

Copyright

by

William Ladd Gurecky

2020

The Dissertation Committee for William Ladd Gurecky  
certifies that this is the approved version of the following dissertation:

**A CFD-Informed Model for Subchannel Resolution  
Crud Prediction**

Committee:

---

Derek Haas, Supervisor

---

Benjamin Leibowicz

---

Sheldon Landsberger

---

Kevin Clarno

---

Stuart Slattery

**A CFD-Informed Model for Subchannel Resolution  
Crud Prediction**

by

**William Ladd Gurecky**

**DISSERTATION**

Presented to the Faculty of the Graduate School of  
The University of Texas at Austin  
in Partial Fulfillment  
of the Requirements  
for the Degree of

**DOCTOR OF PHILOSOPHY**

THE UNIVERSITY OF TEXAS AT AUSTIN

May 2020

*To family and friends*

## Acknowledgments

I would like to acknowledge Dr. Kevin Clarno, Dr. Stuart Slattery, Dr. Derek Haas, Dr. Sheldon Landsberger, Dr. Ben Leibowicz and Dr. Robert Salko for their technical guidance and general advice throughout my graduate studies.

Special thanks to Dr. Erich Schneider who fostered many a students' interest in nuclear engineering & software development, including mine.

This research was supported by and performed in conjunction with the Consortium for Advanced Simulation of Light Water Reactors (<http://www.casl.gov>), an Energy Innovation Hub (<http://www.energy.gov/hubs>) for Modeling and Simulation of Nuclear Reactors under U.S. Department of Energy Contract No. DE-AC05-00OR22725.

# Table of Contents

<b>Acknowledgments</b>	v
<b>Acronyms</b>	vii
<b>Nomenclature</b>	viii
<b>List of Tables</b>	ix
<b>List of Figures</b>	x
<b>Chapter 1. Introduction</b>	1
1.1 Significance and Novelty . . . . .	2
1.2 Crud Background . . . . .	3
<b>Chapter 2. Theory</b>	4
2.1 Model Approach . . . . .	4
2.2 Construction of the Hi2lo Map . . . . .	9
2.2.1 Capturing Dependence Between Random Variables . . . . .	9
2.2.2 Copula . . . . .	12
2.2.3 Fitting Copula . . . . .	17
2.2.4 Sample Quantiles . . . . .	21
2.2.5 Gradient Boosted Regression Trees . . . . .	27
2.2.6 Monte Carlo Crud Estimation . . . . .	36
2.3 Propagating Crud Through Time . . . . .	38
2.3.1 Hot Spot Stationarity . . . . .	38
2.3.2 Time Stepping Scheme . . . . .	41
2.4 Method Summary . . . . .	44
<b>Appendices</b>	48
<b>Appendix A. Rod Surface Heat Transfer</b>	49
A.1 Subcooled Boiling and DNB . . . . .	49
<b>Bibliography</b>	51

## Acronyms

BHF	Boundary Heat Flux
CASL	Consortium for Advanced Simulation of LWRs
CDF	Cumulative Density Function
CFD	Computational Fluid Dynamics
CILC	Crud Induced Local Corrosion
CIPS	Crud Induced Power Shift
CRUD	Chalk River Unidentified Deposit (crud)
CTF	Coolant boiling in rod arrays—Two Fluid (COBRA-TF)
FOI	Field of Interest
GBRM	Gradient Boosted Regression Model
GBRT	Gradient Boosted Regression Tree
HTC	Convective Heat Transfer Coefficient
LANL	Los Alamos National Laboratory
LOO	Leave-one-out
LOOCV	Leave-one-out cross validation
LS	Least Squares
LWR	Light Water Reactor
ML	Maximum Likelihood
ORNL	Oak Ridge National Laboratory
PDF	Probability Density Function
PWR	Pressurized Water Reactor
ROM	Reduced Order Model
RV	Random Variable
TH	Thermal Hydraulic
TKE	Turbulent Kinetic Energy
VERA	Virtual Environment for Reactor Applications

# Nomenclature

## Functions & Maps

$c(\cdot)$	Copula density function
$C(\cdot)$	Copula cumulative density function
$\varphi(\cdot)$	Copula generator function
$f(\cdot)$	Marginal density function
$F(\cdot)$	Marginal cumulative density function
$\mathcal{G}(\cdot)$	Crud generator function
$h(\cdot)$	Joint density function
$H(\cdot)$	Joint cumulative density function
$\mathcal{F}(\cdot)_M$	Gradient boosted model
$\mathcal{R}$	Mapping from sample space to a location on the rod surface

## Symbols

$t$	Time
$T$	Temperature
$k$	Turbulent kinetic energy
$q''$	Boundary heat flux
$\mathbf{p}$	Auxiliary predictive variables
$Q_\tau$	Quantile function (inverse CDF)
$q_\tau$	The $\tau^{th}$ quantile
$\rho_\tau$	Kendall's tau
$\theta$	Marginal distribution parameter
$\theta_c$	Copula shape parameter
$\Theta_c$	Archimedean copula family
$\mathbf{C}$	Crud state vector
$C_m$	Crud mass density
$C_b$	Crud boron density
$C_t$	Crud thickness
$\mathbf{X}$	Random vector
$X$	Random variable

## List of Tables

2.1 Runtime model parameters. . . . .	46
---------------------------------------	----

# List of Figures

2.1	Top-down view of typical subchannel and CFD meshes . . . . .	5
2.2	On a single coarse CTF patch: Differences in crud prediction between CFD and CTF models. . . . .	6
2.3	C-vine on 3 variables: $\{T, q'', k\}$ . . . . .	11
2.4	The conditional $h$ function vs. value of the conditioning variable $u_2$ for a Gaussian copula with $\theta_c = 0.7$ . . . . .	15
2.5	$h^{-1}$ vs. value of the conditioning variable $u_2$ for a Gaussian copula with $\theta_c = 0.7$ . . . . .	15
2.6	Gaussian copula density with $\theta_c = 0.7$ . . . . .	15
2.7	Samples drawn from Gaussian copula with $\theta_c = 0.7$ . . . . .	15
2.8	Samples drawn from Gaussian copula with exponential margins. . . . .	16
2.9	Examples of bivariate copula PDFs. . . . .	18
2.10	Fictitious bivariate data set. . . . .	20
2.11	Graphical comparison of Kendall's distribution for several fitted copula. .	20
2.12	Quantile loss function. . . . .	21
2.13	CDF from quantiles. . . . .	22
2.14	Distribution of quantiles for a normally distributed RV. . . . .	25
2.15	Distribution of quantiles for a beta distributed RV. . . . .	26
2.16	Regression tree stump. . . . .	28
2.17	Gradient boosted quantile regression example. . . . .	31
2.18	Gradient boosted quantile regression residual summary. . . . .	33
2.19	Two-class gradient boosted classifier example. . . . .	35
2.20	Sample remapping. A single CTF face is shown as the bold square with a grid overlay partitioning the patch surface. In this figure the number of samples per CTF face is $N = 30$ . . . . .	40
2.21	Multi-state point time stepping overview. Dashed arrows represent sampling from a distribution. Solid arrows represent a functional mapping or operation. . . . .	42
2.22	Time step procedure depicted for a single CTF face . . . . .	44
A.1	Boiling curve. . . . .	49

# 1 | Introduction

The Consortium for Advanced Simulation of Light Water Reactors (CASL) selected several problems identified by industry partners as critical, inadequately understood, engineering-scale phenomena, which would provide financial and safety benefits to the nuclear power industry if resolved [1]. The problem of interest in this work is the prediction of Chalk River unidentified deposit (crud) growth rates. In an effort to simulate the effects of crud on the power and burnup distribution, a code produced by a Los Alamos National Laboratory (LANL) and Oak Ridge National Laboratory (ORNL) collaboration under the name MAMBA was developed [2]. The development of the MPO Advanced Model for Boron Analysis (MAMBA) and other supporting Virtual Environment for Reactor Applications (VERA) tools provided a starting point for the high-to-low (hi2lo) methods at hand.

A phenomena known as crud-induced power shift (CIPS) is caused by the presence of elevated  $^{10}\text{B}$  concentrations in the crud layer. Since crud is preferentially deposited on the fuel rods in hot regions of the core and  $^{10}\text{B}$  is a strong neutron absorber, the crud buildup leads to a slight shift in power production toward the bottom of the core under steady-state operation. Crud induced power shift impacts the burnup distribution over a cycle, reduces shutdown margin, and is important to account for when computing thermal margins of the fuel [3]. The prediction of CIPS is especially important for older facilities seeking to uprate power output or extend their operational lifetime. Additionally, the presence of crud on the rod surface has been shown to exacerbate local oxide penetration rates of some zirconium alloys [4]. This is known as crud-induced local corrosion (CILC).

Improvements in crud simulation techniques ultimately improve the ability to predict the CIPS and CILC phenomena for a given fuel loading pattern. If significant CIPS or CILC can be accurately predicted provided a candidate loading pattern, significant cost savings are possible by ensuring the target burnup is not missed due to the presence of excess crud in the core [3]. Loading patterns that would yield unfavorable crud buildup could be avoided provided an accurate and robust crud prediction capability is available for use in a production environment.

The Virtual Environment for Reactor Applications (VERA) is a key component of CASL’s technical portfolio. The VERA meta-package integrates a variety of physics packages and multiphysics coupling options to form a robust reactor simulation capability. For multi-cycle depletion computations, VERA relies upon the Michigan Parallel Characteristics Based Transport (MPACT) code, a 2-D/1-D method of characteristics neutronics package, coupled with the subchannel thermal hydraulics code, Coolant Boiling in Rod Arrays–Two Fluid (CTF). An integrated crud modeling capability is provided by MAMBA to address the CIPS challenge problem.

To reduce computation times, the subchannel TH code discretizes the reactor domain into large, centimeter scale finite volumes. As a consequence of this discretization scheme, sub-centimeter scale thermal hydraulic effects of the spacer grids on crud are averaged over large regions on the fuel rods’ surfaces. Though small scale phenomena are not explicitly modeled, they are approximately accounted for in a variety of empirically derived closure relations. In effect, a single constant estimate for the mean thermal hydraulic conditions is obtained in each finite volume.

## 1.1 Significance and Novelty

Crud growth is dominated by threshold physics [5]. Hot and cold spots present downstream of spacer grids must be accurately resolved by the hi2lo model to predict the maximum crud thickness and boron precipitation within the crud layer.

It is challenging to faithfully capture the peaks and valleys in rod surface temperature and TKE distribution by traditional interpolation techniques because such a model must guard against smearing out the sharp peaks present in the spatial distributions. In the present method we forgo a spatial shape function mapping strategy for a statistically driven approach that predicts the fractional area of a rod's surface in excess of some critical temperature but not precisely where such maxima occur.

## 1.2 Crud Background

The buildup of crud results from the deposition of metal particulates and corrosion products entrained in the primary coolant loop of a light water reactor on the exterior surface of the fuel rods. These impurities arise from erosion and corrosion processes elsewhere in the loop. Of all the coolant impurities, the largest contributor the initial formation of a crud layer on the outer cladding surface is nickel ferrite. The initial build up of nickel ferrite may be described by the ordinary differential equation (ODE) shown in equation 1.1.

$$\frac{dN_{\text{NiFe},c}}{dt} = (\alpha_{nb} + \alpha_b q_b'') N_{\text{NiFe,cool}} - \gamma_k k \quad (1.1)$$

Where  $N_{\text{NiFe},c}$  is the concentration of nickel ferrite in the crud within a small finite volume on the cladding surface.  $N_{\text{NiFe,cool}}$  is the concentration of nickel and iron impurities in the coolant.  $\alpha_b$  and  $\alpha_{nb}$  represent boiling and non-boiling rate constants respectively. The boiling component of the boundary heat flux (BHF) on the outer cladding surface is given by  $q_b'' [\text{W}/\text{m}^2]$ . Note that  $q_b''$  is only non-zero when  $T > T_{sat}$ .  $\gamma_k$  is an erosion multiplier and  $k$  is the near-wall local TKE. Crud typically forms where temperatures are high and where subcooled boiling occurs on the rod surface.

## 2 | Theory

A review of the solution detail afforded by CFD and subchannel thermal hydraulic codes is provided to begin this section along with consequences of their respective spatial discretization schemes on crud growth. A simplified method for harnessing CFD results to improve expected-value computation of crud on a given CTF face is provided to introduce the hi2lo strategy. Next, copula and quantile regression are discussed as a means to model the joint distribution of temperature and TKE on the rod surface. This is followed by an introduction to the gradient boosting machine learning method used to predict the behavior of the joint distribution as a function of local core conditions. The Monte Carlo method for estimating the integral required to compute the expected crud value is given. A review of importance sampling is also provided to provide a means to increase the sampling efficiency of the Monte Carlo integration procedure. The section culminates in an integration of the copula, quantile regression, and importance sampling routines into a complete algorithm for time dependent crud prediction.

### 2.1 Model Approach

A fundamental difference between the CFD and CTF computations is the average size of the mesh cells. In the azimuthal coordinate, CTF decomposes a single rod surface into four patches. An example top down view of typical CFD and CTF meshes for a single pin are given in figure 2.1. Though both codes employ a finite volume spatial discretization, CFD can resolve the flow at much smaller length scales. Additionally, each code employs a different set of closure models to the underlying set of coupled energy,

mass, and momentum balances. In practice these differences lead to large discrepancies in boiling, turbulent mixing, and rod surface temperature predictions between the two codes.

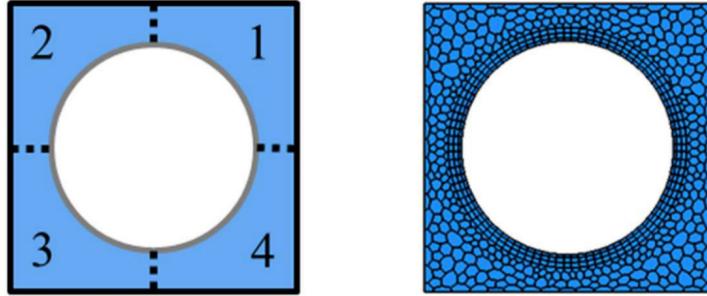


Figure 2.1: Top-down view of typical subchannel (left) and CFD mesh (right) for a single pin [6].

Shown in figure 2.2, on a given CTF rod surface patch, estimates for the surface temperature, TKE, and heat flux are provided as point estimates. The predicted CTF quantities are an estimate for the average thermal hydraulic conditions over that coarse patch. Consequently, CTF crud predictions deviate from reality since crud growth is highly sensitive to the presence of subcooled boiling on the rod surface; if CTF predicts a rod surface temperature less than the saturation point in a given patch little or no crud will form when in reality, a small portion of that rod surface could exist above the saturation point and thus harbor crud. Small localized mistakes in crud predictions compound throughout the core, leading to poor CIPS estimates.

In figure 2.2,  $f$  denotes a probability density function. Integration of this density function may be interpreted as computing a fractional area of the rod surface that exists within the specified integration limits.

