ABSTRACT

BAO, HAN. Development of a Data-driven Framework for Mesh-Model Optimization in System-level Thermal-Hydraulic Simulation. (Under the direction of Dr. Nam T. Dinh).

Over the past decades, several computer codes were developed for simulation and analysis of thermal-hydraulics of reactor cores, reactor coolant systems, and containment behaviors in nuclear reactors under operating, abnormal transient and accident conditions. However, the simulation errors and uncertainties still inevitably exist even while these codes have been extensively assessed and used to support design, licensing, and safety analysis of the plants. Main difficulty comes from the complexity of these multi-phase physical phenomena in the transient scenarios, the inevitable simulation error sources and user effects.

In this work, a data-driven framework (Optimal Mesh/Model Information System, OMIS) for the optimization of mesh and model in system-level thermal-hydraulic simulation is formulated and demonstrated. OMIS framework is developed to estimate simulation error and suggest optimal selection of coarse mesh size and models for low-fidelity simulation, such as coarse-mesh Computational Fluid Dynamics-like (CFD-like) codes, to achieve computationally-effective accuracy comparable to that of high-fidelity simulation, such as high-resolution CFD. It takes advantages of computational efficiency of coarse-mesh simulation codes and regression capability of machine learning algorithms. Instead of expensive computation using fine-mesh as in CFD methods, a cluster of case running with different coarse mesh sizes are performed to obtain the error database between low-fidelity and high-fidelity data. The error database is used to train a machine learning model and find the essential relationship between local simulation error and local physical features, then generate insight and help correct low-fidelity simulations for similar physical conditions. Based on the idea of TDMI (Total Data-Model Integration), the specific closure models, local mesh sizes and numerical solvers are treated as an integrated model. Data obtained from this integrated model is used to construct a library that identifies and stores the local similarities in different physical conditions. This library is self-improvable and automatically updated as new qualified data is available. OMIS framework is completed as a six-step procedure; each step is independent and accomplished with methods and algorithms in the state of the art. A mixed convection case study was designed and performed to illustrate the entire framework.

This work also provides an insight on the development of a data-driven scale-invariant approach to deal with scaling issues. According to the identification of global physics and local

physics, four different Physics Coverage Conditions (PCCs) are classified as Global Interpolation through Local Interpolation (GILI), Global Interpolation through Local Extrapolation (GILE), Global Extrapolation through Local Interpolation (GELI) and Global Extrapolation through Local Extrapolation (GELE). The underlying local physics is assumed to be represented by a set of physical features. GELI condition indicates the situation that the global physical condition of new case is identified as an extrapolation of existing cases, but the local physics are similar. Exploring the local physics with the usage of advanced machine learning techniques makes it possible to bridge the global scale gap. Targeting on "GELI" condition, OMIS framework treats multi-scale data and machine learning techniques in a formulized manner. Different GELI conditions, such as the extrapolation of global parameters, geometry, boundary condition and dimension have been discussed based on the mixed convection case study. The similarity between the training data and testing data is quantified by the defined extrapolation distance. It shows that the prediction by well-trained data-driven model has higher accuracy as the similarity of training data and testing data increases.

Development of a Data-driven Framework for Mesh-Model Optimization in System-level Thermal-Hydraulic Simulation

by
Han Bao

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Nuclear Engineering

Raleigh, North Carolina
2018

APPROVED BY:

_____	_____
Dr. Nam T. Dinh Committee Chair	Dr. Igor A. Bolotnov
_____	_____
Dr. Maria Avramova	Dr. Hong Luo
_____	
Dr. Jeffrey W. Lane	

DEDICATION

To my parents, my wife and my son

BIOGRAPHY

The author, Han Bao, was born in February 1988 in China, Hubei province. His first name "晗 Han" means the dawn of the day in Chinese, which implies a bright morning is coming, followed by good wishes.

Han was brought up in his hometown, an ancient city, Xiangyang. Then he began his college study at Xi'an Jiaotong University, where he obtained his bachelor's degree in Nuclear Engineering in 2010. While he won the scholarship every college year, he was also elected as the vice-president of Student Union of Xi'an Jiaotong University in 2008. In 2013, Han got his Master's degree in Shanghai Jiao Tong University, under the supervision of Prof. Yanhua Yang. Recommended by Prof. Yanhua Yang and Prof. Xu Cheng, Han joined Dr. Nam T. Dinh's group in 2014 and started his Ph.D. study at North Carolina State University. His research focuses on developing and demonstrating a data-driven optimization framework for system-level thermal-hydraulic modeling and simulation using machine learning algorithms.

ACKNOWLEDGMENTS

Firstly, I would like to express my most sincere thanks to Dr. Nam T. Dinh for his enlightenment, guidance and support all through my Ph.D. study in the past five years. His insight on the challenges and opportunities of nuclear energy, his vision on the future development and application of data-driven approaches in thermal-hydraulic analysis, his exploration of applying the state of the art techniques on the management of nuclear power plants are the lights that guide me in my Ph.D. research and the future career. His aspiring character and responsible work attitude worth my life to learn.

I would like to thank Dr. Jeffrey W. Lane and Dr. Robert W. Youngblood for their instructive advice on the development of the proposed framework in this dissertation. I would like to thank Prof. Igor A. Bolotnov, Prof. Maria Avramova and Prof. Hong Luo for providing insightful discussions and suggestions to improve my research. Thank you all for the support and service in my dissertation committee.

I also would like to give my gratitude to Dr. Hongbin Zhang, Dr. Haihua Zhao and Dr. Ling Zou for their expert advice during my internships at Idaho National Laboratory. I really appreciate them sharing valuable experience on research and career. I would like to thank Dr. Olumuyiwa Omotowa for his helpful suggestions on how to initiate and carry on a Ph.D. work. I would like to thank my friends and colleagues at NC State, Guojing Hou, Jinyong Feng, Linyu Lin, Yang Liu, Chih-wei Chang, Yangmo Zhu, Jun Fang, Yuwei Zhu, Hao-Ping Chang, Mengnan Li, Yuqiao Fan, Botros Hanna, Paridhi Athe, Joomyung Lee, Xu Han and many others. There would be no such wonderful work and fun without you in the past five years.

Finally, I would like to give my ultimate thanks and love to my wife, my parents and my family. It is them who give me the power and courage to accomplish this long march. Thank you all for accompanying me.

I gratefully acknowledges the support for my research by the Idaho National Laboratory's National University Consortium (NUC) program and the INL Laboratory Directed Research & Development (LDRD) program under DOE Idaho Operations Office Contract DE-AC07-05ID14517. GOTHIC incorporates technology developed for the electric power industry under the sponsorship of EPRI, the Electric Power Research Institute. This work was completed using a GOTHIC license for educational purposes provided by Zachry Nuclear Engineering, Inc.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACRONYMS

BWR	Boiling Water Reactor	NN	Neural Network
CFD	Computational Fluid Dynamics	NPP	Nuclear Power Plant
CHF	Critical Heat Flux	NRMSE	Normalized Root Mean Squared Error
CSAU	Code Scaling and Applicability Uncertainty	NRS	Nuclear Reactor Safety
DDM	Data-Driven Modeling	OMIS	Optimal Mesh/Model Information System
DNN	Deep Neural Network	OOB	Out-Of-Bag
DNS	Direct Numerical Simulation	PCC	Physics Coverage Condition
EMDAP	Evaluation Model Development and Assessment Process	PCMM	Predictive Capability Maturity Model
EOP	Emergency Operating Procedure	PCMQ	Predictive Capability Maturity Quantification
FNN	Forward Neural Network	PDE	Partial Differential Equation
FOM	Figure of Merit	PDF	Probability Density Function
GELE	Global Extrapolation through Local Extrapolation	PF	Physical Feature
GELI	Global Extrapolation through Local Interpolation	PFC	Physical Feature Coverage
GEP	Gene Expression Programming	PIRT	Phenomena Identification and Ranking Table
GILE	Global Interpolation through Local Extrapolation	PVIM	Permutation Variable Importance Measure
GILI	Global Interpolation through Local Interpolation	PWR	Pressurized Water Reactor
GPR	Gaussian Process Regression	QoI	Quantity of Interest
HF	High-Fidelity	RANS	Reynolds-averaged Navier–Stokes
HL	Hidden Layer	RFR	Random Forest Regression
IC/BC	Initial Condition/Boundary Condition	RISMC	Risk-Informed Safety Margin Characterization
IET	Integral Effect Test	SET	Separate Effect Test
KDE	Kernel Density Estimation	TDMI	Total Data-Model Integration
LF	Low-Fidelity	t-SNE	t-Distributed Stochastic Neighbor Embedding
LOCA	Loss-of Coolant Accident	USNRC	United States Nuclear Regulatory Commission
ML	Machine Learning	VUQ	Validation and Uncertainty Quantification
MSE	Mean Squared Error	V&V	Verification and Validation

NOMENCLATURE

Arabic

d_{Eu}	Euclidean distance
d_{KDE}	KDE distance
d_{Ma}	Mahalanobis distance
F_{LF}	a set of governing equations and constitutive equations in LF simulation
p_{KDE}	KDE probability
$\overrightarrow{R_{LF}}$	solution of the LF simulation
$\overrightarrow{R_T}$	true solution
Re_D	Reynolds number in a pipe with diameter D
t	time
V	simulation variable
$\vec{x}$	space coordinate

Greek Symbols

λ_{LF}	model information (model forms and parameters) used in LF simulation
δ_{LF}	coarse mesh size used in LF simulation
ε	simulation error
ϵ	measurement error
ε_{model}	model error
ε_{mesh}	mesh error
σ	standard deviation

Subscripts

HF	High-Fidelity
LF	Low-Fidelity

CHAPTER 1. INTRODUCTION

1.1. Motivation

Quantification of Nuclear Power Plant (NPP) safety risk requires a systematic and yet practical approach to identification of accident scenarios, assessment of their likelihood and consequences. Instrumental to this goal is Risk-Informed Safety Margin Characterization (RISMC) framework [1], whose realization requires computationally robust and affordable methods for sufficiently accurate simulation of complex multi-dimensional physical phenomena, such as turbulence, heat transfer and multi-phase flow. Thermal-hydraulics is considered as one of the key disciplines essential for progress in nuclear science, making reference to NPP design and Nuclear Reactor Safety (NRS). Over the past decades, a number of computer codes were developed for simulation and analysis of thermal-hydraulics of reactor cores, reactor coolant systems, and containment behaviors in NPP under operating, abnormal transient and accident conditions. However, the simulation errors and uncertainties still inevitably exist even while these codes have been extensively assessed and used to support design, licensing, and safety analysis of the plants.

Main difficulty comes from the complexity of these multi-dimensional multi-phase physical phenomena in the transient scenarios. These phenomena locate in the different NPP components with complex geometries and structures, which makes it impossible to perfectly model and simulate the entire NPP thermal-hydraulic systems in all time and length scales. The nuclear industry and research communities reacted to this challenge in two ways: performing comprehensive experiments to reproduce expectable reactor accident conditions and developing numerical codes to analyze the transient thermal-hydraulic performance by the massive use of computers. Therefore, the development of thermal-hydraulic codes greatly benefits from the massive use of computational capability and fundamental studies based on experimental programs. Normally, there are three types of computational codes used for thermal-hydraulic analysis. The first type is called lumped-parameter code or system code, such as REALP 5, TRAC, ATHLET, MELCOR and MAAP, etc. These codes describe an NPP thermal-hydraulic system as a network of simple control volumes connected with junctions. The control volumes are modeled as homogeneous and described by single values of temperature, pressure and other variables. Turbulence effects are not directly modeled but could be considered using assumed flow-loss coefficients in the momentum equation. [2] Aiming at fast obtaining the overall system response,

much local information is lost when the time and geometry (including volume and area) averaging approaches are applied on the local instantaneous two-fluid models. The second type is Computational Fluid Dynamics (CFD) code that has become commonly used for computer codes that numerically solve the transport equations of fluid mechanics (continuity, momentum and energy) using a local instantaneous formulation. These CFD codes (e.g., STAR-CCM+, OpenFOAM) consider the influence of turbulence using different turbulent models. Thermal-hydraulic analysis using CFD codes is computationally expensive since million cells might be needed even for modeling of a single NPP component.

Different from standard system codes (with much loss of local information) and standard CFD codes (with huge computational cost) for system-level thermal-hydraulic modeling and simulation of NPPs, this dissertation defines the third type as coarse-mesh CFD-like code. These codes with 3D simulation capability and full treatment of momentum transport terms ensure computational efficiency using coarse mesh size and the sub-grid phenomena in the boundary layer that can be captured by adequate constitutive correlations (e.g., wall functions and turbulence models). In contrast to standard CFD codes, these codes, such as GOTHIC [3], do not have a body-fitted coordinate capability: the subdivision of a volume into a multi-dimensional grid is based on orthogonal co-ordinates, and the code uses boundary-layer correlations for heat, mass and momentum exchanges between the fluid and the structures, rather than attempting to model the boundary layers specifically. Therefore, these codes have natural advantages compared with standard system codes and CFD codes to achieve sufficient accuracy for long-term multiple-component system simulation. These CFD-like codes have been extensively used for containment thermal-hydraulic analysis. [4-6]

However, two main error sources exist in the modeling and application of these coarse-mesh CFD-like codes. They solve the integral form of conservation equations for mass, momentum and energy for multi-component, multi-phase flow. Boundary-layer correlations are applied for heat, mass and momentum exchanges between the fluid and the structures, rather than attempting to model the boundary layers specifically. Respective characteristic lengths of these empirical correlations are default calculated using the local mesh size. Therefore, the mesh size greatly affects the performance of the empirical correlations in the local near-wall cells and becomes one key model parameter that determines whether the correlations are applied in their applicable ranges or not. Model error due to physical simplification and mathematical approximation on these

applied models, correlations and assumptions is one of the main error sources. Another one is called “mesh error” that indicates the information loss of conservative and constitutive equations during the application of time and space averaging approaches. The local instantaneous Partial Differential Equations (PDEs) for mass, momentum and energy balance are space averaged to obtain the finite volume equations. Simulation results represent the averaged values of parameters over specified regions, which ignores the local gradient information. Other numerical errors have less influence on the modeling and simulation compared to model error and mesh error.

Since both main error sources are tightly connected with local mesh size, the nodalization of control volumes determines whether the user can get a relatively good simulation result or not. The finite mesh/volume approach, particularly in the coarse scheme of NPP simulations, could fail in not capturing the expected local behaviors of the fluids (sharp gradients of variables) due to limited resolution. On the other hand, a finer nodalization could introduce an improper extending of boundary-layer empirical correlation. All these factors make the selection of mesh size and model information (model parameter and model form) be an important but tricky task in the system-level thermal-hydraulic modeling and simulation using these CFD-like codes. In the current applications, the mesh size and models are selected based on previous modeling experience, this kind of “educated guess” may lead to an error for the new physical conditions. Therefore, a smart guide is urgently needed to provide advice on the optimal selections of coarse mesh size and models.

1.2. Overview of Traditional Model Validation Frameworks

The tight connection between model error and mesh error makes it difficult to perform traditional Verification and Validation (V&V) on these coarse-mesh CFD-like codes to analyze these two errors separately, although the development of system thermal-hydraulic codes greatly benefits from the massive use of computers and fundamental studies based on experimental programs. The evaluation of safety margins, the operator training, the optimization of the plant design and related emergency operating procedures, are some of the applications of these codes, which essentially deal with the solution of the balance equations for steam-liquid two-phase mixtures supplemented by the constitutive equations. [7] The integrated modeling and simulation of primary system and containment was considered necessary for the prediction of the overall system performance. As the rapid and wide use of system-level thermal hydraulic codes, V&V of

these codes have been recognized as a mandatory part before their application on safety analysis and licensing due to the lack of physical knowledge and the lack of confidence on the code capability. Historically, United States Nuclear Regulatory Commission (USNRC) published rules for Loss-of Coolant Accident (LOCA) analysis in 10CFR 50.46 and Appendix K in 1974, which established the initial licensing procedures with a conservative approach. However, the conservative approach brought in some problems: (1) the conservatism proved in scaled-down tests may be not still valid in the full-scale plants; (2) the conservative approach is not suitable for Emergency Operating Procedure (EOP) studies. [8]

- **Code Scaling and Applicability Uncertainty (CSAU)**

Therefore, USNRC initiated an effort to develop and demonstrate a licensing acceptable Best Estimate plus Uncertainty (BEPU) method that provided nuclear plant operators with more economic gains and less conservation. The Code Scaling and Applicability Uncertainty (CSAU) method was formulated to provide more realistic estimates of plant safety margins for large break LOCA in a Pressurized Water Reactor (PWR) in 1990. [9] The first element of CSAU is to specify the requirement and code capability to the transient scenarios. The PIRT (Phenomena Identification and Ranking Table) process was proposed to reduce the complexity of the transient scenarios. Expert judgment was needed to identify and rank different phenomena relevant to the Figures of Merit (FOMs). The phenomena were arranged hierarchically based on transient phase, system components, and underlying phenomena. The second element of CSAU is assessment and ranging of parameters. Identification of relevant Separate Effect Tests (SETs) and Integral Effect Tests (IETs) are involved for code validation. Scaling analysis was introduced here to determine the code scalability. However, there was no a pellucid explanation on how to evaluate the effects of scale distortion on important processes. Besides, as CSAU targets on system codes, the nodalization for NPP calculations was the main uncertainty source and thus fixed before the determination of scale effect. The mesh effect on the code/model accuracy was not fully considered, little attention was put on mesh sensitivity study. This makes CSAU not suitable for the validation of coarse-mesh CFD-like codes since mesh is also one of key model parameters. In the third element, response surface method was used to estimate the overall uncertainties in the prediction of the FOMs.

