} = \left(\frac{\partial Y_n}{\partial \underline{\alpha}}|_{\underline{\alpha}=0}\right)^T d\,\underline{\alpha} = \left(\frac{\partial Y_n}{\partial \underline{\alpha}}|_{\underline{\alpha}=0}\right)^T \underline{\alpha}_n$$

and,

$$0 = dY_n^* = \; Y_{\exp,n} - Y_{code,n}|_{\underline{\alpha}=0} - \left(\frac{\partial Y_n}{\partial \underline{\alpha}}|_{\underline{\alpha}=0}\right)^T \underline{\alpha}_n$$

where $dY_n = Y_{\exp,n} - Y_{code,n}|_{\underline{\alpha}=0}$

We can now formulate the problem to express dY_n as *independent but non-identically* distributed scalar random normal variables. The mean and variance for *each* random variable dY_n is expressed in terms of the mean and covariance matrix of the random vectors of pseudo-parameters $\underline{\alpha}_n$. Since $E[dY_n^*]=0$ and $E[\underline{\alpha}_n]=\underline{b}$, we have the following:

$$\mathrm{E}[\,dY_n\,] = \; \left(\frac{\partial Y_n}{\partial \underline{\alpha}}|_{\underline{\alpha}=0}\right)^T \underline{b}$$

$$\text{and,} \quad var(dY_n) = \left(\left(\frac{\partial Y_n}{\partial \underline{\alpha}}|_{\underline{\alpha}=0}\right)^T \underline{\underline{\Sigma}} \left(\frac{\partial Y_n}{\partial \underline{\alpha}}|_{\underline{\alpha}=0}\right)\right)$$

Since it is assumed that the random vectors $\underline{\alpha}_n$ follow multivariate normal distribution, dY_n follows univariate normal distribution. Then $dY_n \sim N\left(\left(\frac{\partial Y_n}{\partial \underline{\alpha}}|_{\underline{\alpha}=0}\right)^T \underline{b}, \left(\frac{\partial Y_n}{\partial \underline{\alpha}}|_{\underline{\alpha}=0}\right)^T \underline{\underline{\Sigma}} \left(\frac{\partial Y_n}{\partial \underline{\alpha}}|_{\underline{\alpha}=0}\right)\right)$. The likelihood function L is the same as given by Eq. (2.16).

$$L=\left[\prod_{n=1}^{N}\left(\frac{1}{\sqrt{2\pi\left(\left(\frac{\partial Y_n}{\partial \underline{\alpha}}|_{\underline{\alpha}=0}\right)^T \underline{\underline{\Sigma}}\left(\frac{\partial Y_n}{\partial \underline{\alpha}}|_{\underline{\alpha}=0}\right)\right)}}.e^{-\left(\frac{\left(Y_{\exp,n}-Y_{code,n}-\left(\frac{\partial Y_n}{\partial \underline{\alpha}}|_{\underline{\alpha}=0}\right)^T \underline{b}\right)^2}{2.\left(\left(\frac{\partial Y_n}{\partial \underline{\alpha}}|_{\underline{\alpha}=0}\right)^T \underline{\underline{\Sigma}}\left(\frac{\partial Y_n}{\partial \underline{\alpha}}|_{\underline{\alpha}=0}\right)\right)}\right)}\right)\right]$$

The expected value and variance of the random variables dY_n are *not identical*. An arithmetic averaging is performed as shown in Eq. (2.28) and (2.29) to *approximate* the *overall* mean and variance of the random variables dY_n. The motivation behind approximating the *overall* mean (μ) and variance (σ^2) is to treat the random variables dY_n as *approximately identical* such that we could assign reasonable prior distributions on their mean and variance, and hence work towards determining their posterior mode.

$$Mean(\mu)\approx\left(\frac{\partial \bar{Y}}{\partial \underline{\alpha}}\right)^T \underline{b} \tag{2.28}$$

$$Variance(\sigma^2)\approx\left(\frac{\partial \bar{Y}}{\partial \underline{\alpha}}\right)^T \underline{\underline{\Sigma}}\frac{\partial \bar{Y}}{\partial \underline{\alpha}} \tag{2.29}$$

where $\frac{\partial \bar{Y}}{\partial \underline{\alpha}}=\frac{1}{N}\sum_{n=1}^{N}\left(\frac{\partial Y_n}{\partial \underline{\alpha}}|_{\underline{\alpha}=0}\right)$, N is the number of observations.

Furthermore, we want to determine the posterior mode of the mean μ and variance σ^2. For this, conjugate priors are assigned. Since we have the univariate normal dataset dY_n, a normal prior is assigned on the mean μ conditional on the variance, and an inverse gamma prior is assigned to the variance (σ^2) as shown in Eq. (2.30) and Eq. (2.31) respectively. Let these distributions be denoted by g_μ and g_{σ^2} respectively. It should be kept in mind that our ultimate aim is to determine the mean $\underline{b}$ and covariance matrix $\underline{\underline{\Sigma}}$ of the random vectors $\underline{\alpha}_n$. This formulation is merely a way that leads to the quantification of posterior mode of μ and σ^2 .

$$\mu \sim N\left(\mu_p, \frac{\sigma^2}{\kappa_p}\right)$$

$$g_{\mu}|\sigma^2 \propto \left(\sigma^2\right)^{(-1/2)} \exp\left[\frac{-\kappa_p}{\left(2\sigma^2\right)}\left(\mu-\mu_p\right)^2\right] \tag{2.30}$$

$$\text{and} \quad \sigma^2 \sim invGamma\left(\frac{\nu_p}{2}, \frac{\xi_p^2}{2}\right)$$

$$g_{\sigma^2} \propto \left(\sigma^2\right)^{-\frac{\nu_p+2}{2}} \exp\left[-\frac{\xi_p{}^2}{2\sigma^2}\right] \tag{2.31}$$

The four hyper-parameters μ_p, κ_p, ν_p and $\xi_p{}^2$ shown below are the mean of the observed dataset, shrinkage, degrees of freedom and observed variance of the dataset, respectively. Furthermore the following numerical values are used for these hyper-parameters (for univariate densities) [52]:

μ_p :*mean* (data)

κ_p : 0.1

ν_p : 3

$\xi_p{}^2$: *var*(data)

A small value for κ_p is used which reflects lack of precision on the prior mean. ν_p is chosen as the smallest positive integer such that $\nu_p > I + 1$, where $I = 1$ for univariate case [53]. In general, initial hyper-parameters κ_p and ν_p could be freely chosen but in practice, they are chosen such that the model yields a finite posterior variance. Furthermore, dY_n is observed directly hence the mean and variance of the dataset, μ_p and σ^2 respectively, can be computed directly.

Hence the joint prior distribution $g(\mu, \sigma^2)$ for the parameters is given by the following:

$$g(\mu,\sigma^2)=g_{\mu}.g_{\sigma^2}=C\left((\sigma^2)^{(-1/2)}\exp[\frac{-\kappa_p}{(2\sigma^2)}(\mu-\mu_p)^2].(\sigma^2)^{-\frac{\nu_p+2}{2}}\exp[-\frac{\xi_p^{\,2}}{2\sigma^2}]\right) \qquad (2.32)$$

where *C* is a constant quantity that does not depend on μ and σ^2 . From Eq. (2.16), Eq. (2.27) and Eq. (2.32), we write the complete model for the posterior distribution of parameters as follows.

$$f(\mu,\sigma^2|dY_n)=\frac{L(\mu,\sigma^2|dY_n).g(\mu,\sigma^2)}{\int\int\left(L(\mu',(\sigma^2)'|dY_n)g(\mu',(\sigma^2)')\right)d\mu'\,d(\sigma^2)'} \qquad (2.33)$$

We can see Eq. (2.33) above is a function of μ and σ^2 which in turn depends on $\underline{b}$ and $\underline{\underline{\Sigma}}$ as shown in Eq. (2.28) and Eq. (2.29). To obtain the posterior mode we maximize Eq. (2.33) with respect to the parameters μ and σ^2 . Since μ is a linear combination of the elements of $\underline{b}$, in practice maximize Eq. (2.33) with respect to each element of $\underline{b}$.

2.5.1 MAP Estimate of Mean Vector

E-Step: We start with a guess for the covariance matrix $\underline{\underline{\Sigma}}$ first. E-Step *proper* is performed in Section 2.5.3. Note the index of $\underline{\underline{\Sigma}}$ has been changed to *m-1* in the following, where *m-1* represents index for the previous iteration. Ignoring constants, we have the following:

$$\{\log(L)+\log(g)\}=-\log\left[\prod^{N}\left(\sqrt{2\pi\left(\left(\frac{\partial Y_n}{\partial \underline{\alpha}}|_{\underline{\alpha}=0}\right)^{T}{}^{m-1}\underline{\underline{\Sigma}}\left(\frac{\partial Y_n}{\partial \underline{\alpha}}|_{\underline{\alpha}=0}\right)\right)}\right)\right]+$$
$$\log\left[\prod^{N} e^{-\left(\frac{\left(Y_{\exp,n}-Y_{code,n}-\left(\frac{\partial Y_n}{\partial \underline{\alpha}}|_{\underline{\alpha}=0}\right)^{T}\underline{b}\right)^{2}}{2.\left(\left(\frac{\partial Y_n}{\partial \underline{\alpha}}|_{\underline{\alpha}=0}\right)^{T}{}^{m-1}\underline{\underline{\Sigma}}\left(\frac{\partial Y_n}{\partial \underline{\alpha}}|_{\underline{\alpha}=0}\right)\right)}\right)}\right]+ \tag{2.34}$$
$$\frac{\nu_p+3}{2}\log\left(\frac{1}{\sigma^2}\right)-\frac{\mu^2\kappa_p}{2\sigma^2}+\frac{\mu\kappa_p\mu_p}{\sigma^2}-\frac{1}{2\sigma^2}\left(\xi_p^2+\kappa_p\mu_p^2\right)$$

After substitutions, expansion of the log terms and algebraic simplifications we get Eq. (2.35).

$$\{\log(L)+\log(g)\}=-\frac{N}{2}\log[2\pi]-\frac{1}{2}\sum_{n=1}^{N}\log\left(\left(\frac{\partial Y_n}{\partial \underline{\alpha}}|_{\underline{\alpha}=0}\right)^{T}{}^{m-1}\underline{\underline{\Sigma}}\left(\frac{\partial Y_n}{\partial \underline{\alpha}}|_{\underline{\alpha}=0}\right)\right)-$$
$$\frac{1}{2}\sum_{n=1}^{N}\left(\frac{\left(Y_{\exp,n}-Y_{code,n}-\left(\frac{\partial Y_n}{\partial \underline{\alpha}}|_{\underline{\alpha}=0}\right)^{T}\underline{b}\right)^{2}}{\left(\left(\frac{\partial Y_n}{\partial \underline{\alpha}}|_{\underline{\alpha}=0}\right)^{T}{}^{m-1}\underline{\underline{\Sigma}}\left(\frac{\partial Y_n}{\partial \underline{\alpha}}|_{\underline{\alpha}=0}\right)\right)}\right)+\frac{\nu_p+3}{2}\log\left(\frac{1}{\frac{\partial \bar{Y}}{\partial \underline{\alpha}}^{T}{}^{m-1}\underline{\underline{\Sigma}}\frac{\partial \bar{Y}}{\partial \underline{\alpha}}}\right)- \tag{2.35}$$
$$\frac{\left(\frac{\partial \bar{Y}}{\partial \underline{\alpha}}.\underline{b}^{m}\right)^{2}\kappa_p}{2\frac{\partial \bar{Y}}{\partial \underline{\alpha}}^{T}{}^{m-1}\underline{\underline{\Sigma}}\frac{\partial \bar{Y}}{\partial \underline{\alpha}}}+\frac{\left(\frac{\partial \bar{Y}}{\partial \underline{\alpha}}.\underline{b}^{m}\right)}{\frac{\partial \bar{Y}}{\partial \underline{\alpha}}^{T}{}^{m-1}\underline{\underline{\Sigma}}\frac{\partial \bar{Y}}{\partial \underline{\alpha}}}\kappa_p\mu_p-\frac{1}{2\frac{\partial \bar{Y}}{\partial \underline{\alpha}}^{T}{}^{m-1}\underline{\underline{\Sigma}}\frac{\partial \bar{Y}}{\partial \underline{\alpha}}}\left(\xi_p^2+\kappa_p\mu_p^2\right)$$

M-Step: Maximizing Eq. (2.35) with respect to b_i for $i= 1, 2, \ldots I$, we obtain the following:

$$\underset{\underline{\theta}}{arg\,max}(f)=\underset{\underline{\theta}}{arg\,max}\{L(\underline{\theta}|\underline{x}).g(\underline{\theta})\}=\underset{\underline{\theta}}{arg\,max}\{\log(L)+\log(g)\}$$

$$\frac{\partial}{\partial b_i}\{\log(L)+\log(g)\}=0$$

Hence, for a given *i*,

$$0=\sum_{n=1}^{N}\frac{\left(Y_{\exp,n}-Y_{code,n}-\left(\frac{\partial Y_n}{\partial \underline{\alpha}}|_{\underline{\alpha}=0}\right)^T \underline{b}\right)\frac{\partial Y_n}{\partial \alpha_i}}{\left(\frac{\partial Y_n}{\partial \underline{\alpha}}|_{\underline{\alpha}=0}\right)^T c^{m-1}\underline{\underline{\Sigma}}\left(\frac{\partial Y_n}{\partial \underline{\alpha}}|_{\underline{\alpha}=0}\right)}-\frac{\left(\frac{\partial \bar{Y}}{\partial \underline{\alpha}}.\underline{b}^m\right)\kappa_p\left(\frac{\partial \bar{Y}}{\partial \alpha_i}\right)}{\left(\frac{\partial \bar{Y}_n}{\partial \underline{\alpha}}|_{\underline{\alpha}=0}\right)^T {}^{m-1}\underline{\underline{\Sigma}}\left(\frac{\partial \bar{Y}_n}{\partial \underline{\alpha}}|_{\underline{\alpha}=0}\right)}+\frac{\left(\frac{\partial \bar{Y}}{\partial \alpha_i}\right)}{\frac{\partial \bar{Y}}{\partial \underline{\alpha}}^T {}^{m-1}\underline{\underline{\Sigma}}\frac{\partial \bar{Y}}{\partial \underline{\alpha}}}\kappa_p\mu_p \qquad (2.36)$$

Expanding the vector notation in the above equation and dropping the superscript on the components of $\underline{b}$, we obtain the following:

$$\frac{\partial \bar{Y}}{\partial \underline{\alpha}}.\underline{b}=\left(\frac{\partial \bar{Y}}{\partial \underline{\alpha}}\right)^T\underline{b}=\frac{\partial \bar{Y}}{\partial \alpha_1}.b_1+\frac{\partial \bar{Y}}{\partial \alpha_2}.b_2+...+\frac{\partial \bar{Y}}{\partial \alpha_I}.b_I$$

where b_i is the ith element of the mean vector $\underline{b}$. Also recall, for each *n:*

$$\frac{\partial Y_n}{\partial \underline{\alpha}}.b=\left(\frac{\partial Y_n}{\partial \alpha}\right)^T b=\frac{\partial Y_n}{\partial \alpha_1}.b_1+\frac{\partial Y_n}{\partial \alpha_2}.b_2+...+\frac{\partial Y_n}{\partial \alpha_I}.b_I$$

With these substitutions, Eq. (2.36) becomes:

$$\sum_{n=1}^{N}\left(\frac{1}{\left(\frac{\partial Y_n}{\partial \underline{\alpha}}|_{\underline{\alpha}=0}\right)^T {}^{m-1}\underline{\underline{\Sigma}}\left(\frac{\partial Y_n}{\partial \underline{\alpha}}|_{\underline{\alpha}=0}\right)}\left(-\left(\frac{\partial Y_n}{\partial \alpha_1}.b_1+\frac{\partial Y_n}{\partial \alpha_2}.b_2+...+\frac{\partial Y_n}{\partial \alpha_I}.b_I\right)\right)\frac{\partial Y_n}{\partial \alpha_i}\right)-$$

$$\frac{\left(\frac{\partial \bar{Y}}{\partial \alpha_1}.b_1+\frac{\partial \bar{Y}}{\partial \alpha_2}.b_2+...+\frac{\partial \bar{Y}}{\partial \alpha_I}.b_I\right)}{\left(\frac{\partial \bar{Y}}{\partial \underline{\alpha}}^T {}^{m-1}\underline{\underline{\Sigma}}\frac{\partial \bar{Y}}{\partial \underline{\alpha}}\right)}.\left(\kappa_p\frac{\partial \bar{Y}}{\partial \alpha_i}\right)=$$

$$-\left(\frac{\left(\frac{\partial \bar{Y}}{\partial \alpha_i}\right)}{\frac{\partial \bar{Y}}{\partial \underline{\alpha}}^T {}^{m-1}\underline{\underline{\Sigma}}\frac{\partial \bar{Y}}{\partial \underline{\alpha}}}\kappa_p\mu_p\right)-\sum_{n=1}^{N}\frac{1}{\left(\frac{\partial Y_n}{\partial \underline{\alpha}}|_{\underline{\alpha}=0}\right)^T {}^{m-1}\underline{\underline{\Sigma}}\left(\frac{\partial Y_n}{\partial \underline{\alpha}}|_{\underline{\alpha}=0}\right)}\left(Y_{\exp,n}-Y_{code,n}\right)\frac{\partial Y_n}{\partial \alpha_i}$$

On re-arranging,

$$\left(\left(-\sum_{n=1}^{N}\frac{1}{\left(\frac{\partial Y_n}{\partial \underline{\alpha}}\Big|_{\alpha=0}\right)^T {}^{m-1}\underline{\underline{\Sigma}}\left(\frac{\partial Y_n}{\partial \underline{\alpha}}\Big|_{\alpha=0}\right)}\cdot\frac{\partial Y_n}{\partial \alpha_1}\cdot\frac{\partial Y_n}{\partial \alpha_i}\right)-\frac{\frac{\partial \bar{Y}}{\partial \alpha_1}\cdot\left(\kappa_p\cdot\frac{\partial \bar{Y}}{\partial \alpha_i}\right)}{\left(\frac{\partial \bar{Y}}{\partial \underline{\alpha}}^T {}^{m-1}\underline{\underline{\Sigma}}\frac{\partial \bar{Y}}{\partial \underline{\alpha}}\right)}\right)b_1+$$

$$\ldots+\left(\left(-\sum_{n=1}^{N}\frac{1}{\left(\frac{\partial Y_n}{\partial \underline{\alpha}}\Big|_{\alpha=0}\right)^T {}^{m-1}\underline{\underline{\Sigma}}\left(\frac{\partial Y_n}{\partial \underline{\alpha}}\Big|_{\alpha=0}\right)}\frac{\partial Y_n}{\partial \alpha_I}\cdot\frac{\partial Y_n}{\partial \alpha_i}\right)-\frac{\frac{\partial \bar{Y}}{\partial \alpha_I}\cdot\left(\kappa_p\cdot\frac{\partial \bar{Y}}{\partial \alpha_i}\right)}{\left(\frac{\partial \bar{Y}}{\partial \underline{\alpha}}^T {}^{m-1}\underline{\underline{\Sigma}}\frac{\partial \bar{Y}}{\partial \underline{\alpha}}\right)}\right)b_I=$$

$$-\left(\frac{\left(\frac{\partial \bar{Y}}{\partial \alpha_i}\right)}{\frac{\partial \bar{Y}}{\partial \underline{\alpha}}^T {}^{m-1}\underline{\underline{\Sigma}}\frac{\partial \bar{Y}}{\partial \underline{\alpha}}}\kappa_p\mu_p\right)-\sum_{n=1}^{N}\frac{1}{\left(\frac{\partial Y_n}{\partial \underline{\alpha}}\Big|_{\underline{\alpha}=0}\right)^T {}^{m-1}\underline{\underline{\Sigma}}\left(\frac{\partial Y_n}{\partial \underline{\alpha}}\Big|_{\underline{\alpha}=0}\right)}\left(Y_{\exp,n}-Y_{code,n}\right)\frac{\partial Y_n}{\partial \alpha_i}$$

Hence, we get the following system of linear equations. The linear system Eq. (2.37) is solved with the available estimate of the covariance matrix ${}^{m-1}\underline{\underline{\Sigma}}$ to generate a better approximation of $\underline{b}$.

$$\begin{bmatrix} \left(\sum_{n=1}^{N} \frac{\left(\frac{\partial Y_n}{\partial \alpha_1}\right)^2}{\frac{\partial Y_n}{\partial \underline{\alpha}}^T \underline{\underline{\Sigma}}^{m-1} \frac{\partial Y_n}{\partial \underline{\alpha}}}\right) + \frac{\left(\frac{\partial \bar{Y}}{\partial \alpha_1}\right)^2}{\left(\frac{\partial \bar{Y}}{\partial \underline{\alpha}}^T {}^{m-1} \underline{\underline{\Sigma}} \frac{\partial \bar{Y}}{\partial \underline{\alpha}}\right)} \kappa_p & \cdots & \left(\sum_{n=1}^{N} \frac{\frac{\partial Y_n}{\partial \alpha_1}\frac{\partial Y_n}{\partial \alpha_I}}{\frac{\partial Y_n}{\partial \underline{\alpha}}^T \underline{\underline{\Sigma}}^{m-1} \frac{\partial Y_n}{\partial \underline{\alpha}}}\right) + \frac{\left(\frac{\partial \bar{Y}}{\partial \alpha_I} \cdot \frac{\partial \bar{Y}}{\partial \alpha_1}\right)}{\left(\frac{\partial \bar{Y}}{\partial \underline{\alpha}}^T {}^{m-1} \underline{\underline{\Sigma}} \frac{\partial \bar{Y}}{\partial \underline{\alpha}}\right)} \kappa_p \\ \vdots & \ddots & \vdots \\ \left(\sum_{n=1}^{N} \frac{\frac{\partial Y_n}{\partial \alpha_I}\frac{\partial Y_n}{\partial \alpha_1}}{\frac{\partial Y_n}{\partial \underline{\alpha}}^T \underline{\underline{\Sigma}}^{m-1} \frac{\partial Y_n}{\partial \underline{\alpha}}}\right) + \frac{\left(\frac{\partial \bar{Y}}{\partial \alpha_I} \cdot \frac{\partial \bar{Y}}{\partial \alpha_1}\right)}{\left(\frac{\partial \bar{Y}}{\partial \underline{\alpha}}^T {}^{m-1} \underline{\underline{\Sigma}} \frac{\partial \bar{Y}}{\partial \underline{\alpha}}\right)} \kappa_p & \cdots & \left(\sum_{n=1}^{N} \frac{\left(\frac{\partial Y_n}{\partial \alpha_I}\right)^2}{\frac{\partial Y_n}{\partial \underline{\alpha}}^T \underline{\underline{\Sigma}}^{m-1} \frac{\partial Y_n}{\partial \underline{\alpha}}}\right) + \frac{\left(\frac{\partial \bar{Y}}{\partial \alpha_I}\right)^2}{\left(\frac{\partial \bar{Y}}{\partial \underline{\alpha}}^T {}^{m-1} \underline{\underline{\Sigma}} \frac{\partial \bar{Y}}{\partial \underline{\alpha}}\right)} \kappa_p \end{bmatrix}.$$

$$\begin{pmatrix} b_1 \\ \vdots \\ b_I \end{pmatrix} = \begin{bmatrix} \left(\sum_{n=1}^{N} \frac{\frac{\partial Y_n}{\partial \alpha_1}\left(Y_n^{\exp} - Y_n^{code}\right)}{\frac{\partial Y_n}{\partial \underline{\alpha}}^T \underline{\underline{\Sigma}}^{m-1} \frac{\partial Y_n}{\partial \underline{\alpha}}}\right) + \frac{\left(\frac{\partial \bar{Y}}{\partial \alpha_1}\right)}{\left(\frac{\partial \bar{Y}}{\partial \underline{\alpha}}^T {}^{m-1} \underline{\underline{\Sigma}} \frac{\partial \bar{Y}}{\partial \underline{\alpha}}\right)} \kappa_p \mu_p \\ \vdots \\ \left(\sum_{n=1}^{N} \frac{\frac{\partial Y_n}{\partial \alpha_I}\left(Y_n^{\exp} - Y_n^{code}\right)}{\frac{\partial Y_n}{\partial \underline{\alpha}}^T \underline{\underline{\Sigma}}^{m-1} \frac{\partial Y_n}{\partial \underline{\alpha}}}\right) + \frac{\left(\frac{\partial \bar{Y}}{\partial \alpha_I}\right)}{\left(\frac{\partial \bar{Y}}{\partial \underline{\alpha}}^T {}^{m-1} \underline{\underline{\Sigma}} \frac{\partial \bar{Y}}{\partial \underline{\alpha}}\right)} \kappa_p \mu_p \end{bmatrix} \quad (2.37)$$

2.5.2 Direct Estimate of Covariance Matrix

M-Step: We seek to maximize Eq. (2.35) with respect to the covariance matrix (after substituting $\underline{b}$ from Eq. (2.37) into Eq. (2.35)) to generate a better approximation for the covariance matrix.

$$\frac{\partial}{\partial \Sigma_{i,i}}(\log(L) + \log(g)) = 0 \quad \text{, for } i = 1, 2, \ldots I.$$

For the general case with I pseudo-parameters, the off-diagonal elements of the covariance matrix are nominally equal to 0, i.e. the components of random vector of pseudo-parameters are assumed to be statistically uncorrelated..