CTF estimates mean TH conditions everywhere in the core at a low spatial resolution. The CFD informed model provides higher order moments about the mean.

$$\mathbf{X}(\mathbf{p}, \mathbf{z}) = \underbrace{\boldsymbol{\mu}(\mathbf{p}, \mathbf{z})}_{\text{CTF}} + \underbrace{\boldsymbol{\varepsilon}(\theta(\mathbf{p}, \mathbf{z}))}_{\text{CFD Informed}} + \mathbf{b}(\mathbf{p}, \mathbf{z}) \quad (2.1)$$

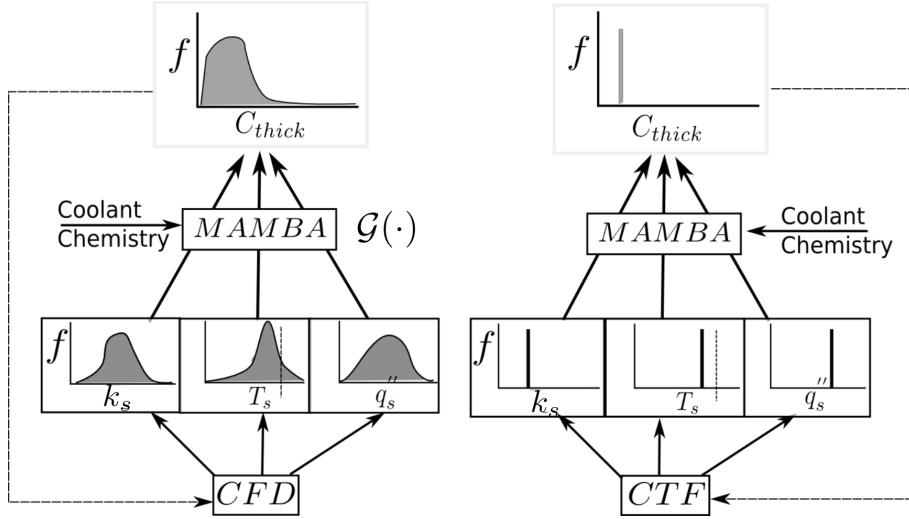


Figure 2.2: On a single coarse CTF patch: Differences in crud prediction between CFD and CTF models.

- $\mathbf{X}$  is a three component vector representing the cladding surface temperature,  $T$ , turbulent kinetic energy,  $k$ , and boundary heat flux,  $q''$ .
- $\mathbf{z}$  denotes spatial coordinates and  $\mathbf{p}$  represents a set of auxiliary predictors. Auxiliary predictors are covariates that describe local core conditions and may be geometric or thermal hydraulic in nature.
- $\boldsymbol{\varepsilon}$  is a three-component random vector comprised of temperature, turbulent kinetic energy and boundary heat flux fields.  $\boldsymbol{\varepsilon}$  is distributed according to a CFD informed model with  $\theta$  representing free model parameters which are determined from the CFD data.
- $\mathbf{b}$  is bias between the low and high fidelity models ( $\boldsymbol{\mu}_{CTF} - \boldsymbol{\mu}_{CFD}$ ). Despite providing identical inlet boundary conditions to both codes bias exists due to algorithmic, closure model and meshing differences between the two codes.
- Field averages,  $\boldsymbol{\mu}$ , are piecewise constant over each CTF patch.

Consider a hypothetical case where the CFD results are normally distributed about

the CTF results such that  $\varepsilon \sim \mathcal{N}(0, \Sigma(\mathbf{p}, \mathbf{z}))$ , where  $\Sigma(\mathbf{p}, \mathbf{z})$  is a covariance matrix that depends on local core conditions. Shifting the distribution by a constant vector  $\mathbf{c} = \mathbf{b} + \boldsymbol{\mu}_{ctf}$ , results in a distribution denoted by  $h$  in equation 2.2.

$$\begin{aligned} h|_{(\mathbf{p}, \mathbf{z})} &= \mathcal{N}(\mathbf{c}, \Sigma(\mathbf{p}, \mathbf{z})) \\ &= \mathcal{N}\left(\begin{pmatrix} c_T \\ c_k \\ c_{q''} \end{pmatrix}, \begin{pmatrix} \Sigma_{TT} & \Sigma_{Tk} & \Sigma_{Tq''} \\ \Sigma_{kT} & \Sigma_{kk} & \Sigma_{kq''} \\ \Sigma_{q''T} & \Sigma_{q''k} & \Sigma_{q''q''} \end{pmatrix}\right) \Big|_{(\mathbf{p}, \mathbf{z})} \end{aligned} \quad (2.2)$$

Where  $\Sigma_{xx} = \sigma_x^2$  and  $\Sigma_{xy} = cov(x, y)$ .

Equation 2.3 estimates the expected crud mass  $C_m$  that accumulates on each CTF patch in time  $\delta t$ . Let the CTF face of interest have area  $A$ . Let  $\mathbf{X} = \{T, k, q''\}$  denote a random vector of temperature, TKE, and BHF.  $\mathbf{I}$  represents additional known crud parameters,  $\mathbf{C}_o$  is the crud state at the start of the time step. Let the joint density function of  $\mathbf{X}$  be denoted by  $h$ , and it's CDF be denoted by  $H$ . The crud model,  $\mathcal{G}(\cdot)$ , is common to all CTF faces. The joint cumulative density's parameters are predicted from the available high resolution CFD data in every CTF face. In the case of an assumed normal distribution model there are nine unknowns which require fitting:  $\theta = \{\sigma_T^2, \sigma_k^2, \sigma_{q''}^2, \Sigma_{kT}, \Sigma_{q''T}, \Sigma_{q''k}, c_T, c_k, c_{q''}\}$ . In the subsequent sections we will seek to relax the normality assumption of the CFD residuals about the CTF result.

$$\begin{aligned} C_m &= A\mu_m \\ &= A\mathbb{E}[\mathcal{G}(\mathbf{X}|\mathbf{C}_o, \mathbf{I}, \delta t)] \\ &= A \iiint \mathcal{G}(\mathbf{X}|\mathbf{C}_o, \mathbf{I}, \delta t)h(\mathbf{X}|\theta)d\mathbf{X} \end{aligned} \quad (2.3)$$

A strategy to compute the unknowns of the joint distribution on each CTF face is required. The current work proposes a data driven model,  $\mathcal{F}_M$ , to predict the unknowns provided a suite of pre-computed CFD results are used to train the machine learning model and local thermal hydraulic conditions provided by CTF at runtime are utilized

to evaluate the model on each CTF face. Algorithm 1 is used to compute the total crud mass in each CTF face.

---

**ALGORITHM 1**

Generic hi2lo method for crud prediction.

---

- 1: **Initialization**
  - 2: (1) Pre-process training set.
  - 3: (1b) Fit the joint distribution parameters,  $\theta$ , to known CFD data.
  - 4: (1c) **def:**  $\theta \leftarrow \mathcal{F}_M(\mathbf{p}, \mathbf{z})$
  - 5: (2) Train model:  $\hat{\mathcal{F}}_M = \operatorname{argmin}_{\mathcal{F}} \mathbb{E} [L(\mathcal{F}_M(\mathbf{p}, \mathbf{z}), \theta)]$
  - 6: **for** CTF face,  $j$  **do**
  - 7:   Evaluate ML model  $\hat{\theta}_j \leftarrow \hat{\mathcal{F}}_M(\mathbf{p}_j, \mathbf{z}_j)$
  - 8:   Reconstruct  $\hat{H}_j(\cdot | \hat{\theta}_j)$
  - 9:   Draw samples  $\mathbf{X} \sim \hat{H}_j$
  - 10:   Evaluate equation 2.3 via Monte Carlo approximation
  - 11: **end for**
- 

Where  $L(\cdot)$  is a generic differentiable loss function. A discussion on the machine learning model and loss function follows in section 2.2.5. The reconstruction of the joint density function  $\hat{H}$  from copula and univariate quantile functions is discussed in section 2.2.4. Monte Carlo and importance sampling are discussed in section 2.2.6.

Algorithm 1 may be broken down into five generic components. *Lines 2-5:* Establish a relationship between the local core geometry and thermal hydraulic conditions to the behavior of surface temperature, TKE and boundary heat flux distributions using a pre-computed suite of CFD results. *Line 7:* The prediction of joint distribution parameters. These parameters will take the form of conditional quantiles and parameters of a copula in the present work. *Line 8:* The reconstruction of the joint temperature, TKE, and boundary heat flux distribution in each CTF face given predictions provided by evaluating the data driven model  $\mathcal{F}_M$  at the local core conditions adjacent to the CTF face under consideration. *Line 9:* Sampling from the reconstructed distribution. *Line 10:* Integration of the crud density by a Monte Carlo procedure.

## 2.2 Construction of the Hi2lo Map

Next, the multivariate Gaussian assumption made to capture the autocorrelation between the surface temperature, boundary heat flux and near wall TKE fields is relaxed. Flexibility is afforded by factoring the multivariate distribution into marginal distributions and a copula. The statistical parameters describing this multivariate distribution are still determined via a data driven model, as in algorithm 1. The result is a semi-parametric model of the conditional joint distribution of temperature, TKE, and boundary heat flux on the rod surface. The model is semi-parametric because the copula is selected from a library of parametric distribution families while the marginal models are constructed using conditional quantile prediction, where the number of quantiles used in the reconstruction is set at runtime. Quantile regression as applied in this work does not assume a priori the univariate distribution family governing the behavior of the surface temperature or near-wall TKE distributions. This is the case because no mechanistic model exists or could be identified which describes the distribution of finding a particular patch of the rod surface in excess of a given temperature. It is unlikely a physics based parametric model could be devised for this purpose due to the general complexity of the underlying governing equations, grid geometry, and since the flow regime of interest is turbulent. In lieu of such a mechanistic model non parametric distributions are adopted to represent the surface temperature and TKE distributions.

### 2.2.1 Capturing Dependence Between Random Variables

Since the outer cladding temperature, near-wall TKE and boundary heat flux are used as boundary conditions to a crud growth package, it is particularly important to understand and capture the relationship between these fields in the hi2lo model. The hi2lo model under consideration is not a dynamic model in the sense that it cannot be expressed as a coupled system of differential equations. Instead, in the purely data driven approach the relationships between the FOI are established through standard statistical

correlation measures.

In a multiphysics simulation of a PWR core the coupled momentum, mass, and energy balances along with the appropriate closure models dictate the rod outer cladding surface temperature. The Dittus Boelter relationship is used to relate the surface heat transfer coefficient with the local Reynolds number (see appendix A). According to this relationship, larger Reynolds numbers corresponds to higher heat transfer coefficients. Newton's law of cooling states  $T_s = q''/h + T_\infty$  and  $h$  may be computed via Dittus Boelter. Therefore, the surface temperature ( $T_s$ ) is negatively correlated with the Reynolds number and where the local turbulent kinetic energy is large the rod surface temperature will be depressed if the local heat flux is held fixed. It is also apparent that the surface temperature is positively correlated with the the local boundary heat flux. In order to simplify the model only the dependency between the surface temperature and local turbulent kinetic energy is considered in hi2lo model.

A statistical relationship between the surface temperature, TKE, and boundary heat flux in each CTF face is sought. To this end vine copula provide a flexible framework to model high dimensional dependence structures [7]. Vine copula are hierarchical tree models in which the edges represent bivariate copula and the nodes are univariate distributions. The canonical vine (C-vine) copula shown in equation 2.4 may be used to express the trivariate ( $n = 3$ ) distribution of temperature, TKE, and  $q''$  on the rod surface.

$$h(T, k, q'') = f_T f_k f_{q''} \prod_{m=1}^{n-1} \prod_{e \in E_m} c_{ij|D_e}(u_i|D_e, u_j|D_e) \quad (2.4)$$

Where  $m$  denotes the tree level in the vine and each bivariate copula model,  $c$  defined on the edge  $\{e\}$  is known as a pair copula. The conditioning set at edge  $e$ , denoted  $D_e$ , is defined by a proximity condition [8].  $E_m$  denotes the set of all edges at level  $m$ . The graphical representation of the vine is provided in figure 2.3. Simplifying independence assumptions can be made based on the heat transfer processes on a fuel rod that lead to

certain copula in the vine to take the form of a uniform density distribution on the unit square.

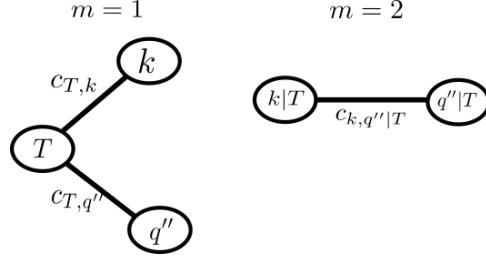


Figure 2.3: C-vine on 3 variables:  $\{T, q'', k\}$ .

In this work it is assumed that the cladding surface temperature and near-wall TKE are uncorrelated with the boundary heat flux. The justification is as follows.

Relative variations in boundary heat flux are very small over a CTF face ( $\pm 5\%$ ) provided that a CTF face represents a small (approximately  $1[cm^2]$ ) localized patch on the rod surface. Large absolute heat fluxes of approximately  $80[W/cm^2]$  are possible on the cladding surface of fuel in a typical PWR. However, high axial and azimuthal gradients of boundary heat flux on the rod surface are not typical under standard operating conditions. High thermal gradients lead to thermal induced stresses on the cladding which can promote rod bowing or, extreme in conditions, fuel failure.

After applying the independence assumptions the original trivariate dependence model between the temperature, TKE, and boundary heat flux is reduced to a bivariate model in which the boundary heat flux is treated independently. Applying the simplifying assumptions results in:  $c_{T,q''}(u_T, u_{q''}) = 1$  and  $c_{k,q''|T}(u_{k|T}, u_{q''|T}) = 1$ . The simplified joint density is given by equation 2.5:

$$h(T, k, q'') \approx f_T f_k f_{q''} c_{T,k}(u_T, u_k) \cdot 1 \cdot 1 \quad (2.5)$$

Where  $u_T = F(T)$  and  $u_k = F(k)$ .  $F(\cdot)$  denotes the CDF. Following the independence assumption, the marginal density function of the outer cladding boundary heat flux is assumed to be a Dirac delta function centered on the value provided by CTF (VERA).

We will apply this assumption in all sections which follow. Then in the  $j^{th}$  CTF face the boundary heat flux marginal distribution is always given by equation 2.6.

$$f_{j,q''} = \delta_{(q''_{j,ctf})} \quad (2.6)$$

### 2.2.2 Copula

A copula is a function which relates marginal probability distributions to a multidimensional joint distribution. Copula provide a flexible alternative to multidimensional Gaussian based models. Copula are utilized in this work because of their ability to capture non-Gaussian dependence structure between two or more correlated random variables, for instance temperature and the TKE at a given point on a rod's surface. Furthermore, Sklar's theorem is used in this work in order to decompose joint distributions into a product of uni-variate marginal distributions and a copula function. In this section, Sklar's theorem is provided along with examples of copula functions and techniques to draw samples from them.