- **Evaluation Model Development and Assessment Process (EMDAP)**

In 2005, USNRC issued another important document Regulatory Guide (RG) 1.203 to provide an acceptable Evaluation Model Development and Assessment Process (EMDAP) for the best estimate calculations of NPP transient and accident analysis. [10] EMDAP aimed to evaluate the adequacy of the applied codes and provide guidance for the following experiment and analytical tool development. The basic principles of EMDAP were developed based on the CSAU methodology, while EMDAP has formal and explicit descriptions for the concepts, definitions and processes, including the PIRT, assessment base, evaluation model, scaling analysis. After the logical and comprehensive validation, the decision process was executed to evaluate whether the code meets the adequacy standard and can be used for plant scenario analysis. However, the acceptance criteria were not clearly defined. One significant difference between EMDAP and CSAU is that the scalability of SET/IET facilities and code/model scalability were evaluated separately in different elements of EMDAP. Scalability analysis was executed for both of data and code. Same as CSAU, the system analysis and scaling analysis in EMDAP are highly heuristic and difficult to implement.

Mesh effect on code/model scalability was not fully considered in CSAU and EMDAP. They both recognized the existence of mesh sensitivity and required a mesh sensitivity study to make sure important FOM (e.g., PCT) are not significantly impacted. They assumed that a "constant" uncertainty was introduced between the scaled tests and plant application by applying the same modeling guidelines and consistent nodalization for both. However, the fact that mesh size could be one of the key model parameters and have an effect on code/model applicability was not fully considered since the mesh sensitivity was performed before the code/model scalability analysis. More discussions on EMDAP is in Chapter 7.

- **Predictive Capability Maturity Model (PCMM)**

In 2007, Predictive Capability Maturity Model (PCMM) was developed by Sandia National Laboratories as a decision model for maturity assessment of modeling and simulation tools. [11] Comparing to CSAU and EMDAP, PCMM explicitly treats the model credibility and uncertainty analysis as a decision-making process with explicit structures. After specifying the target application and corresponding model, eight elements were designed and assessed: (1) Representation and geometric fidelity, (2) Physics and material model fidelity, (3) Software quality

assurance, (4) Code verification, (5) Solution verification, (6) Separate effects model validation, (7) Integral effect model validation, (8) Uncertainty quantification and sensitivity analysis. These eight elements act as decision attributes and form the basis for the decision regarding the maturity of a computer simulation code for the intended use. Then a qualitative assessment for each attribute was performed according to a PCMM matrix and characterized each attribute with different levels of maturity. Lastly, the achieved levels of maturity were compared with the level standards for the final decision making on validation adequacy. One significant benefit of PCMM is that code verification and model validation are explicitly distinguished by a formal definition for credibility assessment. However, the scaling effects were not clearly discussed which limited the PCMM capability in validating complex systems with multiple scales and physics. Recently, a validation framework called Predictive Capability Maturity Quantification (PCMQ) has been developed to support PCMM for maturity estimation of CASL (Consortium for Advanced Simulation of Light Water Reactors) challenging problem and application. [12] It provided a reason-based conceptual approach for structural knowledge representation, evidence incorporation and confidence assessment for implementation of PCMM for the intended applications. Maturity quantification is obtained using different techniques, like Fuzzy logic and Bayesian network that are well known for their ability to integrate subjective information (based on expert knowledge and judgment) with objective data (evidence). [13]

1.3. Overview of Data-driven Applications based on Total Data-Model Integration

In 2013, some new perspectives have been proposed on the nuclear reactor thermal-hydraulics. [14] It was envisioned that "*in the future, the complex and varied issues of nuclear reactor thermal-hydraulic processes could be addressed effectively and efficiently by developing and implementing a data-driven framework for modeling and simulation that brings together and allows for all relevant data and knowledge to be utilized together to enable synergistically predictive tools and processes for nuclear thermal-hydraulics*". The concept of "data-driven modeling and simulation framework" enables the simulation code applying pattern recognition and statistical analysis to obtain required closure information directly from the relevant database generated from huge amounts of experiments and simulations. The core of OMIS framework is the assumptions, methods and tools for Total Data-Model Integration (TDMI) that bring together data, physical models and simulations to effectively support decision-making in engineering applications. This data-driven concept makes great usage of the rich High-Fidelity (HF) data

instead of having to converting the data into separate physical models where a great deal of information has to be abandoned. For conditions where directly applicable data is absent, the information can be predicted based on the near-by conditions included in the database. Uncertainty due to the lack of data can be reduced as new data becomes available. HF data refers to the data that have been adequately validated and has a potential to be used to reduce the simulation uncertainty from Low-Fidelity (LF) modeling and simulation.

Nowadays, the explosive development of Machine Learning (ML) algorithms and massive data available from numerical simulations make the idea of TDMI realistic and feasible. There have already been several efforts to apply ML algorithms on fluid dynamics since the beginning of this century. According to a classification of machine learning frameworks for thermal fluid simulation introduced by Chang et al. [15], current efforts mainly belong to Type I or Type II ML. Type I ML aims at developing new closure models assuming that conservation equations and closure models are scale separable. Type II ML concentrates on reducing the uncertainty of LF simulation by "learning" from HF data. Both of them requires a throughout understanding of the physical system and sufficient prior knowledge on closure models. These limitations make current data-driven approaches for specific local closure laws not applicable to the complex system-level thermal-hydraulic modeling and simulation of plants. The complexity of prior knowledge extremely increases when all the components, processes and involved phenomena in reactor systems should be considered together. A data-driven approach with less knowledge required is urgently needed for complex situations, especially when a great amount of HF data and computation capability are available. Based on the concept of TDMI, a Validation and Uncertainty Quantification (VUQ) framework for Eulerian-Eulerian two-fluid-model based multiphase CFD solver has been formulated. [16] The proposed framework applies Bayesian method to inversely quantify the uncertainty of the solver predictions with the support of multiple experimental data. However, the numerical error introduced in the discretization of the Partial Differential Equations (PDEs) is not considered. Besides, the statistical methods applied in the framework also introduce additional uncertainty which is difficult to be estimated.

In the Integral Research Project of "Development and Application of a Data-Driven Methodology for Validation of Risk-Informed Safety Margin Characterization Models", a validation framework, named Risk-informed Evaluation Model Development and Assessment Process (REMDAP), is proposed for the validation of RISMC models. [17] REMDAP is designed

based on the framework of EMDAP and the methodology of CSAU by combining data-driven and risked-informed concepts. REMDAP aims to shift from current expert-determined validation approach to a data-driven approach including the data-driven closure development, data-driven uncertainty quantification, and PCMQ with Bayesian Network.

Industry also shows interest and provide requirements for the development and demonstration of this data-driven methodology for the validation of Risk-Informed Safety Margin Characterization (RISMC) models for nuclear power plant safety analysis. [18] In order to increase the likelihood of industry acceptance and adoption of a data-driven code framework, following validation requirements that should be satisfied as a foundation have been proposed:

1. Testing/training priorities. The framework should focus first on areas where significant value can be added with a new approach. The model should be considered acceptable for the application if additional information would not affect the performance of this particular computational model.

2. Avoidance of overfitting. Due to the existence of data bias and noise, high-complexity machine learning models may overfit the data into some artificial trends. It should be considered by developers whether the data-driven model fully captures the underlying physics in a sufficient and necessary level of fidelity. The balance between accuracy and computational efficiency should be paid attention.

3. Model evaluation. Evaluations on data-driven models should be cross-benchmarked to traditional methods and clear visual representations should be used to demonstrate compliance.

4. Uncertainty quantification. For the regions where validation data of models is not available, the uncertainty in model/code output should be considered and quantified.

1.4. Dissertation Overview

In order to provide error prediction and advice on the optimal selections of mesh size and models for system-level thermal hydraulic simulation, a data-driven framework (Optimal Mesh/Model Information System, OMIS) is proposed in this work. OMIS framework is developed for the thermal-hydraulic codes that have the following features: using coarse mesh sizes and applying simplified boundary-layer correlations whose applicable ranges depend on respective

characteristic lengths, such as CFD-like codes or coarse-mesh Reynolds-averaged Navier–Stokes (RANS) methods with wall functions.

1.4.1. Objectives of Dissertation

The objective of this work is to develop and demonstrate a data-driven framework to,

1. Estimate the local simulation error using Machine Learning (ML) algorithms;

2. Give advice on the optimal selection of coarse mesh size and models for low-fidelity simulations (e.g., system codes, CFD-like codes or Coarse-Grid CFD codes) to achieve accuracy comparable to that of high-fidelity data (e.g., DNS or high-resolution CFD).

In addition to improve the coarse-mesh simulations, this work also aims to develop a technical basis for the validation and uncertainty quantification of CFD-like codes in system thermal-hydraulic modeling and simulation

1.4.2. Technical Approach

The technical approach to develop and demonstrate the proposed framework includes,

1. Review pros and cons of current traditional and data-driven V&V frameworks if applied to the coarse-mesh CFD-like codes for system thermal-hydraulic modeling and simulation;

2. Considering a multi-disciplinary nature of the proposed development, review the required knowledge and efforts from multidisciplinary fields including system thermal-hydraulic modeling and simulation, statistics, machine learning, and V&V;

3. Investigate necessary technical capabilities and develop a methodology including basic concepts, basic assumptions and hypotheses, and the data-driven optimization framework. Each step of the proposed framework should be sufficiently explicit to be implemented;

4. Construct test matrix and synthetic cases to demonstrate how to apply the framework to a system thermal-hydraulic simulation;

5. Summarize lessons learned from the synthetic cases and plan future work to improve the framework.

1.4.3. Dissertation Structure

The dissertation has following chapters:

Chapter 2 introduces technical background of the proposed OMIS framework including error analysis of current thermal-hydraulic codes, literature review on data-driven applications on fluid dynamics, and the scope of this work.

Chapter 3 reviews the LF simulation tool, GOTHIC, which is a coarse-mesh CFD-like software initially developed for containment thermal-hydraulic analysis. The structure on thermal-hydraulic modeling is described including conservation equations, source terms, and closure models involved. The relationship between mesh and key closure models are discussed. In addition, a qualitative assessment of thermal-hydraulic simulation using GOTHIC is performed including mesh and model sensitivity study.

Chapter 4 reviews the ML algorithms applied in this work: one-layer Forward Neural Network (FNN) and Deep Neural Network (DNN).

Chapter 5 describes methodology of the proposed OMIS framework. Mathematical basis, practical consideration, basic assumptions and hypotheses are introduced. Each step of the framework is explicitly described with the applied methods, algorithms and equations.

Chapter 6 discusses the case study on mixed convection simulation in a cavity. Different global extrapolation conditions are designed to illustrate the proposed data-driven framework and approach. Lessons learned from case study are also recorded for the improvement of the framework in the future.

Chapter 7 represents the effort to integrate OMIS framework and EMDAP. The data-driven OMIS framework has a potential to be a supplement to make the implement of EMDAP feasible and practical.

Chapter 8 summarizes the remarks and the contributions of this work.

1.5. Terminology and Definitions of Related Concepts

- **Data Convergence:**

Normally, convergence is a means of modeling the tendency for the genetic characteristics of populations to stabilize over time. Data convergence in machine learning training and prediction process implies that the prediction for the genetic characteristics of populations tends to stabilize with increasing size of the training database.

- **Data-Driven Modeling (DDM)**

The concept of DDM is based on analyzing the data about a system, in particular finding connections between the system state variables (input, internal and output variables) without explicit knowledge of the physical behavior of the system. DDM focuses on computational intelligence and Machine Learning (ML) algorithms that can be used to build models for complementing or replacing physics-based models. This concept is proposed opposite to the concept of conventional theory-driven modeling.

- **High-Fidelity (HF) Data and Low-Fidelity (LF) Simulation:**

The concept of fidelity represents the degree to which the models or simulations is close to their real world referents or to other simulations in such terms as accuracy, scope, resolution, level of detail, level of abstraction and repeatability. In thermal-hydraulic analysis, experimental data and numerical simulation result using validated code or DNS (Direct Numerical Simulation) can be considered as the High-Fidelity (HF) data for model or code validation.

HF data is regarded as the sufficiently accurate data considering the true physics is unknown. HF data can be the experimental data, DNS simulation result or numerical data from validated thermal-hydraulic simulation codes with fine mesh and HF models. LF simulation implies the un-validated simulations using system-level thermal hydraulic codes with coarse mesh and simplified physical models with low fidelity. HF does not only indicate fine mesh, but also the HF models and algorithms applied in the codes. The application of OMIS framework is trying to minimize the difference between HF data and LF simulation results on the FOMs in system-level thermal hydraulic simulations, so that the LF simulations are computationally cheaper than HF simulations of finer-mesh CFD and meanwhile have higher accuracy than the no-guide one-shot use of system-level thermal hydraulic codes.

- **Coarse-mesh CFD-like Codes:**

Considering the drawbacks of LP codes and CFD codes, in order to obtain computationally efficient, some codes are developed specifically for the analysis of containment thermal hydraulics using coarse-mesh nodalization. Special equipment of a NPP containment like spay, pump, turbine and hydrogen recombiner are modeled in the containment-specific codes. These codes have usually been validated using relevant experiments corresponding to the phenomena occurring in

containment. Most of these containment-specific codes applies porous media approach. The main characteristics of those codes are,

CFD-like: As these codes (e.g., GOTHIC) are developed for the simulation of large containment compartments, coarse numerical grids are applied to divide the large control volumes into several coarse-mesh cells. GOTHIC includes a full treatment of the momentum transport terms in multi-dimensional models, with optional models for turbulent shear, and for turbulent mass and energy diffusion. The hydraulic model of GOTHIC is based on a network of computational volumes (1D, 2D and 3D) connected by flow paths. In contrast to standard CFD codes, those codes do not have a body-fitted coordinates capability: the subdivision of a volume into a multi-dimensional grid is based on orthogonal co-ordinates, and the code uses boundary-layer correlations for heat, mass and momentum exchanges between the fluid and the structures, rather than attempting to model the boundary layers specifically.

- **Mesh Error vs. Discretization Error**

The discretization error is proposed from the classic Verification and Validation (V&V) point of view for the solving of Partial Differential Equations (PDEs), which assumes that when mesh size goes to zero the solution of PDEs converges. However, due to the correlation-based design in the simplified boundary-layer treatment, GOTHIC is not expected to converge when mesh size goes to zero due to very fine mesh may not satisfy the applicability of these empirical correlations. GOTHIC applies finite volume technique with cell volume and surface porosities for complex geometries. The local instantaneous PDEs for mass, momentum and energy are time and space averaged to obtain the finite volume equations. Results from GOTHIC represent the averaged values of parameters over specified regions, not the exact value at the central points of the regions. Therefore, mesh error indicates the information loss of conservative and constitutive equations during the application of time and space averaging approaches. Other numerical errors in the system code due to iterative convergence, algorithm selection, coding error and finite arithmetic also exist in GOTHIC, but have less influence on the modeling and simulation compared to model error and mesh error.

- **Scale Invariant**

Scale invariant represents the entities that are independent of scale, such as physics, DNS models. Scale invariant approaches are the ideal approach to explore and predict behaviors in real

full-scale applications. There are two kinds of scale-invariant approaches: (1) Full-scale (or physics-conserved) experiment, which is (presumably) independent on the facility scale, (2) DNS modeling where the local information is solved accurately with very fine mesh. Reduced-order models e.g., LES (Large Eddy Simulation), RANS models and system codes are not scale-invariant approaches.

- **Scaling Uncertainty**

The empirical models used in system codes are generally developed from the experiments where the IC/BC (Initial Condition/Boundary Condition) and geometry were not the typical NPP operating conditions, so many of them do not have the scalability to the nuclear reactor applications. Besides, some tuning constants, such as flow resistance coefficients, heat transfer fouling factors were used in the validation process to satisfy better agreement between the test data and the simulation. However, in fact, these tuning constants are not scalable for the extrapolation conditions although they could cover up the distortions in some specific conditions. The extending use of developed models from a scaled test to a NPP application requires a great deal of evaluation and calibration. The effect of scaling on the model error/uncertainty calibrated from the data from scaled experiments greatly influences the accuracy of simulation and leads to an unknown error/uncertainty. The uncertainty due to scaling effect is called scaling uncertainty.