Recall, we have the following approximation.

$$Var(dY_n): \sigma^2 = \frac{\partial \bar{Y}}{\partial \underline{\alpha}}^T \underline{\underline{\Sigma}} \frac{\partial \bar{Y}}{\partial \underline{\alpha}}$$

Therefore,

$$\sigma^2 = \begin{pmatrix} a_1 & a_2 & \cdots & a_I \end{pmatrix} \begin{pmatrix} \Sigma_{11} & 0 & \cdots & 0 \\ 0 & \Sigma_{11} & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \Sigma_{II} \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_I \end{pmatrix} = a_1{}^2 \Sigma_{11} + a_2{}^2 \Sigma_{22} + ... + a_I{}^2 \Sigma_{II}$$

where $a_i = \frac{\partial \bar{Y}}{\partial \alpha_i}, \frac{\partial \bar{Y}}{\partial \underline{\alpha}} = \frac{1}{N} \sum_{n=1}^{N} \frac{\partial Y_n}{\partial \underline{\alpha}}|_{\alpha=0}$, N is the number of observations and $\frac{\partial \bar{Y}}{\partial \alpha_i}$ is the ith entry of the column vector $\frac{\partial \bar{Y}}{\partial \underline{\alpha}}$.

Furthermore, the exact covariance $\sigma_n{}^2$ for each dY_n is given as $\sigma_n^2 = \frac{\partial Y_n}{\partial \underline{\alpha}}^T \underline{\underline{\Sigma}} \frac{\partial Y_n}{\partial \underline{\alpha}}$ by definition from Eq. (2.8). Then,

$$\sigma_n{}^2 = \begin{pmatrix} \frac{\partial Y_n}{\partial \alpha_1} & \frac{\partial Y_n}{\partial \alpha_2} & \cdots & \frac{\partial Y_n}{\partial \alpha_I} \end{pmatrix} \begin{pmatrix} \Sigma_{11} & 0 & \cdots & 0 \\ 0 & \Sigma_{11} & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \Sigma_{II} \end{pmatrix} \begin{pmatrix} \frac{\partial Y_n}{\partial \alpha_1} \\ \frac{\partial Y_n}{\partial \alpha_2} \\ \vdots \\ \frac{\partial Y_n}{\partial \alpha_I} \end{pmatrix} = \frac{\partial Y_n}{\partial \alpha_1}{}^2 \Sigma_{11} + ... + \frac{\partial Y_n}{\partial \alpha_I}{}^2 \Sigma_{II}$$

Let $z_i = \frac{\partial}{\partial \Sigma_{ii}} \log(L) + \frac{\partial}{\partial \Sigma_{ii}} \log(g) = 0$ for $i = 1, 2, \ldots I$, which can be written as:

$$\begin{aligned} z_i = 0 = & -\frac{1}{2} \sum_{n=1}^{N} \frac{1}{\sigma_n{}^2} \left(\frac{\partial Y_n}{\partial \alpha_i} \right)^2 - \frac{1}{2} \sum_{n=1}^{N} \left(\left(Y_{\exp,n} - Y_{code,n} - [\frac{\partial Y_n}{\partial \underline{\alpha}}^T \underline{b}] \right)^2 . (-\frac{1}{\sigma_n{}^4}) \left(\frac{\partial Y_n}{\partial \alpha_i} \right)^2 \right) \\ & (\frac{\nu_p + 3}{2}) (-\frac{1}{\sigma^2}) a_i{}^2 + \left(-\frac{\kappa_p \mu^2}{2} + \mu \kappa_p \mu_p - \frac{1}{2} (\xi_p^2 + \kappa_p \mu_p^2) \right) . (-\sigma^{-4}) a_i{}^2 \end{aligned} \quad (2.38)$$

z_1, z_2, … z_I forms a system of non-linear equations in σ^2 and σ_n^2 (the former is

approximate variance of dY_n whereas the latter is the exact variance), and consequently in $\Sigma_{11}, \Sigma_{22}, \cdots \Sigma_{II}$. A Newton-Rhapson non-linear solver can be used to solve for the diagonal elements of the covariance matrix $\underline{\underline{\Sigma}}$.

If $\underline{\underline{J}} = \begin{pmatrix} \frac{\partial z_1}{\partial \Sigma_{11}} & \cdots & \frac{\partial z_1}{\partial \Sigma_{II}} \\ \vdots & \ddots & \vdots \\ \frac{\partial z_I}{\partial \Sigma_{11}} & \cdots & \frac{\partial z_I}{\partial \Sigma_{II}} \end{pmatrix}$ is the Jacobian matrix, then, the following analytical Jacobian can be used to expedite the solution iteration.

$$J_{m,n} = -\frac{1}{2}\sum_{n=1}^{N}\left(-\frac{1}{\sigma_n^{\,4}}\right)\left(\frac{\partial Y_n}{\partial \alpha_m}\right)^2\left(\frac{\partial Y_n}{\partial \alpha_n}\right)^2 - \frac{1}{2}\sum_{n=1}^{N}\left(Y_{\exp,n} - Y_{code,n} - \left[\frac{\partial Y_n}{\partial \underline{\alpha}}^T \underline{b}\right]\right)^2 .\left(\frac{\partial Y_n}{\partial \alpha_m}\right)$$
$$\left(\frac{2}{\sigma_n^{\,6}}\right)\left(\frac{\partial Y_n}{\partial \alpha_n}\right)^2 + \left(\left(\frac{\nu_p + 3}{2}\right) a_m^{\,2}\right)\left(\frac{a_n^{\,2}}{\sigma^4}\right) + \left(-\frac{\kappa_p \mu^2}{2} + \mu\,\kappa_p \mu_p - \frac{1}{2}\left(\xi_p^2 + \kappa_p \mu_p^2\right)\right)\left(a_m^{\,2}\right)\left(2\frac{a_n^{\,2}}{\sigma^6}\right) \quad (2.39)$$

2.5.3 Estimation of Covariance Matrix using Conjugate Prior

An alternate way to estimate the covariance matrix using the mean vector computed in Section 2.5.1 and an estimate of the covariance matrix (from previous iteration) is presented in this Section. The method presented here makes use of the expression for the posterior mean of covariance matrix for multivariate data. A discussion on expressing the joint posterior distribution for univariate normal data was presented in Section 2.5.1. An analogous methodology is followed here for multivariate dataset $\underline{\alpha}_n$. Recall that statistical independence and identical distribution (*i.i.d.*) of random vectors $\underline{\alpha}_n$ is assumed. Furthermore the random vector $\underline{\alpha}_n$ is assumed to follow normal distribution i.e. $\underline{\alpha}_n \sim N(\underline{b}, \underline{\underline{\Sigma}})$. We proceed by treating the parameters $\underline{b}, \underline{\underline{\Sigma}}$ as random variables. To get the posterior mode, a prior multivariate normal distribution is assigned on $\underline{b}$ conditional on the variance $\underline{\underline{\Sigma}}$. An inverted Wishart prior is assigned for $\underline{\underline{\Sigma}}$. This forms a conjugate prior pair on the parameters $\underline{b}$ and $\underline{\underline{\Sigma}}$.

$$\underline{b}|\underline{\underline{\Sigma}} \sim N(\underline{\mu}_p, \underline{\underline{\Sigma}}/\kappa_p) \sim |\underline{\underline{\Sigma}}|^{-\frac{1}{2}} \exp\left(-\frac{\kappa_p}{2} trace\left((\underline{b}-\underline{\mu}_p)^T \underline{\underline{\Sigma}}^{-1}(\underline{b}-\underline{\mu}_p)\right)\right)$$

and

$$\underline{\underline{\Sigma}} \sim invWishart(\nu_p, \underline{\underline{\Lambda}}_p) \sim |\underline{\underline{\Sigma}}|^{-\frac{\nu_p+I+1}{2}} \exp\left(-\frac{1}{2} trace\left(\underline{\underline{\Sigma}}^{-1}\underline{\underline{\Lambda}}_p^{-1}\right)\right)$$

where

$\underline{\underline{\Lambda}}_p$: $var(data)$,

$\underline{\mu}_p$: $mean(data)$,

$\kappa_p = 0.1$,

$\nu_p > I+1$, I is the is the size of the random vector $\underline{\alpha}_n$.

ν_p is the smallest positive integer such that $\nu_p > I + 1$ which gives a finite covariance matrix [53]. Note the numerical value of ν_p here is different than that in Section 2.5.

By definition for multivariate data under the assumption of a normal prior on mean $\underline{b}$ conditional on the covariance $\underline{\underline{\Sigma}}$, and an inverse Wishart prior on the covariance $\underline{\underline{\Sigma}}$, the posterior mean of the covariance matrix is given by Eq. (2.40). For complete derivation see appendix in [54]. The diagonal terms of the following covariance matrix represent the variance of individual components of the mean vector.

$$\underline{\underline{\Sigma}} = \frac{\underline{\underline{\Lambda}}_p^{-1} + (\frac{\kappa_p N}{\kappa_p + N})(\bar{\underline{\alpha}} - \underline{\mu}_p)(\bar{\underline{\alpha}} - \underline{\mu}_p)^T + \sum_{n=1}^{N}(\underline{\alpha}_n - \bar{\underline{\alpha}})(\underline{\alpha}_n - \bar{\underline{\alpha}})^T}{(\nu_p + N + I + 2)} \qquad (2.40)$$

where $\bar{\underline{\alpha}}$ is the mean from the most recent iteration,

The process of forming a posterior distribution for the parameters $\underline{b}, \underline{\underline{\Sigma}}$ is analogous to

that in Section 2.5.1 where similar approach was used for univariate random variable dY_n. However in the multivariate case where $\underline{\alpha}_n$ is assumed as the random vector, the prior mean and variance $\underline{\mu}_p, \underline{\underline{\Lambda}}_p$ can not be computed directly since $\underline{\alpha}_n$ is not observed. Hence some initial approximations are necessary.

Recall Eq. (2.7) and Eq. (2.9) are:

$$Y_{\exp,n} - Y_{code,n}|_{\underline{\alpha}=0} = \left(\frac{\partial Y_n}{\partial \underline{\alpha}}|_{\underline{\alpha}=0}\right)^T d\,\underline{\alpha} = \left(\frac{\partial Y_n}{\partial \underline{\alpha}}|_{\underline{\alpha}=0}\right)^T \underline{\alpha}_n$$

$$\hat{\underline{\alpha}}_n - \bar{\underline{\alpha}} = {}^{m-1}\underline{\underline{\Sigma}}\left(\frac{\partial Y_n}{\partial \underline{\alpha}}\right) \frac{\left(Y_{\exp,n} - Y_{code,n}|_{\underline{\alpha}=0} - \left(\frac{\partial Y_n}{\partial \underline{\alpha}}|_{\underline{\alpha}=0}\right)^T \underline{b}^m\right)}{\frac{\partial Y_n}{\partial \underline{\alpha}}^T {}^{m-1}\underline{\underline{\Sigma}}\frac{\partial Y_n}{\partial \underline{\alpha}}}$$

The error covariance matrix $\underline{\underline{\Sigma}}_n$ for each $\underline{\alpha}_n$ is given by Eq. (2.10):

$$\underline{\underline{\Sigma}}_n = {}^{m-1}\underline{\underline{\Sigma}} - \frac{{}^{m-1}\underline{\underline{\Sigma}}\frac{\partial Y_n}{\partial \underline{\alpha}}\frac{\partial Y_n}{\partial \underline{\alpha}}^T {}^{m-1}\underline{\underline{\Sigma}}}{\frac{\partial Y_n}{\partial \underline{\alpha}}^T {}^{m-1}\underline{\underline{\Sigma}}\frac{\partial Y_n}{\partial \underline{\alpha}}}$$

To determine posterior estimate of $\underline{\underline{\Sigma}}$ using Eq. (2.40), a series of approximations are performed. If $\underline{b}^m$ is the most recent estimate of the mean vector then $\bar{\underline{\alpha}} = \underline{b}^m$. Also, note that the vectors $\underline{\alpha}_n$ are not observed directly so the variance $\underline{\underline{\Lambda}}_p$ and mean $\underline{\mu}_p$ of the dataset can not be computed directly. However, we approximate each $\underline{\alpha}_n$ using the estimator from Eq. (2.9). Hence, we can approximate the variance $\underline{\underline{\Lambda}}_p$ and mean $\underline{\mu}_p$ of the dataset. In addition, we can substitute $(\underline{\alpha}_n - \bar{\underline{\alpha}})(\underline{\alpha}_n - \bar{\underline{\alpha}})^T$ in the summation term in Eq. (2.40) with its expected value.

E-Step:

$$E[(\underline{\alpha}_n - \bar{\underline{\alpha}})(\underline{\alpha}_n - \bar{\underline{\alpha}})^T] = E[\underline{\alpha}_n - \bar{\underline{\alpha}}]E[\underline{\alpha}_n - \bar{\underline{\alpha}}]^T + \underline{\underline{\Sigma}}_n = (\hat{\underline{\alpha}}_n - \underline{b})(\hat{\underline{\alpha}}_n - \underline{b})^T + \underline{\underline{\Sigma}}_n$$

Finally, we arrive at the updated estimate of $\underline{\underline{\Sigma}}$ as follows.

$$\underline{\underline{\Sigma}} = \frac{\underline{\underline{\Lambda}}_p^{-1} + \left(\frac{\kappa_p N}{\kappa_p + N}\right)(\underline{b} - \underline{\mu}_p)(\underline{b} - \underline{\mu}_p)^T + \sum_{n=1}^{N}\left((\hat{\underline{\alpha}}_n - \underline{b})(\hat{\underline{\alpha}}_n - \underline{b})^T + \underline{\underline{\Sigma}}_n\right)}{(\nu_p + N + I + 2)}$$

Substituting the estimators from Eq. (2.9) and Eq. (2.10) in the equation above we obtain the following:

$$\begin{aligned} {}^{m}\underline{\underline{\Sigma}} = {} & \frac{\underline{\underline{\Lambda}}_p^{-1} + \left(\frac{\kappa_p N}{\kappa_p + N}\right)({}^{m}\underline{b} - \underline{\mu}_p)({}^{m}\underline{b} - \underline{\mu}_p)^T + N\,\underline{\underline{\Sigma}}^{m-1}}{(\nu_p + N + I + 2)} + \\ & \sum_{n=1}^{N}\left(\frac{{}^{m-1}\underline{\underline{\Sigma}}\,\frac{\partial Y_n}{\partial \underline{\alpha}}\,\frac{\partial Y_n}{\partial \underline{\alpha}}^T\,{}^{m-1}\underline{\underline{\Sigma}}}{\frac{\partial Y_n}{\partial \underline{\alpha}}^T\,{}^{m-1}\underline{\underline{\Sigma}}\,\frac{\partial Y_n}{\partial \underline{\alpha}}}\left(\frac{\left(Y_{\exp,n} - Y_{code,n}|_{\underline{\alpha}=0} - \left(\frac{\partial Y_n}{\partial \underline{\alpha}}|_{\underline{\alpha}=0}\right)^T \underline{b}^m\right)^2}{\frac{\partial Y_n}{\partial \underline{\alpha}}^T\,{}^{m-1}\underline{\underline{\Sigma}}\,\frac{\partial Y_n}{\partial \underline{\alpha}}} - 1\right)\right) \\ & \frac{}{(\nu_p + N + I + 2)} \end{aligned} \tag{2.41}$$

which completes the MAP update.

2.5.4 Weighting Scheme and Algorithm

An averaging scheme to calculate the final covariance matrix is proposed as a weighted sum between the two covariance matrices found in Section 2.5.2 and Section 2.5.3. If γ denote the weight ($0<\gamma<1$), we obtain the following:

$$\underline{\underline{\Sigma}}_{final} = \gamma\,\underline{\underline{\Sigma}}_1 + (1-\gamma)\,\underline{\underline{\Sigma}}_2 \tag{2.42}$$

where $\underline{\underline{\Sigma}}_1$ and $\underline{\underline{\Sigma}}_2$ are the covariance matrices as found in Section 2.5.2 and Section

2.5.3, respectively. This final weighted covariance matrix is then used for subsequent EM iteration for the mean vector $\underline{b}$ found in Section 2.5.1.

The algorithm to find the MAP estimates can now be summarized as follows:

1. Start with an initial guess for covariance matrix $\underline{\underline{\Sigma}}$. A good initial guess for the covariance matrix is the identity matrix.
2. Maximize the joint posterior distribution of the mean and variance of dY_n with respect to $\underline{b}$. This is done by solving the linear system given by Eq. (2.37) to compute $\underline{b}$.
3. Using the vector $\underline{b}$ found in step 2, maximize the joint posterior distribution of mean and variance of dY_n with respect to $\underline{\underline{\Sigma}}$. This is done by solving a non-linear system of equations given in Eq. (2.38).
4. Using the most recent vector $\underline{b}$, and $\underline{\underline{\Sigma}}$ from the previous iteration, estimate $\underline{\underline{\Sigma}}$ as given by Eq. (2.41). Note that $\underline{\underline{\Sigma}}$ from Section 2.5.2 (Eq. (2.38))is not to be used in this step directly.
5. Calculate $\underline{\underline{\Sigma}}_{final}$ according to Eq. (2.42). This covariance matrix is then used for subsequent iteration.
6. Go to step 2 and iterate until convergence. Convergence can be defined as the maximum difference between two iterations of $\underline{b}$ is below some pre-set threshold value.

Once convergence is achieved, Eq. (2.5) can be used to transform the pseudo parameter α (and associated variance) back to the physical multipliers.

If a random variable X follows normal distribution then the random variable Y defined by $Y=exp(X)$ follows log-normal distribution. If μ is the mean and σ^2 is the variance of X then the probability density functions for X and Y are given by the following.

$$f(x)=\frac{1}{\sqrt{2\pi\sigma^2}}.e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$
$$f(y)=\frac{1}{y\sqrt{2\pi\sigma^2}}.e^{-\frac{(\log y-\mu)^2}{2\sigma^2}} \quad (2.43)$$

If μ_L is the mean and σ_L^2 is the variance of the log-normal random variable *Y* then

$$\mu_L=e^{(\mu+\frac{\sigma^2}{2})}$$

$$\sigma_L^2=(e^{\sigma^2}-1).e^{2\mu+\sigma^2}$$

A naïve way of estimating the mean and variance of the log-normal random variable is to simply take the exponential of the mean and variance of the normal random variable, which is the geometric mean and geometric variance respectively. In that case, we obtain the following estimates for the mean and variance of the physical multipliers p_i, which are used in this thesis for tabulating results in Section 4.4 in Chapter 4.

$$\hat{\mu}_L=e^{\mu}$$

$$\hat{\sigma}_L^2=e^{\sigma^2}$$

2.6 Wrapper Iteration

At this point the concept of repetitive application of EM is introduced, termed *wrapper iteration* which could be thought of as an outer iteration over a complete simulation. The *corrected* (best-fit) coefficients obtained at the end of a complete simulation are used to obtain corrected results which act as the base case for the second *wrapper (outer) iteration*. *(At this point, the coefficients will have been corrected once.)* A new base case is formulated, and the process is repeated until outer convergence is achieved. Convergence is obtained when the multiplier values saturate at 1.0 at which point further application of the EM makes no improvement to the code prediction (i.e. no additional

change to the input models via the physical multipliers p_i). Formulation for *wrapper iteration* is shown below.

At the end of each complete EM (outer) iteration, we obtain the mean values of the pseudo parameters α_i, denoted by $\mu_{i,n}$, and the mean value of the code multiplier p_i denoted by $\mu_{L(i,n)}$, where subscript *L* refers log-normal.

If there are a total of *N* EM (outer) iterations, then the mean value for the overall corrected code multiplier at the end of Nth (outer) iteration is given by the following. (Note: *N* here refers the number of EM outer iterations, not the total number of responses.)

$$E[p_{i,N}] = \prod^{N} \mu_{L(i,n)} \tag{2.44}$$

where $E[p_{i,N}]$ represents 'overall correction till N^{th} iteration'.

The variance of the code multiplier p_i at the end of *N* iterations is calculated as follows.

$$var(p_{n=1}.\, p_{n=2} \ldots p_{n=N}) = E[(p_{n=1} \ldots p_{n=N})^2] - (E[p_{n=1} \ldots p_{n=N}])^2 \tag{2.45}$$

where we have dropped the subscript *i*. Expanding the RHS in Eq, (2.45) assuming independence between successive outer iterations, a closed form expression for overall variance for any given multiplier can be formulated as follows:

$$var(p_1.\, p_2 \ldots p_N) = E[p_1^2.\, p_2^2 \ldots p_N^2] - (E[p_1].E[p_2] \ldots E[p_N])^2$$

$$var(p_1.\, p_2 \ldots p_N) = (E[p_1^2].E[p_2^2] \ldots E[p_N^2]) - (E[p_1]^{2\cdot} E[p_2]^2 \ldots E[p_N]^2)$$

$$var(p_1.\, p_2 \ldots p_N) = \prod_{n=1}^{N} (\sigma_n^2 + E[p_n]^2) - \prod_{n=1}^{N} (E[p_n])^2$$

where σ_n is variance of the physical multiplier in the nth iteration. Hence, the overall variance at the end of N iterations is given by the Eq. (2.46). Note that with $N=1$, the formula simplifies to $\sigma_n = \sigma_N$.

$$So, \sigma_N = \prod_{n=1}^{N} (\sigma_n^2 + \mu_{L,n}^2) - \prod_{n=1}^{N} (\mu_{L,n})^2 \tag{2.46}$$

Eq. (2.44) and Eq. (2.46) provide a complete statistical information for each input parameter. Computational results shown in this thesis are for $N = 1$. The notion of the *wrapper (outer) iteration* is an intuitive one and is presented here just for informative purposes.

CHAPTER 3 FEBA REFLOODING FACILITY AND TRACE MODEL

This chapter describes the experimental facility and TRACE model of the facility. The experimental responses are the temperature measurements at various axial locations during reflood. The experimental and TH code (TRACE) results are shown to illustrate the difference between the two.

3.1 Introduction of FEBA Test Facility

Flooding Experiment with Blocked Array (FEBA) test facility was designed to study fuel pin cladding behavior during Loss of Coolant Accident (LOCA). It was designed for the re-flooding phase of LOCA with a capability of constant flooding rate and back pressure. It consists of a 5×5 rod bundle simulating Pressurized Water Reactor (PWR) fuel pins which are heated electrically and has a cosine power profile. The rod bundle is housed in stainless steel enclosure which is insulated to minimize heat loss. Radial cross-sectional view of FEBA test facility is shown in Figure 3.1. Axial view of a single FEBA heater rod, locations of grid spacers and axial power profile is shown in Figure 3.2. Relevant geometrical dimensions are presented in Table 3.1.