The product rule of probability is shown in equation 2.7. To clarify notation used in this section: The comma denotes the conjunction “and” and the bar,  $|$ , is read “given”.

$$P(x, y) = P(x)P(y|x) \quad (2.7)$$

The marginal distribution of a bivariate joint distribution  $f(x, y)$  is given by equation 2.8. The marginalization process is analogous to projecting the entire joint density onto a single axis.

$$f(x) = \int f(y)f(x|y)dy \quad (2.8)$$

The cumulative density function,  $F$  is defined as:

$$F = \mathbf{P}[X < x] = \int_{-\infty}^x f(x)dx$$

A joint  $d$  dimensional cumulative distribution is given by equation 2.9.

$$H(x_1, \dots x_d) = \mathbf{P}[X_1 \leq x_1, \dots X_d \leq x_d] \quad (2.9)$$

Where  $X_1, \dots X_d$  are random variables.

The process of decomposing a multivariate distribution into uni-variate marginal distributions and an object which describes their conditional dependence was formalized by Sklar [9]. Shown in equation 2.10, Sklar's Theorem defines a *copula* cumulative density function,  $C$ .

$$C(F_1(x_1), \dots F_d(x_d)) = H(x_1, \dots x_d) \quad (2.10)$$

If  $F_1, \dots F_d$  are continuous, then  $C$  is unique. Conversely, if  $C$  is a copula and  $F_1, \dots F_d$  are smooth cumulative destiny functions then the function  $H$  is a joint cumulative distribution with margins  $F_1, \dots F_d$ . A proof is provided in Nelsen's introductory copula text [10].

Sklar also showed that the joint probability distribution,  $h(x_1, \dots x_d)$ , can be computed from constituent marginalized univariate distributions and the copula density,  $c$ .

$$h(x_1, \dots x_d) = c(F_1(x_1), \dots F_d(x_d)) \cdot f_1(x_1) \cdots f_d(x_d) \quad (2.11)$$

For brevity, let  $u_1, \dots u_d$  represent samples from their CDFs as follows:

$$u_1 = F_1(x_1)$$

$$u_d = F_d(x_d)$$

$$u \in [0, 1]$$

Where the joint density of the copula,  $c$ , is given by equation 2.12:

$$c(u_1, \dots u_d) = \frac{\partial C(u_1, \dots u_d)}{\partial u_1 \dots \partial u_d} \quad (2.12)$$

Sklar's theorem enables one to construct models for the margins separately from a model of the dependence structure. When combined, the margins and the copula specify a multivariate probability density function. Compared to rudimentary approaches based on covariance matrix dependence model, a copula based approach can treat skewed dependence structures in which the strength of dependence is allowed to vary depending on location in the parameter space.

## Sampling Copula

For simplicity, this section demonstrates how to draw correlated samples from bivariate copula. Sampling from a bivariate copula is achieved by first defining a conditional distribution function and then applying the inverse probability integral transform. Let  $h$  represent the conditional distribution of  $u_1$  given all other random variables  $\{u_2, \dots, u_d\}$ . In the two dimensional case  $h$  is given by equation 2.13 [10]:

$$h(u_1|u_2) = \frac{\partial C(u_1, u_2)}{\partial u_2} \quad (2.13)$$

If the distribution  $h$  is smooth and monotonic the inverse  $h^{-1}$  exists. For a bivariate Gaussian copula with a shape parameter  $\theta_c = 0.7$  the conditional distribution functions are shown in figures 2.4 and 2.5 for several values of the conditioning variable ( $u_2$ ).

Computing the inverse analytically is oftentimes not possible for some classes of copula and therefore, the more general method shown in equation 2.14 is used. A random vector of length  $N$  is drawn from the uniform distribution  $\in [0, 1]$ :  $\{\mathbf{U}_2\}$ . For each sample,  $u_{2_i}$  in  $\{\mathbf{U}_2\}$  the 1-D line search problem given in equation 2.14 is solved. This produces a sample vector of length  $N$ :  $\{\mathbf{U}_1\}$ .

$$u_{1_i} = \operatorname{argmin}_{u_1} [h(u_1|u_{2_i}) - u_{2_i}], \text{ with } 0 < u_1 < 1 \quad (2.14)$$

The resulting correlated sample vectors  $\{\mathbf{U}_1, \mathbf{U}_2\} \in [0, 1]^2$  are distributed according to the copula,  $c$ , and have uniform margins. An example of random samples drawn from

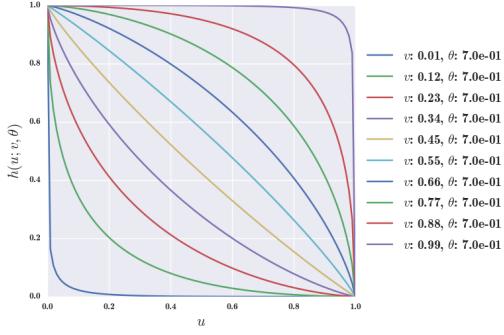


Figure 2.4: The conditional  $h$  function vs. value of the conditioning variable  $u_2$  for a Gaussian copula with  $\theta_c = 0.7$ .

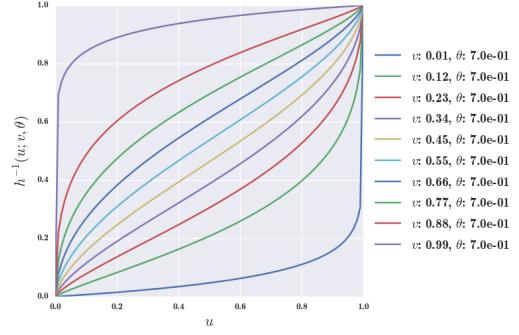


Figure 2.5:  $h^{-1}$  vs. value of the conditioning variable  $u_2$  for a Gaussian copula with  $\theta_c = 0.7$ .

a Gaussian copula are shown in figure 2.7. The smooth Gaussian copula PDF is provided in figure 2.6.

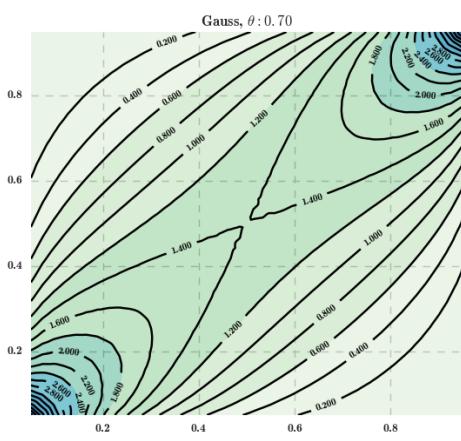


Figure 2.6: Gaussian copula density with  $\theta_c = 0.7$ .

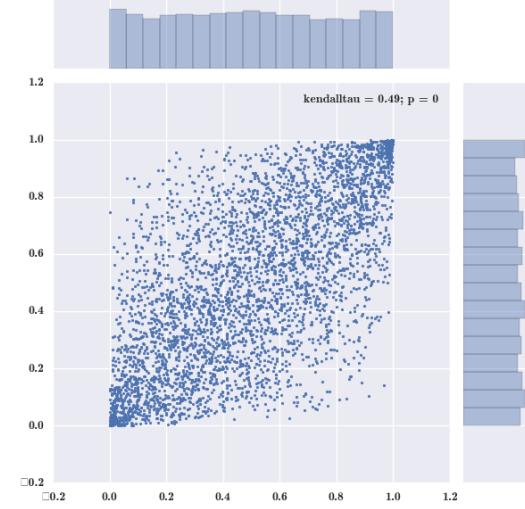


Figure 2.7: Samples drawn from Gaussian copula with  $\theta_c = 0.7$ .

To apply arbitrary margins  $F_1$  and  $F_2$  we employ, again, the inverse probability trans-

form. Correlated samples are then drawn according to:

$$\mathbf{X} = F_1^{-1}(\mathbf{U}_1) \quad (2.15)$$

$$\mathbf{Y} = F_2^{-1}(\mathbf{U}_2) \quad (2.16)$$

The sample vectors  $\{\mathbf{X}, \mathbf{Y}\}$  are distributed according to the joint density,  $C(F_1, F_2)$ . An example bivariate sample set with exponentially distributed margins and a Gaussian copula is shown in figure 2.8. In the example figure both margins follow an exponential distribution given by  $f(x) = \lambda e^{-\lambda x}$  with  $\lambda = 2E-3$ .

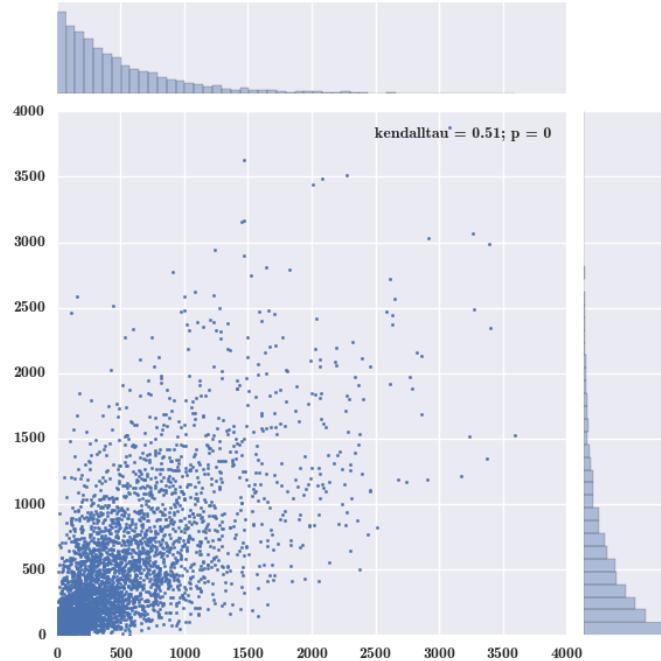


Figure 2.8: Samples drawn from Gaussian copula with exponential margins.

## Copula Families

A wide range of copula functions are available in the literature. In order to satisfy the definition of a copula several criteria must be met:

1. Must integrate to one on  $[0, 1]^n$

2. Must have uniform marginal distributions (as shown in figure 2.7).
3. When one argument to the joint copula CDF is zero, the CDF is zero:

$$C(u_1, u_2, \dots, 0, \dots, u_d) = 0 \quad (2.17)$$

4. When one argument to the joint copula CDF is  $u \in [0, 1]$  and all other arguments are one, the CDF takes a value equal to  $u$ :

$$C(1, 1, \dots, u, \dots, 1) = u \quad (2.18)$$

Examples of valid copula are given in figure 2.9. A wide range of skewed dependence structures can be represented by considering only a few copula families. Each copula can be rotated to accommodate both positive or negative dependence.

### 2.2.3 Fitting Copula

Fitting copula to empirical data can be carried out by the method of maximum likelihood (ML). Consider the bivariate case where  $N$  sample pairs,  $\{w_i, v_i\}_{i \in [1, N]}$  are known. The likelihood function for a copula is given by equation 2.19. The likelihood is a function of the distribution parameter  $\theta_c$  with the data,  $\{\mathbf{w}, \mathbf{v}\}$ , held fixed. When integrated over all possible parameter values the integrated result does not necessarily take on a value of unity and therefore cannot be strictly interpreted as a probability density. Each constituent factor in the likelihood function can be interpreted as the relative likelihood that the sample pair  $\{w_i, v_i\}$  arose from the copula density function with parameter  $\theta_c$ .

$$\mathcal{L}(\theta_c; \mathbf{w}, \mathbf{v}) = \prod_{i=1}^N c(w_i, v_i | \theta_c) \quad (2.19)$$

Where  $\theta_c$  is the free copula shape parameter. Typically the negative log-likelihood,  $-\ln \mathcal{L}$ , is used in when performing ML estimation of a distribution parameter since the

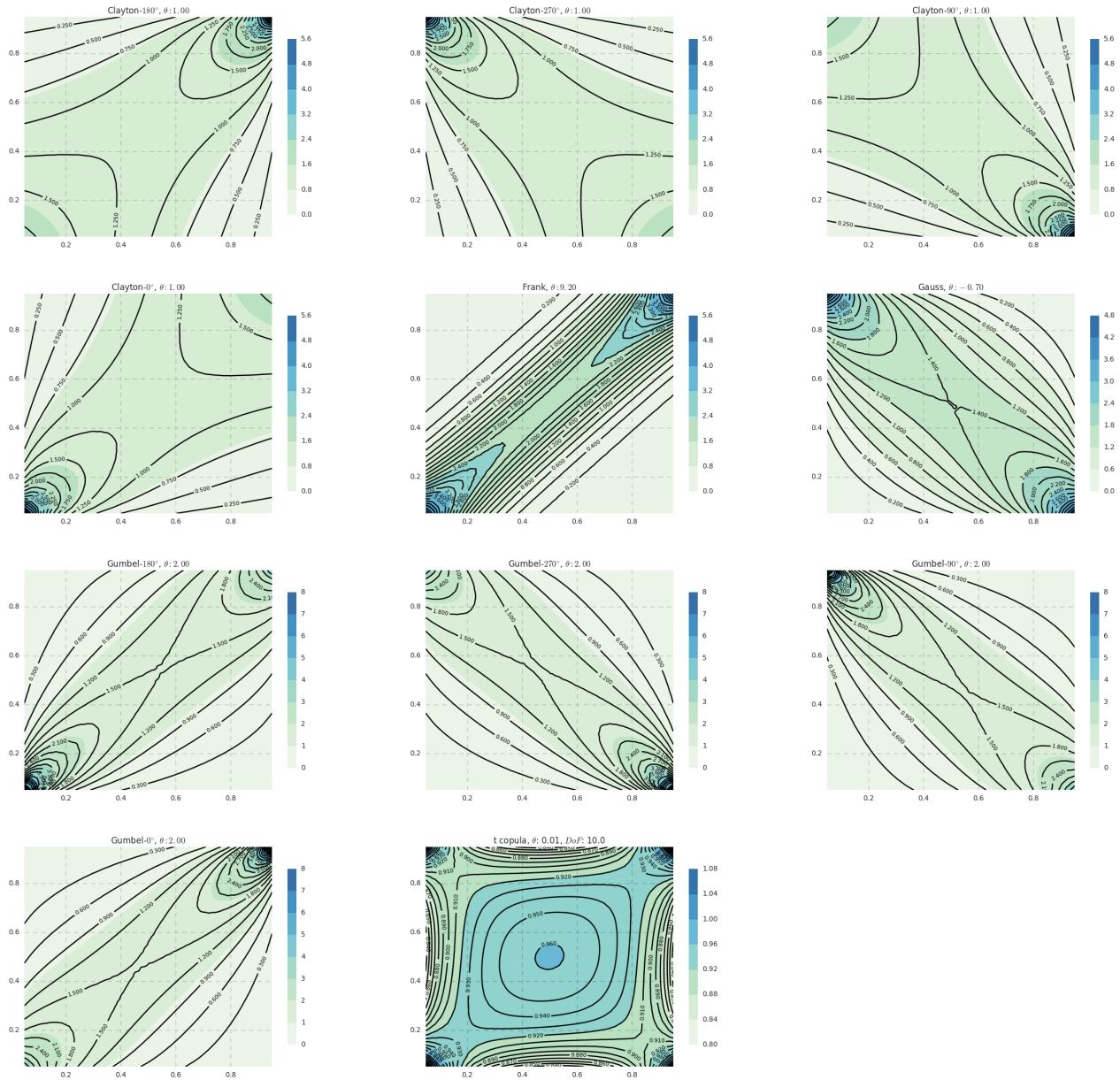


Figure 2.9: Examples of bivariate copula PDFs.

problem is typically cast in terms of a minimization problem.

$$\hat{\theta}_{c,ML} = \operatorname{argmin}_{\theta_c} [-\ln \mathcal{L}(\theta_c; \mathbf{w}, \mathbf{v})] \quad (2.20)$$

To minimize the negative log likelihood in equation 2.20 one computes the partial derivative with respect to  $\theta_c$  and finds the value  $\hat{\theta}_{c,ML}$  for which this expression reaches zero. This can be carried out by Newton's method. If the partial derivatives of the copula's negative log likelihood are impossible to compute analytically one can estimate them by finite difference.