- **Physics Coverage Condition**

According to the identification of global physics and local physics, four different physics coverage conditions are classified as Global Interpolation through Local Interpolation (GILI), Global Interpolation through Local Extrapolation (GILE), Global Extrapolation through Local Interpolation (GELI) and Global Extrapolation through Local Extrapolation (GELE). Global physics indicates the global or macroscopic state, observation and deduction of the simulation target condition, such as the dimension, geometry, structure, boundary condition and non-dimensional parameters that represent the underlying physics; while local physics refers to the microscopic state, observation and deduction of the simulation target condition. For example, the global physics of turbulent flow can be characterized using different values of Re number and geometries. No matter how Re number or geometry changes, the local physics is always turbulence if the Re number is big enough. From the perspective of data characteristics, the underlying local physics is assumed to be represented by a set of Physical Features (PFs).

- **Physical Feature Coverage (PFC)**

As mentioned before, the underlying local physics is assumed to be represented by a set of Physical Features (PFs). The similarity of PFs in training data and testing data can be visualized and measured using dimensionality reduction techniques. Physical Feature Coverage (PFC) implies the similarity or difference between the training data and testing data. The similarity is represented by the coverage (or covered portion) of physical feature between training data and testing data. It is expected that the training data and testing data tends to represent the same local physics as the covered portion approaches to 1. The similarity is depending on the identification of PFs, data quality and quantity. The prediction by well-trained data-driven model is assumed to have higher accuracy as the similarity of training data and testing data increases.

CHAPTER 2. TECHNICAL BACKGROUND

2.1. Introduction

This chapter describes the technical background of the proposed data-driven methodology including error analysis of current thermal-hydraulic codes in Section 2.2, literature review on data-driven applications on fluid dynamics in Section 2.3, and the scope of this work in Section 2.4.

2.2. Error Analysis of Thermal-Hydraulic Codes

These coarse-mesh correlation-base CFD-like codes include a full treatment of the momentum transport terms in multi-dimensional models, with optional models for turbulent shear, and for turbulent mass and energy diffusion. The hydraulic model is based on a network of computational volumes (1D, 2D and 3D) connected by flow paths. This makes these codes become the first choice to achieve sufficient accuracy for long-term multiple-component system simulation. However, there are many challenges in their VUQ process. Those codes are based on equations for two-phase flow which are typically resolved in Eulerian coordinates. The two-phase flow field is described by mass, momentum, and energy conservation equations for the liquid and vapor phases separately and mass conservation equations for some noncondensable gases present in the mixture. Depending on the number of balance equations, different sets of constitutive equations are required to close the equation system. These constitutive equations need to describe the physical phenomena in a wide span of scale, ranging from down-scaled integral system experiments up to full size reactor geometry. Before applying these CFD-like codes into OMIS framework, it is necessary to understand: (I) error sources in traditional V&V perspective, (II) scaling issues in thermal-hydraulic applications and (III) error sources in the applications of these CFD-like codes.

2.2.1. Uncertainty and Error Sources from Verification and Validation Perspective

Due to the numerical approximations and the empirical nature of the included models in the thermal-hydraulic system codes, extensive activities related to Verification and Validation (V&V) of those codes have been pursued during the years. Verification is defined as "*the process of determining that a model implementation accurately represents the developer's conceptual description of the model and the solution to the model*". [19] The fundamental strategy of

verification is the identification and quantification of error in the computational solution through the assessment of coding reliability (as code verification) and the numerical accuracy of computational model (as solution verification). Validation is defined as "*the process of determining the degree to which a model is an accurate representation of the real physics*". [19] The fundamental strategy of validation is the identification and quantification of error and uncertainty in the simulation results by comparison with experiments. Although the huge amounts of financial and human resources spent on the process of V&V, the results predicted by the code are still affected by errors and uncertainties whose sources can be attributed to several reasons as model deficiencies due to physics simplification, approximations in the numerical solution, system nodalization, and the imperfect knowledge of Initial Condition/Boundary Condition (IC/BC). The interchangeable use of the terms "uncertainty" and "error" in many texts and articles leads to a great deal of misinterpretation and confusion of fundamental concepts and simulation analysis.

The concept of uncertainty fundamentally represents whether the source is stochastic in nature or lack of knowledge in nature. During the past decades, the risk assessment community, primarily the Nuclear Reactor Safety (NRS) community, has developed the most workable and effective categorization of uncertainties: aleatory and epistemic uncertainties. Aleatory uncertainty is defined as "*the inherent variation associated with the physical system or the environment such as the variation in thermodynamic properties due to manufacture*". [19] Common sources of aleatory uncertainty include system parameters, IC/BCs that may vary randomly from component to component and/or system to system. Epistemic uncertainty represents the potential deficiency in the modeling due to lack of knowledge. In the risk assessment community, it is common to refer to epistemic uncertainty simply as uncertainty and aleatory uncertainty as variability. *"Epistemic uncertainty is a property of the modeler or observer, whereas aleatory uncertainty is a property of the system being modeled or observed*". [19] One of the main epistemic uncertainties in the thermal-hydraulic modeling and simulation is IC/BC uncertainty, which stems from the imperfect knowledge of the initial status of system component and boundary conditions imposed on the system.

The concept of error commonly means "*a recognizable inaccuracy from the true value in any phase or activity of modeling and simulation that is not due to the lack of knowledge*". [19] It does not address the random nature of the source, but focuses on the identification of the true value. However, in most simulations, the true value is unknown or not representable with finite precision;

and in experimental measurements of engineering and scientific quantities, the true value is never known. Therefore, the definition and accuracy of the true value determines whether the usefulness of the concept of error in practical applications. Sometimes experimental data and numerical simulation result using validated code or DNS can be considered as the HF data for model or code validation. [20] According to the different phases for a computational simulation [21], the sources of errors can be separated into three parts: measurement error, model error and numerical error as shown in Figure 1. If we consider experimental data as the true value of physics of interest, the model error and numerical error can be regarded as the main error sources during the modeling and simulation.

Conceptual modeling phase concentrates on developing a specification of the physics of interest. The specification process includes the determination of physical events, the sequence of events and the way to couple different physical processes. No mathematical equations are written in this phase but the fundamental assumptions and simplifications of the complicated physical system and possible processes should be made. All the key and sensitive system and environment characteristics needs to be considered as detailed as possible. Mathematical modeling phase focuses on developing detailed and precise mathematical models for those characteristics. The mathematical models formulated in this phase include the complete specification of all PDEs, IC/BCs for the system. Meanwhile the mathematical issues such as non-linear problems lead to an acceptable approximation of conceptual models. Mathematical modeling also results in the epistemic uncertainties during the selection of appropriate physical models to represent the physical system and processes of interest. Presumably, only one model is more accurate for the simulation but it is also unknown in the prediction. Discretization phase accomplishes converting the mathematical models into discrete models. This phase deals with questions from the PDEs, stability of the numerical method, approximation of mathematical singularities and differences in zones of influence between the continuum and discrete systems. For the system-level simulation for two-phase flow using porous media approaches, the local instantaneous formulation of the differential balance equations is averaged in time and space with specific averaging methods to obtain time and volume averaged two-fluid model with structural materials in a control volume. Spatial distribution of variables and their effects on the balance and constitutive equations should be considered for volume-averaged model to avoid inaccurate modeling and numerical instabilities. The information is lost during the averaging of source terms and constitutive

equations. Algorithm selection and coding phase focuses on selecting appropriate numerical algorithm and convert the modeling and simulation procedure into a computer code. In this phase, error can be defined as the difference between the exact solution of discrete equation and the exact solution of mathematical model. Numerical solution phase focuses on computing the individual numerical solutions, only discrete values and discrete solutions exist with finite precision. The finite arithmetic on digital computers may lead to round-off errors. Solution representation phase concerns the representation and interpretation of both the individual and collective computational solutions. The collective results are ultimately used by decision makers while the individual results are typically used by engineers, physicists and numerical analysts.

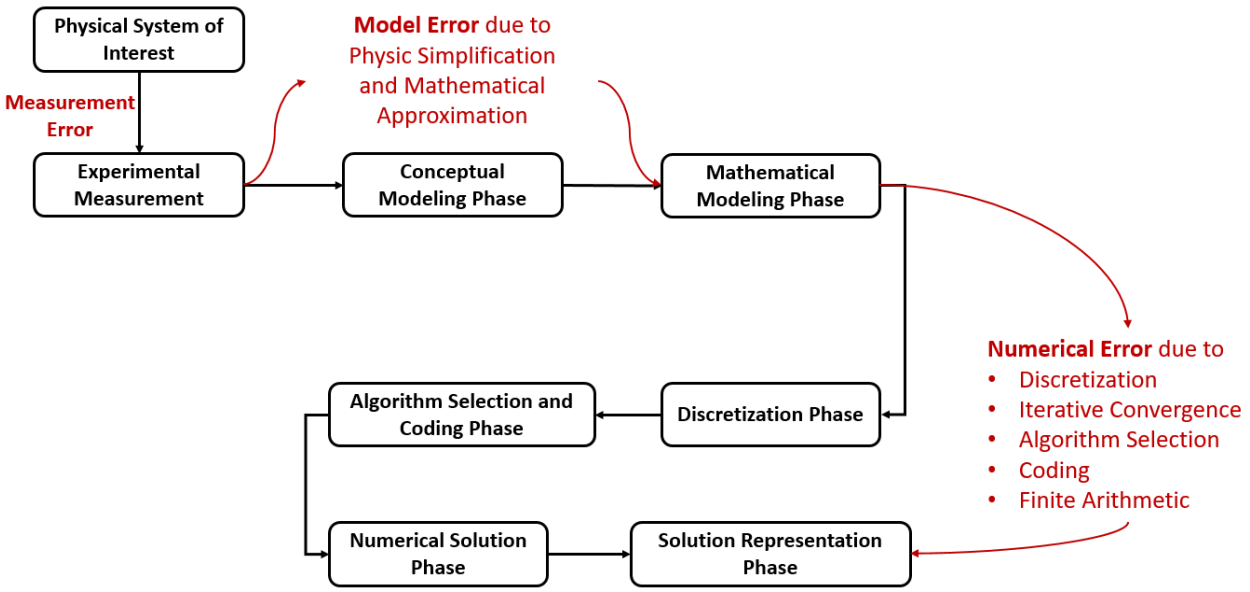


Figure 1. Main Error Sources during Phases for a Computational Simulation from Traditional V&V Perspective of Simulation Codes

As was discussed above, the main uncertainty and error sources in traditional system-level thermal-hydraulic modeling and simulation are (1) IC/BC uncertainty; (2) Model error due to physical simplification and mathematical approximation; (3) Those numerical errors due to discretization, iterative convergence, algorithm selection, coding error and finite arithmetic.

Notably, those uncertainties and errors can be propagated and difficult for the V&V if considering the “user effect”. The user effects can be defined as: Using the same code and same specifications (e.g. initial and boundary condition) the results of the calculations should be similar; otherwise the differences are coming from “user effects". [22] According to this definition, user

effect greatly influences on the selection of model form/parameter and system nodalization (mesh size), which are the main sources of uncertainties and errors in thermal-hydraulic modeling and simulation. Here error is defined as a recognizable inaccuracy in any phase or activity of modeling and simulation that is not due to lack of knowledge. Technically, there are some ways to obtain simulations that are more accurate by using smaller mesh sizes when solving PDEs or using more completed but accurate closure models for the systems. However, the accuracy can be considered acceptable or allowed to remain due to practical constraints such as computation cost or convenient modeling, especially in the analysis for the entire NPP systems.

2.2.2. Scaling Issues in Thermal-hydraulic Application

Scaling is the process of assessing the similarity of phenomena occurred/observed in reduced-scale test facility and the full-scale nuclear reactor application since it is impractical to perform experiments with the same size, pressure and power of the full-scale plants. Normally, the prediction of prototype-scale processes is performed using models developed based on scaled experiments. These scaled experiments are designed by decomposition and down scaling the full scale applications into easy-handling scaled tests with respect to the understanding of one or more of the phenomena involved in the real applications. Scaling issues indicates the difficulties and complexities stemming from the applicability of the data measured in the scaled experiments to the conditions expected in the prototype. The scaling issues arise from the impossibility of obtaining transient data from the prototype system under off-nominal conditions. Solving the scaling issues implies developing approaches, procedures, and data suitable for predicting the prototype's performance utilizing small-scale models or data. [23]

Scale invariant represents the entities that are independent of scale, such as physics, DNS models. Scale invariant approaches are the ideal approach to explore and predict behaviors in real full-scale applications. There are two kinds of scale-invariant approaches: (1) Full-scale (or physics-conserved) experiment, which is (presumably) independent on the facility scale, (2) DNS modeling where the local information is solved accurately with very fine mesh. However, full-scale (fully physics-conserved) experiments are hard to build while many full-scale tests are required. Meanwhile DNS is computationally expensive to deal with the system scenario simulations. Reduced-order models e.g., LES (Large Eddy Simulation), RANS models and system

codes are not scale-invariant approaches. That is where scaling distortion exists, which refers to any discrepancy between the scaled parameter and the referenced plant parameters.

Although scaling distortion exists, computer codes including the three types mentioned before are widely used in NPP safety analysis because of cost and time effectiveness. The empirical models used in system codes are generally developed from the experiments where the IC/BC and geometry were not the typical NPP operating conditions, so many of them do not have the scalability to the nuclear reactor applications. Besides, some tuning constants, such as flow resistance coefficients, heat transfer fouling factors were used in the validation process to satisfy better agreement between the test data and the simulation. However, in fact, these tuning constants are not scalable for the extrapolation conditions although they could cover up the distortions in some specific conditions. The ghost of scalability issues also haunts in the applications of CFD and CFD-like codes. There are several physical models (turbulence models, wall laws) and numerical schemes in these codes, the extending use of which from a scaled test to a NPP application requires a great deal of evaluation and calibration. The effect of scaling on the model error/uncertainty calibrated from the data from scaled experiments greatly influences the accuracy of simulation and leads to an unknown error/uncertainty. The uncertainty due to scaling effect is called scaling uncertainty.

The data-driven framework proposed in this dissertation is expected to have the scalability to improve/correct the scale-distorted approaches that connect scaled data to the real full-scale applications and reduce the uncertainty of scaling.

2.2.3. Error Analysis of CFD-like Codes

Some error sources discussed in the previous part also exist in the modeling and application of these codes. They solve the conservation equations for mass, momentum and energy for multi-component, multi-phase flow. The phase balance equations are coupled by mechanistic models for interface mass, energy and momentum transfer that cover the entire flow regime from bubbly flow to film/drop flow, as well as single-phase flows. Different types of turbulence models are considered, such as mixing length model and two equation k–ε models. Model error due to physical simplification and mathematical approximation on these applied models, correlations and assumptions is one of the main error sources that should be considered. This is same as the system codes and RANS methods with wall functions.

For the thermal-hydraulic simulations using these codes, the key local phenomena in near-wall region are friction, turbulence and heat transfer. Respective correlations are applied for the simulation where characteristic lengths are introduced as one of the key parameters. The calculation of characteristic length is default executed using the local mesh size. Therefore, if users do not set the characteristic lengths by themselves, the mesh size they apply greatly affects the performance of the empirical correlations in the local near-wall cells. Here the mesh size is treated as one of the model parameters that determine whether the correlations are applied in their applicable ranges or not. This is one of the reasons why mesh size greatly affects the modeling and simulation results. Another major error source is also related to mesh size; here we call it "mesh error" to distinguish it with discretization error. The discretization error is proposed from the classic V&V point of view for the solving of PDEs, which assumes that when mesh size goes to zero the solution of PDEs converges. However, due to the correlation-based design in the simplified boundary-layer treatment, these CFD-like codes (e.g., GOTHIC) are not expected to converge when mesh size goes to zero due to very fine mesh may not satisfy the applicability of these empirical correlations. Taking GOTHIC as example, it applies finite volume technique with cell volume and surface porosities for complex geometries. The local instantaneous PDEs for mass, momentum and energy are time and space averaged to obtain the finite volume equations. Results from GOTHIC represent the averaged values of parameters over specified regions, not the exact value at the central points of the regions. Therefore, mesh error indicates the information loss of conservative and constitutive equations during the application of time and space averaging approaches. Other numerical errors in the system code due to iterative convergence, algorithm selection, coding error and finite arithmetic also exist in GOTHIC, but have less influence on the modeling and simulation compared to model error and mesh error.

2.3. Data-driven Modeling Application on Fluid Dynamics

Over the past decades, nuclear reactor thermal-hydraulics was developed successfully to meet most of the engineering practical requirements in nuclear reactor design and safety analysis. However, as discussed above, the difficulties in performing V&V of thermal-hydraulic codes and dealing with those main uncertainty/error sources with taking user effect into consideration still exist. This makes the development of nuclear reactor thermal-hydraulics become sluggish even though the body of knowledge and computer capability are greatly enlarged and improved.