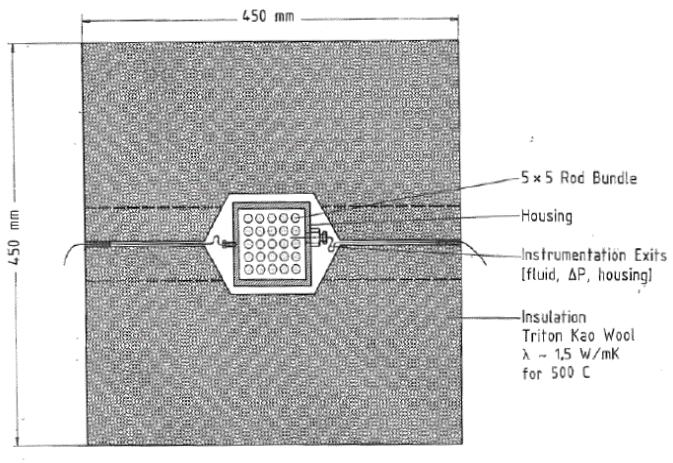

Figure 3.1 Radial cross-section view of FEBA test facility [55]

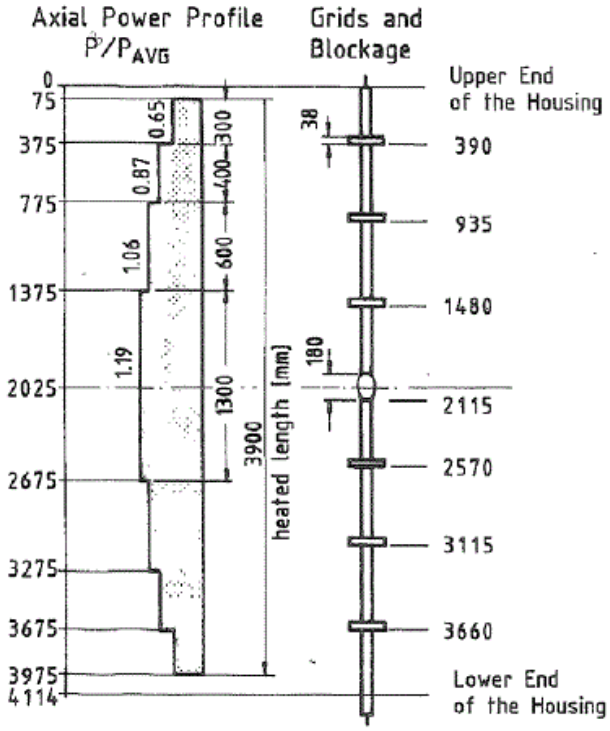

Figure 3.2 Axial cross-section view of the simulator heater rod [55]

Table 3.1 Dimensions of heater rod

Stainless steel housing thickness (mm)	6.5
Outer diameter of heater rod (mm)	10.75
Heater rod pitch (mm)	4.3
Inner length of square housing (mm)	78.5

3.2 Experiment Description

Tests with different flooding rate and back-pressure were performed. These tests simulated the re-flooding phase of Large Break LOCA in PWRs. Test #216 is chosen as the base case for which the computational (TRACE) model is built. Prior to the start of the transient, the rods are heated in stagnant steam under low power until the desired initial cladding temperature is reached. The stainless steel housing is heated up only by radiation from the rods. At the start of the transient, the heater rod power is increased to power level that simulates decay heat according to 120% ANS-standard about 40 seconds after reactor shutdown. Relevant initial and boundary conditions corresponding to Test #216 are presented in Table 3.2. The initial temperature distribution of the heater rods and housing are taken from Karlsruhe Research Center Report 3657 [55].

Table 3.2 System operating conditions

Fluid inlet velocity (cm/s)	3.8
System outlet pressure (bar)	4.12
Feed water temperature (^{o}C) (0-30 s)	48
Feed water temperature (^{o}C) (30s-end)	37
Bundle power (kW) at 0 s	200
Bundle power (kW) during transient	120% ANS standard

The transient temperature of the fuel pin cladding at various axial locations is presented in Figure 3.3. In this figure, the reference level (0 mm) corresponds to the upper end of housing (see Figure 3.2). The coolant enters the test section at axial level 4114 mm (which corresponds to the lower end of housing (see Figure 3.2).

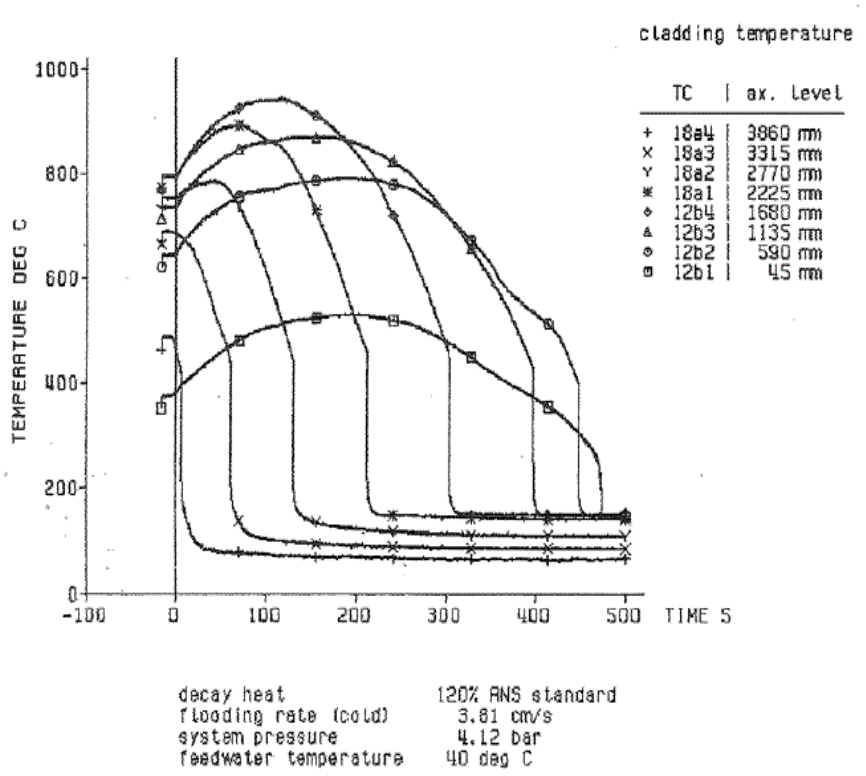

Figure 3.3 Transient temperature distribution [55]

3.3 Computational Model In TRACE

TRACE (TRAC/RELAP Advanced Computational Engine) is a modernized thermal-hydraulics code designed to merge and extend NRC's three legacy safety codes – TRAC-P, TRAC-B and RELAP5. It can be used to analyze large and small break LOCA and system transients in PWRs and BWRs. TRACE models most of the thermal-hydraulics

phenomena in one-dimension (1D), and it includes 3D flow capability though the VESSEL component.

The computational model for TRACE is built with SNAP (Symbolic Nuclear Analysis Package). The FEBA test facility is modeled using a single vertical pipe (PIPE component) with appropriate inlet (FILL component) and outlet (BREAK component) boundary conditions. The heater rods simulating fuel pins are modeled using heat structure (HTSTR component). A second heat structure is also used to model the stainless steel enclosure. Heat loss from the outer surface of the housing is not modeled, it is assumed to be a perfect thermal insulator. The total axial length of the pipe is divided into 41 cells. The first (bottom-most) and the last (top-most) cells in the PIPE component correspond to the unheated portion of the pipe. The remaining 39 cells in the middle of the pipe correspond to the heated section. The inlet, outlet and operating conditions are the same as experimental conditions (See Table 3.2). Figure 3.4 shows the TRACE nodalization. The nodalization dimensions are presented in Table 3.3.

Junctions where grid spacers are present are modeled with a reduced flow area. Pressure loss at grid spacers is modeled via abrupt area change model and additional form loss coefficients are set to 0.In addition, fine mesh reflood is activated. All other internal model coefficients are kept constant at their respective nominal values.

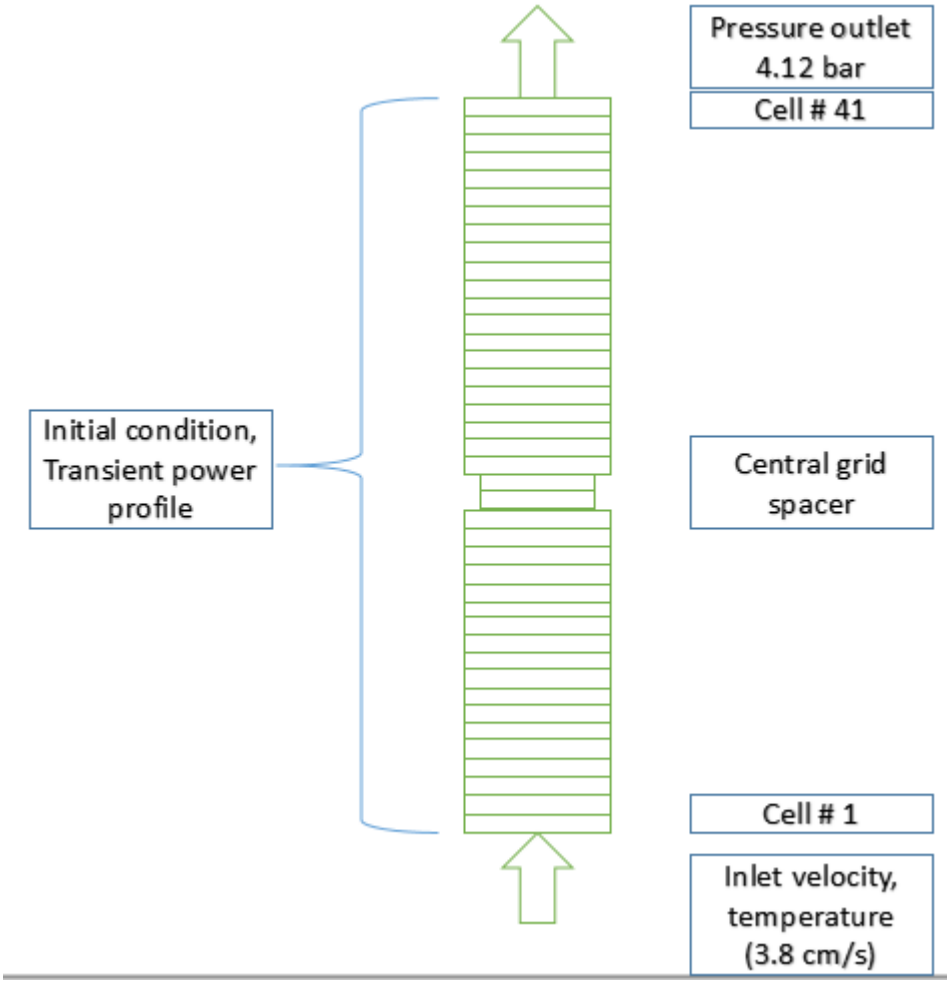

Figure 3.4 Nodalization diagram in TRACE

Table 3.3 Computational model dimensions

Cell 1 length (m) (lower unheated portion)	0.139
Cells 2-40 length (m) (middle heated portion)	0.1
Cell 41 length (m) (upper unheated portion)	0.075
Cells 1-18, 21-41 flow area (m^2)	$3.893\ 10^{-3}$
Cells 19, 20 flow area (m^2)	$3.115\ 10^{-3}$
Junction hydraulic diameter (m) (cells 3-4, 8-9, 14-15, 18-19, 19-20, 24-25, 30-31, 35-36)	0.01075
Junction hydraulic diameter (m) (Remaining junctions)	0.01344
Junction area (m^2) (cells 3-4, 8-9, 14-15, 18-19, 19-20, 24-25, 30-31, 35-36)	$3.115\ 10^{-3}$
Junction area (m^2) (remaining cells)	$3.893\ 10^{-3}$
Wall roughness (Cells 1-41)	$1.0\ 10^{-5}$
Hydraulic diameter (m)	0.01344
Heated diameter (m)	0.01844

3.4 TRACE Results of the Transient

The transient temperature of the fuel pin cladding at various axial locations calculated by TRACE is shown in Figure 3.5. The selected axial locations are the same as measured values shown in Figure 3.3.

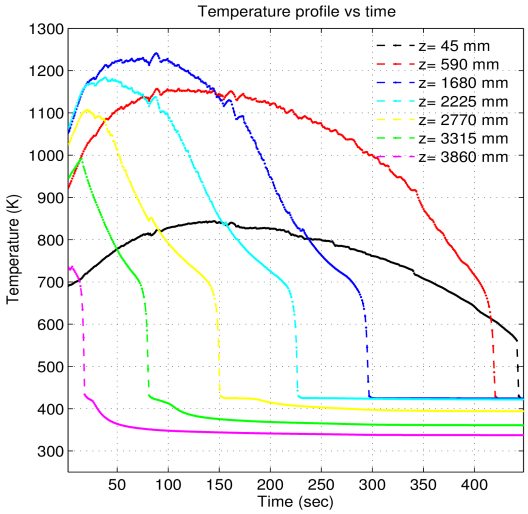

Figure 3.5 Transient clad temperature calculated using TRACE [56]

3.4.1 Results Comparison

Figure 3.6 shows a comparison between measured and TRACE calculated temperatures for three axial elevations. The selected axial locations correspond to top, middle and bottom part of the fuel rods.

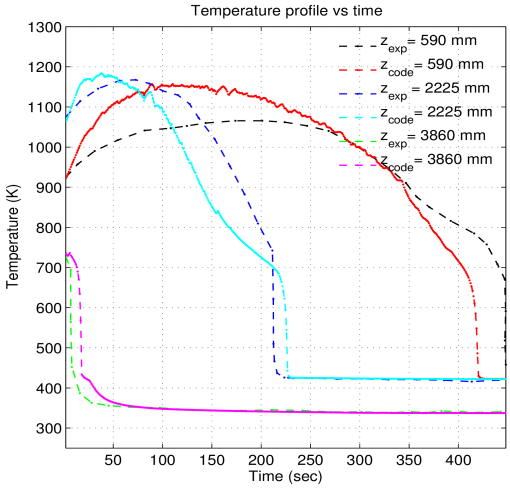

Figure 3.6 Comparison between measured and TRACE calculated cladding temperatures

Based on results presented in Figure 3.6, the initial rate of cladding temperature increase is over-predicted by TRACE, especially for elevations *z = 590 mm* and *z = 2225 mm*. The difference in Peak Cladding Temperature (PCT) between the experiment and TRACE code is larger for cells that are located closer to the top (outlet). For example, $\delta(PCT_{z=590\,mm}) \approx 90\,K$ whereas $\delta(PCT_{z=2225\,mm}) \approx 25\,K$. For locations close to the bottom (inlet) these deviations are much lower, for instance $\delta(PCT_{z=3860\,mm}) < 5\,K$. Table 3.4 shows the comparison of maximum cladding temperature and quench time between the experiment and TRACE for all axial levels. In addition, the PCT occurs earlier in TRACE simulation compared to the experiment for all axial levels. The discrepancies in PCT time and magnitude prediction may be attributed to the limitation in modeling heat losses. The failure to dissipate heat through enclosure walls may be the cause for a faster increase in cladding temperature. Small discrepancy in prediction of quenching time can also be seen in Table 3.4. The largest difference $\delta t_{quench} = 35\,sec$ is observed for *z = 1135 mm*. The quenching times are predicted fairly accurately for lower axial levels i.e. *z >1135 mm*.

Table 3.4 Comparison of Peak Cladding Temperature

Axial height (mm)	TRACE		Experiment	
	PCT (K)	Quench time (sec)	PCT (K)	Quench time (sec)
45	850	450	802	477
590	1155	425	1060	456
1135	1227	367	1135	400
1680	1240	304	1210	316
2225	1185	231	1168	218
2770	1100	154	1050	138
3315	980	80	955	75
3860	730	60	730	64

The observations above show that even under well-defined simple geometry and accurate boundary and initial conditions, the best-estimate thermal-hydraulics code TRACE yields differences of up to 100 K in the PCT prediction. Considering that a high accuracy is desired to predict fuel pin failure during LOCA, any significant deviation from the experimental measurement needs to be accounted for in the form of quantification of code input uncertainty.

CHAPTER 4 RESULTS

This Chapter presents specific examples of implementation and results of methodology developed in Chapter 2. In Section 4.1, analytical function is considered to demonstrate the methodology. Noise is introduced to the analytic function to simulate noisy experimental responses. The sensitivity of the analytical function with respect to its parameters is also quantified in a sensitivity matrix. Based on (i) artificially generated experimental responses, (ii) precise synthetic function responses (namely base code responses) and (iii) the sensitivity matrix, function parameters viz. the mean and variance are calculated according to the methodology in Section 2.3. Second, we take the TRACE model for FEBA reflooding experiment described in Chapter 3 and calculate the sensitivities of the TRACE code parameters (physical models), hence creating a ranking of the most important parameters. This ranking is presented in Section 4.2. In Section 4.3, the input parameters (TRACE physical models) are artificially altered by some arbitrary values such that the modified simulation acts as a synthetic experiment. This is analogous to the process of forming experimental responses by introducing noise to analytical function in Section 4.1. Then, starting from the base TRACE model, methodologies developed in Section 2.3, Section 2.4 and Section 2.5 are used to verify if these methodologies successfully find the *correct* parameters. Note that the correct parameters are the arbitrary values set by the user (known *prior* to us but not used anywhere in the computation) with which the experimental (synthetic) responses are generated. Details are provided in Section 4.3. Section 4.1 and Section 4.3 can be thought of as a *verification* step that needs to be performed before actual experimental data are used for *parameter uncertainty quantification*.

In Section 4.4, actual experimental data from FEBA Test 216 is used and estimates of the TRACE input (physical models) parameters are tabulated for several important physical models. The tabulated values from Section 4.4 can then be used for future *forward uncertainty quantification*. Finally, validation is performed using FEBA Test 234 in

Section 4.5, using input parameter uncertainty determined in Section 4.4.

4.1 Maximum Likelihood Estimate of Analytical Functions

4.1.1 Linear Function

Consider the following function.

$$Y(n) = n^3 x_1 + n^2 x_2 + n^3 x_3 \tag{4.1}$$

Here, x_1, x_2, x_3 are the code input parameters (e.g. physical model coefficients) that are fixed and $n = 1, 2, 3 \ldots 9$ is the independent test condition. Therefore, if $n = 1$ we obtain the first response, say $Y_{1,code}$, with $n = 2$, we get the second response, say $Y_{2,code}$ and so on.

Now, consider random numbers in (0.6, 1.0) and multiply $Y_{1,code}$, $Y_{2,code}$ … with these random numbers to generate noisy experimental data from the given function $Y(n)$. These are assumed to be the experimental measurements and are denoted as $Y_{1,\exp}, Y_{2,\exp} \ldots$. Now we have $Y_{n,exp}, Y_{n,code}$ for $n = 1, 2, 3 \ldots 9$. Then the problem can be posed as follows.

We want to find multipliers p_1, p_2 *and* p_3 of the code input parameters (e.g. physical model coefficients) x_1, x_2 *and* x_3 such that the $Y_{n,exp} \approx Y_{n,code}$ for all n i.e. we are looking for the code input parameters $x_{1,new}$, $x_{2,new}$ *and* $x_{3,new}$ given by

$$\begin{aligned} x_{1,new} &= x_1 . p_1 \\ x_{2,new} &= x_2 . p_2 \\ x_{3,new} &= x_3 . p_3 \end{aligned} \tag{4.2}$$

Since x_1, x_2 and x_3 are fixed, we want to determine p_1, p_2, p_3. To this end, we write the function as follows.

$Y(n, p_1, p_2, p_3) = n^3 p_1 . x_1 + n^2 p_2 . x_2 + n^3 p_3 . x_3$ where p_1, p_2, *and* p_3 are multipliers. It is again stressed that the base code calculations are preformed with p_1, p_2, *and* p_3 equal to 1.0 (i.e. the nominal code input values).

We can now create the sensitivity matrix $\frac{\partial Y_n}{\partial p_i}$ which is given by $\frac{\partial Y(n, p_1, p_2, p_3)}{\partial p_i}$ *for n = 1,2,3...9,* and *i = 1,2,3.* The derivatives are taken around the base value of p_i = *1.0.* As mentioned in Chapter 2, the multipliers p_i are related to pseudo parameters α_i by

$$p_i = \exp(\alpha_i)$$

where i = 1, 2, 3.

Furthermore, by the chain rule,

$\frac{\partial Y_n}{\partial \alpha_i} = \frac{\partial Y_n}{\partial p_i} . \frac{\partial p_i}{\partial \alpha_i}$ for any given *i, n*

Eq. (4.1) combined with $Y_{n,\exp}, Y_{n,code}$ for *n = 1, 2, 3, ..9* is used as input to the algorithm developed in Chapter 2 to find the statistical parameters for p_i. Note that the code input parameters x_1, x_2 and x_3 have not been prescribed yet. For this example the following values of code input parameters are taken.

x_1 = 1.2

x_2 = 1.2

x_3 = 1.2

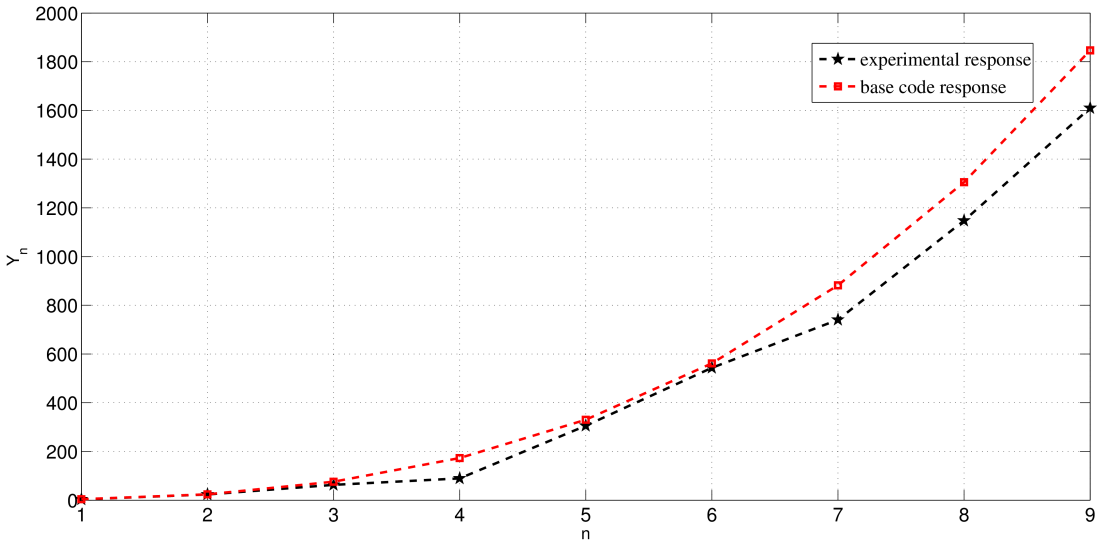

Figure 4.1 Comparison between base code responses and experimental responses

Figure 4.1 shows the comparison between the base code response and experimental response. Application of MLE linear method developed in Section 2.3 on this example yields the results shown in Table 4.1 It shows the mean values and variances of the pseudo parameters α_i and the corresponding quantities for the physical code multipliers p_i. Since the parameters α_i follow normal distribution (by assumption), it follows that the physical multipliers follow log-normal distribution.

Table 4.1 Statistical parameters for Linear function

	α_1	α_2	α_3	p_1	p_2	p_3
mean	-0.16	0.046	-0.16	0.85	1.04	0.84
variance	0.038	$5.4\ 10^{-5}$	0.038	0.17	0.007	0.17

Figure 4.2 shows the comparison of the base code response ($p_i = 1$) with the code responses corrected according to the MLE linear method (see Table 4.1). The experimental (synthetic) responses are also shown in Figure 4.2. It can be seen that the

base code responses are corrected to match experimental results with better accuracy. In effect, the MLE linear algorithm finds the code input parameters that maximizes the likelihood of producing code responses as close as possible to the experimental ones.