In order to determine which copula family best represents the data, an arsenal of statistical tests can be applied to select the copula which best fits the data. Here, we consider two methods: (1) Comparing Akaike information criterion (AIC) and (2) graphically comparing each fitted copula.

The AIC is computed by equation 2.21.

$$\text{AIC} = 2k - 2\ln(\mathcal{L}) \quad (2.21)$$

Where  $k$  is the number of free parameters in the model. The AIC penalizes models with larger numbers of parameters. Automated copula selection is achieved by selecting the copula that obtains the lowest AIC score.

A graphical method of copula selection was proposed by Barbe et. al. (1996) [11]. In this method each trial copula's Kendall's function,  $K_c(t)$  is plotted against an empirical estimate of this function,  $\hat{K}_c$ . Given  $d$  random variables  $\mathbf{U} = \{U_1, \dots, U_d\}$  distributed according to some  $d$  dimensional copula,  $C$ , Kendall's function is given by 2.22 [7].

$$K_c(t; C) = \operatorname{P}[C(\mathbf{U}) \leq t; \mathbf{U} \sim C] \quad (2.22)$$

For example, by graphical inspection of figure 2.11, the Gumbel copula is the best fit to the original data set. This visual process can be automated by computing and comparing  $\|\hat{K}_c(t) - K_c(t)\|$  for each trial copula.

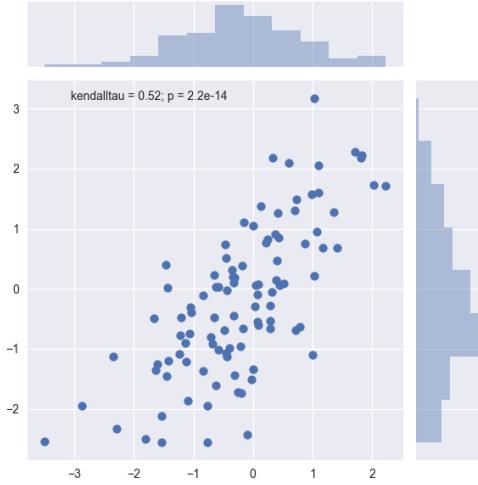


Figure 2.10: Fictitious bivariate data set.

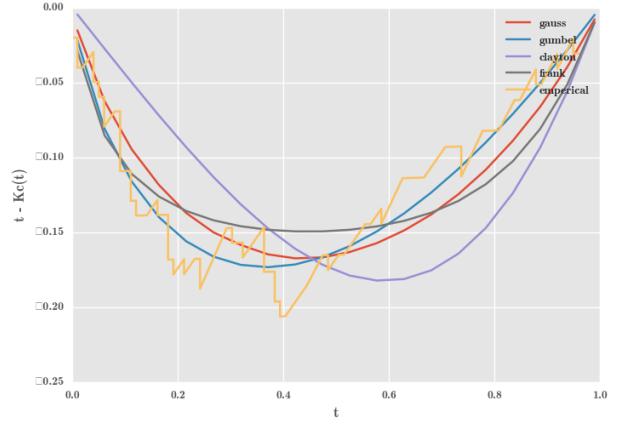


Figure 2.11: Graphical comparison of Kendall's distribution for several fitted copula.

## Kendall's Tau

Kendall's tau is a measure of concordance. Consider two correlated and uniformly distributed random variables,  $X, Y$ . Let  $(X_1, Y_1)$  and  $(X_2, Y_2)$  be identically distributed random vectors from some joint cumulative distribution so that individual samples from  $X$  and  $Y$  will take on values in  $[0, 1]$ , then Kendall's tau is given by equation 2.23 [10].

$$\rho_\tau = P[(X_1 - X_2)(Y_1 - Y_2) > 0] - P[(X_1 - X_2)(Y_1 - Y_2) < 0] \quad (2.23)$$

In the case of Archimedean copula,  $\rho_\tau$  is directly related to the copula's parameter,  $\theta_c$ . Equation 2.24 relates an Archimedean copula's parameter to  $\rho_\tau$ . This is useful since if one can estimate  $\rho_\tau$  from the empirical data and the copula type is known, one can quickly compute the copula's shape parameter without resorting to the method of ML.

$$\rho_\tau = 1 + 4 \int_0^1 \frac{\varphi(\theta_c, t)}{\varphi'(\theta_c, t)} dt \quad (2.24)$$

Where  $\varphi(t)$  is the copula's generator function and  $\varphi'(t)$  is the first derivative of the generator function with respect to  $t$ . A list of copula generator functions may be found in introductory copula texts [10].

## 2.2.4 Sample Quantiles

Here a non parametric representation of univariate distributions is given by utilizing a set of sample quantiles. Let the quantile function be represented by  $Q = F^{-1}$  where  $F$  is a cumulative density function (CDF).

The  $\tau^{th}$  quantile is  $q_\tau = Q(\tau)$ ; where  $F(t) = P[T \leq t]$ .  $\tau \in [0, 1]$ . The quantile loss function is given by equation 2.25 and depicted in figure 2.12.

$$\mathcal{W}_\tau(u) = u \cdot (\tau - \mathbb{I}_{(u<0)}) \quad (2.25)$$

Where  $\mathbb{I}$  is the indicator function which returns 1 if the argument is true and 0 otherwise. In order to estimate a sample quantile,  $\hat{q}_\tau$ , given the empirical CDF  $F$ , minimize:  $\mathbb{E}[\mathcal{W}_\tau(T - q_\tau)]$  where  $T$ , the temperature, is treated as a random variable distributed according to  $F_T$ . Considering a sample set  $\{T_0, T_1, \dots, T_N\}$  the desired quantile  $q_\tau$  may be estimated by equation 2.26.

$$\begin{aligned} \hat{q}_\tau &= \operatorname{argmin}_{q_\tau} \mathbb{E}[\mathcal{W}_\tau(u)]; \quad u = T - q_\tau \\ &\approx \operatorname{argmin}_{q_\tau} \frac{1}{N} \sum_i^N \mathcal{W}_\tau(u_i); \quad u_i = T_i - q_\tau \\ &\approx \operatorname{argmin}_{q_\tau} \left[ (1 - \tau) \sum_{T \leq q_\tau} (T - q_\tau) - \tau \sum_{T > q_\tau} (T - q_\tau) \right] \end{aligned} \quad (2.26)$$

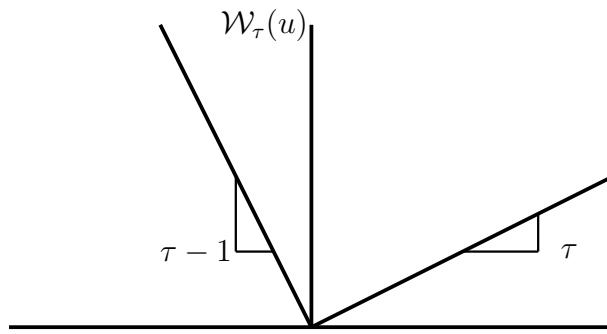


Figure 2.12: Quantile loss function.

In this work, a set of sample quantiles,  $\hat{\theta}_\tau = \{\hat{q}_{\tau_0}, \dots, \hat{q}_{\tau_i}, \dots, \hat{q}_{N_Q}\}$ , are used to construct step-wise cumulative distributions,  $\hat{F}$ , for the cladding outer temperature and

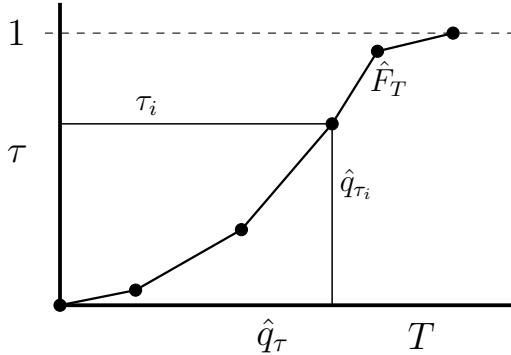


Figure 2.13: Piecewise linear CDF interpolated from a set of quantiles.

near-wall TKE surface fields.  $N_Q$  denotes the number of quantiles used in the CDF reconstruction and is a user set parameter. The stepwise quantile function with prescribed quantiles is given by equation 2.27 and depicted in figure 2.13.

$$\hat{F}_T = Q^{-1}(T; \{\hat{\theta}_\tau\}) \quad (2.27)$$

The reconstructed CDF can then in turn be used to build a histogram. In place of the stepwise representation, a monotone piecewise cubic hermite interpolating polynomial (PCHIP) may be fit to the stepwise conditional quantile distribution to generate a differentiable CDF. The PCHIP interpolation preserves monotonicity of the CDF if the provided quantiles are strictly monotone [12]. This condition is enforced in the software; any violation of the monotone restriction would indicate a software bug in the quantile regression code. Inverse transform sampling is used to draw samples from the univariate distributions.

Other strategies to approximate univariate density functions from a set of sample statistics are abundant. One alternative to quantile regression is to relate sample moments to the parameters of some parametric distribution. In this procedure a set of moment conditions is defined for the target parametric model. Then the method of moments (MoM) can be used to determine the parameters of the parametric model which best satisfy the moment conditions in the L2 norm sense. The MoM is applicable when

the number of defined moment conditions is greater than or equal to the number of free distribution parameters.

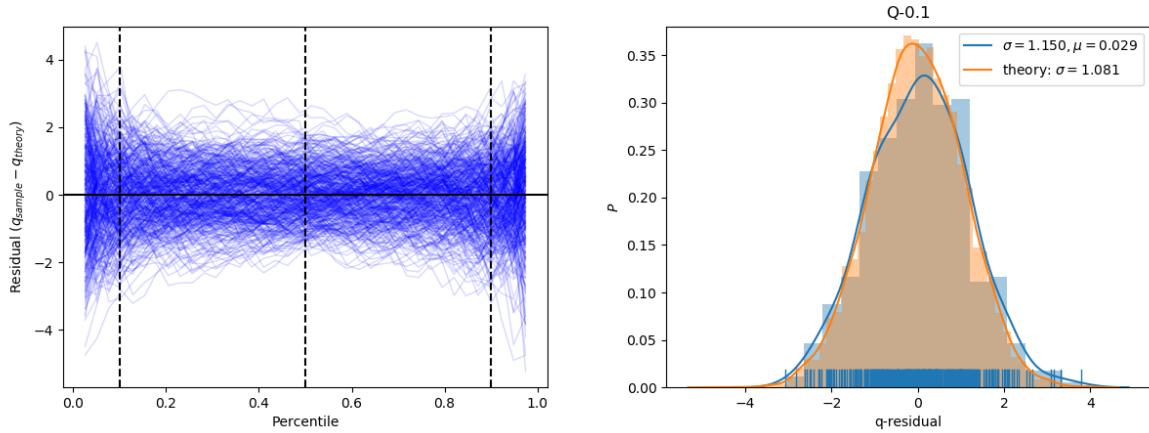
Alternatively, one may construct a non-parametric univariate density given some set of predictive features by utilizing a set of predicted distribution cumulants. The estimated cumulants, produced by evaluating a trained machine learning model, may be used to build an Edgeworth series expansion [13]. It remains as future work to determine if cumulants or traditional moments behave in a predictable manner as a function of local core conditions.

It can be shown that the sample quantiles are distributed according to equation 2.28. This directly follows from the distribution of the order statistics [14]. The theoretical distribution of the quantiles is Gaussian in the large sample limit. The large sample behavior of the quantiles is demonstrated for a normal random variable in figure 2.14 and a beta random variable in figure 2.15. When benchmarking or validating quantile predictions it is essential to check that the predictions follow the expected Gaussian behavior. For the quantile regression procedures employed in this work, this check is performed in section 2.2.5. Deviance from the expected behavior would indicate additional model-induced quantile prediction variance or if skewness is observed, bias introduced by the quantile regression model. In the current application it is important to estimate uncertainties inherent to the sample quantile estimation procedure so such uncertainties may be propagated into the crud integration procedures.

$$\begin{aligned} q_p &\sim \mathcal{N}\left(F_T^{-1}(p), \sigma_{q_p}^2\right) \\ \sigma_{q_p}^2 &= \frac{p(1-p)}{n[f_T(F_T^{-1}(p))]^2} \end{aligned} \quad (2.28)$$

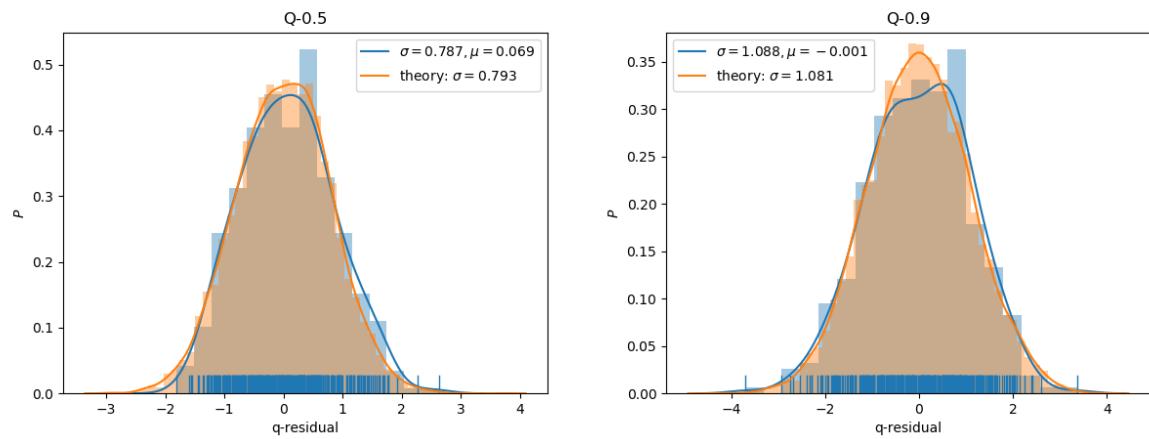
In equation 2.28 the variance associated with a sample quantile depends on the CDF and PDF of the true distribution function of interest. The underlying CDF from which a sample population is typically unknown and the goal is often to infer properties of this

distribution from the sample population. Therefore, it is not possible to employ equation 2.28 directly to compute the variance of a given sample quantile unless the underlying CDF is known. Note that this is indeed the case in section 2.2.5 where the large sample quantile theory is employed to check the residuals of a quantile regression procedure because the underlying distribution of the data was prespecified to be Gaussian. However, from more complex cases where the nature of the distribution functions is unknown, such as with raw CFD data, equation 2.28 is not directly useful but still can be used to aid interpretation of the quantile regression predictions.