Many contributions have been made on the development of data-driven approaches in the study of fluid dynamics, especially the data-driven turbulence closures to deal with the issues from model form uncertainty and knowledge lack of turbulence. Early in 2002, Milano used DNS results as HF data to train a Neural Network (NN) to replicate near-wall channel flows, but did not apply these NNs on forward models for turbulent flow prediction. [24] More recently in fluid dynamics, the rise of performance computing has led to the large HF data from DNS and well-resolved LES for the training and development of data-driven turbulence closures. RANS simulations, which provides significant savings in computational cost in comparison with DNS, have been used as LF model and coupled with these data-driven closures. Tracey and Duraisamy used NNs to predict the Reynolds stress anisotropy and source terms for turbulence transport equations. [25] Parish and Duraisamy introduced a multiplicative correction term for the turbulence transport equations using Gaussian Process Regression (GPR) with the uncertainty of this correction term quantified. [26] Zhang and Duraisamy also applied NNs to predict a correction factor for the turbulent production term in channel flow, which could affect the magnitude but not the anisotropy of the predicted Reynolds stress tensor. [27] Ling proposed the training of Random Forests (RFs) to predict the Reynolds stress anisotropy. [28] However, Ling and Templeton explored the capability of RFs and NNs in learning the invariance properties and concluded that RFs are limited in their ability to predict the full anisotropy tensor because they cannot easily enforce Galilean invariance for a tensor quantity. [29] So later Ling and Templeton used deep NNs with embedded invariance to predict the Reynolds stress anisotropy. [30] Different with those data-driven approaches above that directly predict Reynolds stress, Wang and Xiao proposed to apply RFs to predict the Reynolds stress discrepancy. [31] It should be noted that several well-selected physical features are used as the training input instead of physical coordinates in this approach. Another ML algorithm, Gene Expression Programming (GEP) was applied by Weatheritt and Sandberg to formulate the non-linear constitutive stress-strain relationships for turbulence modeling. [32] Recently, Zhu and Dinh performed a data-driven approach to model turbulence Reynolds stress leveraging the potential of massive DNS data. [33] The approach is validated by a turbulence flow validation case: a parallel plane quasi-steady state turbulence flow case. These efforts mainly contributes on improving the RANS capability for turbulence modeling, not for the application of commercial codes with fixed model forms for system-level large-space thermal-hydraulics such as containment thermal-hydraulics.

Most of these approaches focused on how to deal with model form uncertainty of RANS turbulence modeling without considering the numerical error due to discretization. Hanna and Dinh investigated the feasibility of a Coarse Grid CFD (CG-CFD) approach by utilizing ML algorithms to produce a surrogate model that predicts the CG-CFD local errors to correct the variables of interest. [34] This work focused on the correction of discretization error of CG-CFD without considering the model errors that may be introduced in CFD applications on thermal-hydraulic analysis. According to a classification of machine learning frameworks for thermal fluid simulation [15], the CG-CFD approach belongs to Type V ML which does not have requirement for prior knowledge. Type V ML fully relies on ML algorithms to discover the underlying physics directly from data.

2.4. Scope of This Work

The significance of this work stems from the application of data-driven modeling approach based on the concept of TDMI. The work scope mainly focuses on three parts:

2.4.1. Validation of Coarse-mesh CFD-like Codes

All these data-driven approaches reviewed in Section 2.3 are not designed for CFD-like or coarse-mesh CFD codes. These efforts analyzed model error and mesh error separately with another fixed, the logic of which is impractical to the coarse-mesh methods where mesh size is treated as a model parameter and mesh convergence is not expected. To overcome this difficulty in the V&V and application of CFD-like codes, OMIS is developed to deal with these two error sources together, as shown in Figure 2. OMIS framework is considered as a Type V ML framework since it treats the physical models, coarse mesh sizes and numerical solvers as an integrated model, which can be considered as a surrogate of governing equations and closure correlations of LF code. The development of this integrated model does not need relevant prior knowledge, and purely depends on existing data. Besides, compared to current data-driven efforts, OMIS framework is successfully applied in thermal-hydraulic modeling and simulations, not only in adiabatic fluid dynamics where previous efforts were focused on.

In some respects, OMIS is expected to provide a potential data-driven approach for the validation of these CFD-like codes in the system-level thermal-hydraulic modeling and simulations. As the response of trained data-driven model, simulation error in each cell is estimated according to the Physical Features (PFs) in each local cell. By introducing the concept of TDMI

and various PFs, the prediction of simulation error takes all the error sources into accounts and has a promising accuracy even for extrapolative conditions where validation data is not available. This scalability by exploring the local physics is discussed in following section.

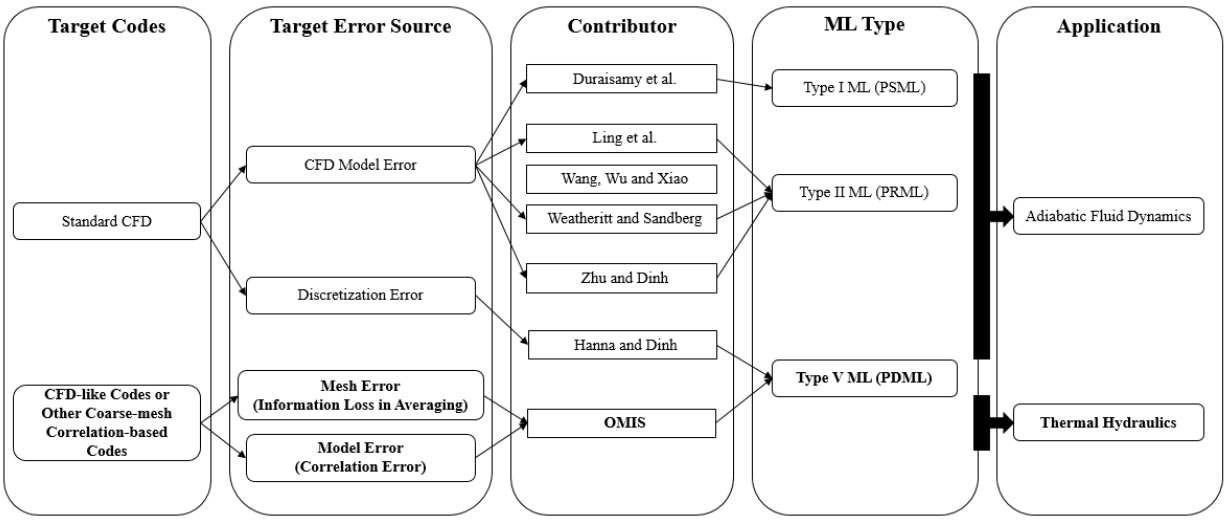


Figure 2. Review of Machine Learning Applications on Thermal-Hydraulic Modeling

2.4.2. Development of A Data-driven Scale-invariant Approach

Over the past few decades, many concepts of nuclear reactor have been proposed with different components, geometries and powers. The respective global physical conditions might be an "extrapolation" to previous simulations, which seems to bring large uncertainty into the demonstration simulations. The relevant thermal-hydraulic experiments with a wide range of scale and structures have to be designed for respective code development and validation, the data generated from previous simulations have to be abandoned in the corner. However, no matter how much the extrapolation of global physics is, local physics such as the interaction between liquid, vapor and heat structure may not change. This makes it possible that local physical parameters or variables in the local cells are similar even the global physical condition totally changes.

Firstly we need to identify the definitions of global physics and local physics: the former one indicates the global or macroscopic state, observation and deduction of the simulation target condition, such as the dimension, geometry, structure, boundary condition and non-dimensional parameters that represent the underlying physics; while the latter one refers to the microscopic state, observation and deduction of the simulation target condition. For example, the global physics of turbulent flow can be characterized using the value of Re number and geometries. No matter

how Re number or geometry changes, the local physics is always turbulence if the Re number is big enough.

According to the identification of global physics and local physics, four different physics coverage conditions are classified as Global Interpolation through Local Interpolation (GILI), Global Interpolation through Local Extrapolation (GILE), Global Extrapolation through Local Interpolation (GELI) and Global Extrapolation through Local Extrapolation (GELE).

For instance, there are several cases for single-phase fully developed flow in a pipe of diameter D: the local physical conditions and the values of Re_D which represent the global physical conditions are listed in Table 1. Assume some of cases as existing data and others as target simulation, then four different physics coverage conditions are specified with both of global and local physics taken into consideration, as shown in Figure 3. GILI condition represents the situation where the both global and local physical conditions of target case (Case 4) are identified as an interpolation of existing cases (Case 3 and 5). In this situation, the physics of target case is globally and locally "covered" by existing cases, the model developed using the sufficient data from Case 3 and 5 is reliable to predict the condition of Case 4. However, in GILE condition, even if the global physical condition of target case (Case 2) is covered by existing cases (Case 1 and 3), the data from existing cases is not able to inform the prediction of target case since the local physics are totally different. As proved in reality, the models developed from experiments of laminar flow or turbulent flow are not applicable for the transition prediction. GELE condition has the same problem, the models developed from the experiments of laminar flow is not applicable for the turbulence prediction. In these two conditions, the existing data does not contain the instructive information of the target so that it is useless no matter how much the data is used for model development.

Table 1. Example of Different Global and Local Physical Conditions

Case	Global Physical Condition	Local Physics
1	$Re_D = 10^2$	Laminar Flow
2	$Re_D = 2.5 \times 10^3$	Laminar–Turbulent Transition
3	$Re_D = 1 \times 10^4$	Turbulent Flow
4	$Re_D = 2 \times 10^4$	
5	$Re_D = 3 \times 10^4$	

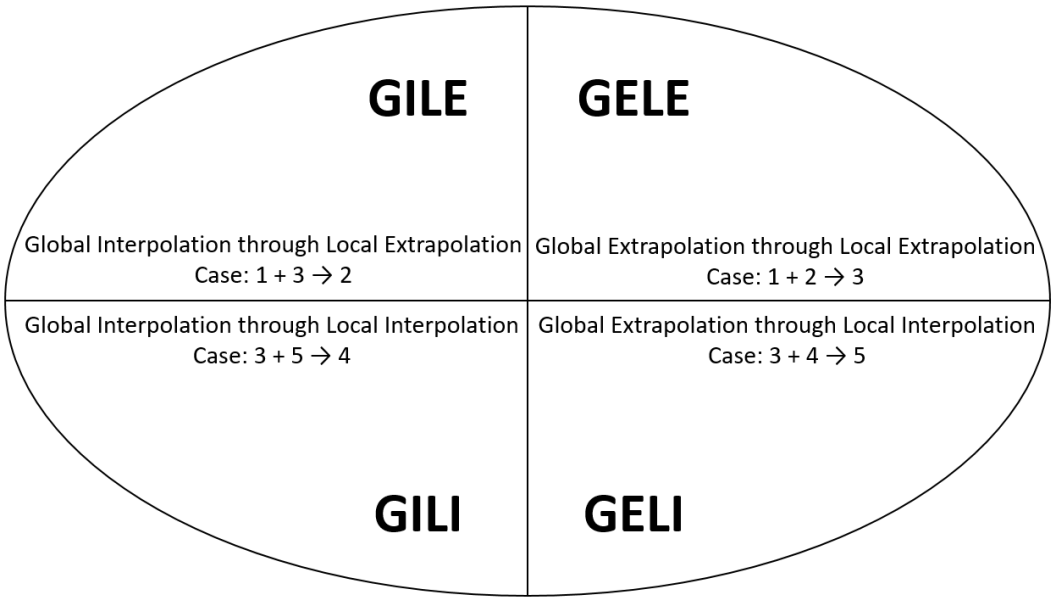


Figure 3. Illustration of Physics Coverage Condition Considering Global and Local Physics

GELI condition indicates the situation where the global physical condition of target case (Case 5) is identified as an extrapolation of existing cases (Case 3 and 4) but the local physics are similar, turbulent flow. The values of some representative parameters (e.g., local velocity gradients) are even interpolative in the existing cases. Unlike in GILE or GELE conditions, the local similarity in GELI condition provides the feasibility to take great benefits from the existing data to estimate the target case. Instead of endlessly evaluating the applicable ranges of models and scaling uncertainty, exploring the similarity of local physics opens another door to overcome the problems in global extrapolations. Based on the idea of TDMI, the specific physical models, local mesh sizes and numerical solvers are treated as an integrated model, where the interactions among different physical or numerical models are also taken into consideration. Data obtained from this integrated model can be used to construct a library that identifies and stores the local similarities in different physical conditions. This library is self-improvable and automatically updated as new qualified data is available. Once the library is built, ML algorithms are applied to find natural patterns in data that generate insight and help make better predictions.

Currently, the most comfortable physics coverage condition for thermal-hydraulic modeling or simulation is GILI condition where the existing data or experience has the capability to estimate the target case. And GELI condition has this potential capability when appropriate data and ML algorithm are coupled. The extrapolation of global physics indicates different global

physical conditions such as a set of characteristic non-dimensional parameters, or different IC/BCs, or different geometries/structures, or dimensions. The interpolation of local physics implies two definitions:

1. From the perspective of traditional knowledge on physics: the existing cases (no matter experimental or numerical tests) and target case are designed for the local physics with similar length scales and time scales, such as the turbulence example discussed above;

2. From the perspective of data characteristics: the underlying local physics of these cases is assumed to be represented by a set of Physical Features (PFs), and the PF data of target case is mostly covered or similar to the PF data of existing cases. This similarity is depending on the identification of PFs, data quality and quantity.

Targeting on the "GELI" condition, OMIS framework is developed as a TDMI approach that deals with data, physical model and coarse-mesh simulation in an integrated manner using ML algorithms. By concentrating on the similarity of local physics, OMIS framework has a potential scalability to the globally extrapolative conditions. The application of ML algorithms realizes the data-driven concept of OMIS by using computational methods to "learn" information directly from data without assuming a predetermined equation as a model. These algorithms adaptively improve their performance as the size of training data increases. The key outcomes of OMIS framework are (1) quantitatively measuring the PF similarity of existing data and target data, and (2) identifying the relationship between these local PFs and local simulation error for future predictions. After all, OMIS is promising to develop a data-driven scale-invariant approach to deal with scaling issues and provide evidence for the generation of validation data.

2.4.3. Supplement to Evaluation Model Development and Assessment Process

As reviewed in Section 1.2, Evaluation Model Development and Assessment Process (EMDAP) is too high-level and heuristic to implement even if it has formal and explicit descriptions for the concepts, definitions and processes. Besides, the acceptance criteria were not clearly defined. Same as CSAU, the system analysis and scaling analysis in EMDAP are highly heuristic and difficult to implement, and the mesh effect on code/model scalability was not fully considered. Especially in EMDAP Step 19: assess scalability of integrated calculations and data for distortions, necessary techniques are deficient on data assimilation or scalability assessment. By treating mesh error and model error together and introducing machine learning algorithms to

explore the local physics, OMIS framework has the potential to bridge the scale gap and work as a supplement to the implementation of EMDAP considering the industry requirements on validation of RISMC models. Mode details are discussed in Chapter 7.

2.5. Chapter Summary

This chapter introduces the technical background of OMIS framework. In Section 2.2, uncertainty and error sources in system-level thermal-hydraulic modeling and simulation are discussed from traditional Verification and Validation (V&V) perspective, followed by the overview of scaling issues and user effects in thermal-hydraulic applications. It implies that the traditional V&V frameworks are not applicable to these coarse-mesh CFD-like codes, whose mesh error and model error are tightly connected since mesh size is treated as one of the key model parameters. By performing a brief discussion on the error analysis of these codes, it is found that mesh convergence is not available due to the integral form of conservation equations. Therefore, a smart guide is urgently needed to provide advice on the optimal selections of coarse mesh size and models before the application of these codes.

Section 2.3 reviews the data-driven modeling applications in fluid dynamics. Most of these approaches focused on how to deal with model form uncertainty of RANS turbulence modeling without considering the numerical error due to discretization. All these data-driven approaches are not designed for CFD-like or coarse-mesh CFD codes. These efforts analyzed model error and mesh error separately with another fixed, the logic of which is impractical to the coarse-mesh methods where mesh size is treated as a model parameter and mesh convergence is not expected.

Section 2.4 describes the scope of this work from three respects: (1) provide a potential data-driven approach for the validation of these CFD-like codes in the system-level thermal-hydraulic modeling and simulations; (2) develop a data-driven scale-invariant approach to deal with scaling issues and provide evidence for the generation of validation data; (3) work as a supplement to the implementation of EMDAP.

CHAPTER 3. REVIEW OF COARSE-MESH SIMULATION TOOL: GOTHIC

3.1. Introduction

This chapter reviews the Low-Fidelity (LF) simulation tool, GOTHIC, which is a coarse-mesh CFD-like software initially developed for containment thermal-hydraulic analysis. The structure on thermal-hydraulic modeling is described in Section 3.2 including conservation equations, source terms, and closure models involved. The relationship between mesh and key closure models are discussed. Section 3.3 describes a qualitative assessment of thermal-hydraulic simulation using GOTHIC including mesh and model sensitivity study.