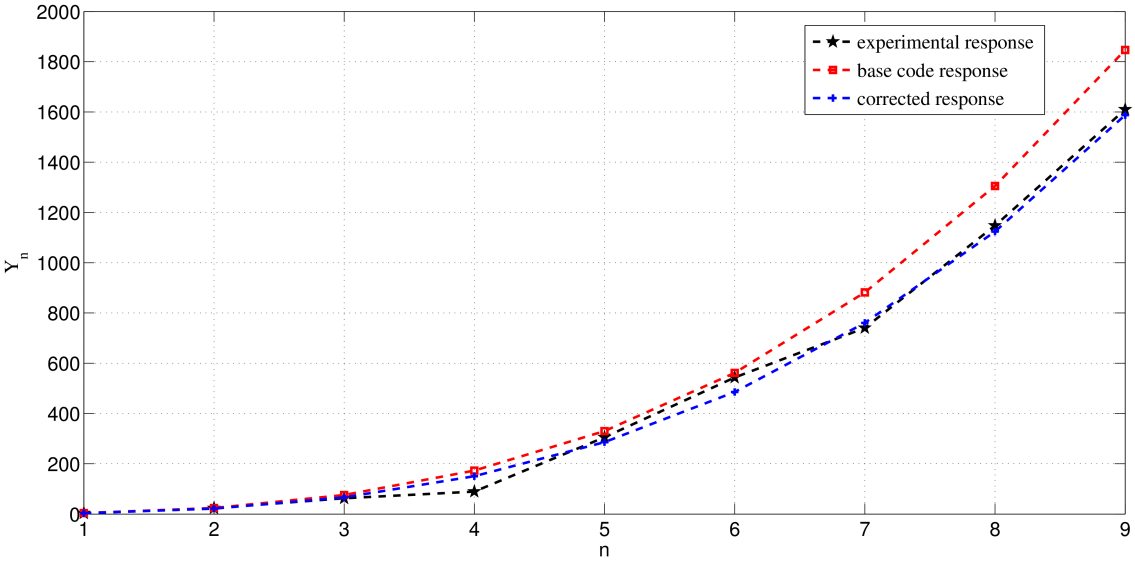

Figure 4.2 Comparison between base code, corrected code and experimental responses

In addition, Figure 4.3 shows the upper limit and lower limit of the code responses calculated at 1σ from the mean of the multipliers. The standard deviation σ is simply the square root of variance given in Table 4.1. The results show that the extreme bounds capture the noisy experimental responses, thus enabling the user to successfully characterize the uncertainty in the code input parameters.

Finally, a non-linear least square fit is performed. The converged results from the non-linear least-square fit are plotted on the same figure to compare with the MLE linear algorithm. As seen from Figure 4.3, a very good agreement is observed between the results from *maximum likelihood estimates* and non-linear least square fits. The non-linear least square code is written based on Newton's method where we minimize the following function.

$$g(x_1, x_2, x_3) = \sum_{n=1}^{9} (Y_{n,exp} - Y_{n,code})^2 \qquad (4.3)$$

where $Y_{n,code} = n^3 . x_1 + n^2 . x_2 + n^3 . x_3$

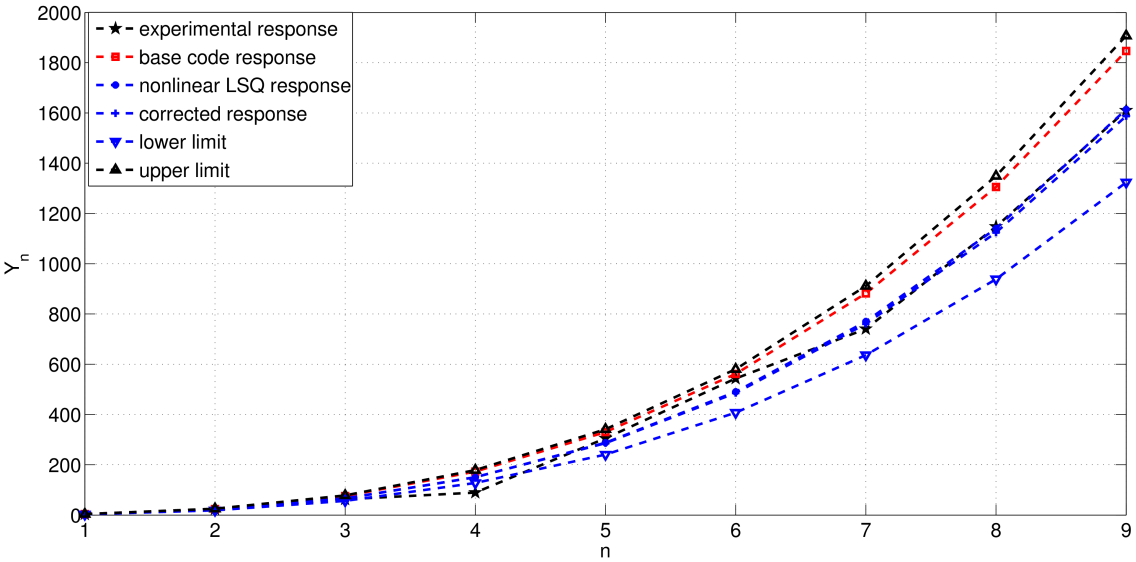

Figure 4.3 Comparison of calibrated results with extreme code responses

4.1.2 Non-linear Function

Consider the function $Y(n) = 100 \sin(\frac{\pi}{n}) x_1^2 + (100 - n^2) x_2 + (\frac{n^3}{10}) x_3$. Base code responses $Y_{n,code}$ are calculated by substituting $n = 1, 2, ...22$ in the function above. Furthermore x_1, x_2 *and* x_3 are assumed to have the same values as in Section 4.1.1. Twenty-two random numbers in (0.6,1.0) are taken and multiplied to $Y_{n,code}$ to generate $Y_{n,exp}$. Then we write the function *f* as follows:

$Y(n, p_1, p_2, p_3) = 100\sin(\frac{\pi}{n})\, p_1 x_1^2 + (100 - n^2)\, p_2 x_2 + (\frac{n^3}{10})\, p_3 x_3$ where we seek to find the statistical parameters of the multipliers p_1, p_2 *and* p_3. Figure 4.4 shows the comparison between base code and experimental responses.

It is interesting to note that the first term in Y is non-linear in the parameter x_1. However, the multiplier assigned to it is linear! Analyzing the expression for Y it is evident that the contribution of the first term decreases monotonically for $n > 2$. It might seem erroneous that a single multiplier has been assigned to a squared term and consequently, subsequent inference about the squared parameter might be incorrect. A preview of the final results shown in Figure 4.5 is helpful. It shows a comparison between corrected code response (obtained using MLE linear algorithm) and results obtained from non-linear least-square fit. As seen from Figure 4.5, there is a very good agreement between the two methodologies. As mentioned earlier, the impact of non-linear term is minimal in a statistical sense and hence the user can lump the non-linear term as a single linear term. However, this approximation should be used with caution, only when the user is certain the impact of non-linear terms is small. Unfortunately, in most cases of practical interest (e.g. large-scale computer simulations) there are no explicit analytical expressions like the one analyzed in this Section. Blindly lumping non-linear parameters into a single parameter and treating them as linear may produce incorrect inference.

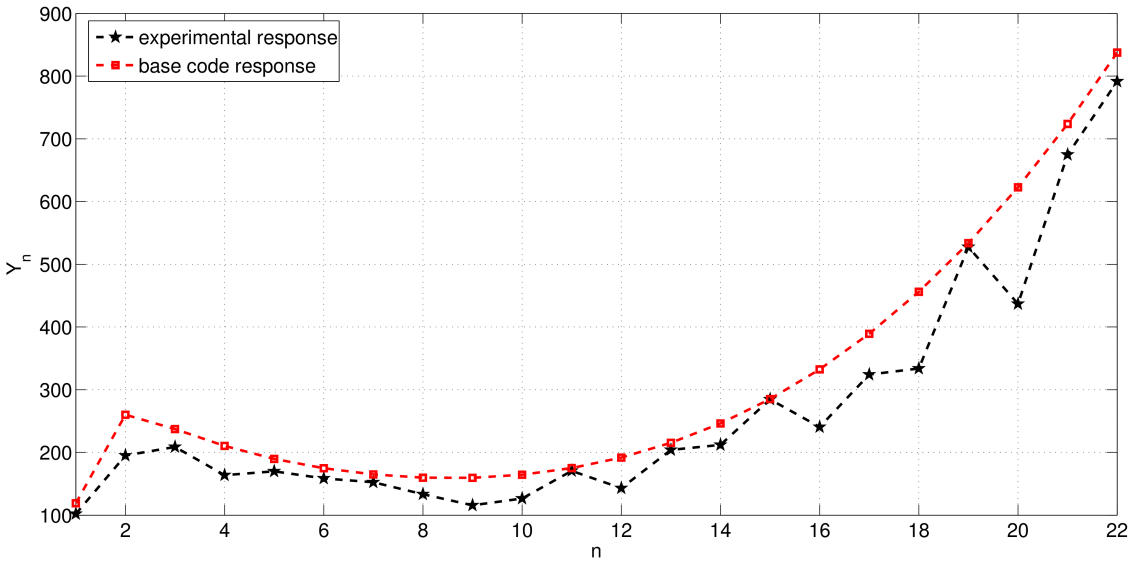

Figure 4.4 Comparison between base code responses and experimental responses

Table 4.2 shows the mean value and variances of the input parameters. Figure 4.5 shows the comparison between corrected code response and converged results from non-linear least-square fit. It also shows the extreme code limits – upper and lower bounds. These extrema are calculated at 1σ deviation from the mean p_i values.

Table 4.2 Statistical parameters for Non-linear function

	α_1	α_2	α_3	p_1	p_2	p_3
Mean	-0.17	-0.13	-0.14	0.84	0.87	0.87
variance	0.022	$8.8\ 10^{-12}$	0.005	0.016	$6.7\ 10^{-12}$	0.004

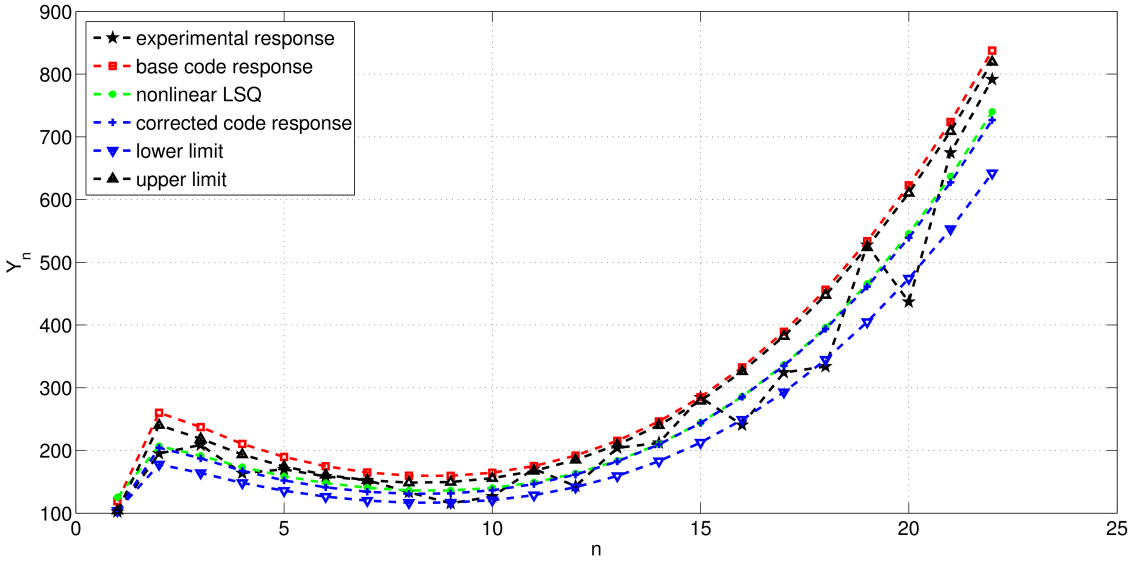

Figure 4.5 Comparison of base code, corrected code and experimental responses with code extrema

4.2 Sensitivity Analysis

Sensitivity of TRACE code parameters are quantified using FEBA Test 216 described in Chapter 3. Sensitivity analysis is performed to rank TRACE thermal-hydraulics input parameters (physical models) in order of importance as shown in Table 4.3. Change in peak cladding temperature at $z = 1680\ mm$ is taken to determine sensitivity of the input parameters (physical models). The formula belows shows the calculation of absolute relative difference in peak cladding temperature that is used to rank the model parameters. Each parameter is perturbed by 20% and the absolute value of the relative difference in the peak cladding temperature between two cases (base and perturbed case) is calculated to obtain the ranking of the parameters. ($p = 1.2$ refers to the perturbed calculation and $p = 1.0$ refers to nominal code calculation.) The ranking thus obtained holds for reflooding experiments. Parameters corresponding to absolute relative difference of more than 0.90% are grouped as the most important parameters corresponding to absolute relative difference between 0.90% and 0.1% are grouped as relevant, the remaining parameters are grouped as insignificant.

$$Absolute\ Rel.difference(\%) = \left| \frac{T_{peak}|_{p=1.2} - T_{peak}|_{p=1.0}}{T_{peak}|_{p=1.0}} \right| * 100\%$$

Table 4.3 Ranking of TRACE code input parameters

Parameter Description	Parameter ID	Peak Temperature (*z = 1680 mm)*	Absolute Relative Difference *(%)*
Dispersed flow film boiling heat transfer coefficient	1	1220.2	1.68
Interfacial drag (dispersed flow film boiling) coefficient	2	1255.1	1.12
Single phase vapor to wall heat transfer coefficient	3	1229.5	0.93
Departure from nucleate boiling critical heat flux	4	1244.8	0.29
Film to transition boiling T_{min} criterion temperature	5	1237.4	0.29
Vapor to wall inverted annular heat transfer coefficient	6	1242.5	0.11
Liquid to wall inverted annular heat transfer coefficient	7	1242	0.07
Interfacial drag (inverted slug flow) coefficient	8	1241.9	0.06
Interfacial drag (inverted annular flow) coefficient	9	1241.5	0.03
Wall drag coefficient	10	1241.5	0.03
Liquid to interface bubbly-slug heat transfer coefficient	11	1240.9	0.01
Sub-cooled boiling heat transfer coefficient	12	1241.0	0.008

4.3 Synthetic Reflooding Experiment

FEBA Test 216 reflooding experiment described in Chapter 3 is considered. Specific TRACE physical models are taken as uncertain input parameters (equivalent to x_1, x_2 and

x_3 in Section 4.1). These code coefficients are multiplied with an arbitrary pre-determined multipliers p_i to generate synthetic experimental responses. Starting from these experimental observations *(now, without the knowledge of what multipliers i.e. physical models generated them)* and the base TRACE model for FEBA Test 216, we seek to determine the statistical parameters of the multipliers.

The experimental responses ($Y_{n,exp}$) consist of transient cladding temperature measurements at $z = 45$ *mm* for $t = 20, 40, 60, \ldots 360$ *secs.*

It should be further noted that each TRACE code input parameter/coefficient has a unique multiplier. For example, calculation of wall-fluid heat transfer coefficient (HTC) is dependent on the flow regime. In a transient that involves depressurization and subsequent re-flooding, various flow regimes are encountered at different axial levels. Hence there may be many heat transfer correlations active at any given time, depending on the flow regime. In this thesis, a unique multiplier is assigned to each correlation, even for the same physical model (e.g. different input parameter multiplier for bubbly flow HTC and annular flow HTC).

4.3.1 Synthetic Example I: Dispersed flow film boiling HTC and interfacial drag coefficient

We use the methodology developed in Chapter 2 to determine the physical multipliers *(unknown a priori)* for a thermal-hydraulics transient modeled with the TRACE code. Two code parameters (i) dispersed flow film boiling heat transfer coefficient and (ii) interfacial drag coefficient are considered as unknown in this example. Experimental (synthetic) response is generated by multiplying these coefficients by 0.7 and 1.3 respectively. Given the experimental (synthetic) responses, we seek to determine the multipliers. In addition to $Y_{n,exp}$, we need two further quantities viz. the sensitivity matrix

$\frac{\partial Y_{n,code}}{\partial \alpha_i}$ and $Y_{n,code}$, for $i = 1, 2$ and $n = 1, 2, \ldots N$. The sensitivity matrix could be easily constructed using finite difference. $Y_{n,code}$ is obtained directly from the base TRACE code simulation (i.e. with nominal values $p_i = 1.0$).

A comparison between experimental data and base code response for $z = 1680$ *mm* and $z = 2770$ *mm* is presented in Figure 4.6. It is useful to remember that the cold water flows from the bottom ($z = 4114$ *mm*) at a constant rate and gradually rises up. Hence the bottom portion of the rod bundles closest to the bottom quenches first. Figure 4.6 shows the quenching of both axial levels. At quenching, there is a sharp decrease in the cladding temperature, after which it attains a nearly constant value close to the saturated liquid temperature. This is shown by the constant temperature in the figure. For the upper level ($z = 1680$ *mm*),the cladding temperature continues to increase until mist cooling heat transfer is greater than power deposited to the fuel rod. Sub-cooled mist rising generated upstream of the quench-front travels downstream faster than liquid interface and hence, gradually cools the rod cladding before quenching takes place.

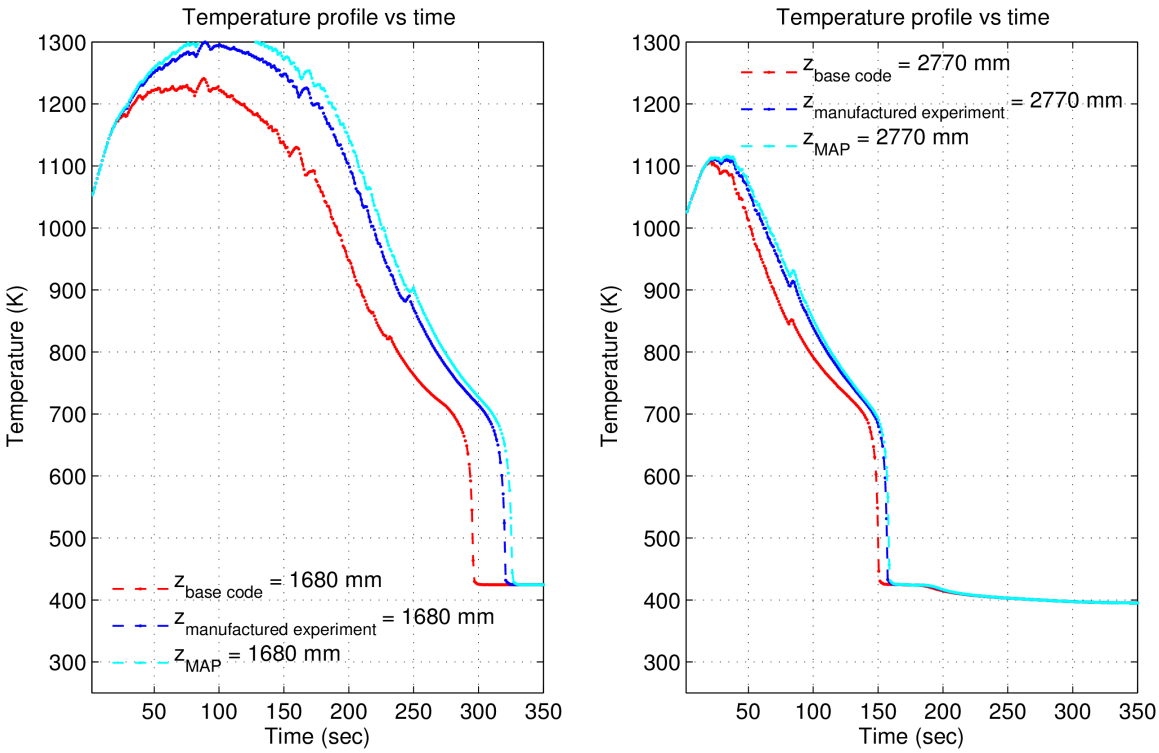

Figure 4.6 Cladding temperature profile during transient (dispersed flow film boiling HTC and interfacial drag coefficient)

Figure 4.6 shows that the base code cladding temperature is underpredicted for the axial level *z = 1680 mm* and *z = 2770 mm*. By seeking the correct multipliers to the dispersed flow film boiling HTC and interfacial drag coefficient, we aim to arrive closer to the experimental (synthetic) measurements of the cladding temperature. Figure 4.6 also shows the corrected code response obtained by using the Maximum a Posteriori (MAP) estimates of the multipliers. A very good agreement is observed between the corrected MAP estimate and experimental (synthetic) data. Associated statistical parameters of the physical multipliers for dispersed flow film boiling HTC and interfacial drag coefficient are presented in Table 4.4.

Table 4.4 Mean value and variance (dispersed flow film boiling HTC and interfacial drag coefficient)

	Mean				Variance		
	True mean	MLE linear	MLE quadratic	MAP	MLE linear	MLE quadratic	MAP
dispersed flow film boiling HTC	0.7	0.66	0.70	0.66	1.00	1.02	1.00
interfacial drag coefficient	1.3	1.44	1.22	1.26	1.01	1.03	1.03

Table 4.4 shows that all three methodologies (MLE - linear, MLE - quadratic and MAP) yield similar values for the mean and variance of the two input parameters under consideration. Moreover, the mean values calculated by each of the three methods is close to the original multiplier values (0.7 and 1.3) with which the experimental (synthetic) data is generated.

Uncertainty quantification in the inverse domain is very sensitive to the set of input arguments supplied to the underlying algorithm [57]. Detailed discussions on stability and sensitivity of inverse problems are presented in Section 4.6. Results presented in Table 4.4 are obtained by taking $n = 18$, where n is the number of experimental observations (equivalently base code responses) considered for the algorithm. Table 4.5 summarizes the statistical properties of the code multipliers obtained using Maximum a Posteriori (MAP) estimates as a function of number of experimental observations considered.

Table 4.5 Mean and variance as a function of number of experimental responses for MAP estimates (dispersed flow film boiling HTC and interfacial drag coefficient)

Number of experimental observations (n)	Mean (dispersed flow film boiling HTC)	Variance (dispersed flow film boiling HTC)	Mean (interfacial drag coefficient)	Variance (interfacial drag coefficient)
18	0.66	1.00	1.26	1.03
34	0.65	1.02	1.36	1.02
51	0.64	1.01	1.39	1.02
65	0.64	1.01	1.40	1.05

As seen in Table 4.5, the mean multipliers for respective coefficients are close to the true multiplier values (0.7 and 1.3). It is helpful to remember that the framework for inverse uncertainty quantification developed in Chapter 2 are built on a specific set of user assumptions. As discussed later in the thesis in Section 4.6, the assumptions on input arguments (experimental observations) and their validity play an important role in determining the accuracy and quality of the predicted mean and variance of multiplier values.

4.3.2 Synthetic Example II: Interfacial drag coefficient and single phase vapor to liquid HTC

Here, we consider interfacial drag coefficient and single phase vapor to liquid HTC as the two input parameters. The experimental (synthetic) response is generated the same way as described in Section 4.3.1 with pre-determined multipliers. In this example, multipliers of 0.6 and 1.5 are used for interfacial drag coefficient and single phase vapor to liquid HTC, respectively. Figure 4.7 shows a comparison between experimental (synthetic) results, base code results (nominal value for physical multipliers p_i=1.0) and corrected code results for *z = 1135 mm* and *z = 2225 mm.* A near perfect agreement is observed between the experimental data and the corrected code response with Maximum a

Posteriori (MAP) estimates of the input multipliers. Statistical properties of the multipliers obtained from different methodologies are compared in Table 4.6.