(a) Empirical quantiles for a normal distribution.  
500 trials shown. 0.1, 0.5, 0.9 quantiles  
denoted by dashed verticle lines.

(b) Empirical vs theoretical 0.1 quantile  
distribution.



(c) Empirical vs theoretical 0.5 quantile  
distribution.

(d) Empirical vs theoretical 0.9 quantile distribution.

Figure 2.14: Distribution of quantiles for a normally distributed RV:  $X \sim \mathcal{N}(0, 10)$ .  
Theoretical quantile standard deviation given by equation 2.28.

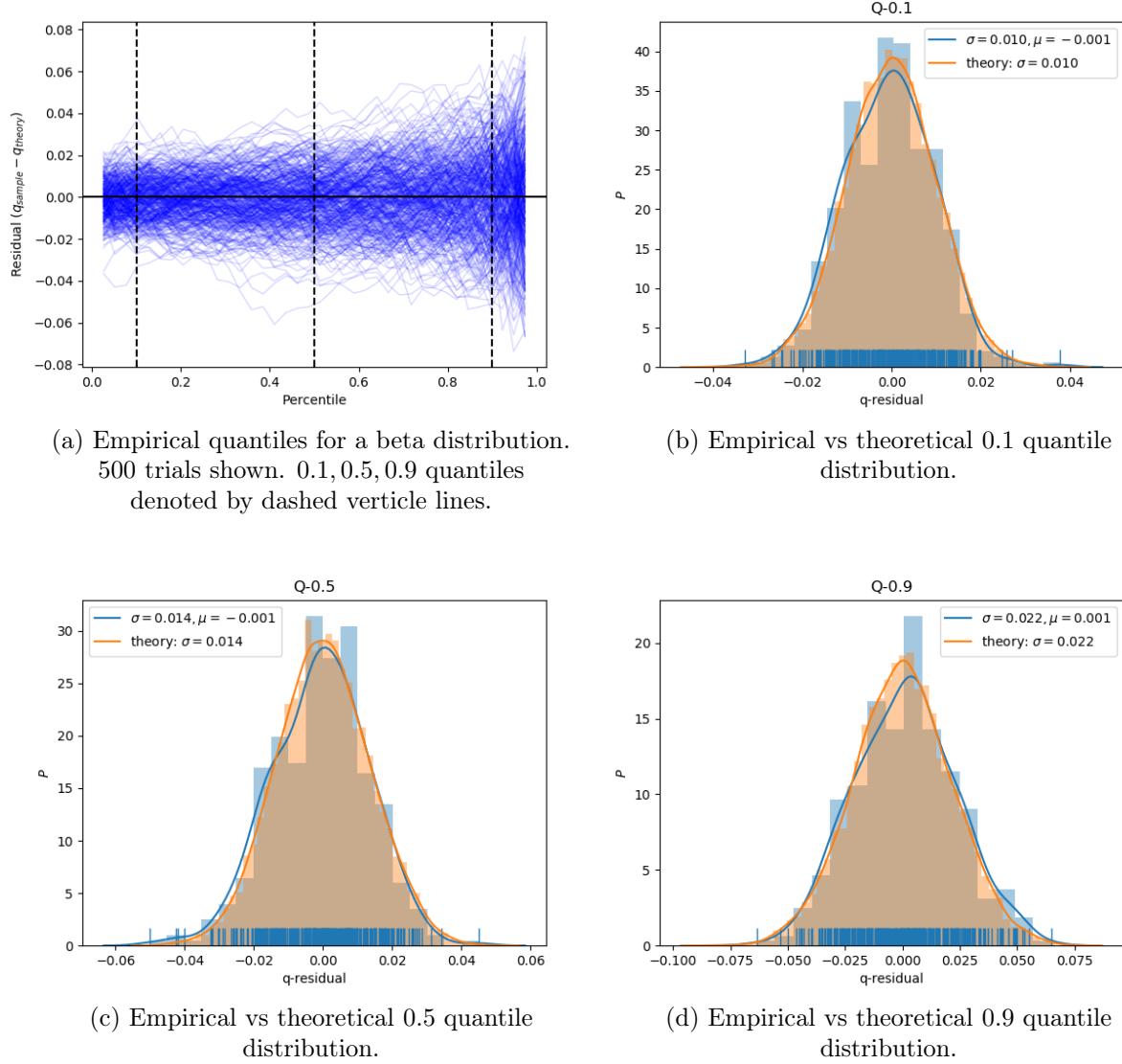


Figure 2.15: Distribution of quantiles for a beta distributed RV:  $X \sim \beta(2, 5)$

The sampled temperatures may be tallied over each CTF face to estimate the fractional area that exceeds some threshold temperature. The probability of exceeding a threshold temperature is shown in equation 2.29.

$$p_e = Pr(T > T^*) = 1 - \int_0^{T^*} f_T dT \quad (2.29)$$

Let  $q_p = F_T^{-1}(1-p_e)$  denote the quantile associated with the threshold probability,  $p_e$ .

$F_T^{-1}$  is the inverse CDF function and  $f_T$  is the probability density function of temperature on the patch.

The sample quantile corresponding to  $p_e$  is distributed according to:

$$q_p \sim \mathcal{N}\left(F_T^{-1}(p), \sigma_{q_p}^2\right) \quad (2.30)$$

$$\sigma_{q_p}^2 = \frac{p(1-p)}{n[f_T(F_T^{-1}(p))]^2} \quad (2.31)$$

The variance of upper tail probability mass estimate can be found by standard propagation of uncertainty principles:

$$\sigma_p^2 = \left(\frac{\partial p_e}{\partial q_p}\right)^2 \cdot \sigma_{q_p}^2 + HOT. \quad (2.32)$$

Where

$$\begin{aligned} \frac{\partial p_e}{\partial q_p} &= \frac{\partial}{\partial q_p} \left(1 - \int_0^{q_p} f_T dT\right) \\ &= \frac{\partial}{\partial q_p} (-F_T(q_p) + F_T(0)) \\ &= -f_T(q_p) \end{aligned} \quad (2.33)$$

This dictates that estimates of extreme upper tail integrals carry large relative uncertainties.

### 2.2.5 Gradient Boosted Regression Trees

A data driven machine learning model is used to predict quantiles,  $\{\hat{\theta}_\tau\}$ , required to reconstruct cumulative density functions as shown in equation 2.27. The desired quantiles of the cladding surface temperature and TKE fields are conditioned upon the local core state denoted here by  $\mathbf{p}$ . The vector  $\mathbf{p}$  contains the local thermal hydraulic conditions in the core and local geometric grid and pin factors. The purpose of this vector is to uniquely identify each CTF face in the core though some set of explanatory features. The problem is then of conditional quantile prediction and this is performed by gradient boosted regression trees in this work.

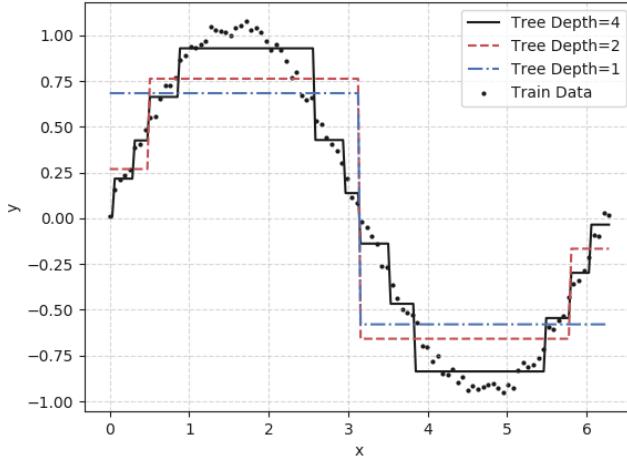


Figure 2.16: Single CART regression tree stump comparing a fit of depth 1 and 2 to the function  $y = \sin(x) + \varepsilon$ ,  $x \in [0, 2\pi]$ ,  $\varepsilon \sim \mathcal{N}(0, 0.001)$ .

The modern gradient boosting algorithm was developed by Friedman et. al. (1998) [15], [16]. The development of generalized boosting was significant because it re-envisioned previous boosting algorithms such as AdaBoost as special cases of gradient boosting with specific loss functions. Gradient boosting is a numerical optimization procedure carried out in function space with the goal of finding a function  $\mathcal{F}_M$  that maps the inputs  $\mathbf{p}$  to  $y$  where  $\mathcal{F}_M$  is given by equation 2.34.

$$\mathcal{F}_M = \operatorname{argmin}_{\mathcal{F}} \mathbb{E}_{y, \mathbf{p}} (L(y, \mathcal{F}(\mathbf{p}))) \quad (2.34)$$

Where  $L(\cdot)$  is a differentiable loss function.  $\mathbf{p}$  are known as predictive features and  $y$  is the response. The paired set  $\{\mathbf{p}, y\}$  is referred to as the training data set. In the standard gradient boosting algorithm the functional form of  $\mathcal{F}$  is chosen to be an additive model of the form 2.35:

$$\mathcal{F}_M(\mathbf{p}, \gamma, \mathbf{a}; \mathbf{b}) = \sum_{m=0}^M \gamma_m h_m(\mathbf{p}, a_m; b_m) \quad (2.35)$$

Where  $M$  is the number of constituent sub-models and  $\{\gamma_m, a_m\}$  are coefficients and free parameters requiring fitting in each sub-model.  $b_m$  represent sub-model hyperparameters

that are fixed at user set values. Rather than fitting all sub-models simultaneously, gradient boosting greedily fits the sub-models to the gradient of the loss function in a stage wise fashion as shown in algorithm 2. The loop depicted in line number 8-10 performs the computation of the loss function gradient at each boosted iteration  $m$  in the algorithm. In the literature the vector of gradients is sometimes referred to as the pseudo residuals vector since if the loss function is taken to be the L2 loss, the gradient of the loss is proportional to the residual vector as shown in equation 2.36.

$$\begin{aligned} (y_i - \mathcal{F}_m(x_i)) &\propto \frac{\partial[(y_i - \hat{y}_i)^2]}{\partial \hat{y}_i} \\ &\propto \frac{\partial(y_i^2 + \hat{y}_i^2 - 2y_i\hat{y}_i)}{\partial \hat{y}_i} = 2\hat{y}_i - 2y_i \end{aligned} \quad (2.36)$$

Each sub-model,  $h_m$ , is referred to as a weak learner and is defined to be a typical classification or regression tree (CART) depending on the problem context. Methods for fitting, pruning the decision trees and optimizing the numerical implementation of finding the best splits when fitting the tree to the pseudo-residuals are left to Friedman et. al. [17]. Provided CART trees are used for the weak learners, the free (sub-) model parameters  $a_m$  take the form of split locations and regional constants which are fitted to produce a piecewise constant predictor. In this case the user set hyperparameters,  $b_m$ , are taken to be the maximum tree depth allowed when fitting each weak learner to the pseudo-residuals.

Examples of single dimensional fitted CART trees with a depth of 1 and 2 are shown in figure 2.16. As the depth of a tree is increased the decision tree prediction converges onto the data; however, a single decision tree of large depth is prone to over fitting the data. Decision trees produce a piecewise constant prediction, and therefore since the final boosted model is a linear combination of decision tree models (weak learners), the fitted boosted model is piecewise constant.

---

**ALGORITHM 2**Gradient boosting algorithm [17].

---

- 1: **Initialization**
  - 2: (1) Training set  $\{(p_i, y_i)\}_{i=1}^n$ .
  - 3: (2) Differentiable loss function  $L(y, \mathcal{F}(p))$ .
  - 4: (3) Number of iterations  $M$ .
  - 5: (4) Initialize model with a constant value:  $\mathcal{F}_0(p) = \arg \min_{\gamma} \sum_{i=1}^n L(y_i, \gamma)$ .
  - 6: **for**  $m = 1$  to  $M$  **do**
  - 7:   Compute the pseudo-residuals:
  - 8:   **for**  $i = 1, \dots, n$  **do**
  - 9:      $r_{im} = -\frac{\partial L(y_i, \mathcal{F}_{m-1}(p_i))}{\partial \mathcal{F}_{m-1}(p_i)}$
  - 10:   **end for**
  - 11:   Fit a weak learner  $h_m(p; a_m)$  to pseudo-residuals,  $r_m$ :  
     
$$h^* = \arg \min_{a_m} (\|h_m(p; a_m) - r_m\|)$$
  
     Training data set is  $\{(p_i, r_{im})\}_{i=1}^n$
  - 12:   Compute multiplier  $\gamma_m$ :  $\gamma_m = \arg \min_{\gamma} \sum_{i=1}^n L(y_i, \mathcal{F}_{m-1}(p_i) + \gamma h_m^*(p_i))$
  - 13:   Update the model:  $\mathcal{F}_m(p) = \mathcal{F}_{m-1}(p) + \nu \gamma_m h_m^*(p)$ .
  - 14: **end for**
  - 15: Output  $\mathcal{F}_M(p)$ .
- 

Where  $\nu$  is a tunable constant in  $(0, 1]$  called the learning rate. A value of  $\nu < 1$  reduces the contribution of each weak learner in the final model. Reducing the learning rate increases resilience to over fitting but results in a proportional increase in the number of boosted iterations to achieve the same level of convergence of the boosted tree chain. Smaller values for the learning rate result in performing smaller steps in function space so we are less likely to overshoot the optimal function. Provided boosting produces a stage-wise additive model, overshoots are problematic since the final model carries memory of previous iterations. A typical learning rate is  $\nu \leq 0.05$  but a balance between computation time and model prediction accuracy on a testing set should be considered when setting this value.

For gradient boosted quantile regression, the loss function given by equation 2.25 is

substituted into algorithm 2.

$$L(y, \mathcal{F}(p); \tau) = \left[ (1 - \tau) \sum_{y \leq \mathcal{F}(p)} (y_i - \mathcal{F}(p)) \right] - \left[ \tau \sum_{y > \mathcal{F}(p)} (y_i - \mathcal{F}(p)) \right] \quad (2.37)$$

Where  $\tau$  is the user set quantile of interest. For instance, if the 95% percentile is desired then  $\tau = 0.95$ .