GOTHIC allows users to build models for solving complex thermal hydraulics problems involving multiphase flow of steam, water drops, liquid water and non-condensing gases with interface heat and mass transport. Fluid regions can be represented by lumped parameter nodes, 1D, 2D and 3D grids. Physical models are included for interphase drag and heat and mass transfer to model boiling, evaporation and condensation in a wide range of flow regimes including single phase, bubbly and film/drop flows. Each phase is tracked with its own set of mass, energy and momentum balance equations to allow modeling thermal nonequilibrium, phase slip and counter current flows. The vapor phase is made up of steam and any number of non-condensing gas components. [3]

As a coarse-mesh thermal-hydraulic analysis software for modeling and simulation of containment processes, GOTHIC code has been evaluated, validated, and applied in deterministic safety analysis. In particular, GOTHIC code has been successfully employed for analysis of containment thermal-hydraulic during loss of coolant accidents (LOCAs). [5]

In the previous study, a demonstration GOTHIC model has been developed for BWR Mark I containment and successfully applied to investigate the performance of reactor safety system and containment venting processes during SBO accident scenario. [4,35,36] GOTHIC has the capability to simulate the dynamical performance of reactor systems needed for analysis of reactor depressurization and containment venting. It allows an effective description and integration of plant components in 0-D (i.e., lumped parameter), 1-D (e.g., piping network), and 3-D (recirculation flow). This advanced capability in GOTHIC allows analysis of complex thermal-hydraulic scenarios involving 3-D flow patterns (e.g., in containment) and 1-D pipe network.

3.2. GOTHIC Structure on Thermal-Hydraulic Modeling and Simulation

GOTHIC structure on thermal-hydraulic modeling and simulation contains conservation equations, source terms and closure models as shown in Figure 4. The conservation equations build the physics basis for the multiphase thermal-hydraulic analysis. The source terms in these averaged conservation equations respectively represent the mass/momentum/energy sources from external environment (thermal boundaries, plant equipment or chemical reactions) or due to phase change. In order to solve those source terms, several closure models are introduced and coupled.

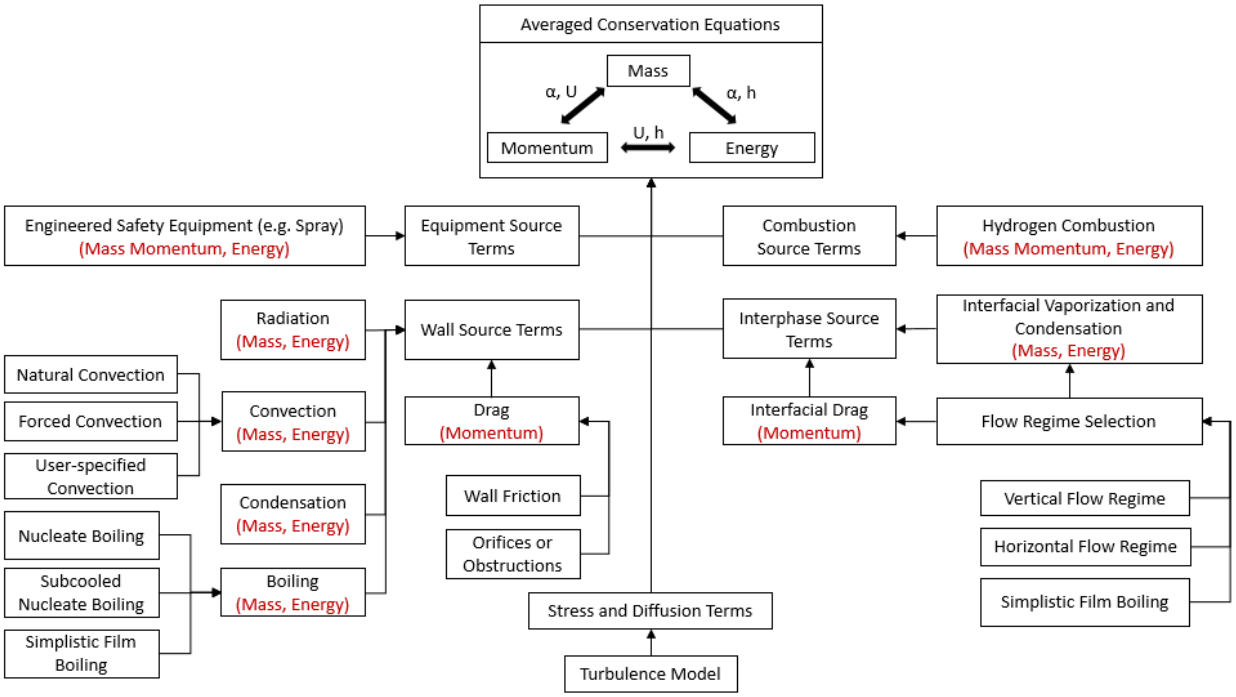


Figure 4. GOTHIC Structure on Thermal-Hydraulic Modeling and Simulation

3.2.1. Conservation Equations

The conservation equations are written in integral form because this form is closely related to the finite volume numerical method used to solve the equations. Mass, momentum and energy conservation equations are derived from local instantaneous formulation of differential balance equations using time and volume averaging methods, as shown in Equation (1), (2) and (3). Mass conservation is expressed as below,

$$\frac{\partial}{\partial t}\int_V \theta\alpha_\varphi\rho_{\varphi\zeta}dV = -\int_A \Psi\alpha_\varphi\rho_{\varphi\zeta}\overrightarrow{u_\varphi}\cdot\vec{n}dA + \int_{A_f}\Psi\alpha_\varphi\rho_\varphi D_\varphi^c\vec{\nabla}\left(\frac{\rho_{\varphi\zeta}}{\rho_\varphi}\right)\cdot\vec{n}dA$$

Storage *Convection* *Diffusion*

$$+\int_{A_w} s_{\varphi\zeta}^c dA + S_{\varphi\zeta}^c + E_{\varphi\zeta}^c + C_{\varphi\zeta}^c \quad (1)$$

Boundary (Wall) Source *Interface Source* *Equipment Source* *Combustion Source*

The subscript φ refers to the phase and takes on the values v (vapor), l (liquid). The subscript ζ refers to a component of the vapor (ζ = s for the steam component, ζ= n for a single component of the noncondensing gas mixture, and ζ = g for the noncondensing gas mixture). GOTHIC applies porous media approach, θ is the volume porosity and is Ψ the area porosity factor. The porosity factors range from 0 to 1 with a value of 1 for a completely unobstructed volume or area. α is the volume fraction, ρ is the density, u is the velocity, is outward normal to the surface dA. A_f is that portion of the total surface area in contact with adjacent fluid volumes. D^c is the mass diffusion coefficient, including turbulence effects only. s^c is the mass source per unit area generated at, or passing through, bounding wall A_w. S^c is the mass source due to interaction with other phases (e.g., evaporation, condensation, drop entrainment deposition), E^c is the mass source from engineered safety equipment and C^c is the mass source from hydrogen combustion. Momentum conservation is expressed as below,

$$\frac{\partial}{\partial t}\int_V \theta\alpha_\varphi\rho_\varphi\overrightarrow{u_\varphi}dV = -\int_A \Psi\alpha_\varphi\rho_\varphi\overrightarrow{u_\varphi}\left(\overrightarrow{u_\varphi}\cdot\vec{n}\right)dA + \int_{A_f}\Psi\alpha_\varphi\sigma_\varphi\cdot\vec{n}dA + \int_V \theta\vec{g}\alpha_\varphi\rho_\varphi dV$$

Storage *Convection* *Surface Stress* *Body Force*

$$+\int_{A_w}\overrightarrow{s_\varphi^m}dA + \overrightarrow{S_\varphi^m} + \overrightarrow{E_\varphi^m} \quad (2)$$

Boundary (Wall) Source *Interface Source* *Equipment Source*

σ includes the static pressure and the viscous and Reynolds stress terms, $\vec{g}$ is the gravitational acceleration, $\overrightarrow{s_\varphi^m}$ is the momentum source per unit wall area, $\overrightarrow{S_\varphi^m}$ is the momentum

source due to interphase exchange (drag and phase transition) and $\overrightarrow{E_{\varphi}^{m}}$ is the momentum source from equipment. All components of the vapor are assumed to move at the same velocity. The density in the vapor momentum equation includes the steam and gas component densities, (each at their own partial pressure). Energy conservation is expressed as below,

$$\underbrace{\frac{\partial}{\partial t}\int_{V}\theta\alpha_{\varphi}(\rho_{\varphi}(h+ke)_{\varphi}-P)dV}_{\textit{Storage}} = \underbrace{-\int_{A}\Psi\alpha_{\varphi}\rho_{\varphi}(h+ke)_{\varphi}\overrightarrow{u_{\varphi}}\cdot\vec{n}dA-\int_{V}P\frac{\partial}{\partial t}(\theta\alpha_{\varphi})dV}_{\textit{Convection and Work}}$$

$$+\underbrace{\int_{A_f}\Psi\alpha_{\varphi}\rho_{\varphi}c_{p\varphi}D_{\varphi}^{e}\vec{\nabla}T_{\varphi}\cdot\vec{n}dA}_{\textit{Thermal Diffusion}} \quad \underbrace{+\sum_{\zeta}\int_{A_f}\Psi\alpha_{\varphi}D_{\varphi}^{c}\vec{\nabla}\left(\frac{\rho_{\varphi\zeta}}{\rho_{\varphi}}\right)h_{\varphi\zeta}\cdot\vec{n}dA}_{\textit{Mass Diffusion}} \tag{3}$$

$$\underbrace{+\int_{A_w}s_{\varphi}^{e}dA}_{\substack{\textit{Boundary (Wall)}\\ \textit{Source}}} \quad \underbrace{+S_{\varphi}^{e}}_{\substack{\textit{Interface}\\ \textit{Source}}} \quad \underbrace{+E_{\varphi}^{e}}_{\substack{\textit{Equipment}\\ \textit{Source}}} \quad \underbrace{+C_{\varphi}^{e}}_{\substack{\textit{Combustion}\\ \textit{Source}}}$$

where h is enthalpy, ke is the kinetic energy, P is the static pressure, D_{φ}^{e} is the thermal diffusion coefficient, s_{φ}^{e} is the energy source per unit wall area, S_{φ}^{e} is the interphase energy source, E_{φ}^{e} is the equipment energy source and E_{φ}^{e} is the energy source from hydrogen combustion. Kinetic energy is included or neglected by user selection, and all other energy forms not explicitly represented above are neglected. Viscous dissipation is also neglected. The kinetic energy is defined as $ke_{\varphi}=\frac{{u_{\varphi}}^{2}}{2}$. All components of the vapor are assumed to be at the same temperature. The enthalpy in the vapor energy is the mixture energy of the steam and noncondensing gas mixture. The energy transported with the mass through mass diffusion is included only for the vapor.

3.2.2. Source Terms

I. Boundary (Wall) Source Terms

The boundary (wall) source terms in the conservation equations above include convection and radiation heat transfer, condensation and boiling at the wall (as mass and energy sources) and

friction and orifice drag (as momentum sources). The boundary mass source terms provide mass sources and sinks due to phase change at a non-fluid surface. For example, condensation heat transfer on conductors results in a mass sink for the steam and a mass source for the liquid. The boundary energy source terms include convection and radiation heat transfer from walls and the energy associated with any surface mass source terms. It is assumed that all wall heat transfer is between the walls and the liquid and vapor phases. For each conductor, the energy source depends on the particular heat transfer option selected by the user. The logic for selecting the heat transfer coefficients for heat transfer between the conductor surface and the fluid in GOTHIC is shown in Figure 5. α_{l-lim} is the limit value for liquid fraction to determine whether the fluid in the cell is vapor only or two-phase fluid. $T_{sat}(P_{vs})$ is the saturation temperature under the steam pressure. T_w is the wall temperature and T_l is the liquid temperature. The heat transfer correlations built into GOTHIC that are accessed with this model apply to the heat transfer at the conductor surface, covering the portion of the boiling curve which spans single phase heat transfer to pre-CHF (Critical Heat Flux) heat transfer.

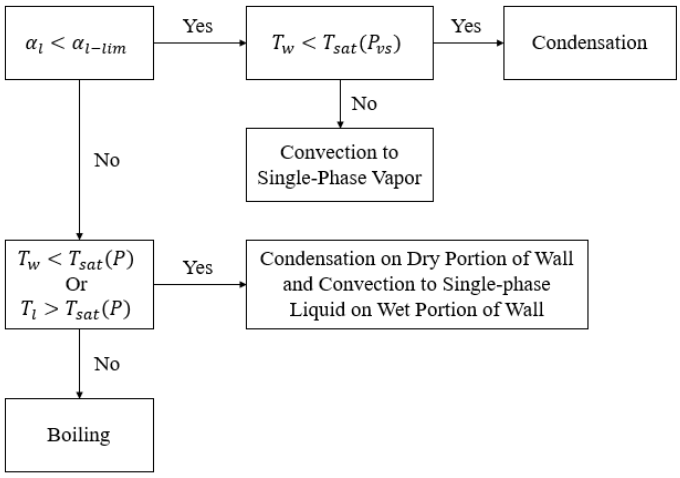


Figure 5. Heat Transfer Selection Logic in GOTHIC

- **Wall Convection Heat Transfer Model**

The boundary source terms for the fluid energy equations include convection and radiation heat transfer from walls and the energy associated with any surface mass source terms. The convection energy source terms for the conductor surfaces are,

$$Q_{w_l} = \lambda_t \lambda_{w_l} A_{cn} H_{conv_l} (T_w - T_l) \qquad (4)$$

And

$$Q_{w_v} = \lambda_t \lambda_{w_v} A_{cn} H_{conv_v} \Delta T_{conv_v} \quad (5)$$

where Q_{w_v} and Q_{w_l} represent the heat flux from the wall to the vapor and liquid. A_{cn} is the conductor surface area, λ_t is a user-defined multiplier on the heat transfer coefficient with a default value of 1.0. λ_t is input as a forcing function. λ_w is the user-defined heat transfer option for different phases. $\lambda_{w_l} = 1$ means only liquid is considered for heat transfer, $\lambda_{w_l} = 0$ means only vapor is considered for heat transfer. $\lambda_{w_v} = 1 - \lambda_{w_l}$. If both of liquid and vapor are considered for heat transfer then the portion of the conductor surface covered by liquid is assumed to be given by the fraction indicated in following equation with lower and upper limits on the liquid volume fraction specified by the user. ΔT_{conv_v} represents the temperature difference for convection between the surface and the vapor. The convection heat transfer coefficient may be specified by the user or it is calculated from correlations for natural and forced convection as,

$$H_{conv} = Max \begin{bmatrix} Natural\ Convection, H_{nc} \\ Forced\ Cnvection, H_{fc} \\ H_{conv_{min}} \end{bmatrix} \quad (6)$$

The user may supply a minimum value for the convection heat transfer coefficient as $H_{conv_{min}}$, otherwise the default lower limit for convective heat transfer is based on simple conduction through stagnant fluid and is calculated as $H_{conv_{min}} = \frac{k}{L_c}$, where *k* is the fluid thermal conductivity and L_c is the user specifiable effective conduction length. The default value for L_c is $\frac{D_h}{8}$ where D_h is the cell hydraulic diameter. There are several correlations available in GOTHIC to define a convection heat transfer coefficient on the surface of a thermal conductor, which is defined as,

$$H_{nc} = \frac{k}{l} Cf(Gr, Pr) \quad (7)$$

$$H_{fc} = \frac{k}{l} C Re^m Pr^n \quad (8)$$

where *C*, *m* and *n* are constants. The local heat transfer coefficient is calculated using respective correlations for natural convection and forced convection with local characteristic length l, Gr and Pr, or Re and Pr. The local characteristic length for all of the above correlations

is the cell hydraulic diameter D_h, unless the user provides a specific value. The calculation of local Re and Gr are based on the same characteristic length. Figure 6 shows the schematic of how GOTHIC calculates the boundary energy source terms for convection heat transfer.