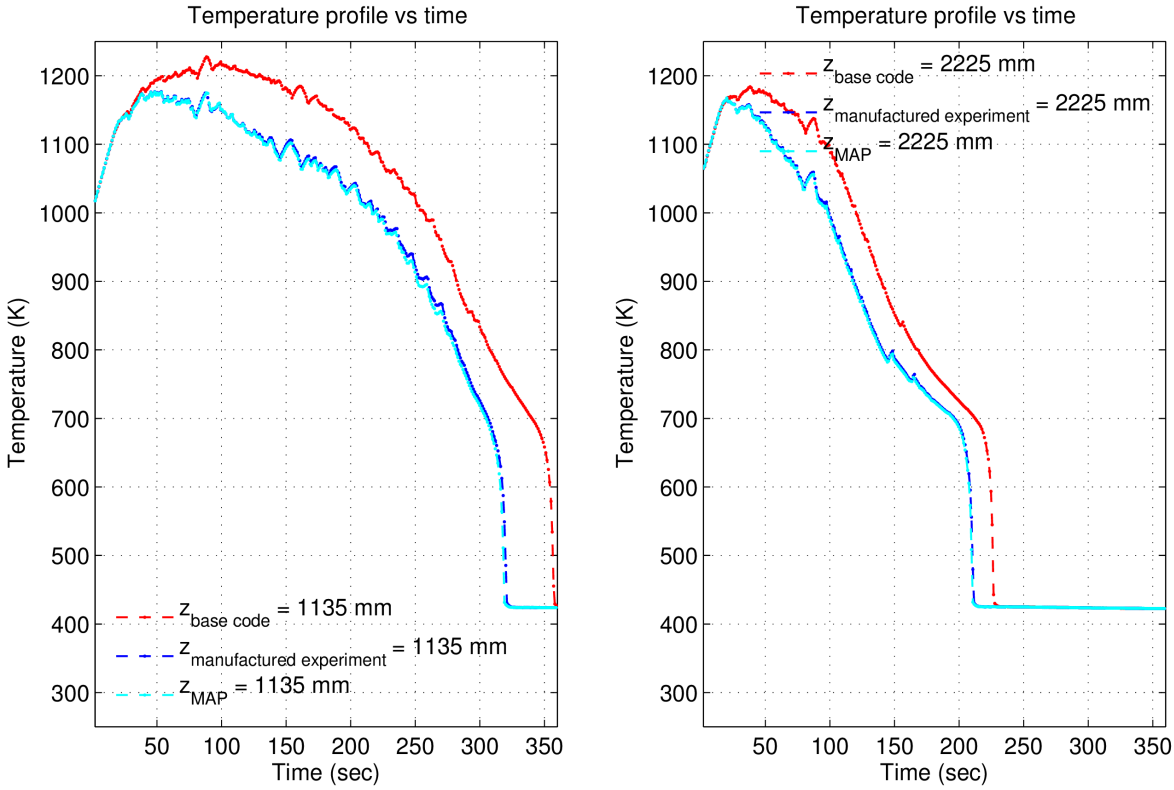

Figure 4.7 Cladding temperature profile during transient (interfacial drag coefficient and single phase vapor to liquid HTC)

Table 4.6 Mean value and variance (interfacial drag coefficient and single phase vapor to liquid HTC)

	Mean				Variance		
	True mean value	MLE linear	MLE quadratic	MAP	MLE linear	MLE quadratic	MAP
interfacial drag coefficient	0.6	0.56	0.58	0.57	1.01	1.00	1.03
single phase vapor to liquid HTC	1.5	1.41	0.97	1.47	1.00	1.02	1.3

From Table 4.6, we can see that the mean values of the multipliers obtained using MLE linear and MAP closely match the input multipliers (0.6 and 1.5) that were used to generate the experimental (synthetic) data. However, the mean value of the interfacial drag coefficient multiplier predicted by MLE quadratic is quite different. This can be attributed to two main factors. First, it is imperative that a non-linear solver start close to the true solution to enable it to converge to a global minimum. Moreover, as detailed in Section 2.4, MLE quadratic tries to find the mean value through a series of double approximations. First, the non-linear solver performs a series of linear approximations to step through one iteration of Expectation-Maximization algorithm, which itself also starts with an approximation. Hence it is not difficult to see the calculation converging to a wrong solution. Second, numerical approximations of mixed derivatives in Eq. (2.24) and Eq. (2.25) are very sensitive to finite-difference step size and require special monitoring. Combined effect of these factors have led to incorrect inference when using MLE quadratic.

Table 4.7 summarizes the statistical properties of the code multipliers obtained using Maximum a Posteriori (MAP) estimates as a function of number of experimental observations considered. Discussions of the impact on the computed statistical parameters due to the assumptions on user-supplied input dataset (experimental observation) are presented in Section 4.6.

Table 4.7 Mean and variance as a function of number of experimental responses for MAP estimates (interfacial drag coefficient and single phase vapor to liquid HTC)

Number of experimental observations (n)	Mean (interfacial drag coefficient)	Variance (interfacial drag coefficient)	Mean (single phase vapor to liquid HTC)	Variance (single phase vapor to liquid HTC)
18	0.57	1.03	1.47	1.3
33	0.61	1.03	1.59	1.01
47	0.61	1.14	1.66	1.01
66	0.56	1.12	1.33	1.11

4.4 Model Uncertainty for LOCA Re-flooding

Different combination of important TRACE input parameters (physical models) shown in Table 4.3 are considered for calibration and inverse uncertainty quantification. For each combination of TRACE input parameters (physical models), MLE linear, MLE quadratic and MAP estimates of the mean and variance are quantified and tabulated. In addition, each sub-section shows comparison of transient temperature and axial quench location obtained using the methodologies developed in Chapter 2. Section 4.4.10, includes comparison of error norms between experimental data and TRACE results (base and calibrated).

All figures shown in the following sub-sections are generated with $\gamma = 0.5$ (equal weight) in Eq. (2.42) for calculating MAP estimates of the input parameters. The sensitivity matrices are calculated using forward difference using a step-size of $\Delta p = 0.2$ around the nominal value of the physical multipliers p_i.

As seen in Figure 4.8 (input model ID 1 and ID 4, see Table 4.3), the corrected transient temperature profile predicted by MAP estimate mimics the experimental reading much better than the base code transient profile. A slight improvement is observed in the case of a MLE – linear prediction. However, it is seen that the MLE quadratic prediction is actually no better than the base case reading. Moreover, since measured data from $z = 1680\ mm$ and $z = 2225\ mm$ are used for calibration purposes we can see that the profile for those axial levels fits better than $z = 2770\ mm$. In addition, the standard deviation for physical multiplier predicted by MLE linear and MLE quadratic is much higher than MAP estimate. High standard deviation reflects a low precision on the estimation of the mean and its confidence interval while estimating MLE. Figure 4.8 shows that the hypothesis (belief) on the assumptions of normality used on prior mean and inverse

gamma (and inverse Wishart) on prior variance improves the simulated solution. However it should be stressed that these assumptions are assigned according to the belief of the user prior to collecting the data (evidence) and hence should not be adjusted later merely to seek a better agreement with the observed data.

In Figure 4.9 (input model ID 1 and ID 2, see Table 4.3), we can see that MLE - linear and MAP estimates are approximately the same. However the transient predicted by MLE quadratic slightly underpredicts the temperature profile for $z = 1680\ mm$ and $z = 2225\ mm$ relative to MLE linear and MAP. Discrepancies in the prediction between MLE quadratic and the other two methods (MLE linear and MAP) is attributed to the discussions presented in Section 4.6. Moreover, temperature profiles and quench front location obtained using MAP estimates are a better fit to the experiment data (as in Figure 4.8) compared to MLE linear and MLE quadratic. Similar observation is made in Figure 4.10 (input model ID 2 and ID 3, see Table 4.3). The MLE linear and MAP method predict similar transient profile. However MLE – quadratic does not provide any improvement, and sometimes it is the worst of the three methods. For example, the quench front location as predicted by the MLE quadratic is the highest as compared to MLE linear and MAP.

MLE quadratic is not estimated when number of input model parameters is greater than two. Computational complexity for MLE quadratic parameter estimation increases sharply with the number of input model parameters. Derivation for the case of three or more input model parameters is not shown explicitly in Chapter 2 but is a straightforward extension of the case with two input model parameters.

Figure 4.13 (input model ID 3, ID 4 and ID 1, see Table 4.3) shows the temperature comparison when three parameters are considered together. It shows a significant improvement for MLE linear and MAP prediction, particularly in the time interval

between the Peak Cladding Temperature and quenching, for all three axial levels. The plot for $z = 2225\ mm$ shows that the base code quench time estimate is higher than either from MLE linear or MAP. The MAP estimate is seen to be better than the MLE estimate. Similar conclusions can be drawn from the remaining plots in Figure 4.13. The larger standard deviation of the single phase vapor to wall HTC multiplier implies that the confidence in the value of its multiplier is weak. It is expected that a larger set of independent observations (Y_n) would alleviate this problem of a larger standard deviation.

Figure 4.14 (input model ID 3, ID 4 and ID 2, see Table 4.3) shows significant improvement over the base code results. The plot for $z = 1680\ mm$ in the figure shows a clear improvement in the quench time prediction as obtained from MLE linear and MAP estimate.

Figure 4.16 shows results when four parameters are considered together. The corrected (calibrated) code calculations show a remarkable similarity to the experimental data. The same inference can be deducted from the plots for error norms in Figure 4.19. These are the plots for error norms obtained from different methodologies for specific sets of parameters considered. Error from each method is defined as the difference in experimental data and prediction from respective method.

4.4.1 Dispersed flow film boiling HTC and departure from nucleate boiling critical heat flux

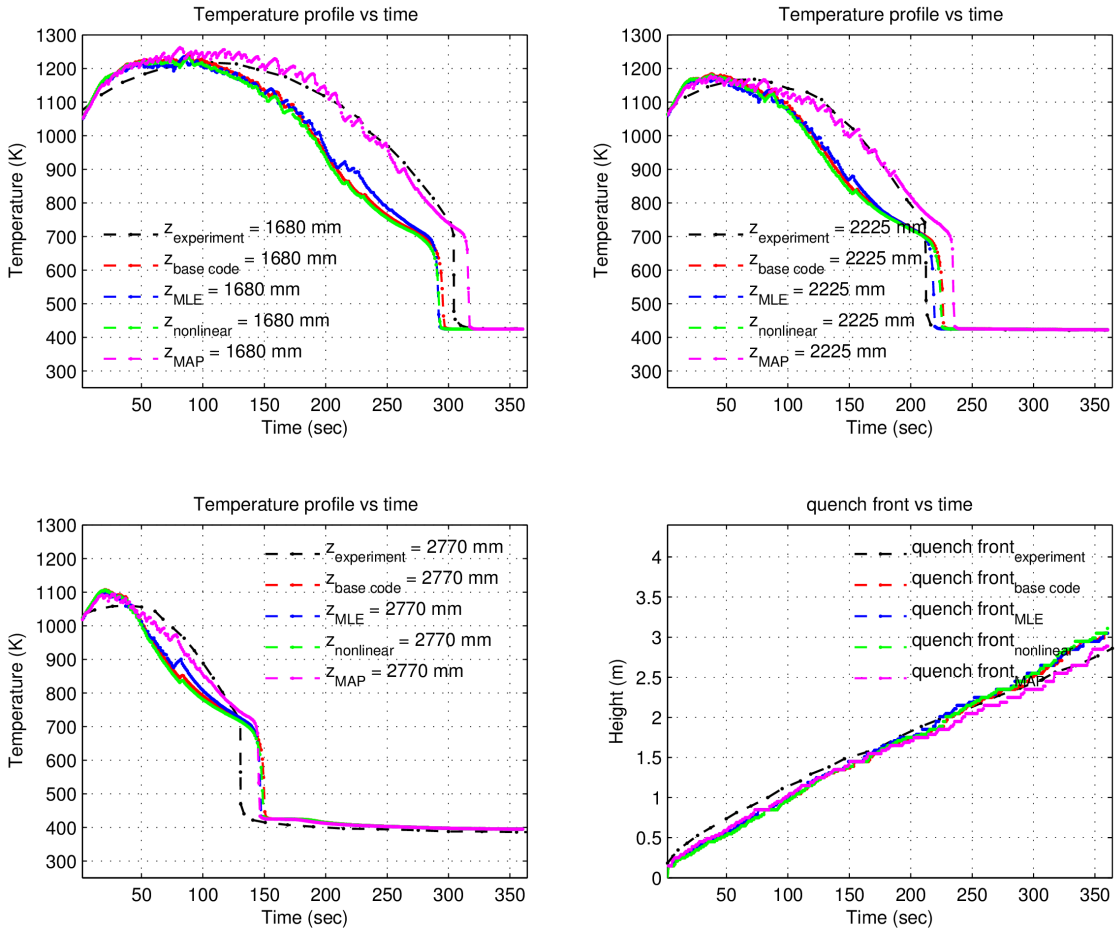

Figure 4.8 Comparison of calibrated results (dispersed flow film boiling HTC and departure from nucleate boiling critical heat flux)

Table 4.8 Comparison of calibrated results (Dispersed flow film boiling HTC and Departure from nucleate boiling critical heat flux)

	Mean			Variance		
	MLE linear	MLE quadratic	MAP	MLE linear	MLE quadratic	MAP
Dispersed flow film boiling HTC	1.28	1.08	1.48	20.6	1.1	1.68
Departure from nucleate boiling critical heat flux	1.63	0.97	4.26	>> 1	13.4	1.47

4.4.2 Dispersed flow film boiling HTC and interfacial drag coefficient

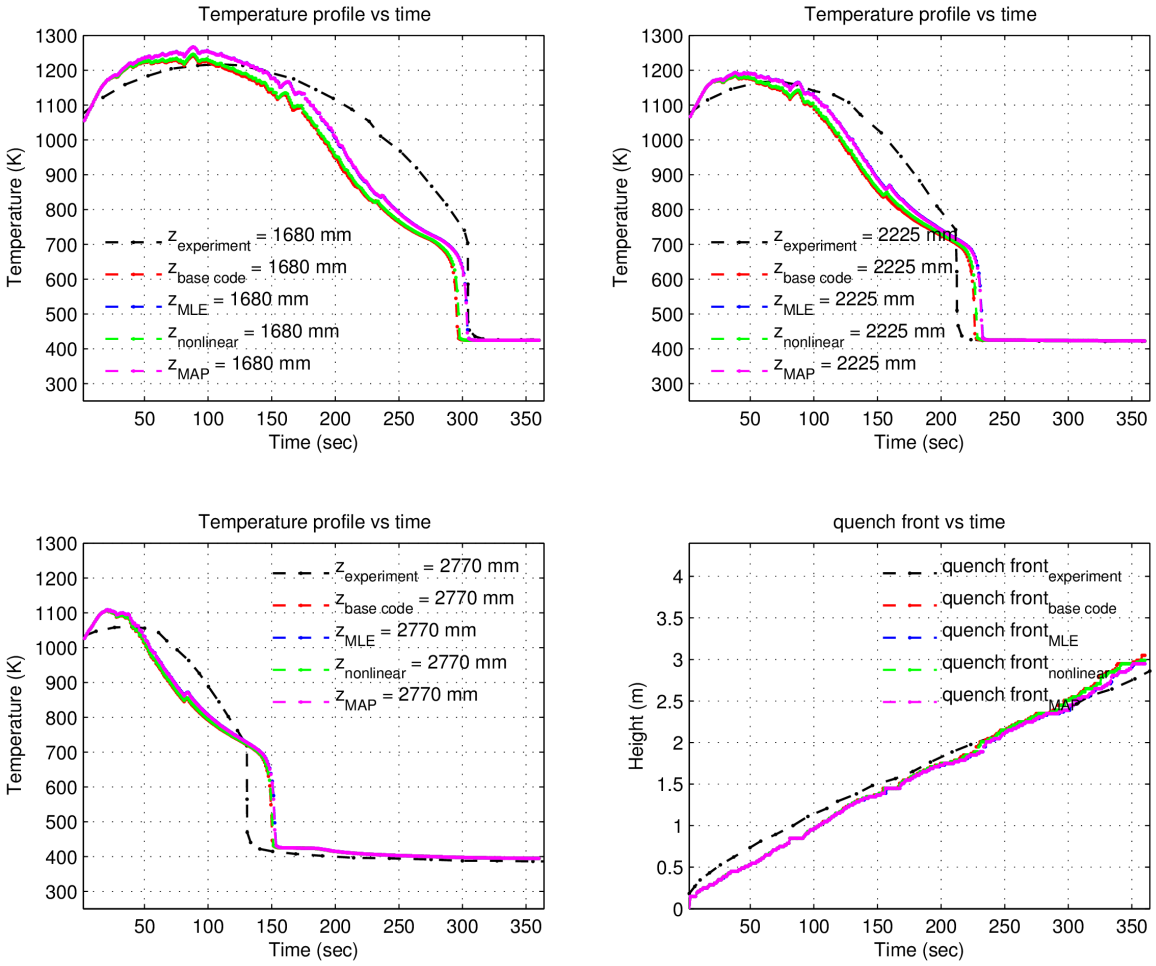

Figure 4.9 Comparison of calibrated results (dispersed flow film boiling HTC and interfacial drag coefficient)

Table 4.9 Comparison of calibrated results (dispersed flow film boiling HTC and interfacial drag coefficient)

	Mean			Variance		
	MLE linear	MLE quadratic	MAP	MLE linear	MLE quadratic	MAP
Dispersed flow film boiling HTC	0.93	.98	0.95	3.8	1.75	3.0
Interfacial drag coefficient	1.26	1.03	1.30	4.5	9.0	2.7

4.4.3 Interfacial drag coefficient and single phase vapor to wall HTC

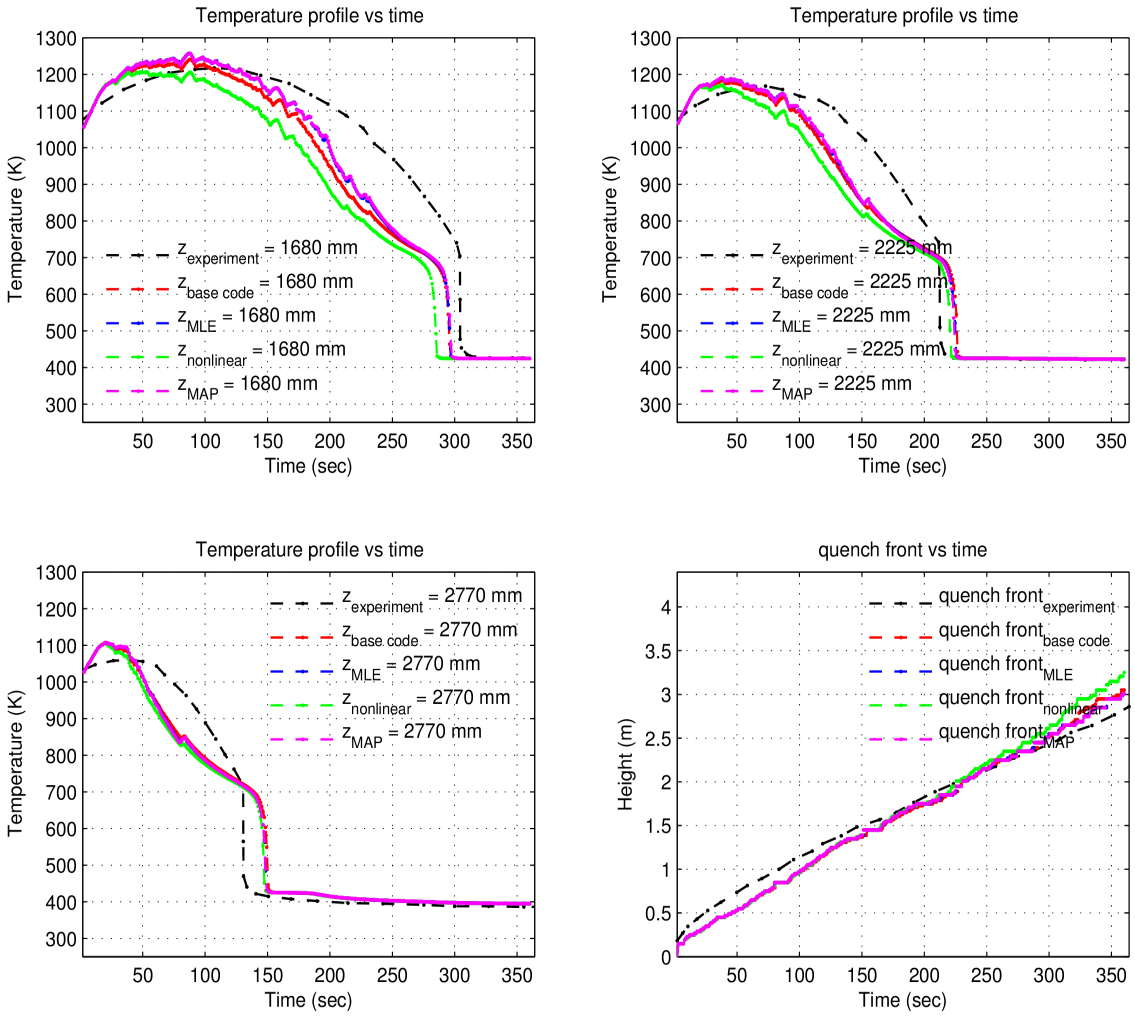

Figure 4.10 Comparison of calibrated results (interfacial drag coefficient and single phase vapor to wall HTC)

Table 4.10 Comparison of calibrated results (Interfacial drag coefficient and Single phase vapor to wall HTC)

	Mean			Variance		
	MLE linear	MLE quadratic	MAP	MLE linear	MLE quadratic	MAP
Interfacial drag coefficient	1.5	0.7	1.53	2.4	3.6	1.7
Single phase vapor to wall HTC	1.31	1.05	1.37	16.4	14.8	14.7

4.4.4 Interfacial drag coefficient and departure from nucleate boiling critical heat flux

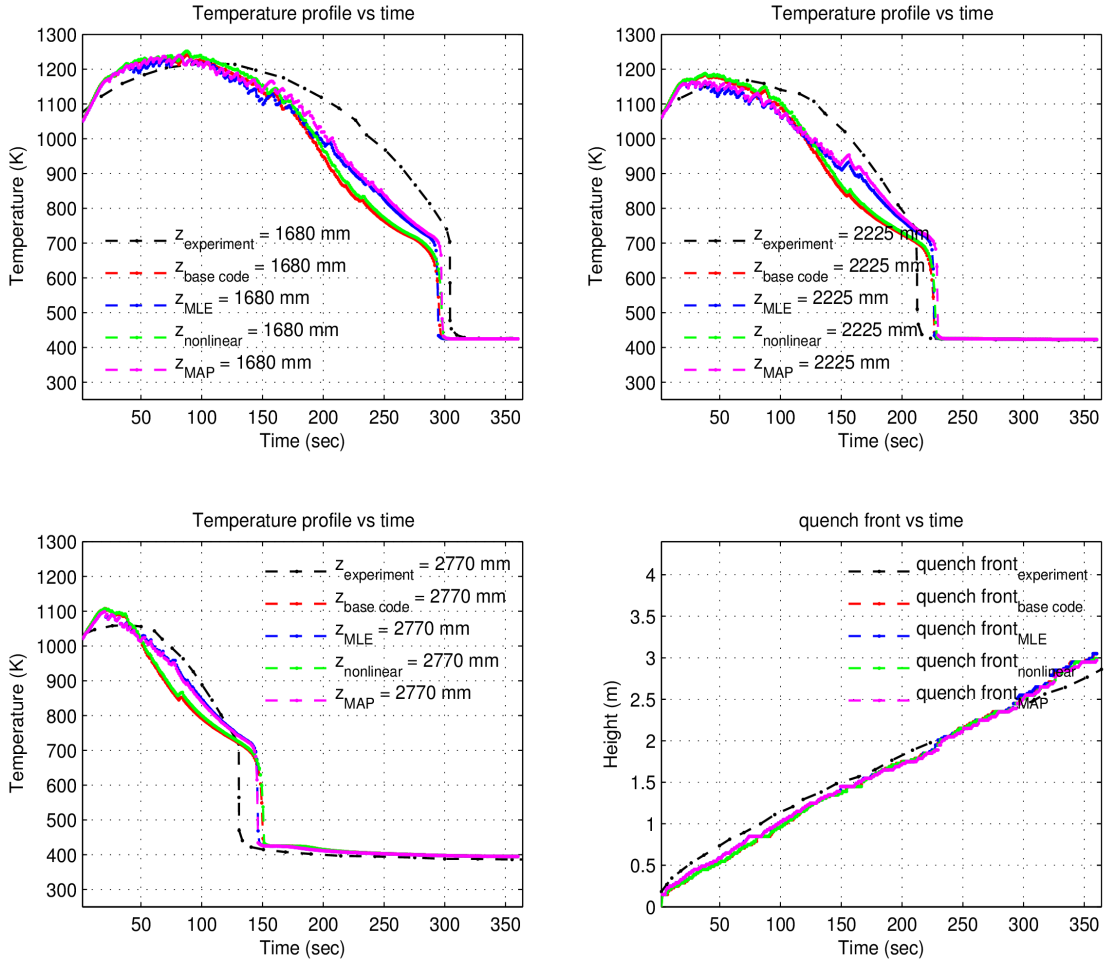

Figure 4.11 Comparison of calibrated results (interfacial drag coefficient and departure from nucleate boiling critical heat flux)

Table 4.11 Comparison of calibrated results (interfacial drag coefficient and departure from nucleate boiling critical heat flux)

	Mean			Variance		
	MLE linear	MLE quadratic	MAP	MLE linear	MLE quadratic	MAP
Interfacial drag (dispersed flow film boiling) coefficient	0.48	1.06	0.48	4.1	6.0	4.3
Departure from nucleate boiling critical heat flux	4.26	1.03	4.18	8.6	12.4	7.3