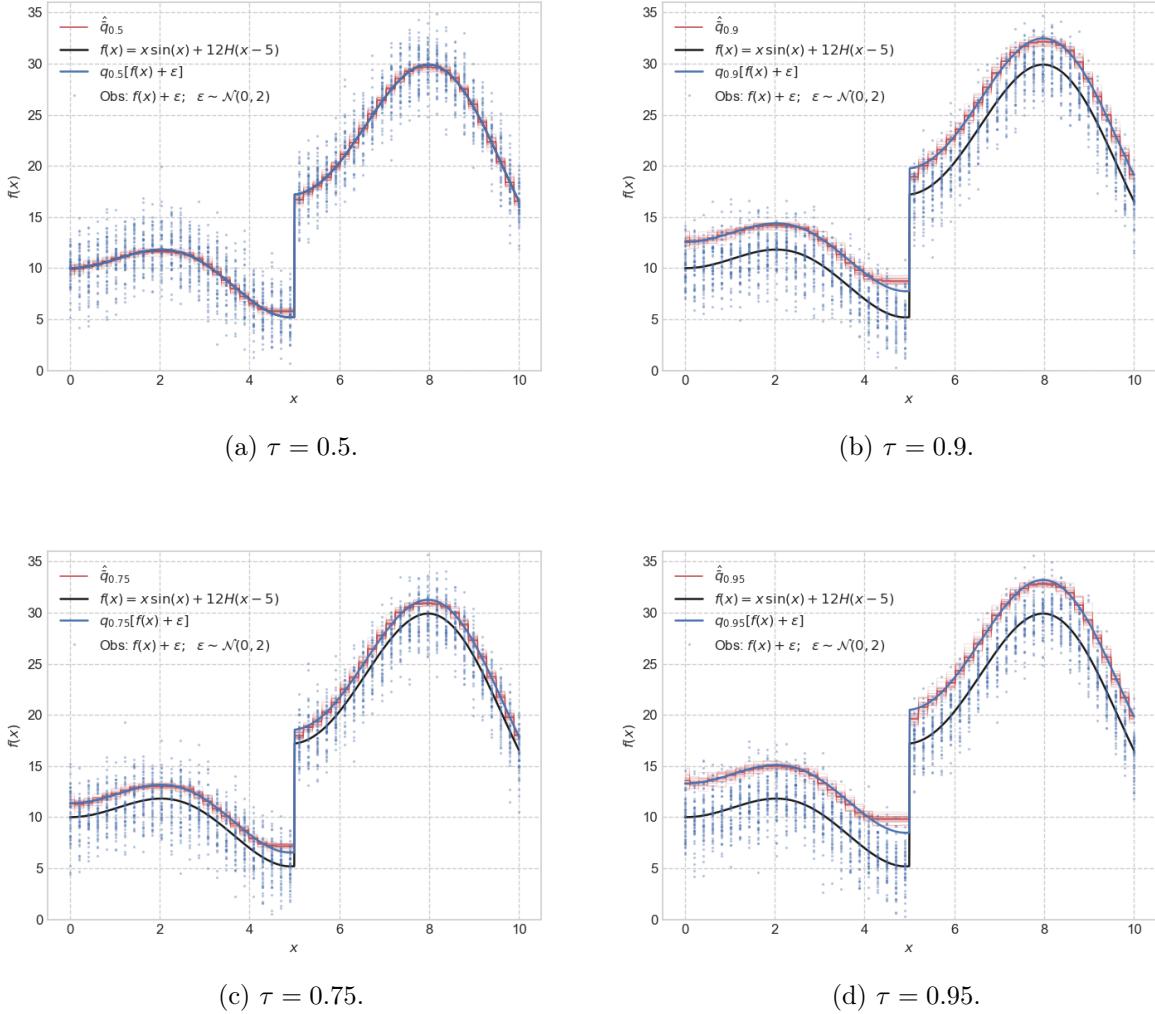


Figure 2.17: Gradient boosted quantile regression example.

Two candidate gradient boosting implementations were evaluated for use in this work. A test problem was constructed to check that the standard gradient boosting regression

technique is resilient to discontinuities in the data since the response fields are expected to abruptly change when moving across a spacer grid. The function  $y = x\sin(x) + 12\mathcal{H}(x - 5) + \varepsilon$  with  $x \in [0, 10]$  was used for testing.  $\mathcal{H}$  denotes the Heaviside function. Synthetic noise,  $\varepsilon \sim \mathcal{N}(0, 2)$ , was applied to the piecewise smooth test function. 5000 total samples were drawn from the test function. The test data was spatially aggregated to 100 axial levels to mimic CTF axial grid spacing. Four separate quantile regressors ( $\tau = \{0.5, 0.75, 0.9, 0.95\}$ ) were then fit to the data by algorithm 2. The predicted quantiles, conditioned on  $x$ , were compared to the expected results, as shown in figure 2.17. For this problem results from the scikit-learn gradient boosting implementation are compared to a custom boosting implementation, named *pCRTree*, developed specifically to solve the classification and quantile regression problems in this work. The residuals provided figure 2.18 show that the custom implementation performs similarly to the well-tested scikit-learn implementation with both models agreeing with the theoretical large-sample quantile distributions provided by equation 2.28.

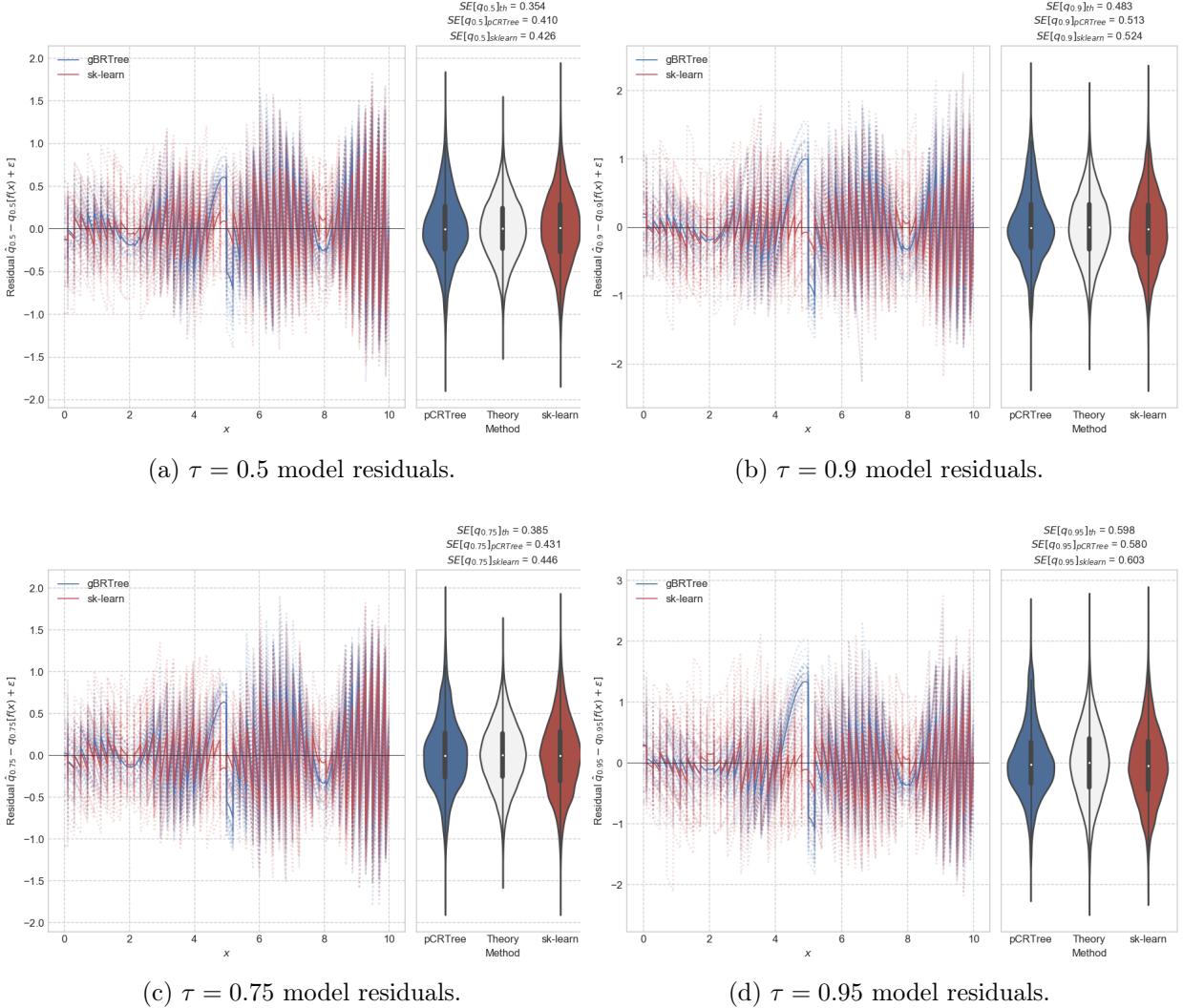


Figure 2.18: Gradient boosted quantile regression residual summary. The theoretical residual distribution were computed according to equation 2.28.

The parametric copula family which corresponds to a given set of local core conditions must also be determined in order to fully specify the copula required in the definition of the joint density function given in equation 2.5. This gives rise to a typical supervised classification problem which will be solved using the gradient boosting method. The family of copula,  $\Theta_c$ , i.e. either Frank, Clayton, or Gumbel, should be determined by the classifier provided a local core thermal hydraulic state vector  $\mathbf{p}$ . The gradient boosted classification algorithm can be recovered by substituting the exponential loss function

shown in equation 2.38 into algorithm 2.

$$\begin{aligned} L(y, F(\mathbf{p})) &= \mathbb{E} [e^{-yF(\mathbf{p})}] \\ &= \frac{1}{N} \sum_i^N e^{-y_i F(\mathbf{p}_i)} \end{aligned} \quad (2.38)$$

Where  $y$  takes on integer values in  $\{-1, 1\}$  for the classification problem.

A two-class test case was devised to validate the custom boosting implementation against the scikit-learn implementation. This is meant as a demonstration for qualitative understanding of the boosted classification algorithm output. In this problem red and blue points were emplaced in a two dimensional space as shown in figure 2.19. These predefined points form the necessary training data for the test problem. The goal of this test is to segregate the input space,  $\{x, y\}$ , into regions which are most likely to contain only one color of points. Results from the test 2-D classification test problem are shown in the figure. In much the same way regression trees produce piecewise constant predictions, the boosted classifier partitions the input space by segregating the space along orthogonal splits.

Classifier predictions are made by tallying the weighted predictions of each constituent fitted classification weak learner (CART tree) in the boosted model. By tallying the predictions of all trees in the model one can obtain a probability mass function over all possible classes. The probability mass function is visualized in figure 2.19(b) wherein the predicted probability of the “blue” class obtained from the fitted boosted model is shown. The most likely class at each requested point is taken as the output of the classification model.

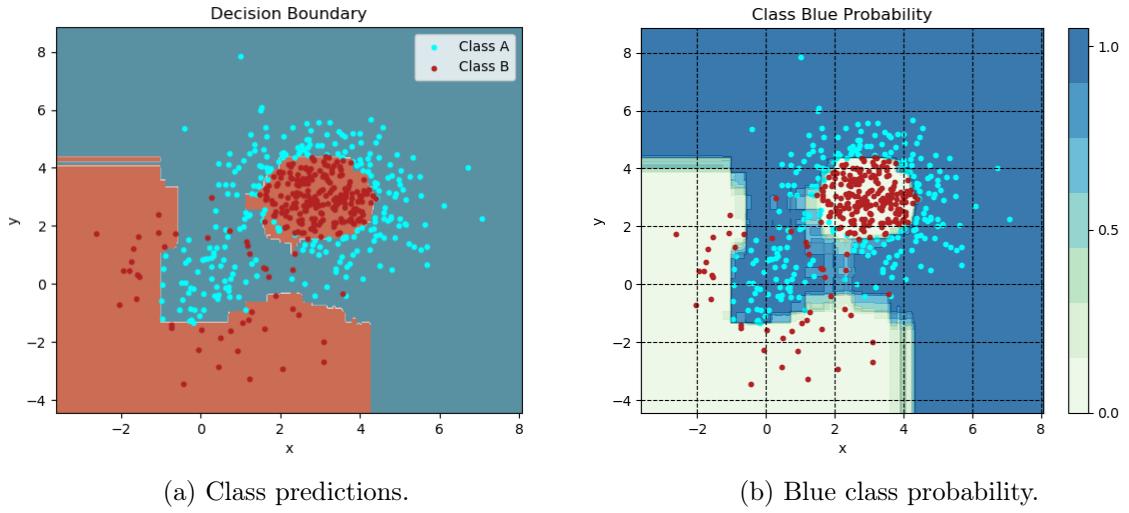


Figure 2.19: Two-class gradient boosted classifier example.

Finally, gradient boosting can be used to estimate the relative explanatory power of each predictive variable included in the model. This is done by tallying the number of times a particular dimension is split upon in the CART fitting process and weighting these splits by a gain measure. The gain can be interpreted as the benefit of making a particular split in the decision tree as measured by improvement in  $R^2$  for regression problems, or improvement in class purity as measured by the Gini impurity ratio for classification problems. Finally, a weighted sum of the split gains of each tree is performed over all boosted iterations to obtain relative variable importances. Friedman provides a detailed explanation of this relative variable importance computation [16]. This ability of gradient boosting to identify important predictors has been exploited in email filter and web page ranking applications [18], [19]. In the case of copula prediction, exogenous variables which are extraneous can be detected and eliminated from the model to save computation time.

### 2.2.6 Monte Carlo Crud Estimation

Before applying a sampling procedure to the temperature and TKE surface fields on a given CTF face, the joint density function is reconstructed from copula and marginal distributions. On CTF face  $j$ , the gradient boosted model is queried for the conditional quantiles and the copula shape parameter:

$$\begin{aligned}\hat{\theta}_j &\leftarrow \mathcal{F}_M(\mathbf{p}_j, \mathbf{z}_j) \\ \hat{\theta}_j &= \{\hat{\theta}_{j,c}, \hat{\theta}_{j,\{T,k\}}\}\end{aligned}\quad (2.39)$$

The CTF face index is dropped in the remainder of this section to reduce subindex clutter.

Provided estimates for the conditional quantiles,  $\hat{\theta}_{\{T,k\}}$  in each face, the margins are defined by their quantile functions as shown in equation 2.27. After separate reconstruction of the margins and applying the boundary heat flux independence assumption shown previously in equation 2.5, equation 2.40 defines the joint density.

$$h(T, k, q'') = f_T(T; \hat{\theta}_T) f_k(k; \hat{\theta}_k) f_{j,q''}[c_{T,k}(F_T(T; \hat{\theta}_T) F_k(k; \hat{\theta}_k); \hat{\theta}_c)] \quad (2.40)$$

After reconstruction of the joint temperature, boundary heat flux and turbulent kinetic energy distribution on each patch, the goal is to draw samples from the copula based distribution with arbitrary margins. This is accomplished through the methods outlined by section 2.2.2 in equation 2.14.