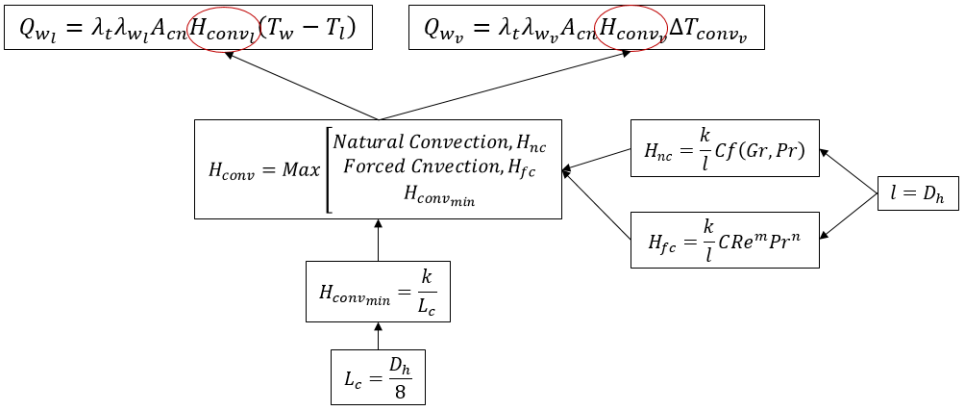


Figure 6. Schematic of the Calculation of Boundary Energy Source Terms

- **Wall Drag Force Model**

The boundary momentum source includes friction and form drag due to walls, orifices and obstructions. Drag force on each phase contains drag from orifices/obstructions and wall friction, the drag force coefficients are calculated using experience correlations for different conditions (laminar and turbulence). The total drag force on phase ϕ is,

$$D_\phi = D_\phi^o + D_\phi^f \qquad (9)$$

where the subscript ϕ refers to the phase and takes on the values *v* (vapor), *l* (liquid). D_ϕ^o is the drag from orifices or obstructions and D_ϕ^f is the drag force from wall friction. The user supplies a drag coefficient for each flow connection. The total orifice or obstruction drag force on each phase is assumed to be proportional to the area fraction of the phase, giving

$$D_\phi^o = A\alpha_\phi \frac{K}{2} \rho_\phi u_\phi |u_\phi| \qquad (10)$$

where K is the user-specified drag coefficient, A is the free area of the flow connection and u_ϕ is the velocity component normal to A. Wall friction drag is calculated for the vapor and liquid phases only and for co-flow is given by

$$D_{\phi}^{f} = \lambda_{f_{\phi}} {\lambda_{\phi o}}^{2} A \frac{l_w \cdot f(Re_{G_{\varphi}})}{2D_h} \rho_{\phi} u_{\phi} |u_{\phi} \lambda_{exp}| \tag{11}$$

where l_w is the wall length and D_h is the cell hydraulic diameter; $\lambda_{f_{\phi}}$ represents a ramp functions that puts all of the drag on the liquid phase until the flow is in the single-phase vapor regime; λ_{exp} is an expansion factor that approximates the increase in frictional drag due to the expansion of the fluid as the pressure falls along the length of the duct; The phase dependent multiplier ${\lambda_{\phi o}}^{2}$ is set to 1 for the vapor phase and is defined for the liquid phase by the Fridel correlation [37]. The expression $f(Re_{G_{\varphi}})$ is the friction factor that is dependent on the local Reynolds number of the phase.

$${\lambda_{lo}}^{2} = E + \frac{3.24FH}{Fr^{0.045} We^{0.035}} \tag{12}$$

$$E = (1 - x_f)^2 + (x_f)^2 \frac{\rho_l f(Re_{G_{\varphi}})}{\rho_v f(Re_{G_{\varphi}})} \tag{13}$$

$$F = (x_f)^{0.78} (1 - x_f)^{0.242} \tag{14}$$

$$H = \left(\frac{\rho_l}{\rho_v}\right)^{0.91} \left(\frac{\mu_v}{\mu_l}\right)^{0.19} \left(1 - \frac{\mu_v}{\mu_l}\right)^{0.7} \tag{15}$$

$$Fr = \frac{G^2}{g D_h \rho_{tp}^2} \tag{16}$$

$$We = \frac{G^2 D_h}{\rho_{tp} \sigma} \tag{17}$$

$$x_f = \frac{\alpha_v \rho_v u_v}{G} \tag{18}$$

$$\rho_{tp} = \left(\frac{x_f}{\rho_v} + \frac{1 - x_f}{\rho_l}\right)^{-1} \tag{19}$$

$$f(Re_{G_{\varphi}}) = \max \begin{cases} f(Re_{G_{\varphi}})_{lam} \\ f(Re_{G_{\varphi}})_{turb} \end{cases} \tag{20}$$

$$f(Re_{G_{\varphi}})_{lam} = \frac{64 \lambda_G}{Re_{G_{\varphi}}} \tag{21}$$

$$\frac{1}{\sqrt{f(Re_{G_\varphi})_{turb}}} = -2\log\left[\frac{\frac{\varepsilon}{D_h}\lambda_G}{3.7} - \frac{4.518\lambda_G}{Re_{G_\varphi}}log\left(\frac{6.9\lambda_G}{Re_{G_\varphi}} + \left(\frac{\frac{\varepsilon}{D_h}\lambda_G}{3.7}\right)^{1.11}\right)\right] \quad (22)$$

where ε is the wall roughness, λ_G is the geometry adjustment factor that accounts for non-circular geometry effects and is given by the following equation $\lambda_G = \frac{G_L}{64}$. G_L is the laminar friction geometry factor and has a default value of 64 for circular pipes. The Reynolds number is calculated as though the total mass flux consists of only that phase.

$$Re_{G_\varphi} = \frac{GD_h}{\mu_\phi} \quad (23)$$

$$G = \alpha_l\rho_l u_l + \alpha_v\rho_v u_v \quad (24)$$

Here the correlations of D_ϕ^f and $f(Re_{G_\varphi})$ implies that the control volume is considered as an equivalent pipe with inside diameter as D_h and height as l. Wall length l_w can be considered as the mesh size of the local cell along the friction direction. The calculation of D_h is different for different local mesh sizes and wall conditions. D_h works as a parameter to the wall friction model. Figure 7 shows the schematic of how GOTHIC calculates the boundary energy source terms for convection heat transfer.

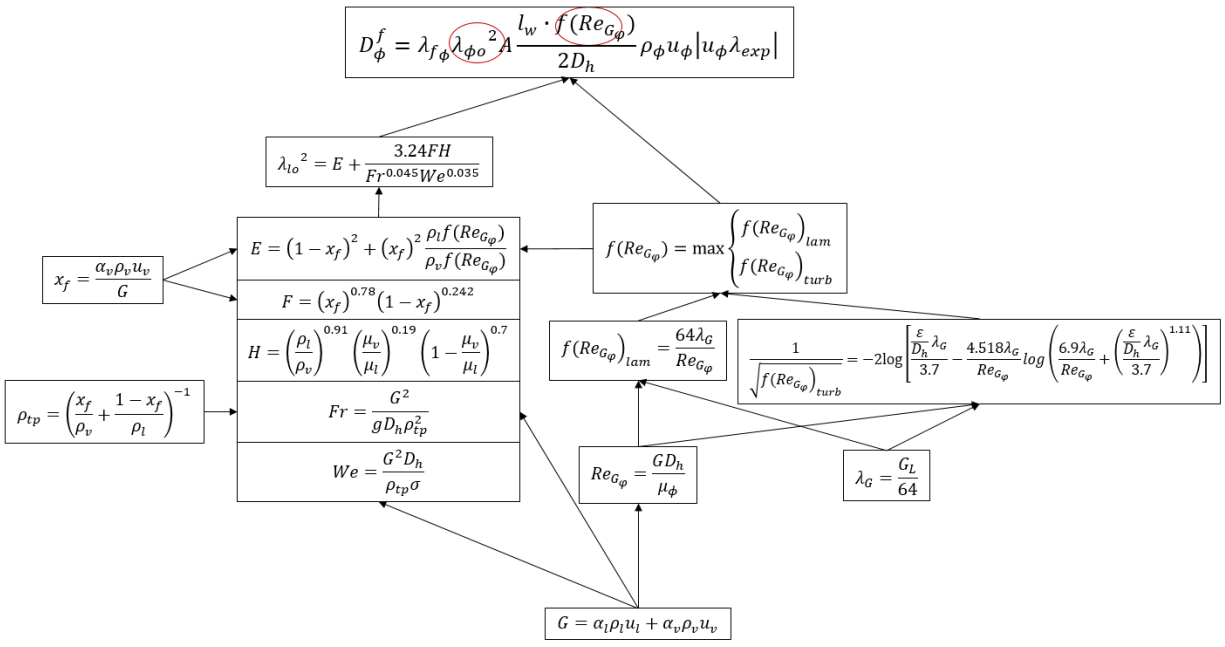


Figure 7. Schematic of the Calculation of Boundary Momentum Source Terms

II. Stress and Diffusion Terms

- **Turbulence Model**

The stress tensor σ_ϕ in the surface stress term of the momentum equation, Equation (2), includes the effects of static pressure, viscous shear and turbulent diffusion of momentum. The components of the mass and energy diffusion that are due to turbulence are closely related to that for turbulent momentum diffusion and, therefore, are also discussed in this section. Mass diffusion coefficient D_φ^c in Equation (1), and in the energy diffusion coefficient D_φ^e in Equation (3) can be obtained by using turbulence models in GOTHIC. σ_ϕ includes the effects of static pressure, viscous shear and turbulent diffusion of momentum.

$$\sigma_\phi = \sigma_{ij} = -P\delta_{ij} + (\mu + \mu^T)\left(\bar{u}_{i,j} + \bar{u}_{j,i}\right) - \frac{2}{3}\delta_{ij}\rho k \qquad (25)$$

where P is static pressure, δ_{ij} is the special tensor, μ is molecular viscosity, μ^T is turbulent viscosity, $\bar{u}_{i,j} = \frac{\partial \bar{u}_i}{\partial x_j}$, $\bar{u}_i$ is the time-averaged velocity in the x_i direction, k is the turbulent kinetic energy, $k = \frac{1}{2}\overline{u_i' u_i'}$. The mass diffusion coefficient and thermal diffusion coefficient include two parts: molecular diffusivity and turbulent diffusivity:

$$D^c = D^{c,m} + D^{c,T} \qquad (26)$$

$$D^e = D^{e,m} + D^{e,T} \qquad (27)$$

The molecular diffusivity is determined by the property of fluid, while the mass and energy turbulent diffusivity are respectively defined as the following equations.

$$D^{c,T} = \frac{\mu^T}{\rho Sc^T} \qquad (28)$$

$$D^{e,T} = \frac{\mu^T}{\rho Pr^T} \qquad (29)$$

Experimental evidence indicates that the turbulent Schmidt Sc^T and Prandtl number Pr^T vary only slightly within a flow field or from flow to flow and can usually be treated as constants. In GOTHIC, the recommended value for Sc^T and Pr^T is 1. Therefore, turbulence effects in the local control volumes can be modeled by obtaining expressions for μ^T and k. Two types of turbulence models are considered in GOTHIC, mixing length model [38] and two equation

k–ε models [39]. The Prandtl mixing length model was developed for unidirectional flow over a flat plate in the *x* direction. Because of these simplistic origins, the model provides reasonable results for situations involving uniform geometries and flow fields. The model is dependent on a user-specified mixing length which is highly problem dependent. However, the model is easy to use and does not require the solution of additional transport equations to obtain the essential turbulent parameters noted in the preceding section. The shortcomings of the mixing length model have led to development of the two-equation k–e model which solves partial differential equations that model the transport of effective parameters for calculating local turbulence, these being the velocity scale and length scale. Although this model solves transport equations to get the spatial and temporal distribution of the turbulent velocity and length scale, it still relies on empirically determined coefficients to close the model. Therefore, the model is only strictly reliable within the range of problems for which the empirical coefficients have been verified. The eddy viscosity is given in terms of the kinetic energy and the dissipation by

$$\mu^T = C_\mu \rho \frac{k^2}{\varepsilon} \tag{30}$$

- **Near-wall Treatment in GOTHIC**

For computational cells adjacent to walls, the turbulence parameters are obtained using an approach using the logarithmic law of the wall which gives the near wall velocity profile as, [40]

$$U = \frac{U_f}{K} \ln\left(\frac{Ey\rho U_f}{\mu}\right) \tag{31}$$

where U is the velocity parallel to the wall at a distance y from the wall, K is the von Karman constant ($K \approx 0.4$) and E is a roughness parameter ($E \approx 9.0$). U_f is the friction velocity that is defined as,

$$U_f = \frac{\sigma_w}{\mu} \tag{32}$$

where σ_w is the wall shear stress. Equation (31) is used for the calculation of U_f where U is set as the cell center velocity and y as the distance for cell center to wall. In the near wall region, it is assumed the turbulent shear stress is equal to the wall stress and that the production and destruction of turbulence are in equilibrium. Under this condition,

$$k_w = \frac{U_f}{\sqrt{C_\mu}} \tag{33}$$

$$\varepsilon_w = \frac{{U_f}^3}{Ky} \tag{34}$$

where y, the distance to the wall, is assumed to be proportional to the hydraulic diameter as,

$$y = \frac{D_h}{8} \tag{35}$$

This gives the correct wall to cell center distance for a rectangular cell with a wall on one side. The kinetic energy and turbulent dissipation for cells adjacent to walls are defined as,

$$k = {k_w}^\gamma {k_f}^{1-\gamma} \tag{36}$$

$$\varepsilon = {\varepsilon_w}^\gamma {\varepsilon_f}^{1-\gamma} \tag{37}$$

where k_f and ε_f are the turbulence parameters calculated for a free cell without wall connected. The transition parameter γ is given by,

$$\gamma = Min\left[Max\left(0, \frac{1}{\frac{w}{y} - 1}\right)\right. \tag{38}$$

where w is the width of the cell, as shown in Figure 8. This forces the turbulence parameters to the wall values if y is less than one-half the cell width and relaxes them to the free mesh values as y becomes greater than the cell width.

$D_h = \frac{4abc}{bc} = 4a$

$y = \frac{D_h}{8} = \frac{a}{2}$

(a is cell width normal to the wall)

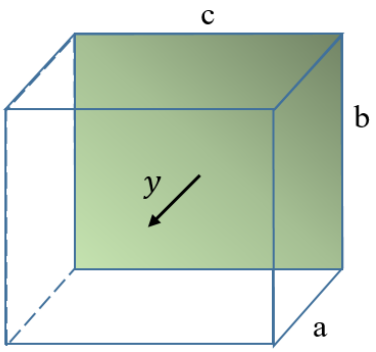


Figure 8. Calculation of Wall Distance in GOTHIC

Figure 9 summarizes how GOTHIC calculates the stress and diffusion terms to close the conservation equations by using two-equation *k*–ε model. For free cells without walls connected,

the turbulence model is directly applied to solve k and ε to obtain μ^T. For the cells adjacent to walls, k and ε are respectively divided into two parts, one part is calculated as the free cell condition, another part is calculated based on the friction velocity U_f, the distance to the wall y and local cell width w. The values of y and w are related to the local mesh size, as shown in Figure 8. The friction velocity U_f is calculated based on the velocity in the cell center parallel to the wall.

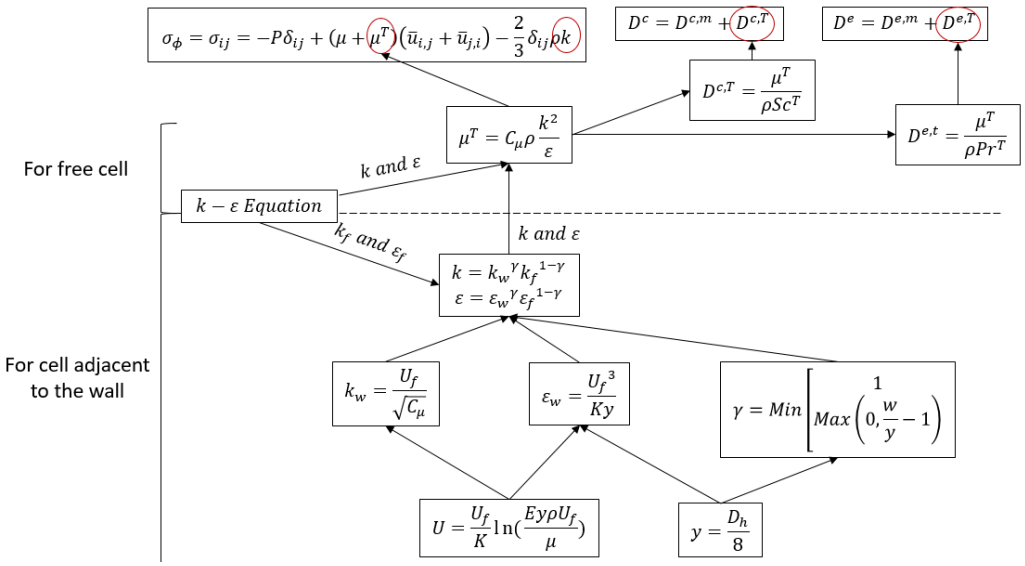


Figure 9. Schematic of Calculation of Stress and Diffusion Terms in GOTHIC

III. Other Source Terms

The interface source terms include the mass, energy and momentum transfer from one phase to another due to vaporization or condensation. The source terms for phase transition are obtained by mass and energy balances for the interfaces. It is assumed that no mass or energy is stored at the interface. The selection of interfacial heat transfer coefficients, drag coefficients and areas greatly depends on the geometry of the flow. While many of flow regime maps are useful within the range of the data for which they were developed, they cannot be generally applied to all two-phase flow problems. The accurate prediction of exactly which flow regime can be expected under a given set of flow conditions is beyond the current understanding of two-phase flow. Furthermore, the flow regime selection must be applicable to the discrete representation of the flow field imposed by the grid of computational volumes. With this in mind, the physical basis of existing flow regime maps was used to develop a widely applicable and yet simple flow regime map for use in GOTHIC. For each computational cell in a model, the flow regime must be decided

upon before correlations for mass, energy and momentum transfer can be applied. The flow regime is determined using only information for the cell and its immediate neighbors. Different flow regime maps are provided in GOTHIC for interface heat, mass and momentum transfer: Vertical flow regimes, horizontal flow regimes, lumped parameter flow regimes, junction flow regimes. Equipment source terms provide the mass/energy/momentum specification form engineered safety equipment, such as spray, pump/fan, heat exchanger, cooler/heater. Combustion source terms introduce the inputs from hydrogen combustion. [3]

3.2.3. Relationship between Mesh and Key Closure Models

According to the reviews on these three models (wall friction, convection heat transfer and turbulence model), the local mesh size directly influences the performance of empirical correlations applied. Figure 10 shows the relationship between mesh size and these models. Friction model provides wall friction drag force to momentum conservation equation, which provides velocity in the cell adjacent to the wall to turbulence model to calculate the kinetic energy and turbulent dissipation. All these models require the information relevant to mesh size: hydraulic diameter, wall length or the distance to the wall. The calculation of these parameters is based on the mesh size of the local cell. However, there are different wall or flow conditions for the wall friction where the calculation of cell hydraulic diameter should apply different formulas. In the turbulence modeling, the distance to the wall and the cell width determine whether the turbulent parameters are more close to the ones in free cell or the ones in near-wall region. The relationship among the length, width and height of local cell also should be taken into consideration.