4.4.5 Single phase vapor to wall HTC, dispersed flow film boiling HTC and interfacial drag coefficient

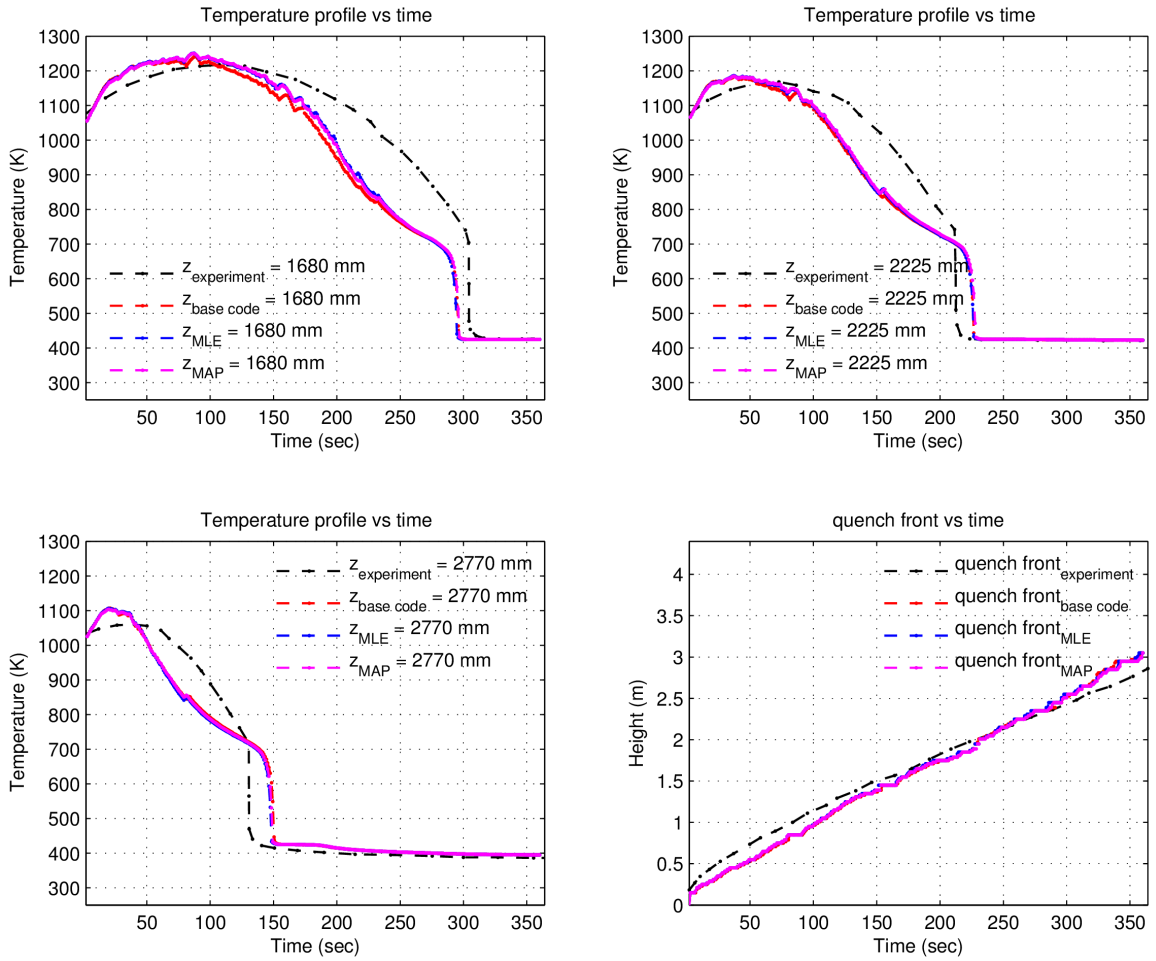

Figure 4.12 Comparison of calibrated results (single phase vapor to wall HTC, dispersed flow film boiling HTC and interfacial drag coefficient)

Table 4.12 Comparison of calibrated results (single phase vapor to wall HTC, dispersed flow film boiling HTC and interfacial drag coefficient)

	Mean		Variance	
	MLE linear	MAP	MLE linear	MAP
Single phase vapor to wall heat transfer coefficient	3.27	4.28	16.4	15.3
Dispersed flow film boiling heat transfer coefficient	0.42	0.29	1.00	1.02
Interfacial drag (dispersed flow film boiling) coefficient	1.02	0.86	2.46	2.56

4.4.6 Single phase vapor to wall HTC, departure from nuclear boiling critical heat flux and dispersed flow film boiling HTC

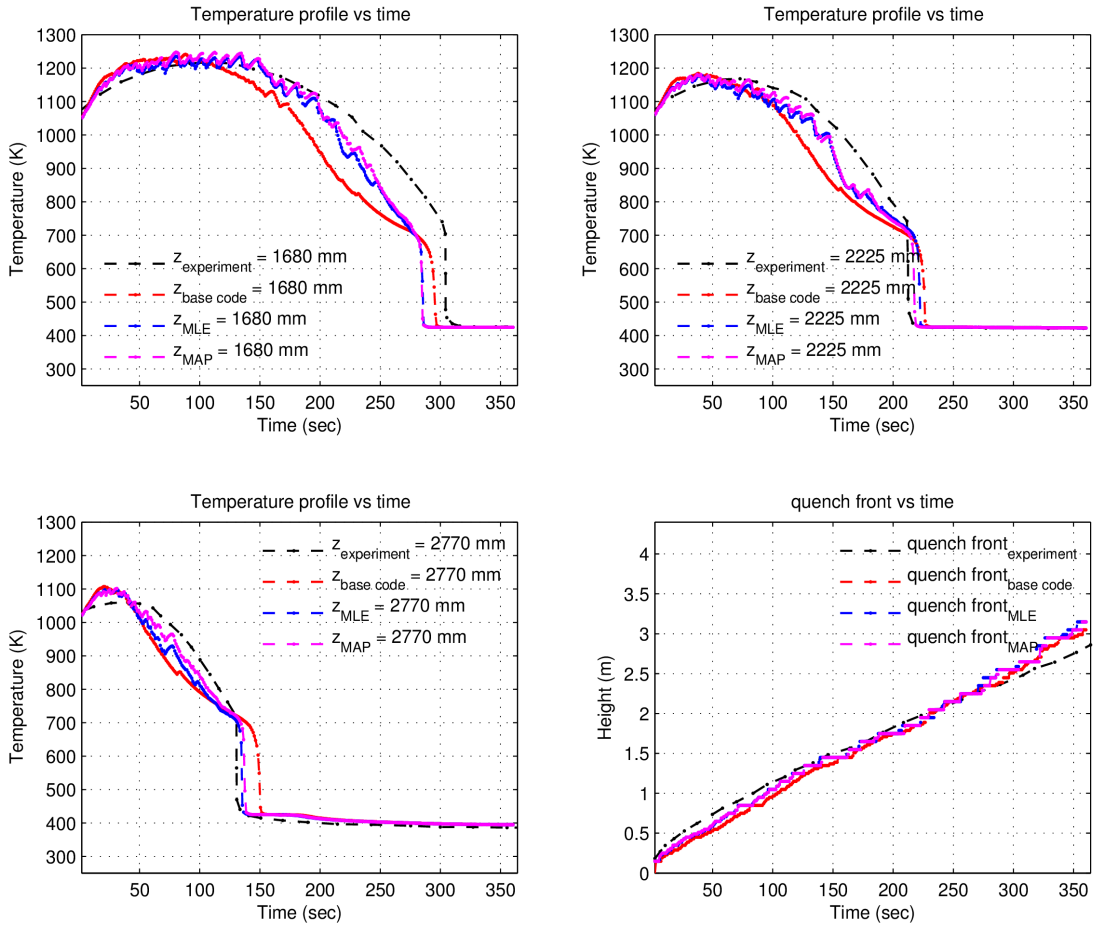

Figure 4.13 Comparison of calibrated results (single phase vapor to wall HTC, departure from nuclear boiling critical heat flux and dispersed flow film boiling HTC)

Table 4.13 Comparison of calibrated results (single phase vapor to wall HTC, departure from nuclear boiling critical heat flux and dispersed flow film boiling HTC)

	Mean		Variance	
	MLE linear	MAP	MLE linear	MAP
Single phase vapor to wall heat transfer coefficient	3.46	3.87	>> 0	>> 0
Departure from nucleate boiling critical heat flux	2.48	2.80	4.95	1.15
Dispersed flow film boiling heat transfer coefficient	0.75	0.71	1.00	2.36

4.4.7 Single phase vapor to wall HTC, departure from nuclear boiling critical heat flux and interfacial drag

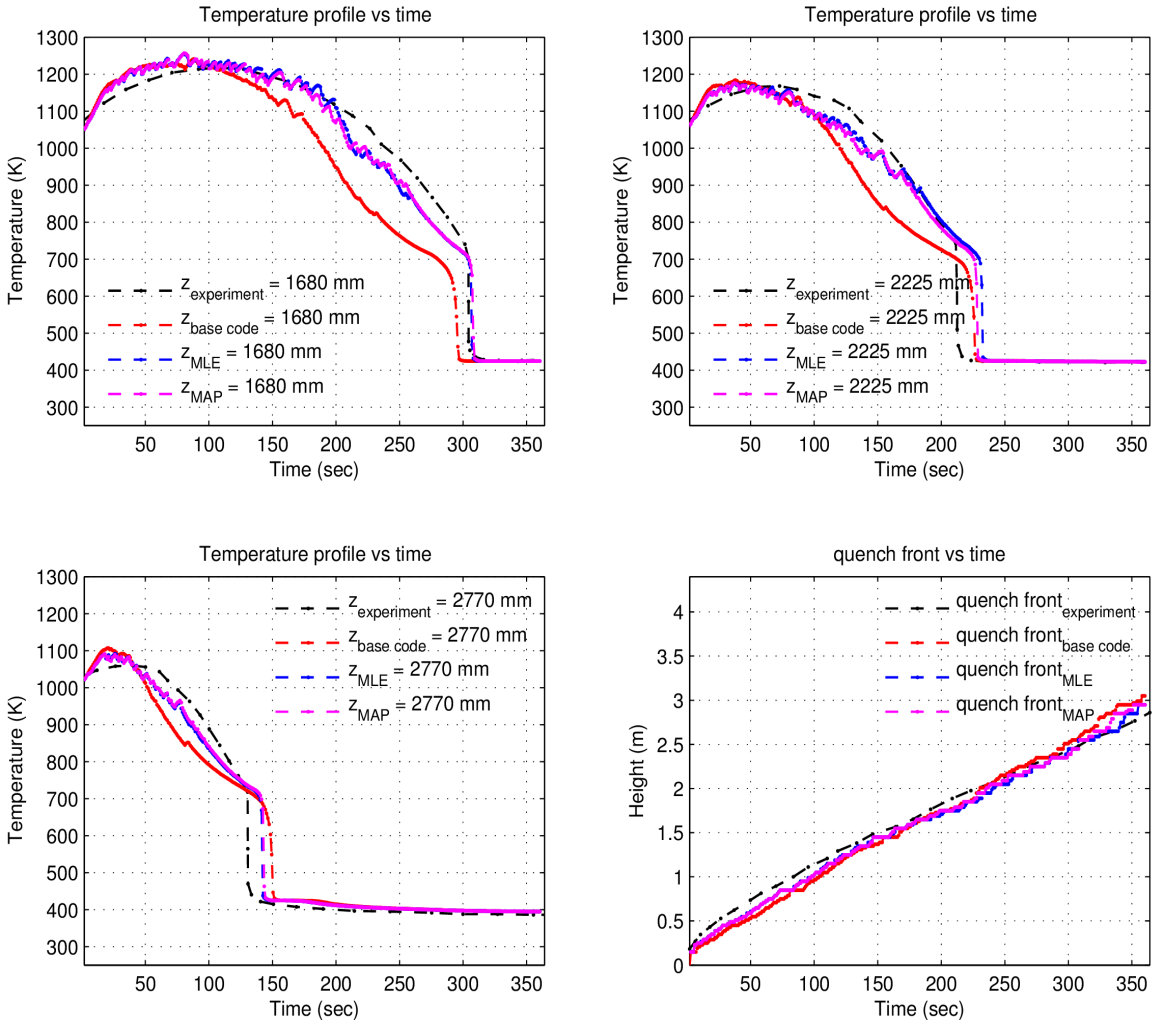

Figure 4.14 Comparison of calibrated results (single phase vapor to wall HTC, departure from nuclear boiling critical heat flux and interfacial drag)

Table 4.14 Comparison of calibrated results (single phase vapor to wall HTC, departure from nuclear boiling critical heat flux and interfacial drag)

	Mean		Variance	
	MLE linear	MAP	MLE linear	MAP
Single phase vapor to wall heat transfer coefficient	1.52	1.43	4.05	3.32
Departure from nucleate boiling critical heat flux	4.60	4.48	1.52	1.43
Interfacial drag (dispersed flow film boiling) coefficient	0.69	0.68	2.09	1.97

4.4.8 Departure from nuclear boiling critical heat flux, dispersed flow film boiling HTC and interfacial drag

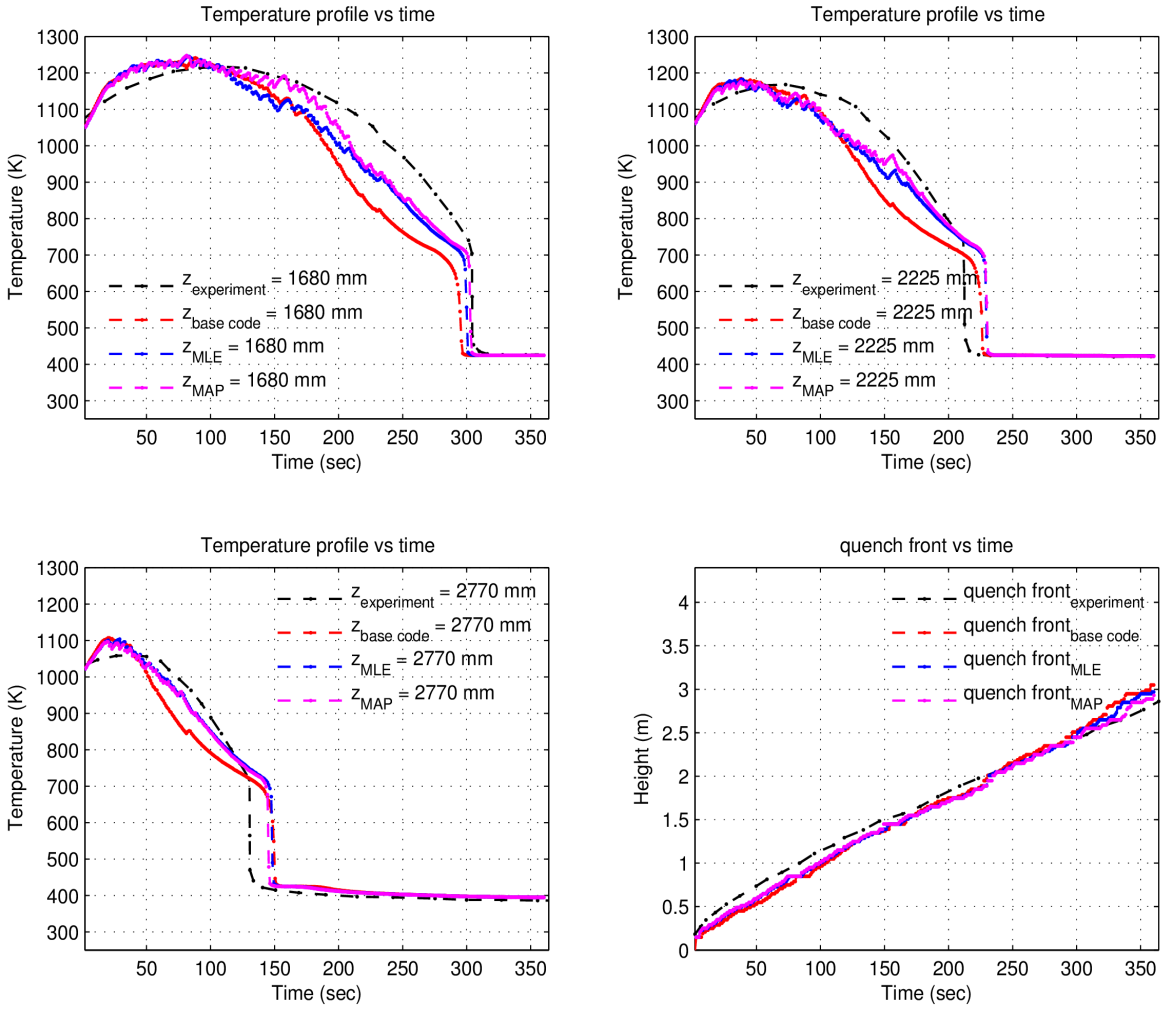

Figure 4.15 Comparison of calibrated results (departure from nuclear boiling critical heat flux, dispersed flow film boiling HTC and interfacial drag)

Table 4.15 Comparison of calibrated results (departure from nuclear boiling critical heat flux, dispersed flow film boiling HTC and interfacial drag)

	Mean		Variance	
	MLE linear	MAP	MLE linear	MAP
Departure from nucleate boiling critical heat flux	3.77	4.42	7.3	2.36
Dispersed flow film boiling heat transfer coefficient	0.86	1.1	1.09	1.49
Interfacial drag (dispersed flow film boiling) coefficient	0.48	0.61	3.38	2.61

4.4.9 Single phase vapor to wall HTC, departure from nuclear boiling critical heat flux, dispersed flow film boiling HTC and interfacial drag

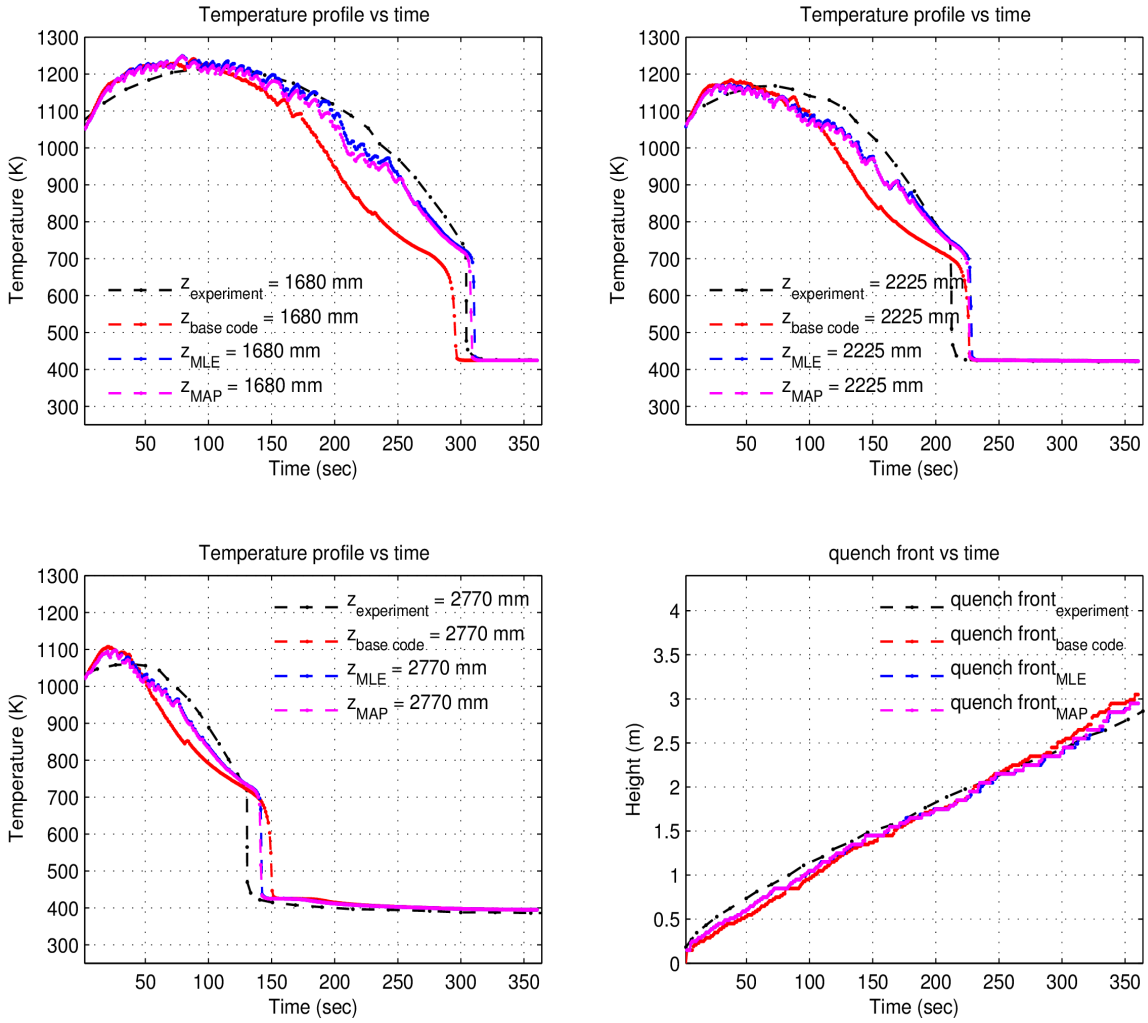

Figure 4.16 Comparison of calibrated results (single phase vapor to wall HTC, departure from nuclear boiling critical heat flux , dispersed flow film boiling HTC and interfacial drag)

Table 4.16 Comparison of calibrated results (single phase vapor to wall HTC, departure from nuclear boiling critical heat flux , dispersed flow film boiling HTC and interfacial drag)

	Mean		Variance	
	MLE linear	MAP	MLE linear	MAP
Single phase vapor to wall heat transfer coefficient	3.66	4.29	4.05	2.48
Departure from nucleate boiling critical heat flux	3.79	3.90	1.66	1.40
Dispersed flow film boiling heat transfer coefficient	0.38	0.34	1.00	1.03
Interfacial drag (dispersed flow film boiling) coefficient	0.46	0.48	2.11	1.82

4.4.10 Error Norm Comparison

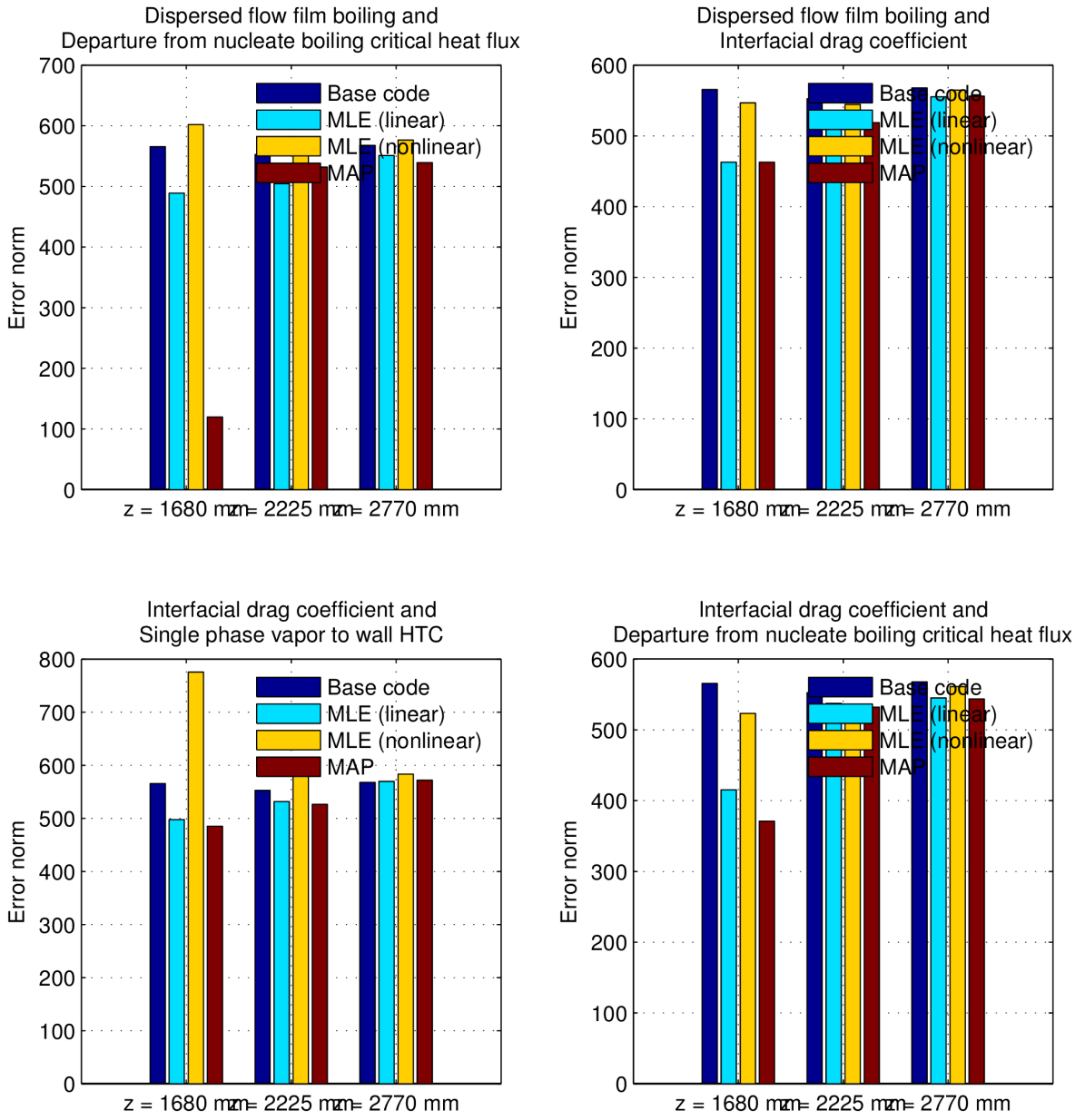

Figure 4.17 Comparison of error norm (two parameters)