With the ability to draw samples from a multivariate distribution now established, the Monte Carlo approximation of integral 2.3 follows and is represented by equation 2.41. Let  $X = \{T, k, q''\}$ .

$$\mathbb{E}(\mathcal{G}(X)) \approx \frac{1}{N} \sum_i^N \mathcal{G}(X_i), \quad \mathbf{X} \sim h \quad (2.41)$$

Rather than sampling from the density function  $h$ , one may draw samples from an alternate proposal density distribution denoted,  $\tilde{h}$ , and appropriately weight the samples by the probability ratio of the original density to the proposal density,  $h/\tilde{h}$ , so to avoid introducing bias in the approximation of the expected value. This leads to the importance sampling formulation of approximating integral 2.3 which is given in equation 2.42.

$$\mathbb{E}(\mathcal{G}(X)) \approx \frac{1}{N} \sum_i^N \mathcal{G}(X_i) \frac{h(X_i)}{\tilde{h}(X_i)}, \quad \mathbf{X} \sim \tilde{h} \quad (2.42)$$

In principal the proposal distribution,  $\tilde{h}$ , may be radically different from the target density,  $h$ . In practice computing the optimal choice for the proposal density distribution is non-trivial. The generic original and proposal densities can be written simply as shown in equation 2.43.

$$\begin{aligned} h(T, k, q'') &= c(F_T(T), F_k(k)) f_T f_k f_{j,q''} \\ \tilde{h}(T, k, q'') &= \tilde{c}(\tilde{F}_T(T), \tilde{F}_k(k)) \tilde{f}_T \tilde{f}_k f_{j,q''} \end{aligned} \quad (2.43)$$

$$\mathbb{E}(\mathcal{G}(X)) \approx \frac{1}{N} \sum_i^N \mathcal{G}(X_i) \omega(X_i), \quad \mathbf{X} \sim \tilde{c}(\tilde{F}_T(T), \tilde{F}_k(k)) \tilde{f}_T \tilde{f}_k f_{j,q''} \quad (2.44)$$

with the probability ratio of the target density to the proposal given by 2.45:

$$\omega_i = \frac{h_i}{\tilde{h}_i} = \frac{f_T(T_i) f_k(k_i) c(F_T(T), F_k(k))}{\tilde{f}_T(T_i) \tilde{f}_k(k_i) \tilde{c}(\tilde{F}_T(T), \tilde{F}_k(k))} \quad (2.45)$$

In some scenarios the proposal or target density is only known up to a constant, e.g.  $h^* = ch(X)$ . In this case the probability ratio is known up to a constant of proportionality and requires renormalization:

$$\mathbb{E}(\mathcal{G}(X)) \approx \frac{\sum_i^N \mathcal{G}_i(X) \omega_i(X)}{\sum_i^N \omega_i(x)}, \quad \mathbf{X} \sim \tilde{c}(\tilde{F}_T(T), \tilde{F}_k(k)) \tilde{f}_T \tilde{f}_k f_{j,q''} \quad (2.46)$$

This is known as self normalizing importance sampling. Traditional importance sampling may be applied in this case since  $h$  and  $\tilde{h}$  are properly normalized density functions in this hi2lo application which integrate to 1 over their respective support.

## 2.3 Propagating Crud Through Time

In the construction of the hi2lo method an assumption is made about the location of hot and cold spots on the rod surface as a function of time. The presence of hot and cold spots downstream spacer grids and the location on the rod surface these spots occupy are assumed to be principally governed by the geometry of the mixing vanes and geometric layout of the fuel and guide tubes. The influence of flow rate and core power on the relative location of the hot spots on the rod surface are assumed to be second order effects and are not explicitly captured by the hi2lo methodology at present.

### 2.3.1 Hot Spot Stationarity

The assumption that the location of eddy regions which develop downstream of spacer grids near the rod surface are principally governed by the grid geometry coupled with an assumption of steady state constant flow conditions leads to the notion of hot spot stationarity in time since the geometry of the core does not change throughout a cycle. To achieve a stable location of hot and cold spots on the rod surface through time, the hi2lo methodology establishes a mapping from the sample space containing all possible outcomes of surface temperature and TKE sample pairs to a location on the rod surface. To accomplish this, the order statistics of the joint temperate and turbulent kinetic energy distribution in each CTF patch are computed and the condition that the highest order statistic always falls on the same location within a given CTF patch is enforced.

Order statistics are well defined for a single dimensional random variable but in higher dimensions a consistent ordering is not possible. Therefore, as shown in equation 2.47 a convex combination of the surface fields with user specified weights is used to reduce

the random vector of T, TKE, and  $q''$  samples on any given CTF patch into a univariate random variable denoted by  $\mathbf{m} = \{m_0, m_1, \dots, m_N\}$ .

$$m_i = w_T \left( \frac{T_i - T_{min}}{T_{max} - T_{min}} \right) + w_k \left( \frac{k_i - k_{min}}{k_{max} - k_{min}} \right) + w_{q''} \left( \frac{q''_i - q''_{min}}{q''_{max} - q''_{min}} \right) \quad (2.47)$$

Where the sample remapping coefficients  $\{w_T, w_k, w_{q''}\}$  are set at runtime by the hi2lo model user and sum to 1:

$$w_T + w_k + w_{q''} = 1 \quad (2.48)$$

Next the order statistics of  $\mathbf{m}$  are computed such that  $\mathbf{m}' = \{m_{(0)} < m_{(1)} < \dots < m_{(N)}\}$ .

As shown in figure 2.20, the ordered samples are then emplaced on the CTF patch in an organized manner. The sample space is denoted by  $\Omega$  and a specific temperature, TKE, and boundary heat flux sample is denoted by  $\mathcal{F}_i$  as a single dot residing inside the sample space. The path taken on the patch surface is user controllable and is taken to be a simple serpentine left-to-right pattern in this work. Typical values for the remapping coefficients are  $w_T = 0.6, w_k = 0.4, w_{q''} = 0$ . With this setting, relatively high temperate and low TKE samples are likely to remain in the same location on the rod surface over multiple resampling events.

Since the crud simulation package utilized in this work is one-dimensional, the pattern chosen for sample emplacement has no influence on the integrated crud result over a patch. This would not be the case if the crud simulation package modeled a fully 3-D crud layer as the state of the neighboring crud nodes in that case would matter. Interestingly, since the hi2lo model user specifies the sample remapping pattern at run time, a physically realistic pattern could be prescribed on the rod surface - even prescribed as a function of local core conditions leading to a hybrid strategy between the current pure statistical hi2lo procedure and the spatial remapping procedure implemented by Salko et. al [20].

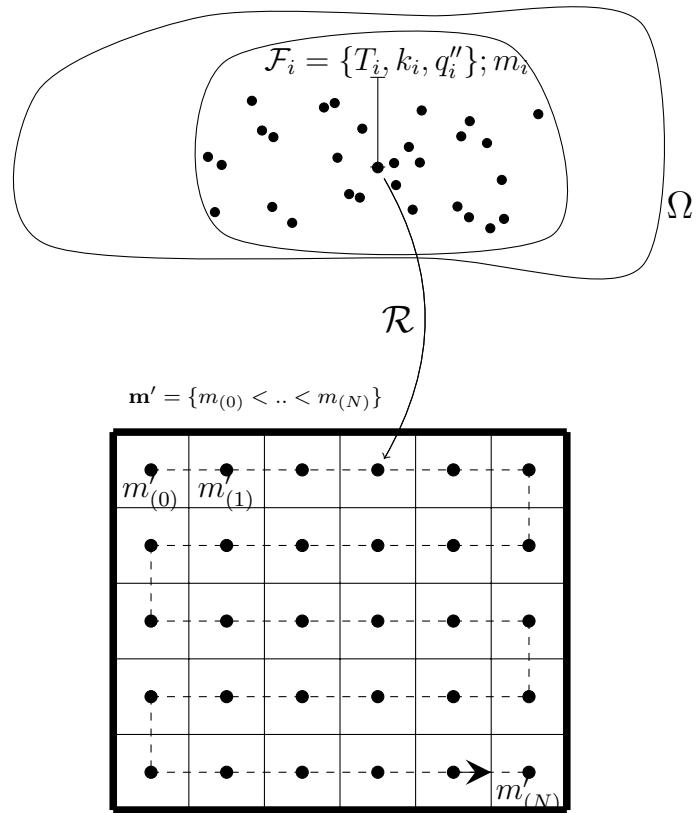


Figure 2.20: Sample remapping. A single CTF face is shown as the bold square with a grid overlay partitioning the patch surface. In this figure the number of samples per CTF face is  $N = 30$ .

### 2.3.2 Time Stepping Scheme

Propagating the importance weights through discrete resampling steps requires special attention to ensure bias is not introduced into the crud results when moving through multiple time steps and VERA states. The key observation is that the importance weights computed at a given resampling event should neither be double-counted nor discarded in subsequent resampling steps. Instead, the importance weights are time averaged to achieve the correct importance weights used to evaluate the total crud mass on the rod surface at any given time step (resampling step).

Figure 2.21 depicts the time stepping scheme for a single patch. The patch index is not shown to reduce visual clutter. The VERA state-point index is denoted by  $\ell$ . At the beginning of each VERA state-point the power distribution and thermal hydraulic conditions in the core change. Here it is assumed there is a constant power profile and constant flow conditions over a VERA depletion step.

In figure 2.21 dotted arrows represent sampling from a probability distribution.  $\mathcal{R}$  is the spatial reordering map from sample space to a position on the rod surface.  $\mathcal{G}$  represents the crud generation function provided by the crud simulation package which takes thermal hydraulic boundary conditions as input in addition to the previous crud state and produces a new crud state. Resampling events are performed at time intervals  $\Delta t_s$  and this interval is set at runtime by the user. The resample index is denoted by  $s$ . The subscript,  $(\cdot)_s$ , denotes evaluation at a resampling event. A prime-notated variable,  $(\cdot)'$ , represents a spatially re-mapped sample as in figure 2.20. At each resampling event the temperature, TKE and boundary heat flux samples are independently drawn from the reconstructed joint probability  $\hat{H}^\ell$ . Assume that the resample time step is constant and equal to  $\Delta t_s$ .

A dependency in time is introduced by the crud growth step because the current crud growth rate depends on the previous crud state (e.g. a severely thick crud deposit will behave differently than a small crud deposit). This time dependence is denoted by the

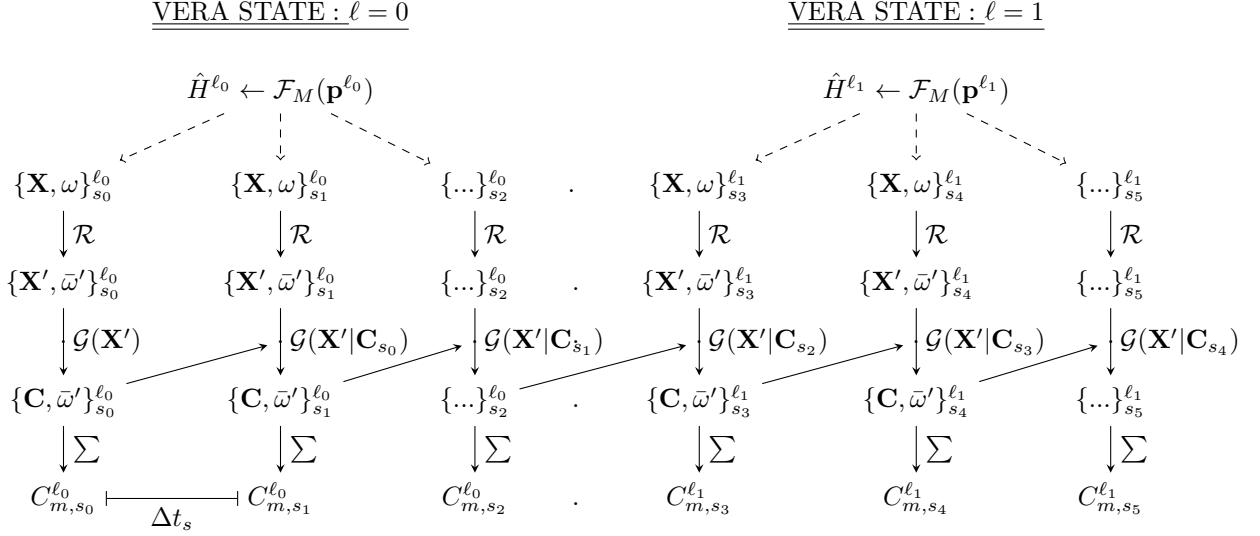


Figure 2.21: Multi-state point time stepping overview. Dashed arrows represent sampling from a distribution. Solid arrows represent a functional mapping or operation.

diagonal arrows linking the crud states together across resampling events in figure 2.21.

Since the resampling time step size is constant the time corresponding to any given sample event is computed by multiplying the resampling event index by the constant resampling step size:

$$t_{s_n} = n\Delta t_s \quad (2.49)$$

The time averaged sample weights are in general updated by equation 2.50 at each resampling step. Recall that the ratio of the target density and the sampling distribution density is denoted by  $\omega$  and is given in equation 2.45.

$$\bar{\omega}'_{s_n,i} = \left( \frac{\sum_{l=1}^{n-1} \Delta t_{s_l}}{\sum_l^n \Delta t_{s_l}} \right) \bar{\omega}'_{n-1,i} + \left( \frac{\Delta t_{s_n}}{\sum_l^n \Delta t_{s_l}} \right) \omega'_{n,i} \quad (2.50)$$

Where  $i$  is the sample index within a single CTF face.

After applying the constant resample step size assumption the time averaged sample weights are updated at each resampling step according to equation 2.51.

$$\begin{aligned}\bar{\omega}'_{s_n,i} &= \left( \frac{(n-1)\Delta t_s}{n\Delta t_s} \right) \bar{\omega}'_{n-1,i} + \left( \frac{\Delta t_s}{n\Delta t_s} \right) \omega'_{n,i} \\ &= \left( \frac{(n-1)}{n} \right) \bar{\omega}'_{n-1,i} + \left( \frac{1}{n} \right) \omega'_{n,i}\end{aligned}\quad (2.51)$$

After each resampling event the total crud mass,  $C_m$  on a CTF face, is computed by a weighted sum given in equation 2.52. The sampling and weighting procedure is depicted in figure 2.22.

$$C_m = \left( \frac{A}{\sum_i^N \bar{\omega}'_i} \right) \sum_i^N C'_i \bar{\omega}'_i \quad (2.52)$$

Where  $N$  is the number of samples drawn per CTF face and  $A$  is the surface area of the  $j^{th}$  CTF face. The re-stepping index  $n$  and the patch index  $j$  are dropped to reduce clutter in equation 2.52.

The crud stepping strategy is explicit in time as the next crud samples are drawn from knowledge of the previous crud state and the current thermal hydraulic conditions. Feedback between the crud state and the underlying density functions are not captured in the hi2lo stepping scheme. The presence of crud influences the rod surface temperature distribution due to two competing effects: Providing favorable conditions for bubble nucleation and an increased thermal resistance. Though these effects are slight they impact not only the mean rod surface temperature but also higher moments about the mean of the probability densities of surface temperature. These higher order feedbacks are not currently considered, however it is possible to incorporate the current crud state into the explanatory variable set used by the machine learning model to resolve this feedback between the crud thickness and the higher order moments of temperature and TKE about the CTF prediction. This remains as future work as the impact of the crud layer on the flow field is hypothesized to only significantly matter if the crud deposits significantly shifts the locations of the onset of nucleate boiling within the core.

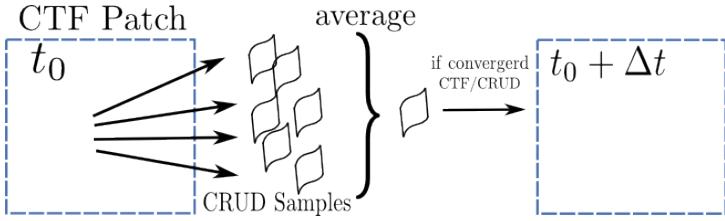


Figure 2.22: Time step procedure depicted for a single CTF face. A single resampling event is shown.

## 2.4 Method Summary

The time stepping, importance sampling and surface sample remapping strategies may be included into algorithm 1 to provide a detailed overview of the hi2lo procedure. The expanded version of the hi2lo model is given in algorithm 3.