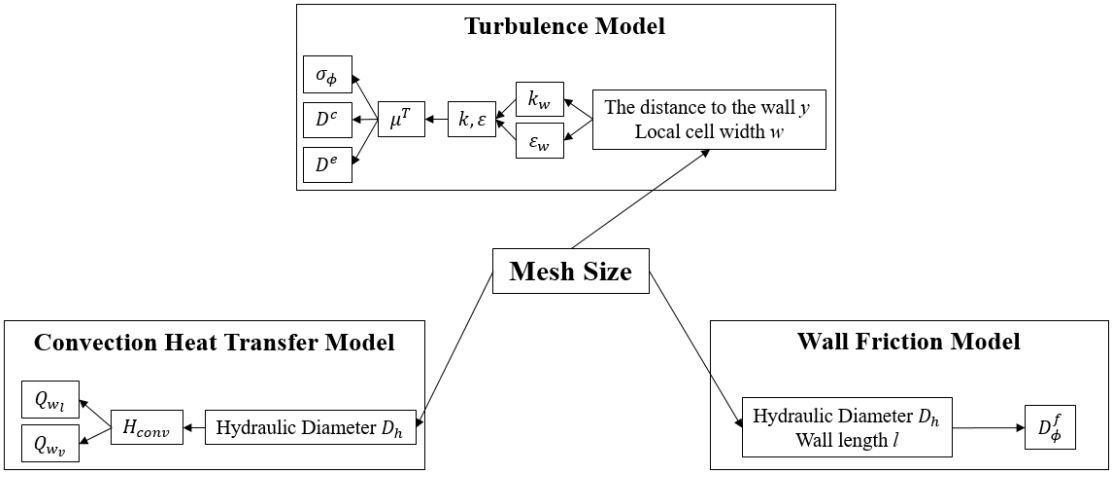


Figure 10. Relationship between Mesh size and Closure Models

3.3. Qualitative Assessment of Thermal-Hydraulic Simulation using GOTHIC

As discussed before, the main error sources when using GOTHIC for system-level thermal-hydraulic simulation include model error due to physical simplification and mathematical approximation, mesh error due to the information loss in applying averaging methods on balance/constitutive equations, and other numerical errors. All these error sources lead to not only the error for single phenomena simulation, but also the error propagation then further to the simulation error for system scenario simulation. In this section, qualitative assessment of thermal-hydraulic modeling and simulation using GOTHIC is performed based on several case tests for system-level thermal hydraulics including mesh and model sensitivity study.

3.3.1. Natural convection with Heat Source in A Cavity

Natural convection flow and heat transfer in a fluid layer with a volumetric heat source are of interest in certain geophysical, astrophysical, and technological problems. Here the natural convection study with volumetric heat in a horizontal fluid (water) layer was performed using GOTHIC. The cubic cavity has the boundary condition: a rigid and insulated bottom boundary, a cold upper boundary (20 ^{o}C) and periodic boundary at each side. The length of the cubic is 2 inch and the volumetric heat generated with the desired value of Rayleigh number (9.3×107). The standard two-equation k–ε turbulence model was applied for this case. As shown in Figure 11, different mesh sizes were used for the GOTHIC 2D model: $\Delta x/H = \Delta z/H = 0.1$ for 10*10 node, $\Delta x/H = \Delta z/H = 0.05$ for 20*20 node, $\Delta x/H = \Delta z/H = 0.033$ for 30*30 node, $\Delta x/H = \Delta z/H = 0.025$ for 40*40 node, $\Delta x/H = \Delta z/H = 0.02$ for 50*50 node, $\Delta x/H = \Delta z/H = 0.01$ for 100*100 node.

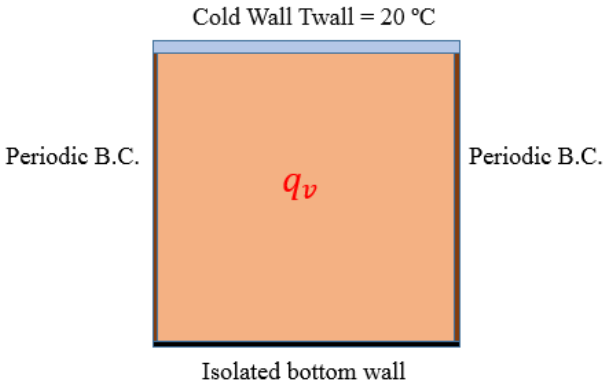


Figure 11. The Illustration of 2D Model for the GOTHIC Simulation of Natural Convection

After quasi-steady state was reached, the non-dimensional temperature profile $T^* = \frac{2k(T-T_w)}{q_v H^2}$ was obtained. Figure 12 displayed the time-averaged non-dimensional temperature profile in the center horizontal location with different mesh sizes, and also included the data from DNS simulation [41] and results based on OpenFOAM RANS fine-mesh simulation [42]. The temperature profile with 30*30 node fitted best with the high-fidelity results. It should be noted that the finer mesh size did not provide higher accuracy in GOTHIC modeling and simulation in this case. And the results did not show a convergence as the mesh size decreased.

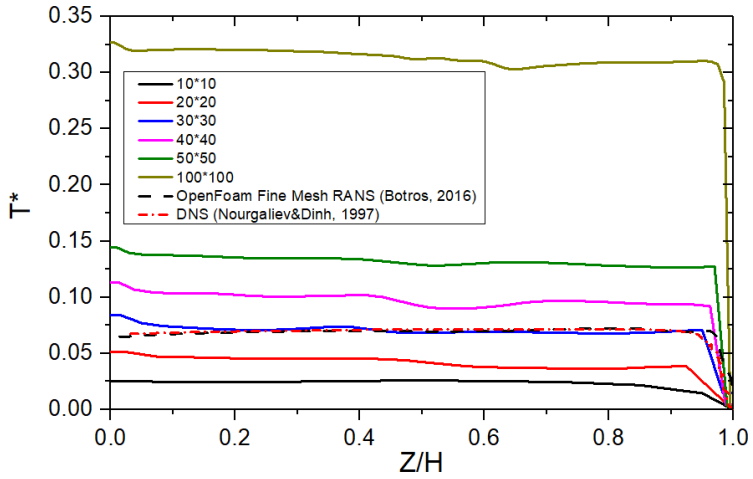


Figure 12. Time-Averaged Non-dimensional Temperature Profiles with Different Meshes

In Figure 13, different results using 2D and 3D models with same mesh size were compared with the high-fidelity data. Based on the results of Figure 12, 2D model with 30*30 node had the least simulation error of non-dimensional temperature profile compared to other nodes. A 3D model with same mesh size is applied for the same natural convection case to check whether the same mesh size was also the optimized option for 3D simulation. However, the non-dimensional temperature profile using 3D model is much different with the DNS and RANS high-fidelity simulations. One reason is that the information lost during the space and time averaging in different dimensions are different. Here the same characteristic length was applied for the heat transfer model due to the same mesh size, which indicated that model error should be similar and mesh error had greater impact on the simulation. The averaged values losses the local information obviously; therefore, even 3D fine-mesh result may not be reliable without choosing the "right" mesh size.

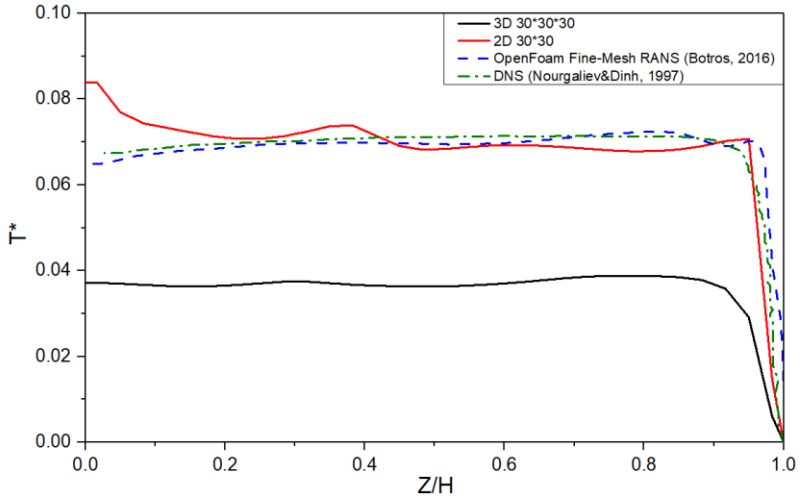


Figure 13. Time-Averaged Non-dimensional Temperature Profiles with Same Mesh Size using 2D and 3D GOTHIC Models

3.3.2. Mixed convection with Hot Air Injection

A mixed convection case with hot air injection on bottom of one side wall and a vent on the other side wall was simulated using a GOTHIC 2D model to investigate the mesh size and model sensitivity in GOTHIC modeling and simulation, as shown in Figure 14. There is no volumetric heat source in this case. Three different mesh sizes (10*10, 20*20, 30*30) and four different forced convection heat transfer models were used and temperature distributions in vertical centerline were compared. Table 1 listed these four heat transfer models applied. Standard k–ε turbulence model was applied in this case.

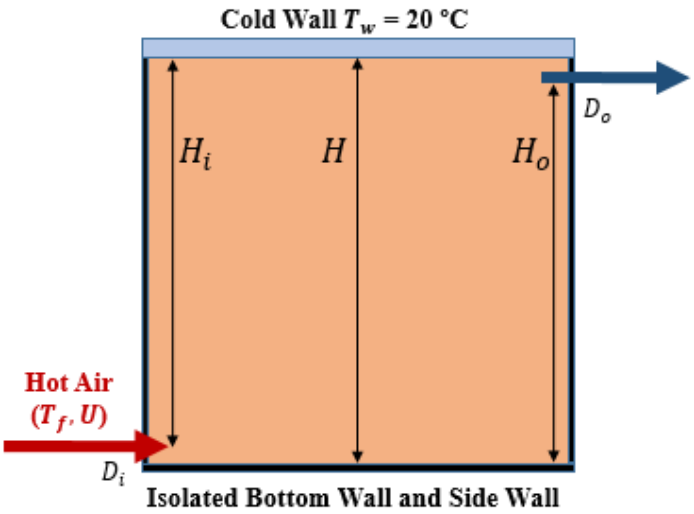


Figure 14. The Illustration of GOTHIC Model for Mixed Convection

Table 2. Different Heat Transfer Correlations applied in Forced Convection Case

Model NO.	Nu	Re	For GOTHIC			Experimentation
			B	C	D	
1	$Nu = 0.023Re^{0.8}Pr^{0.3}$	10^3 - 10^5	0.023	0.8	0.3	Cooling inside tubes
2	$Nu = 0.683Re^{0.466}Pr^{0.333}$	10^1 - 10^4	0.683	0.466	1/3	Heating and cooling outside tubes
3	$Nu = 0.246Re^{0.588}Pr^{0.333}$	10^3 - 10^5	0.246	0.588	1/3	
4	$Nu = 0.228Re^{0.731}Pr^{0.333}$	10^3 - 10^4	0.228	0.731	1/3	Heating or cooling over vertical plate

The simulation on mixed convection using GOTHIC was greatly sensitive to the selections of mesh size and heat transfer model. Figure 15 showed the temperature distributions in the centerline with different mesh sizes and same heat transfer model. The temperature distribution with 20*20 nodes, not the finer one (30*30), was much closer to the High-Fidelity (HF) result with fine mesh size. The results did not show the significant convergence as the fine meshes were applied. Figure 16 shows the temperature distributions in the centerline with different models and same mesh size, compared with the HF data. It proved that the heat transfer in the top layer determined the heat removal and the temperature distribution. The simulation was also very sensitive to the model selection.

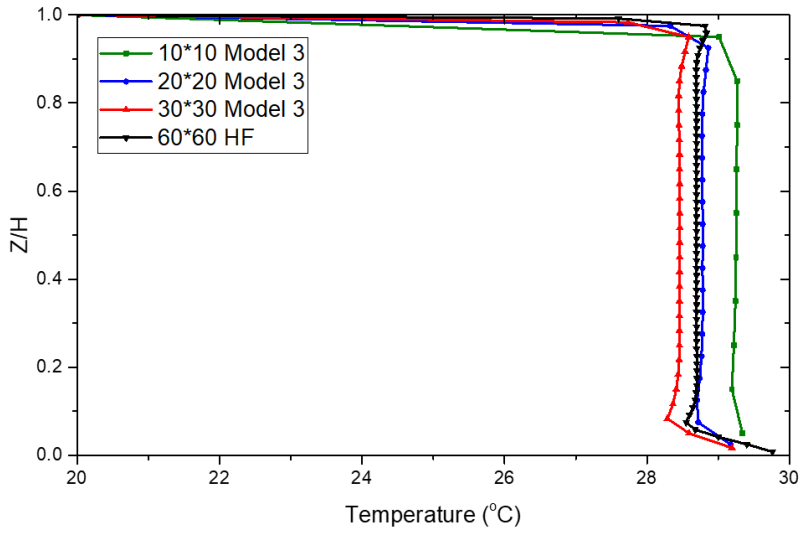


Figure 15. Comparison of Temperature Distribution with Different Mesh Sizes and Same Heat Transfer Model

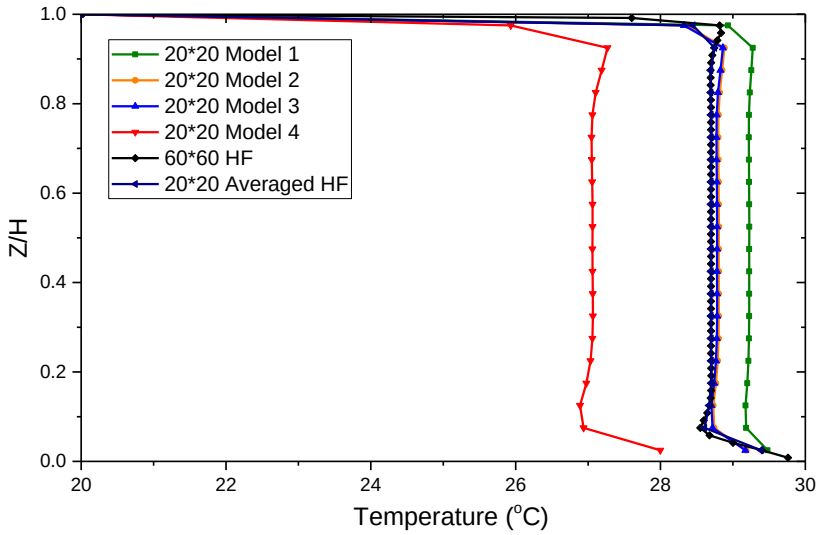


Figure 16. Comparison of Temperature Distribution with Different Heat Transfer Models and Same Mesh Size

3.4. Chapter Summary

This chapter reviews the technical basis of GOTHIC in system-level thermal-hydraulic modeling and simulation. The main error sources include model error due to physical simplification and mathematical approximation, mesh error due to the information loss in applying averaging methods on balance/constitutive equations, and other numerical errors. All these error sources lead to not only the error for single phenomena simulation, but also the error propagation then further to the simulation error for system scenario simulation. For the thermal-hydraulic simulations using GOTHIC, the mesh size greatly affects the performance of the empirical correlations in the local near-wall cells since mesh size is treated as one of the model parameters that determine whether the correlations are applied in their applicable ranges or not. Meanwhile, the local instantaneous PDEs for mass, momentum and energy are time and space averaged to obtain the integral of finite volume equations. Simulation results from GOTHIC represent the averaged values of parameters over specified regions, which ignores the local gradient information. Mesh error indicates the information loss of conservative and constitutive equations during the application of time and space averaging approaches. The tight connection between these two main error sources and mesh size makes it difficult to perform traditional Verification and Validation (V&V) on these codes to analyze the model error and mesh error separately. The mesh

and mode sensitivity in the qualitative assessment proves that the mesh convergence does not apply for GOTHIC, the selection of mesh and model greatly affects the simulation performance.

CHAPTER 4. MACHINE LEARNING ALGORITHMS

4.1. Introduction

This chapter reviews the Machine Learning (ML) algorithms applied in fluid dynamics, especially the one-layer Forward Neural Network (FNN) and Deep Neural Network (DNN) which are applied in this work.

Machine Learning (ML) teaches computers to do what comes naturally to humans and animals: learn from experience. ML algorithms use computational methods to "learn" information directly from data without assuming a predetermined equation as a model. These algorithms adaptively improve their performance as the number of samples available for learning increases. The goal is to find natural patterns in data that generate insight and help make better decisions and predictions. There are two types of ML techniques: supervised learning that trains a model on known input and output data so that it can predict future outputs, and unsupervised learning that finds hidden patterns or intrinsic structures in input data. The selection of an appropriate ML algorithm always puzzles users because there are dozens of supervised and unsupervised algorithms with different approaches to learning, and there is no “best” algorithm that can fits all problems.