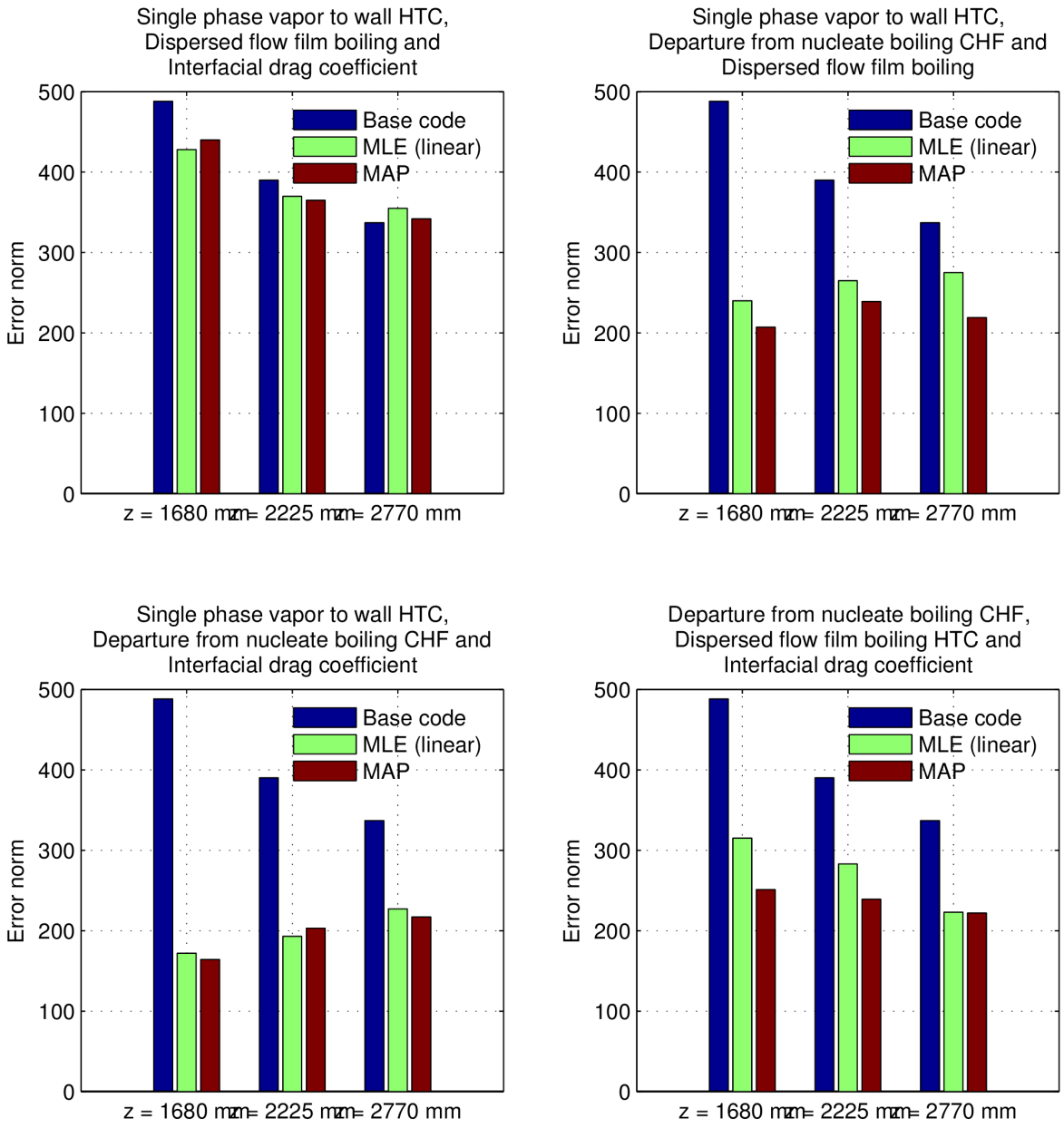

Figure 4.18 Comparison of error norm (three parameters)

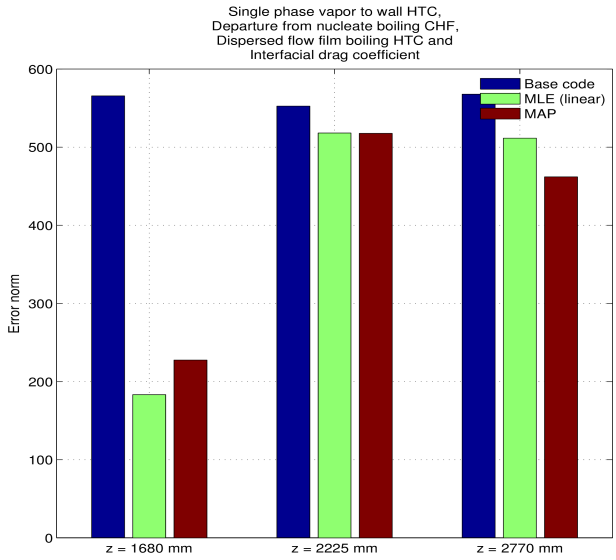

Figure 4.19 Comparison of error norm (four parameters)

The comparisons shown in Figure 4.8 through 4.19 show that the maximum likelihood estimates (MLE) and maximum a posterior (MAP) estimates improve the code prediction. In most cases, the calibrated results from MAP estimates with the assumption of normality on the mean and inverse gamma on the variance (inverse Wishart for multivariate data) are found to reduce the error norm relative to maximum likelihood estimates (MLE) calibrated results. The methodology was validates using test that was not used to generate calibrated parameters (Section 4.5).

4.5 Validation using FEBA Test 234

Validation of the input parameter estimates (physical models multipliers) obtained in the preceding section is carried out using FEBA Test 234. It has the same geometrical configuration as FEBA Test 216 described in Chapter 3 except for the lack of the grid spacers at axial midplane. This is modeled in TRACE by adjusting the cross sectional flow area at the corresponding axial locations. In addition, the boundary conditions (back pressure, inlet water velocity, inlet water temperature) are changed to those corresponding to Test 216. Table 4.17 lists the operating conditions.

Table 4.17 System operating conditions for FEBA Test 234

Fluid inlet velocity (cm/s)	3.6
System outlet pressure (bar)	2.02
Feed water temperature (°C) (0-30 s)	40
Feed water temperature (°C) (30s-end)	40
Bundle power (kW) at 0 s	200
Bundle power (kW) during transient	120% ANS standard

MAP estimate of input parameters (physical model multipliers) calculated in Section 4.4.7 are used to calibrate the base code results. Calibrated (MAP) temperature profile is compared with uncalibrated (nominal) temperature profile in Figure 4.20 for $z = 2225\ mm$ and $z = 2770\ mm$. The figure also shows experimental data and lower bound and upper bounds of calculated temperature profile using multipliers at 1σ from the mean.

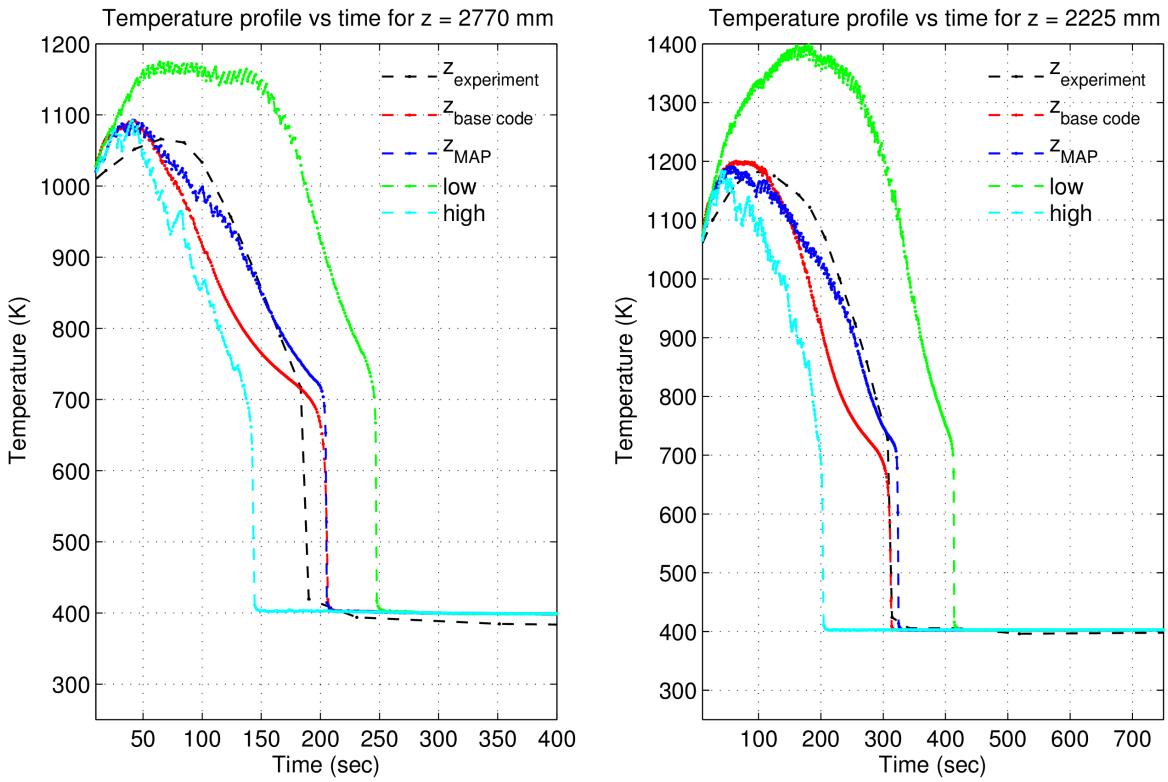

Figure 4.20 Comparison of calibrated results for FEBA Test 234

Figure 4.20 shows that the temperature profile obtained after parameter calibration using Maximum a Posteriori (MAP) estimates matches the experimental data better than base (uncalibrated) code results. A comparison of the error norm between the two cases (base code and MAP estimate) is shown in Figure 4.21. Note that the norm is taken with respect to the experimental data in both cases (base code and MAP estimate). In addition, the upper and lower bounds of temperature estimates envelope most of the experimental data, as expected.

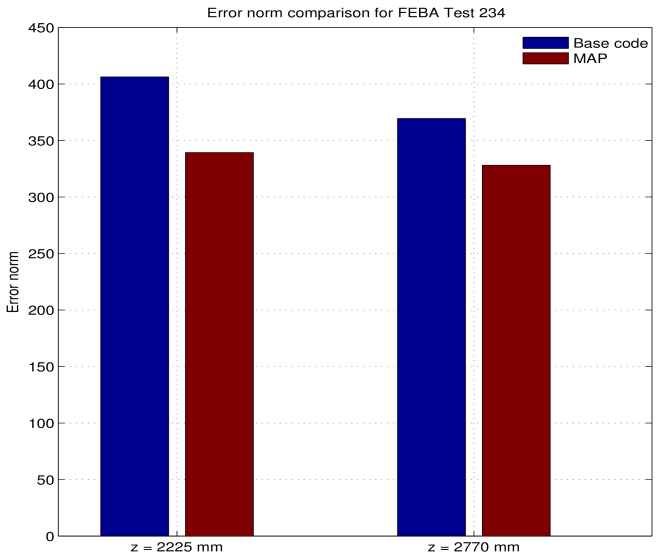

Figure 4.21 Comparison of error norm for FEBA Test 234

4.6 Discussion

The Y_n that form the observables are *not* completely independent since they are axial temperature readings from the same levels (in most cases) and only differ in time. In case of time series, it is intuitively obvious that the next time observable depends on the previous one. This makes adjacent columns in the sensitivity matrix *identical*. This is to say that the sensitivity dY_n/dp_i is approximately the same for every reading n for each input parameter. Although it does not pose a problem while forming the system of linear equations (in the case of MLE linear and MAP), it poses a serious threat while solving for these system of linear equations in Eq. (2.18) and Eq. (2.37) respectively, because of linearly dependent columns. In other words, non-independent readings Y_n cause a bad conditioning of the coefficient matrix. In such case, even though a unique solution can be calculated, it is highly sensitive to perturbations. The result is that a different mean vector $\underline{b}$ is observed for each specific set of experimental readings Y_n. Ideally, it is desirable that the mean vectors are close to each other regardless of the experimental reading

considered, but the algorithms developed in Chapter 2 can not guarantee that. Robustness (e.g. repeatability, convergence, user effect independence) of the algorithm depends on the experimental data and input parameters considered. It is important to remember that the derivation carried out in Chapter 2 assumed independent observations thus eliminating the risk outlined above. However, considering available experimental data, particularly when taking data from a transient, the assumptions of independent observations is weak.

While estimating the mean and variance of input parameters multipliers under a quadratic model assumption in non-Bayesian framework (MLE quadratic), a series of double approximations are used. It is helpful to remember that each EM algorithm starts with an initial guess. Inside each step of EM algorithm, a system of equations are solved to determine the mean vector of pseudo parameters. This system of equations is linear in case of MLE linear and MAP but are non-linear in case of MLE quadratic. For the latter it is vital that Newton iteration starts at a point close to the correct solution. Since the algorithm implemented for the MLE quadratic algorithm uses starting values provided to the EM, care must be taken to monitor for diverging solution. Since more than one set of input parameter values p_i can yield a minimum, chances are that the algorithm converges to a local minimum instead of a global minimum. There are no robust checks against this behavior. A practical solution is that a series of different starting values are considered and monitored for convergence to the same solution while using quadratic model in non-Bayesian framework (MLE quadratic).

Under the Bayesian framework there are a couple of ways to predict the covariance matrix besides using weights given in Eq. (2.42). Remember that the two quantities to be weighted in Eq. (2.42) are obtained as a result of direct maximization in Eq. (2.39) and as an approximation to posterior covariance in Eq. (2.41). Both of these matrices are formulated using prior information, albeit interpreted differently. If the user chooses not

to incorporate any prior information while constructing the covariance matrix then Eq. (2.14) could be substituted for Eq. (2.42). Hence this effectively forms a hybrid approach where no prior information about the observables (Y_n) is used while determining the covariance matrix. However, prior assumptions about the unobserved mean vector ($\underline{\mu}_p$) will still be used. (Note that results from this hybrid approach are not calculated in this thesis.)

As mentioned earlier, violation of independence assumption in case of a transient time series experimental data makes the inverse problem unstable. In addition, the current methodology explicitly focuses on input parameter calibration and does not take *model inadequacy* into account. *Model inadequacy* refers to the inherent inability of the model to predict the experiment correctly. It should be corrected using additional function, also known as *bias correction.* It is not unreasonable to imagine that TRACE code may not be able to accurately predict certain experimental data regardless of the accuracy of each parameter in the code, i.e. inadequacy of the model (Taylor series - linear or quadratic) itself must be addressed using *bias*. For these reasons the input parameter estimations (calibration) can fail to accurately reproduce the experimental data in some cases. This could also be viewed as follows. Estimating (calibrating) the parameters of a problem (aiming to exactly reproduce experiment) knowing that no such set of parameter value exists is futile. It is the very nature of inverse problems. This is in contrast with Section 4.3 where we know with a *certainty* that there exists a set of input parameter values such that the observed experimental (synthetic) data are reproduced exactly. Not surprisingly, an excellent agreement is observed between true and predicted (MLE/MAP) values in Section 4.3.

It may be surprising that the prediction of the statistical properties of the multipliers differ when different combinations of input models are taken. The reasons are due to the assumptions used by the MLE and MAP method derived in this thesis. It is helpful to

remember that the most complete model (linear or quadratic) incorporates all possible input parameters in Eq. (2.7) and Eq. (2.21) and as such, may include hundreds or even thousands of parameters. When a limited number of input parameters are considered, it is implied that other input parameters are nominally correct and their calibration is not needed. (Note that this has nothing to do with sensitivity at this point, which may be large or small, accurate or not.) Moreover, inverse problems are ill-posed [57]. Ill-posedness of a problem refers to the lack of one of the following – existence, uniqueness and stability of the solution. Inferring statistical parameters from a partial or incomplete information like a limited dataset (e.g. finite number of code result and experimental data) in statistical estimation is an example of inverse problem.

In addition, the estimation procedure in this thesis is developed such that the estimation of unknown unobserved parameters (physical multipliers and their standard deviation) is carried out on the basis of some tractable observation. This inference procedure could be thought of as introducing additional level of ill-posedness. The sensitivity of such inferred estimates is usually significant to small perturbations in dataset [58][59]. Table 4.5 and Table 4.7 in Section 4.3 show estimated values of corresponding physical multipliers as a function of the number of experimental observations (dataset) considered. For example in Table 4.7, the mean values of the physical multiplier for interfacial drag coefficient fluctuate around the true value of 0.6 for different number of experimental observations. On the other hand, the multiplier for single phase vapor to liquid HTC seems to converge to true value of 1.5 with increasing number of experimental observations. However as seen in the table, the estimate suddenly drops to attain a value of 1.33 for a larger dataset (n=66, Table 4.7). Since the quality of the estimates depend on the dataset (number of experimental observation), it is imperative that the assumption of independence among experimental observations hold true for more accurate results.

In some cases, MLE performs better than MAP. However, it should be noted that an

assumption of normality on the prior mean was used for MAP derivation. The user must choose wisely the kind of assumptions and number of parameters to consider before implementing this algorithm. It is also useful to note that the derivation so far has been agnostic to the physics of the parameters. It is recommended that the process of forming the most important input parameters and their ranking be carried out prior to performing inverse uncertainty quantification, i.e. individual physics of the parameters must be considered before performing inverse and forward uncertainty quantification for a given class of problem. For example, in the context of the reflooding experiment given in Chapter 3, the dispersed flow film boiling HTC (input model ID 1, see Table 4.3) was found to have the highest impact on the PCT as is evident from the sensitivity calculations. However, declaring this parameter to be the most important *universally* i.e. for all future simulations is incorrect.

For example, consider a case of water under high pressure conditions flowing through adiabatic pipe. It is easy to see that the water inside the pipe will not undergo phase change at all, let alone traverse into the dispersed flow film boiling flow regime. Thus, input parameter ID 1 (see Table 4.3) has no impact on such experiment. If the user was to choose input parameter ID 1 as one of the relevant input parameters , we can trace the consequences by going through the derivation steps in Chapter 2. First, the entries in the sensitivity matrix with respect to the input parameter ID 1 will be zero. This will introduce a series of linearly dependent columns in the system of equations in Eq. (2.18) (for MLE linear) and hence make it intractable. Thus care must be taken before performing inverse uncertainty quantification with this algorithm. It is recommended the methodology proposed here not be used like a black-box.

As discussed in Section 4.4 the algorithms developed here employ several assumptions and hence the results and their interpretation are only as good as the assumptions themselves. Moreover, inverse problems are highly sensitive to the input data. It is

important to understand additional details about the algorithm that specifically deals with choosing the input data (experimental responses) for calibration purposes. Improper consideration of seemingly relevant input data (experimental responses) may yield unreasonable inference. It may even lead to a worse predictions, failing to provide any improvement from the nominal case. Calibration of physical model parameters using experimental responses that are significantly different than their respective computed values would significantly modify the resulting adjusted parameters (calibrated physical model). Such adjustments are the primary causes of obtaining unreasonable inference. For example consider Figure 3.6. Assume code calculated result and experimental data are considered for $z = 590\ mm$ at $t = 440\ sec$ as the input for the MLE/MAP algorithm. We can see that the code predicted temperature for the considered axial location and that time corresponds to the steady quenched value while the observed experimental reading has not yet reached quenched state. Now, if this dataset was to be considered for the MLE/MAP algorithm (under the assumption of a linear model for MAP estimates), one would skew model parameter estimates (physical model multipliers p_i and their variance). A direct way to observe this effect is to check the corresponding entry in the sensitivity matrix. A large dY_n for a given $\Delta\alpha$ yields a very high estimate of corresponding sensitivity ($dY/d\alpha$), thus skewing subsequent calculations for the mean vector $\underline{b}$. It is important to note that the approach developed here considers the nominal values of the model parameters ($p_i = 1.0$) to be *acceptable* and any further calibration done (with MLE or MAP) is meant to improve their accuracy, not to fix the physical model itself or make it model new physics. Thus, specific quantitative measures should be implemented as a part of the overall inverse uncertainty quantification methodology with special provisions to reject such dataset (e.g. unreasonably large deviation between code calculated result and experimental observation). It is helpful to understand that the algorithm itself does not require the rejection of such dataset. In case of large datasets (large experimental database) a probabilistic treatment of all available datasets make such anomaly less serious, especially if isolated to a few data points ($Y_{exp,n}$). Nevertheless such anomalous datasets should be rejected. This can be thought of as preconditioning the matrix (improving condition number) before performing large linear operations.

CHAPTER 5 CONCLUSION

This research presents a systematic mathematical methodology to calculate uncertainty distribution of TRACE code input parameters. Historically, the probability density function (p.d.f.) of physical models and its statistical parameters (mean, variance etc.) have been derived from 'expert judgment'. The methodology developed in this thesis is aimed to replace such ad-hoc approach with rigorous mathematical formulation.

TRACE code physical model parameters (e.g. heat transfer coefficient) are not observed directly. Thus it is impossible to directly collect data points and infer statistical properties of the underlying distribution. Hence, the relationship between the observables (difference between experimental measurement and TRACE code prediction) and TRACE code model parameters are expressed either as a linear model or a quadratic model. Normally during an inference process, the user collects readings and calculates statistical properties about the collected sample. However, in the context of this thesis the goal is not to deduce statistical conclusions about the sample observed (observables). Rather, by utilizing the model (linear or quadratic), statistical properties of the physical models of TRACE code parameters are calculated. This constitutes an inverse problem.

Methodology based on estimating the statistical parameters (mean, variance etc.) of the physical models of TRACE code parameter by maximizing the (i) likelihood function and (ii) posterior density are discussed. In both cases, Expectation-Maximization (EM) algorithm is implemented to calculate the converged values of the statistical parameters.

Maximum Likelihood Estimates (MLE) are calculated under a linear and a quadratic model assumption. This is carried out under a non-Bayesian framework where no information about the prior is assumed. Simply put, the statistical parameters of physical models are assumed some unknown constants which we seek. The methodology takes code sensitivity matrix and experimental results from Separate Effect Test (SET) facilities as input. A constraint is put on the physical multipliers such that attainable values by

these random variables are always positive. This is a reasonable constraint and suggests that the physical multipliers follow a log-normal distribution.

The framework of parameter estimation is extended further wherein the user can accommodate any known prior information about the observables into the calculation. A novel methodology based on Maximum a Posteriori (MAP) estimate developed here enables the user to estimate posterior mode of statistical parameters of the physical models. Though the methodology developed in this thesis is aimed specifically at calibrating physical models in thermal-hydraulics codes (such as RELAP5, TRACE), the methodology is generic and can be readily applied to other models and codes.

Comparisons presented in this thesis suggest that the assumption of normality on the mean and inverse gamma on the variance of the observables with properly chosen hyperparameters yield MAP estimates that produce consistently better calibrated results when compared to those obtained from Maximum Likelihood Estimate. Here, 'better' is objectively used on the basis of comparison of error norm between the two methodology (See Chapter 4 for results). The novel method developed in Section 2.5 is observed to yield consistently lower error norm as seen in Section 4.4.10. It is also validated using a different Test problem in Section 4.5 (other than the one which was used to calibrate the parameters). A series of tabulated results are presented in Chapter 4 showing a comparison of the calibrated statistical properties of the physical multipliers for some important TRACE code parameters.