Algorithm 3 begins with the same pre-processing and machine learning procedure outlined in the generic hi2lo algorithm 1. *Lines 2-5:* *Line 9:* Reconstructs the temperature and TKE CDFs from quantiles. The details of reconstructing a cumulative density function using a set of predicted quantiles was discussed in section 2.2.4. *Line 10:* The simplified treatment of the boundary heat flux distribution was discussed in section 2.2.1 and the application of this simplification resulted in a compact representation of the joint distribution of temperature, TKE, and BHF in each CTF face given in equation 2.5. *Lines 15-16:* Involves the construction of the importance sampling distributions. *Line 17-18:* Draws samples from the reconstructed joint distribution and follows the standard importance sampling procedure outlined in section 2.2.6. *Line 19:* Performs the spatial remapping procedure required to preserve hot spot stationarity. The details of this procedure may be found in section 2.3.1. *Line 20:* The importance weights are time averaged according to method provided in section 2.3.2 before the crud distribution is integrated in each CTF face at each resampling step.

---

**ALGORITHM 3**

Statistically based hi2lo method for time dependent crud prediction.

---

```

1: Initialization
2: (1) Pre-process training set.
3: (1b) Fit the joint distribution parameters to known CFD data:  $\theta(\mathbf{p}, \mathbf{z})$ .
4: (1c) def:  $\theta \leftarrow \hat{\mathcal{F}}_M(\mathbf{p}, \mathbf{z})$ 
5: (2) Train model:  $\hat{\mathcal{F}}_M = \operatorname{argmin}_{\mathcal{F}} \mathbb{E} [L(\mathcal{F}_M(\mathbf{p}, \mathbf{z}), \theta(\mathbf{p}, \mathbf{z}))]$ 
6: for VERA State,  $v$  do
7:   for CTF face,  $j$  do
8:     Evaluate ML model  $\hat{\theta}_j \leftarrow \hat{\mathcal{F}}_M(\mathbf{p}_j, \mathbf{z}_j)$ 
9:      $\hat{\theta}_j = \{\hat{\theta}_{j,c}, \hat{\theta}_{j,\{T,k\}}\}$ 
10:    Reconstruct margins (CDFs) from quantiles
11:     $\hat{F}_{j,T} = Q^{-1}(T; \{\hat{\theta}_{j,\{T,k\}}\})$ ,  $\hat{F}_{j,k} = Q^{-1}(k; \{\hat{\theta}_{j,\{T,k\}}\})$ 
12:    Def  $q''$  margin  $f_{j,q''} = \delta_{(q''_{j,ctf})}$ 
13:    Reconstruct joint distribution  $\hat{h}_j(\cdot | \hat{\theta}_j) = \hat{f}_{j,T} \hat{f}_{j,k} f_{j,q''} c(\hat{F}_{j,T}, \hat{F}_{j,k}; \hat{\theta}_{j,c})$ 
14:   end for
15:   for Resample time step,  $s$ ,  $\Delta t_s$  do
16:     for CTF face,  $j$  do
17:       Def importance mixture quantile functions by equation  $\tilde{Q}_{j,k} = \lambda_{0,k} \hat{Q}_{j,k} + \lambda_{1,k} Q_{\beta_k}(k; \vartheta_k)$ ,  $\tilde{Q}_{j,T} = \lambda_{0,T} \hat{Q}_{j,T} + \lambda_{1,T} Q_{\beta_T}(T; \vartheta_T)$ ;
18:        $\sum_i \lambda_i = 1$ ,  $\tilde{F}_{j,T} = \tilde{Q}_{j,T}^{-1}$ ,  $\tilde{F}_{j,k} = \tilde{Q}_{j,k}^{-1}$ 
19:       Def importance sampling distribution  $\tilde{h}_j = \tilde{f}_{j,T} \tilde{f}_{j,k} c(\tilde{F}_{j,T}, \tilde{F}_{j,k}; \hat{\theta}_{j,c})$ 
20:       Draw samples  $\mathbf{x} \sim \tilde{h}_j$ 
21:       Compute importance weights  $\omega = \hat{h}_j(\mathbf{x}) / \tilde{h}_j(\mathbf{x})$ 
22:       Re-map samples  $\mathbf{x}', \omega' \xleftarrow{\mathcal{R}} \mathbf{x}, \omega$ 
23:       Update importance weights by equation 2.51
24:       Evaluate equation 2.3 via importance sampling
25:        $\mathbf{C}_s = \mathcal{G}(x'_i; \mathbf{C}_{s-1}, \mathbf{I}, \Delta t_s)$ 
26:       Crud mass at step  $s$  in patch  $j$ :  $C_{s,j,m} = \left( \frac{A_j}{\sum_i^N \bar{\omega}'_i} \right) \sum_i^N C_{s,i,j,m} \bar{\omega}'_i$ 
27:     end for
28:   end for
29: end for

```

---

All model variables which are set at runtime are provided in table 2.1. Recommended settings are also provided by each model parameter.

Table 2.1: Runtime hi2lo model parameters.

Sym	Default Value	Purpose
$\Delta t_s$	25 [days]	Resampling time step size.
$N$	400	Number of samples drawn per CTF face.
$N_{Q_k}$	20	Number of quantiles used in TKE marginal distribution reconstruction.
$N_{Q_T}$	20	Number of quantiles used in temperature marginal distribution reconstruction.
$w_k$	0.6	TKE weighting factor used to remap samples on the rod surface.
$w_T$	0.4	Temperature weighting factor used to remap samples on the rod surface.
$\vartheta_T$	{1.0, 0.9}	Beta distribution parameters for temperature importance distribution.
$\vartheta_k$	{1.1, 1.2}	Beta distribution parameters for TKE importance distribution.
$\lambda_{0,T}, \lambda_{1,T}$	{0.6, 0.4}	Mixture weighting parameters for temperature importance distribution.
$\lambda_{0,k}, \lambda_{1,k}$	{0.7, 0.3}	Mixture weighting parameters for TKE importance distribution.
$b_m$	4	Gradient boosting model parameter: Max CART tree depth. A larger CART tree depth value improves the quality of fit of each weak-learner but increases overfitting.
$\nu$	0.01	Gradient boosting model parameter: Learning rate.
$M$	4000	Gradient boosting model parameter: Number of boosted iterations.

Finally, the simplifications and assumptions made in the construction of the hi2lo model are reviewed. The model requires that within a CTF face no spatial information is retained. The crud simulation package, MAMBA, used in this work is a one-dimensional code and therefore this simplification has no influence on the integrated crud results within each CTF face. The boundary heat flux is treated independently from the rod surface temperature or TKE. This simplifies the copula model used to reconstruct the joint distribution in each face. Hot spots are assumed to remain stationary on the rods'

surface as a function of time. This assumption can be slightly relaxed with tuning of the sample remapping procedure; however, this assumption simplifies the treatment of the time dependent nature of crud build up in the core. The assumption takes into account that as time marches forward, the rate of crud growth depends on the previous crud state. The viability of these assumptions will be investigated in the following sections. The aim of the hi2lo model is to reproduce gold standard crud results provided by high fidelity CFD coupled crud simulations. The application of the simplifications should not impede the ability of the model to reproduce the correct axial and total crud deposition on the rod surface.

## Appendices

# A | Rod Surface Heat Transfer

## A.1 Subcooled Boiling and DNB

The relationship between the surface temperature of an internally heated object and the heat flux from the surface into the surrounding fluid is shown in figure A.1.

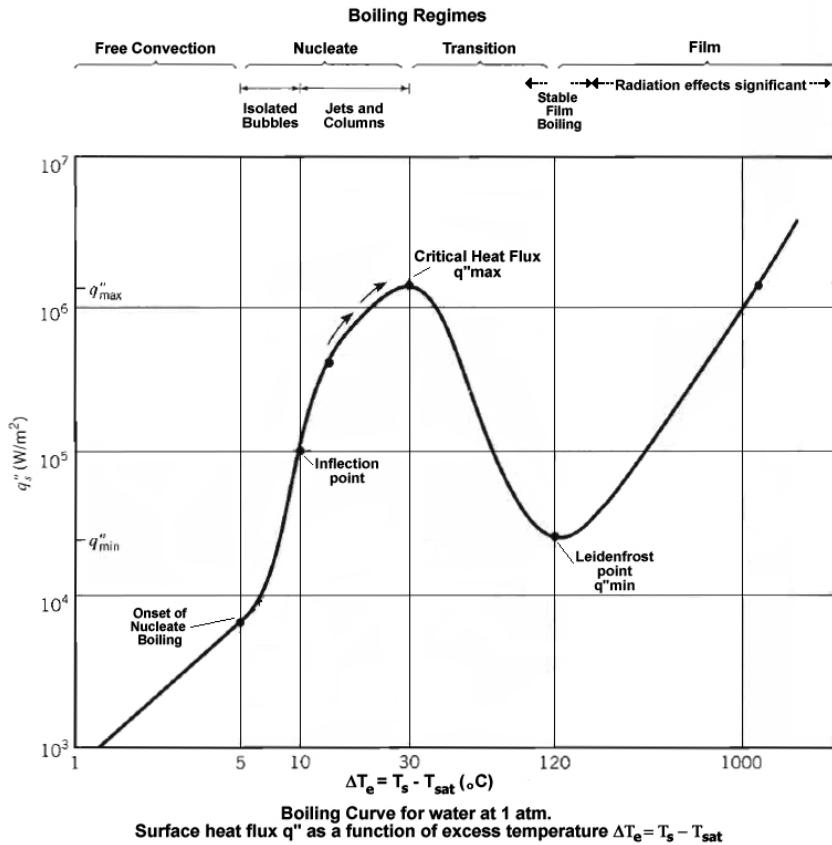


Figure A.1: Boiling curve.

The curve can be approximated by equation A.1. Note that surface temperature  $T_s$  is equivalent to the wall temperature  $T_w$  in the equations which follow. The critical heat

flux (CHF) is the point at which film boiling begins to dominate and is accompanied by a precipitous drop in the heat transfer and a rise in the surface temperature. This condition is known as departure from nucleate boiling (DNB) and must be avoided when operating a PWR.

$$q''(T_w) = \begin{cases} h(T_w - T_\infty), & \text{if } T_w < T_{sat} \\ h(T_w - T_\infty) + q''_{nb}, & \text{if } T_{sat} \leq T_w < T_{CHF} \end{cases} \quad (\text{A.1})$$

Where  $h$  is the single phase convective heat transfer coefficient which is in turn a function of the Nusselt number given in equations A.3 and A.4. The contribution of nucleate boiling to the heat transfer can be approximated by the Rohsenow model given in A.2 [21].

$$q''_{nb} = \mu_L h_{fg} \left[ \frac{g(\rho_L - \rho_v)}{\sigma} \right]^{1/2} \left[ \frac{c_{pL}(T_w - T_{sat})}{C_{sf}h_{fg}Pr_L^n} \right]^3 \quad (\text{A.2})$$

Where  $h_{fg}$  is the latent heat of vaporization,  $\mu_L$  is the liquid viscosity,  $\rho_v$ ,  $\rho_L$  are the vapor and liquid phase densities,  $c_{pL}$  is the specific heat of the liquid phase, and  $C_{sf}$  is a tunable empirical constant.

$$h = \frac{k_l Nu}{L} = \frac{q''}{T_w - T_\infty} \quad (\text{A.3})$$

Where  $k_l$  is the thermal conductivity of the liquid,  $L$  is the characteristic length scale, and  $Nu$  is the Nusselt number. For non-boiling flows over a flat vertical surface, the Nusselt number can be approximated by the Dittus-Boelter equation:

$$Nu = 0.023 Re^{4/5} Pr^n \quad (\text{A.4})$$

Where  $Re$  is the Reynolds number and  $Pr$  is the Prandtl number.  $n$  is an empirically derived constant and is typically 0.4 for a heated flow.

# Bibliography

- [1] P. Turinsky and D. Kothe. Update on capabilities development at CASL. In *International Congress on Advances in Nuclear Power Plants*. American Nuclear Society, May 2015.
- [2] B. Collins, R.K. Salko, S. Stimpson, K. Clarno, and A. Godfrey. Simulation of Crud Induced Power Shift using the VERA Core Simulator and MAMBA. In *PHYSOR 2016*, 2016.
- [3] T. Lange. *Methodology for an Advanced Risk Assessment of Crud Induced Power Shift using Coupled Multi-Physics Simulations and a Monte Carlo Scenario Analysis of the Potential Financial Benefits*. PhD thesis, University of Tennessee, Knoxville, 2017.
- [4] R. Adamson, B. Cox, A. Strasser, and P. Rudling. Corrosion mechanisms in zirconium alloys. Technical report, Advanced Nuclear Technology International, 2007.
- [5] B. Collins and J. Galloway. *Mongoose Methods and Theory*. Consortium for Advance Simulation of Light Water Reactors.
- [6] R.K. Salko and M.N. Avramova. *CTF Theory Manual*. The Pennsylvania State University.
- [7] H. Joe. *Dependence Modeling with Copulas*. CRC Press, 2015.

- [8] T. Bedford and R.M. Cooke. Probability density decomposition for conditionally dependent random variables modeled by vines. *Annals of Mathematics and Artificial Intelligence*, 32, 2001.
- [9] A. Sklar. Fonctions de répartition à n dimensions et leurs marges. *Publ. Inst. Statist. Univ. Paris*, 8, 1959.
- [10] R. Nelsen. *An Introduction to Copulas*. Springer, 2006.
- [11] P. Barbe, C. Genest, K. Ghoudi, and B. Remillard. On Kendall's process. *Journal of Multivariate Analysis*, 58, 1996.
- [12] F. Fritsch and R. Carlson. Monotone piecewise cubic interpolation. *Numerical Analysis*, 1980.
- [13] P. Hall. *The Bootstrap and Edgeworth Expansion*. Springer, 1997.
- [14] F. Mosteller. On some useful inefficient statistics. *Ann. Math. Statist.*, 17(4):377–408, 12 1946.
- [15] J.H. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: A statistical view of boosting. *Annals of Statistics*, 28, August 2000.
- [16] J.H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29:1189, October 2001.
- [17] J.H. Friedman. Stochastic gradient boosting. *Computational Statistics and Data Analysis*, 38:367, Febuary 2002.
- [18] O. Chapelle and Y. Chang. Learning to rank challenge overview. *Machine Learning Research, Workshop and Conference Proceedings*, 14, 2011.

- [19] S. Tyree, K. Weinberger, and K. Agrawal. Parallel boosted regression trees for web search ranking. In *Proceedings of the 20th International conference on World Wide Web*, January 2011.
- [20] R.K. Salko, W. Gurecky, S. Slattery, K. Clarno, D. Pointer, D. Walker, and V. Petrov. Implementation of a Grid Heat Transfer and Turbulent Kinetic Energy Hi2Lo Remapping Capability into CTF in support of the CIPS Challenge Problem. Technical Report CASL-U-2017-1322-000, Consortium for Advanced Simulation of Light Water Reactors, 2017.
- [21] W.M. Rohsenow. A method of correlating heat transfer data for surface boiling of liquids. Technical report, Office of Naval Research, 1951.