It is important to note that care must be taken while deriving and interpreting calibrated results from the algorithm developed here. The calculated estimates are only as good as the assumptions that go into the algorithm. Datasets that severely violate the assumptions should be rejected. Haphazard incorporation of experimental data violating assumptions of independence may produce flawed estimates. The assumption of linearity is a weak one and care must be confirmed while implementing the methodology derived here to problems where observables show strong non-linear variation with respect to TRACE

code parameters. Moreover, the EM algorithm converges to the nearest stationary point of the log-posterior. Convergence to local maxima are frequently observed which might lead to misleading conclusions. A remedy to this is to consider several initial values during the first E-Step. As a practical solution to this problem, it is recommended to vary the weights to observe the final covariance matrix (Section 2.5.4) i.e. to compare the covariance matrices obtained from Section 2.5.2 and Section 2.5.3.

Recommendations for Future Work

As seen from the results in the preceding discussions, calibrating the parameters using MAP estimates consistently improve code predictions. However, they still fail to accurately reproduce experimental data in most cases. To achieve this goal, the methodology developed here can be further refined by adding a model correction term to the Taylor series expansion, called *bias correction*, which quantifies model inadequacy. Under this paradigm, it is accepted that the base model (TRACE code) lacks sufficient accuracy and thus introduction of an additional analytical term to address this discrepancy is warranted. The author recommends further research on this approach, particularly for a complicated TH code like TRACE, which sometimes does not reproduce experimental data to satisfactory accuracy. Though there have been developments in determining the exact analytical form of this bias correction expression for simple analytical models (linear or otherwise), the same can not said when the responses are derived from a black-box computational code like TRACE.

APPENDIX A. EXPECTATION-MAXIMIZATION ALGORITHM

Let $\underline{Y}$ be the random vector corresponding to the observed data $\underline{y}$, with probability density function $g(\underline{y}; \underline{\theta})$ where $\underline{\theta}$ is a vector of unknown parameters. The EM algorithm is a broadly applicable algorithm that provides an iterative procedure for computing maximum likelihood estimate (MLE) in situations where, but for the absence of some additional data, ML estimation would be straightforward. Hence in this context, the observed data vector $\underline{y}$ is viewed as being incomplete and is regarded as an observable function of the so-called complete data. The notion of 'complete data' includes conventional sense of missing data, but it also applies to situations where the complete data represent what would be available from some hypothetical experiment. In the latter case, the complete data may contain some variables that are never observed in a data sense. Within this framework, we let $\underline{x}$ denote the vector containing the augmented or so-called complete data, and we let $\underline{z}$ denote the vector containing the additional data, referred to as the unobservable or missing data.

In many cases, even when a problem does not at first appear to be an incomplete data one, computation of the MLE is often greatly facilitated by artificially formulating it to be as such. This is because the EM algorithm exploits the reduced complexity of ML estimation given the complete data. For many statistical problems the complete data likelihood has a nice form.

Let $g_c(\underline{x}; \underline{\theta})$ denote the p.d.f. of the random vector $\underline{X}$ corresponding to the complete-data vector $\underline{x}$. Then the complete-data log likelihood function that could be formed for $\underline{\theta}$ if $\underline{x}$ were fully observed is given by the following.

$$\log L_c(\underline{\theta}) = \log g_c(\underline{x}; \underline{\theta})$$

Formally, we have two samples spaces *X* and *Y* and a many-to-one mapping from *X* to *Y.* Instead of observing the complete-data vector $\underline{x}$ in *X*, we observe the incomplete-data

vector $\underline{y}$ in *Y*. It follows:

$$g(\underline{y};\underline{\theta})=\int_{X(\underline{y})} g_c(\underline{x};\underline{\theta})\,d\underline{x}$$

where $X(\underline{y})$ is the subset of *X* determined by the equation $\underline{y} = \underline{y}(\underline{x})$. The EM algorithm approaches the problem of solving the incomplete-data likelihood equation indirectly by proceeding iteratively in terms of the complete-data log likelihood function, $\log L_c(\underline{\theta})$. As it is unobserved, it is replaced by its conditional expectation given $\underline{y}$, using the current fit for $\underline{\theta}$. More specifically, let $\underline{\theta}^{(0)}$ be some initial value for ($\underline{\theta}$). Then on the first iteration, the E-step requires the calculation of the following.

$$Q(\underline{\theta};\underline{\theta}^{(0)})=E_{\underline{\theta}^{(0)}}[\log L_c(\underline{\theta})|\underline{y}]$$

The M-step requires the maximization of $Q(\underline{\theta}, \underline{\theta}^{(0)})$ with respect to $\underline{\theta}$ over the parameter space. That is, we choose $\underline{\theta}^{(1)}$ such that

$$Q(\underline{\theta}^{(1)};\underline{\theta}^{(0)})\geq Q(\underline{\theta};\underline{\theta}^{(0)})$$

for all $\underline{\theta}$. The E and M-steps are then carried out again, but this time with $\underline{\theta}^{(0)}$ replaced by the current fit $\underline{\theta}^{(1)}$. On the *(k+1)*th iteration, the E and M-steps are defined as follows.

E-Step. Calculate $Q(\underline{\theta}, \underline{\theta}^{(k)})$, where

$$Q(\underline{\theta};\underline{\theta}^{(k)})=E_{\underline{\theta}^{(k)}}[\log L_c(\underline{\theta})|\underline{y}]$$

M-Step. Choose $\underline{\theta}^{(k+1)}$ to be any value of $\underline{\theta}$ that maximizes $Q(\underline{\theta}, \underline{\theta}^{(k)})$; that is

$$Q(\underline{\theta}^{(k+1)};\underline{\theta}^{(k)})\geq Q(\underline{\theta};\underline{\theta}^{(k)})$$

The E and M-steps are alternated repeatedly until the difference

$$L(\underline{\theta}^{(k+1)})-L(\underline{\theta}^{(k)})$$

changes by an arbitrarily small amount in the case of convergence of the sequence of likelihood values $\{L(\underline{\theta}^{(k)})\}$.

As pointed out by a referee of the DLR paper, the use of the term “algorithm” to describe this procedure can be criticized, “because it does not specify the sequence of events actually required to carry out a single E- or M-step.” The EM algorithm is really a generic device.

APPENDIX B. MINIMUM MEAN SQUARE ERROR

Inference is the process of systematic estimation or prediction based on information obtained from measurement of other quantities. Hence inference could be viewed as an inversion process. Estimates are usually chosen to minimize the expected value of the square of the error, also known as Minimum Mean Square Error (MMSE) criteria. If $\underline{\boldsymbol{Y}} \in \mathbb{R}^m$ be a random vector that we want to determine and $\underline{\hat{\boldsymbol{y}}}$ be the estimate that we want to obtain. Then, based on the PDF of *Y,* we seek such $\underline{\hat{\boldsymbol{y}}}$ that minimizes the Mean Square Error (MSE) given by the following.

$$E[(\underline{\boldsymbol{Y}} - \underline{\hat{\boldsymbol{y}}})^2] = \int (\underline{\boldsymbol{y}} - \underline{\hat{\boldsymbol{y}}})^2 \, f_y(\underline{\boldsymbol{y}})\, dy$$

If $\underline{\boldsymbol{X}} \in \mathbb{R}^n$ represent the random vector of observations, *m* not necessarily the same as *n*. In the presence of this set of observation, we seek such $\underline{\hat{\boldsymbol{y}}}$ that minimizes the following conditional expectation.

$$E[(\underline{\boldsymbol{Y}} - \underline{\hat{\boldsymbol{y}}})^2 | \underline{\boldsymbol{X}} = \underline{\boldsymbol{x}}] = \int \|\underline{\boldsymbol{y}} - \underline{\hat{\boldsymbol{y}}}\|^2 f_{(y|x)}(\underline{\boldsymbol{y}}|\underline{\boldsymbol{x}})\, dy$$

where $f_{(y|x)}(\underline{\boldsymbol{y}}|\underline{\boldsymbol{x}})$ is the *a posteriori* density. The above equation is minimized by the posterior mean estimator given by the following.

$$\underline{\tilde{\boldsymbol{y}}}_{PM} = \int \underline{\boldsymbol{y}}\, f_{(y|x)}(\underline{\boldsymbol{y}}|\underline{\boldsymbol{x}})\, \underline{\boldsymbol{dy}}$$

where the subscript *PM* refers to *posterior mean.* To see this, we differentiate the conditional expectation with respect to $\underline{\hat{\boldsymbol{y}}}$.

$$\frac{\partial}{\partial \underline{\hat{\boldsymbol{y}}}} \int \|\underline{\boldsymbol{y}} - \underline{\hat{\boldsymbol{y}}}\|^2 f_{(y|x)}(\underline{\boldsymbol{y}}|\underline{\boldsymbol{x}})\, \underline{\boldsymbol{dy}} = \int 2(\underline{\boldsymbol{y}} - \underline{\hat{\boldsymbol{y}}})(-1) f_{(y|x)}(\underline{\boldsymbol{y}}|\underline{\boldsymbol{x}})\, \underline{\boldsymbol{dy}} = 0$$

$$\hat{\underline{\boldsymbol{y}}}\int f_{(y|x)}(\underline{\boldsymbol{y}}|\underline{\boldsymbol{x}})\underline{\boldsymbol{dy}}-\int \underline{\boldsymbol{y}}\, f_{(y|x)}(\underline{\boldsymbol{y}}|\underline{\boldsymbol{x}})\underline{\boldsymbol{dy}}=0$$

$$\hat{\underline{\boldsymbol{y}}}=\int \underline{\boldsymbol{y}}\, f_{(y|x)}(\underline{\boldsymbol{y}}|\underline{\boldsymbol{x}})\underline{\boldsymbol{dy}}=\tilde{\underline{\boldsymbol{y}}}_{PM}$$

Hence we want to find the mean of the *posterior density* to use it as an estimator to our unknown variables. Bayes' Theorem gives the *a posteriori* PDF in terms of prior and likelihood. Dropping the vector notation, we write Bayes' Theorem as follows.

$$f(y|x)=\frac{f(x|y)f(y)}{f(x)}$$

In general, the conditional expectation is difficult to compute. Monte-Carlo integration algorithm effectively computes the integral but it can be very costly for higher dimensions. However, if *X* and *Y* are *jointly Gaussian* random vectors *(RV)* given by the following equation, a relatively simplified formulation could be derived.

$$\begin{bmatrix}X\\Y\end{bmatrix}=\boldsymbol{N}\left(\begin{bmatrix}\mu_x\\\mu_y\end{bmatrix},\begin{bmatrix}\Sigma_{xx} & \Sigma_{xy}\\\Sigma_{yx} & \Sigma_{yy}\end{bmatrix}\right)$$

Without loss of generality, assume *X* and *Y* be zero-mean *RV*. Then using the law of probability we have the following.

$$P(Y|X)=\frac{P(X,Y)}{P(X)}=\frac{(2\pi)^{\frac{-n}{2}}(2\pi)^{\frac{-n}{2}}|\Sigma|^{-1}\exp\left(-\frac{1}{2}\begin{bmatrix}x & y\end{bmatrix}\Sigma^{-1}\begin{bmatrix}x & y\end{bmatrix}^T\right)}{(2\pi)^{\frac{-n}{2}}|\Sigma_{xx}|^{-1}\exp\left(\frac{-1}{2}x^T\Sigma_{xx}^{-1}x\right)}$$

where $\Sigma=\begin{bmatrix}\Sigma_{xx} & \Sigma_{xy}\\\Sigma_{yx} & \Sigma_{yy}\end{bmatrix}$. and we have implemented the following notation to denote

the covariance matrices.

$$cov(X,X)=\Sigma_{xx}, cov(Y,Y)=\Sigma_{yy}, cov(X,Y)=\Sigma_{xy}.$$

Note $\begin{bmatrix}\Sigma_{xx} & \Sigma_{xy}\\ \Sigma_{yx} & \Sigma_{yy}\end{bmatrix}^{-1}=\begin{bmatrix}\Sigma_{xx}^{-1} & 0\\ 0 & 0\end{bmatrix}+\begin{bmatrix}-\Sigma_{xx}^{-1}\Sigma_{xy}\\ I\end{bmatrix}Q^{-1}\begin{bmatrix}-\Sigma_{yx}\Sigma_{xx}^{-1} & I\end{bmatrix}.$

where $Q=\Sigma_{yy}-\Sigma_{yx}\Sigma_{xx}^{-1}\Sigma_{xy}.$

On substitution we get the following.

$$P(Y|X)=(2\pi)^{\frac{-n}{2}}|Q|^{-1}\exp\left[\frac{-1}{2}(y-\Sigma_{yx}\Sigma_{xx}^{-1}x)^{T}Q^{-1}(y-\Sigma_{yx}\Sigma_{xx}^{-1}x)\right].$$

This shows the mean value of the estimator $\hat{Y}$ when X is given is equal to $\Sigma_{yx}\Sigma_{xx}^{-1}x$ with Q being the auto-covariance.

For general cases, whenever $E[X]=\mu_x, E[Y]=\mu_y$ the estimator is given by the following.

$\hat{Y}|X=N(\mu_y+\Sigma_{yx}\Sigma_{xx}^{-1}(x-\mu_x),Q).$ This is also known as the *Gauss-Markov Theorem*.

BIBLIOGRAPHY

1. W. L. Oberkampf, T. G. Trucano, Verification and Validation in Computational Fluid Dynamics, Progress in Aerospace Sciences, 38, 209-272, 2002

2. AIAA, Guide for the Verification and Validation of Computational Fluid Dynamics Simulations, American Institute of Aeronautics and Astronautics, AIAA-G-077, Reston, VA, 1998

3. F. D'Auriaa, O. Mazzantini, The Best-Estimate Plus Uncertainty (BEPU) Challenge in the Licensing of Current Generation of Reactors, International Conference on Opportunities and Challenges for Water Cooled Reactors in the 21. Century, Vienna, Austria, 2009

4. U.S. Nuclear Regulatory Commission, Regulatory Analysis Guidelines of the U.S. Nuclear Regulatory Commission, NUREG/BR-0058 Revision 4, 2004

5. S. N. Aksan, F. D'Auriaa, H. Stadtke, User Effects on the Thermal-Hydraulic Transient System Code Calculations, Nuclear Engineering and Design, 145, 159–174, 1993

6. C. Unal, B. Williams, F. Hemez, S. H. Atamturktur, P. McClure, Improved Best Estimate Plus Uncertainty Methodology, Including Advanced Validation Concepts, to License Evolving Nuclear Reactors, Nuclear Engineering and Design, 241, 1813-1833, 2011

7. C. Unal, K. Pasamehmetoglu, Transmutation Fuels Campaign Modeling and Simulation Roadmap, GNEP, Idaho National Laboratory Report, 7th International Workshop Materials Models and Simulations for Nuclear Fuels, Karlsruhe, Germany, 2008

8. J. Carmack, Advanced Fuels Fundamental Behavior Phenomena Research and Development Plan, Idaho National Laboratory report, INL/LTD-10-18313, 2010

9. H. Glaeser, GRS Analyses for CSNI Uncertainty Methods Study (UMS), NEA-CSNI/R(97)35/V.2-3.1 Report, 1998

10. B. Boyack, R. Duffey, P. Griffith, G. Lellouche, S. Levy, U. Rohatgi, G. Wilson, W. Wulff, N. Zuber, Quantifying Reactor Safety Margins: Application of Code Scaling, Applicability, and Uncertainty Evaluation Methodology to a Large Break Loss of Coolant

Accident, Technical Report NUREG/CR-5249, 1989
11. R. P. Martin, L. D. O'Dell, AREVA's Realistic Large Break LOCA Analyses Methodology, Nuclear Engineering and Design, 235, 1713-1725, 2005
12. M. Y. Young, S. M. Bajorek, M. E. Nissley, L. E. Hochreiter, Application of Code Scaling Applicability and Uncertainty Methodology to the Large Break Loss of Coolant, Nuclear Engineering and Design, 186, 39-52, 1998
13. C. Frepoli, Realistic Large Break LOCA Analysis of AP1000 with Astrum, The Sixth International Conference on Nuclear Thermal Hydraulics, Operations and Safety (NUTHOS-6), Japan, 2004
14. F. D'Auria, W. Giannotti, Development of Code with Capability of Internal Assessment of Uncertainty, Journal of Nuclear Technology, 131, 159-196, 2000
15. W. Jaeger, Uncertainty and Sensitivity Study with TRACE-DAKOTA and TRACE-SUSA: A Comparison Based on NUPEC BFBT Experimental Data, CAMP, Slovenia, 2012
16. M. S. Eldred, Overview of Uncertainty Quantification Algorithm R&D in the Dakota Project, 2011
17. L. P. Swiler, T. L. Paez, R. L. Mayes, Epistemic Uncertainty Quantification Tutorial, Proceedings of the IMAC-XXVII: Conference & Exposition on Structural Dynamics - Model Verification & Validation, 2009
18. S. H. Lee, W. Chen, A Comparative Study of Uncertainty Propagation Methods for Black-box type Functions, ASME, International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, Las Vegas, 2007
19. J. Won, C. Choi, J. Choi, Improved Dimension Reduction Method (DRM) in Uncertainty Analysis using Kriging Interpolation, Journal of Mechanical Science and Technology, 23, 1249-1260, 2009
20. L. Jaulin, M. Kieffer, O. Didrit, E. Walter, Applied Interval Analysis, Berlin: Springer, 1-85233-219-0, 2001
21. A. P. Dempster, Upper and Lower Probabilities Induced by a Multivalued Mapping, The Annals of Mathematical Statistics, 38, 325-339, 1967
22. Shafer, Glenn, A Mathematical Theory of Evidence, Princeton University Press, 0-

608-02508-9, 1976

23. Fine, L. Terrence, Review: Glenn Shafer, A Mathematical Theory of Evidence, Bull. Amer. Math. Soc., 83, 667-672, 1977

24. M. Pourgol-Mohamad, A. Mosleh, M. Modarres, Methodology for the Use of Experimental Data to Enhance Model Output Uncertainty Assessment in Thermal-Hydraulics Codes, Reliability Engineering and System Safety, 95, 77-86, 2010

25. J. Aldrich, R. A. Fisher and the Making of Maximum Likelihood 1912-1922, Statistical Science, 12, 162-176, 1997

26. A. P. Dempster, N. M. Laird, D. B. Rubin,, Maximum Likelihood from Incomplete Data via the EM Algorithm, Journal of the Royal Statistical Society, Series B (Methodological), 39, 1-38, 1977

27. R. A. Redner, H. F. Walker, Mixture Densities, Maximum Likelihood and the EM Algorithm, SIAM Review, 26, 195-239, 1984

28. G. J. Mclachlan, D. Peel, Finite Mixture Models, John Wiley & Sons Inc., 2000

29. J. L. Gauvain, C. H. Lee, Maximum a Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains, IEEE Transactions on the Speech and Audio Processing, 2, 291-298, 1994

30. M. R. Gupta, Y. Chen, Theory and Use of the EM Algorithm, 1601984308, 2011

31. Z. Liang, S. Wang, An EM approach to MAP Solution of Segmenting Tissue Mixtures: A Numerical Analysis, IEEE Trans. Med. Imag., 28, 297-310, 2009

32. J. Zhou, J. L. Coatrieux, A. Bousse, H. Shu, L. Luo, A Bayesian MAP-EM Algorithm for PET Image Reconstruction using Wavelet Transform, IEEE Trans. Nucl. Sci., 54, 1660-1669, 2007

33. B. E. Boyack, I. Catton, R. B. Duffey, P. Griffith, K. R. Katsma, G. S. Lellouche, S. Levy, U. S. Rohatgi, G.E. Wilson, W. Wulff,N. Zuber, Quantifying Reactor Safety Margins Part 1: An Overview Of The Code Scaling, Applicability, And Uncertainty Evaluation Methodology, Nuclear Engineering and Design, 119, 1-16, 1990

34. A. de Crecy, Determination of the Uncertainties of The Constitutive Relationships of The CATHARE 2 Code, Mathematics and Computation, Salt Lake City, 2001

35. A. de Crecy, P. Bazin, H. Glaeser,Uncertainty and Sensitivity analysis of the LOFT

L2-5 test: Results of the BEMUSE Programme, Nuclear Engineering and Design, 238, 3561-3578, 2008

36. D. G. Cacuci, Predictive Modeling of Coupled Multi-physics Systems: I. Theory, Annals of Nuclear Energy, 70, 266-278, 2014

37. D. G Cacuci, M. C. Badea, Predictive Modeling of Coupled Multi-physics Systems: II. Illustrative Application to Reactor Physics, Annals of Nuclear Energy, 70, 279-291, 2014

38. D. G. Cacuci, M. Ionescu-Bujor, Model Calibration and Best-Estimate Prediction through Experimental Data Assimilation: I. Mathematical Framework, Nuclear Science and Engineering, 165, 18-44, 2010

39. A. Bui, B. Williams, N. Dinh, Advanced Calibration and Validation of a Mechanistic Model of Subcooled Boiling Two-Phase Flow, International Congress on Advances in Nuclear Power Plants, Charlotte, 2014

40. B. Adams, W. Bohnhoff, K. Dalbey, J. Eddy, M.Eldred, D. Gay, K. Haskell, P. Hough, L. Swiller, DAKOTA: A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 5.0 Reference Manual, Sandia Tech Report, SAND2010-2184, 2013

41. I. Jolliffe, Principal Component Analysis, Springer, 2002

42. H. Lee, C. Holloman, C. Calder and D. Higdon, Flexible Gaussian Process via Convolution, Duke University, 2002

43. F. D'Auria, Overview of Uncertainty Issues and Methodologies, Proceedings of the ECD/CSNI Seminar on Best Estimate Methods in Thermal Hydraulic Analysis, 437-458, Ankara, 1998

44. M. Pourgol-Mohamad, M. Modarres, A. Mosleh, A General Thermal Hydraulics Uncertainty Analysis Methodology, PSAM8 conference, 14-19, New Orleans, 2006

45. I. J. Myung, Tutorial on Maximum Likelihood Estimation, Ohio State University, Columbus, OH 43210-1222, USA

46. M. Bagnoli, T. Bergstrom, Log-Concave Probability and its Applications, Economic Theory, 26, 445-469, 2005

47. J. A. Bilmes, A Gentle Tutorial of the EM Algorithm and its Application to Parameter

Estimation for Gaussian Mixture and Hidden Markov Models, Tech. Rep. ICSI-TR-97-021, 1998

48. C. Do, S. Batzoglou, What is the Expectation Maximization Algorithm?, Nature Biotechnology, 26, 897-899, 2008

49. G. J. McLachlan, T. Krishnan, S. K. Ng, The EM Algorithm, Humboldt-Universität Berlin, Center for Applied Statistics and Economics (CASE), 2004

50. P. L. Odell, Gauss–Markov Theorem, Encyclopedia of Statistical Sciences, 4, 2006

51. L. L. Scharf, Statistical Signal Processing: Detection, Estimation, and Time Series Analysis, Addison-Wesley, 1991

52. C. Fraley, A. Raftery, Bayesian Regularization for Normal Mixture Estimation and Model-Based Clustering, Journal of Classification, 24, 151-181, 2007

53. J. L. Schafer, Analysis of Incomplete Multivariate Data by Simulation, Chapman and Hall, 1997

54. C. Fraley, A. Raftery, Bayesian Regularization for Normal Mixture Estimation and Model-Based Clustering, Technical Report No. 486, 2009

55. P. Ihle, K. Rust, FEBA-Flooding Experiments with Blocked Arrays Evaluation Report, 1984

56. R. Shrestha, T. Kozlowski, Verification of Axial Cladding Temperature Distribution Using RELAP5 and TRACE for FEBA/SEFLEX LOCA Experiments, The 15th International Topical Meeting on Nuclear Reactor Thermal-hydraulics, NURETH-15, Pisa, 2013

57. J. Hadamard, Lectures on Cauchy's Problem, Yale University Press, 1923

58. F. O'Sullivan, A Statistical Perspective on Ill-Posed Inverse Problems, Statistical Science, 1, 502-518, 1986

59. A. Tikhonov, V. Arsenin, Solutions of Ill-Posed Problems, Wiley, New York, 1977