In order to meet the prediction needs, the application of supervised ML algorithms are more popular to train a model to generate reasonable predictions for the response to new data. Supervised learning uses classification and regression techniques to develop predictive models. Classification techniques normally work for the data that can be separated into specific groups or classes and predict discrete responses, while regression techniques mainly predict continuous responses. Neural Networks (NNs) [43,44], Gaussian Process Regression (GPR) [45], Random Forests (RFs) [46], Gene Expression Programming (GEP) [47] used in the data-driven modeling for fluid dynamics mentioned in last chapter all belong to supervised learning approaches. The main considerations in choosing the supervised learning method are the dimensionality of input, the quantity of training datasets, and the capability of prediction with quantified uncertainty, training speed and memory usage. The first three terms are mostly considered for prediction purpose. GPR is good at dealing with small datasets and low dimensionality, for high dimensionality problems, the Principal component analysis (PCA) method can be used for dimensionality reduction. Besides, GPR also has the capability for prediction on new data with quantified uncertainty. NNs

work well for high dimensionality problems with large datasets while little knowledge about the underlying process or suitable physical features exist. FNN can be very efficiently done on GPU platforms that accelerate the learning process quite a lot. FNN also has the capability of deep learning in which low-level features can be combined and transformed into high-level features. This capability allows them to learn meta-properties like symmetry or invariance more easily, however, a network with more hidden layers can raise the risk of overfitting the training data. [48] After evaluating these existing supervised learning methods, a Deep NN (DNN), which is a multi-layer FNN, is identified as the currently efficient ML algorithm for OMIS approach.

4.2. Feedforward Neural Network (FNN)

A Feedforward Neural Network (FNN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems in parallel. As in nature, the connections between elements largely determine the network function. You can train a neural network to perform a particular function by adjusting the values of the connections (weights) between elements. Typically, neural networks are adjusted and trained so that a particular input leads to a specific target output, as illustrated in Figure 17. Here, the network is adjusted, based on a comparison of the output and the target, until the network output matches the target. Normally, many such input/target pairs are separately required for the training and testing of a network. [49] A FNN is an artificial neural network wherein connections between the units do not form a cycle.

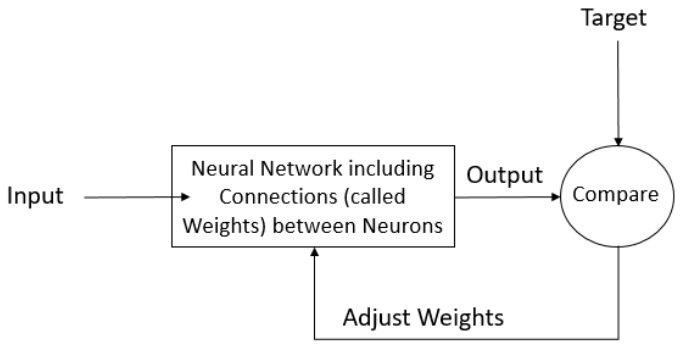


Figure 17. Schematic of How to train Neuron Networks [49]

Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an "expert" in the category of information it has been given to analyses. This expert can then be used to provide projections given new situations of interest and answer "what if" questions. Neural networks have been trained to perform complex functions in various fields, including pattern recognition, identification, classification, speech, vision, and control systems.

I. Neuron Model

A neuron with a single R-element input vector p ($\vec{P}_{R*1}$) is shown below in Figure 18. These individual element inputs are multiplied by weights W ($\vec{\omega}_{1*\mathrm{R}}$) respectively, and the weighted values are fed to the summing junction. Their sum is simply Wp, the dot product of the (single row) matrix W and the vector p. The neuron has a bias b which is summed with the weighted inputs to form the net input n. This sum, n, is the argument of the transfer function f, a is the output of neuron.

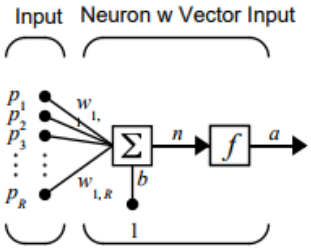


Figure 18. Illustration of Neural Model with with R Inputs [49]

$$a = f(n)\ and\ n = \boldsymbol{W} * \boldsymbol{p} + b \qquad (39)$$

II. Network Architecture

Two or more neurons can be combined in one layer to obtain one output. A network could contain one or more such layers. Here we firstly discuss single layer of neurons, as shown in Figure 19, a one-layer network with R input elements and S neurons. In this network, each element of the input vector **p** ($\vec{P}_{R*1}$) is connected to each neuron input through the weight matrix **W** ($\vec{\omega}_{S*R}$). The i^{th} neuron has a summation that gathers its weighted inputs and bias to form its own scalar output n(i). The various n(i) taken together form an S-element net input vector n. Finally, the neuron layer outputs form a column vector **a** ($\vec{a}_{S*1}$).

$$\boldsymbol{a} = \boldsymbol{f}(\boldsymbol{W} * \boldsymbol{p} + \boldsymbol{b}) \quad (40)$$

And the predicted scalar output, y, by the entire FNN can be expressed as,

$$y = \boldsymbol{\theta a} + \varepsilon \quad (41)$$

Where $\boldsymbol{\theta_{1*S}}$ is the weights and ε is the bias for a.

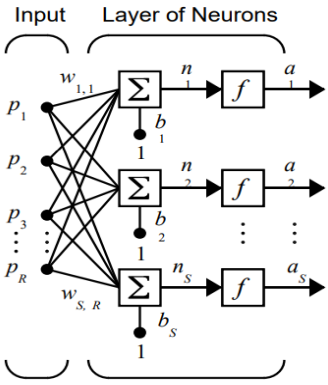


Figure 19. Illustration of a One-layer Network [49]

III. Fit Data with a Neural Network

The application of NNs for this proposed approach is to fit the function between simulation error and several inputs. The workflow contains several steps as: (1) collect and process data, (2) create the network, (3) configure the network, (4) initialize the weights and biases, (5) train the network, (6) validate the network, (7) use the network for test or prediction. There are many algorithms to adjust the weights and biases. Levenberg-Marquardt method [50] is recommended for most problems, but for some noisy and small problems Bayesian Regularization [51] can take longer but obtain a better solution. For large problems, however, Scaled Conjugate Gradient method [52] is recommended as it uses gradient calculations which are more memory efficient than the Jacobian calculations the other two algorithms use. The evaluation metric of the function fitting is the Mean Squared Error (MSE) at each test point,

$$mse = \frac{1}{n}\sum_{k=1}^{n}(T_k - Y_k)^2 \quad (42)$$

Where, T_k is the target output data considered as HF data while Y_k is the predicted output from the fit function by FNN, n is the number of the samples. Overfitting is an important issue that the model fits the training dataset well but fails in prediction. In order to avoid the possibility of

overfitting data, the HF data is divided into two sets as the training dataset and test dataset. The model is trained only using the training data and evaluated by the test data. It is also worth to note the number of coefficients that are calibrated during the FNN training. For R inputs and S neurons, the number of adjusted coefficients is S*(R+2)+1, where S*(R+1) for weights and S+1 for biases. More neurons means more coefficients needed to be calibrated. Overfitting may occur when the number of coefficients is greatly larger than the number of data points.

4.3. Deep Neural Network (DNN)

Deep learning or so-called Deep Neural Network (DNN) essentially refers to multilayer neural networks with more than two Hidden Layers (HLs). In DNNs, there are multiple layers of nodes, with the outputs of one layer becoming the inputs to the next layer. By using multiple layers of transformations, deep neural networks are able to capture complex, hierarchical interactions between features.

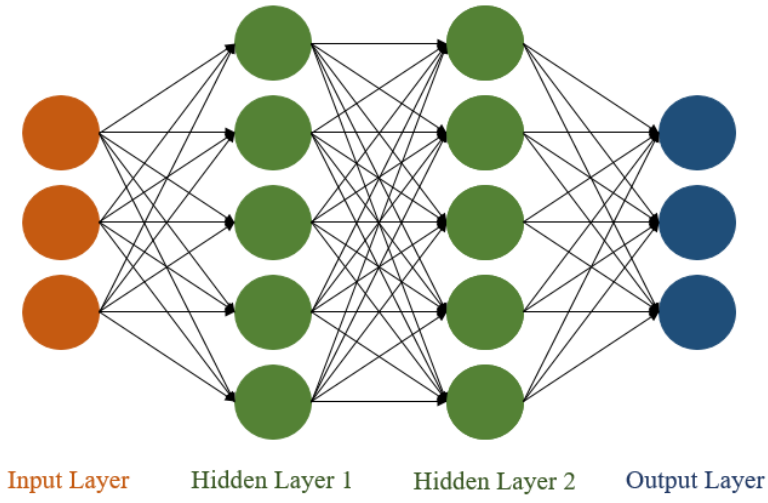


Figure 20. Schematic of a Three-layer Network with R Input Elements

Figure 20 illustrated the schematic of a typical three-layer feedforward network with one input layer, two HLs, and one output layer. The information flow is straightforward from input to output, so it is called feedforward network. For both neural network architectures, there were three main hyper-parameters: the number of hidden layers, the number of nodes per hidden layer and the learning rate in the gradient descent training algorithm. All three of these parameters can have a significant effect on model performance. Larger networks (with more hidden layers and more nodes per layer) can fit more complex data, but are also more prone to overfitting the data.

4.4. Chapter Summary

In this chapter, the Machine Learning (ML) algorithms applied in fluid dynamics have been reviewed. Compared to other ML algorithms, the Feedforward Neural Network (FNN) works well for high dimensionality problems with large datasets while little knowledge about the underlying process or suitable physical features exist. FNN also has the capability of deep learning in which low-level features can be combined and transformed into high-level features. This capability allows it to learn meta-properties like symmetry or invariance more easily. After evaluating these existing supervised learning methods, the multi-layer Deep FNN (DNN), is identified as the currently efficient ML algorithm for OMIS approach.

There are three main hyper-parameters: the number of hidden layers, the number of nodes per hidden layer and the learning rate in the gradient descent training algorithm. All three of these parameters can have a significant effect on model performance. Therefore, several DNN structures with different hidden layers and neuron numbers can be constructed as the potential ML method for data training and prediction. Larger networks (with more hidden layers and more neurons per layer) can fit more complex data but are more prone to overfitting the data and more computationally expensive.

CHAPTER 5. METHODOLOGY OF THE PROPOSED DATA-DRIVEN FRAMEWORK

5.1. Introduction

This chapter describes the methodology of the proposed OMIS framework. Mathematical basis, practical consideration, basic assumptions and hypotheses are introduced in Section 5.2 and 5.3. Each step of the framework is explicitly illustrated in Section 5.4 with the applied methods, algorithms and equations.

5.2. Mathematical Basis and Practical Consideration

Consider a physical system that is governed by a set of non-linear equations, the physical system can be simulated using a coarse-mesh CFD-like code as,

$$F_{LF}\left(\overrightarrow{V_{LF}}(\vec{x},t),\lambda_{LF},\delta_{LF}\right)=0 \tag{43}$$

where F_{LF} is the set of governing equations and constitutive equations as a LF model, $\overrightarrow{V_{LF}}$, λ_{LF} and δ_{LF} represent the model variables, the model information (model forms and relative parameters), and the coarse mesh size used in the LF simulation. Simulation error (ε) including model error, mesh error and other numerical errors exists, even if the best possible set of parameters, models and mesh sizes have been inferred. Given the true solution as $\overrightarrow{R_T}$ for the same physical condition, then the output quantities of interest can be expressed as,

$$\overrightarrow{R_T}=\overrightarrow{R_{LF}}\left(\overrightarrow{V_{LF}},\lambda_{LF},\delta_{LF}\right)+\varepsilon+\epsilon \tag{44}$$

where $\overrightarrow{R_{LF}}$ represents the output of the LF simulation. ϵ is the measurement error. Then we can find ε is expressed as below for a given physical condition.

$$\varepsilon=(\overrightarrow{\mathrm{R_T}}-\epsilon)-\overrightarrow{\mathrm{R_L}}\left(\overrightarrow{\mathrm{V_{LF}}},\lambda_{\mathrm{LF}},\delta_{\mathrm{LF}}\right) \tag{45}$$

Or in the following expression if the measurement error is considered to be negligible,

$$\varepsilon=\overrightarrow{R_T}-\overrightarrow{R_{LF}}\left(\overrightarrow{V_{LF}},\lambda_{LF},\delta_{LF}\right) \tag{46}$$

One can conclude that the LF simulation error ε is determined by physical condition represented by $\overrightarrow{R_T}$ and $\overrightarrow{V_{LF}}$, model information (model form and parameter, λ_{LF}) and the coarse

mesh size δ_{LF} used for the LF simulation. However, the relationship shown in Equation (46) is difficult to explore.

From the V&V point of view, the practical consideration of modeling and simulation is how to quantify the uncertainties and estimate the errors involved during the modeling and simulation process. As discussed in previous sections, the total simulation error (ε) for the physics of interest using these coarse-mesh CFD-like codes integrates the model error (ε_{model}), mesh error (ε_{mesh}) and other numerical error where the former two error sources have heavier weights on the total simulation error. Model error is somehow related to the model information for the physics of interest and local mesh sizes while mesh error is determined by the mesh size. Therefore, the relationship implied in Equation (46) is essentially the relationship between ε and ε_{model}, ε_{mesh}. The ideal way is to estimate these two error sources and find this relationship. However, these two error sources cannot be quantified separately due to their tight connections with the mesh size. By treating these two error sources together, the central idea of is to develop a surrogate model to identify the relationship between ε and specific local Physical Features (PFs), as shown in Figure 21. The identification of PFs integrates the physical information of the system of interest, model information and the effect of mesh size.

Once the error function $\varepsilon = f(PFs)$ is developed and evaluated based on existing data and the application of Ml algorithms, the simulation error for new condition with the specific mesh and model is predictable. The mesh size and model with least simulation error are identified as the optimal mesh size and model for the specific physical system, which means that, they are the "best" choice for the simulation for this condition.

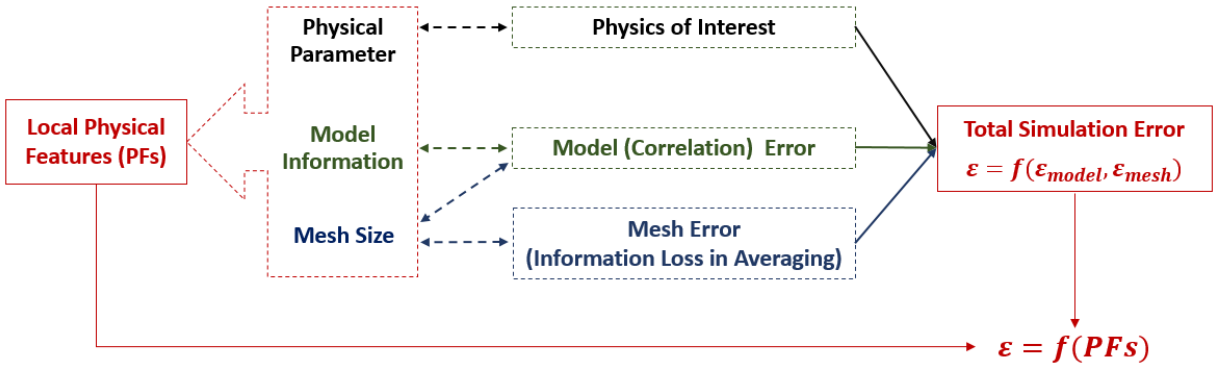


Figure 21. Central Idea of OMIS: Local Data Training for Error Estimation

5.3. Basic Assumptions and Hypotheses

The basic assumptions for the development and application of OMIS framework are described:

1. Technical tools:
    - Length scale of the physics of interest is large enough to be captured by coarse-mesh modeling and simulation.
    - The selected coarse-mesh LF computational tool is able to capture the basic physical behaviors of the system of interest, even with a large uncertainty.
    - The simulation error using coarse-mesh LF computational tool is mainly impacted by model error and mesh error. They cannot be quantified separately since mesh size is one of key model parameters that makes them tightly connected.
    - Always an appropriate ML algorithm has the basic capability to explore the local patterns and lost information in LF simulations.
2. Data:
    - Training data is qualified and sufficient for Machine Learning (ML) algorithm to learn from and find the intrinsic knowledge of the physics.
    - A group of local PFs is able to represent the characteristics of local physics of interest.
    - The generation, process and classification of training and testing data ensure that they have the similar physical meanings.

The hypotheses of the framework that need to be evaluated are:

1. The simulation error can be represented as a function of key Physical Features (PFs) which integrate the information from local physics, applied models and local mesh sizes.

2. "$\varepsilon = f(PF)$" is not a fixed correlation, it just represents the relationship between simulation errors and physical features, which is improvable and compatible when new qualified data or physical conditions are added into training data.

3. The similarity of training data and testing data determines the predictive capability of trained ML algorithms on testing case.

5.4. Framework Formulation

This data-driven mesh-model optimization framework contains six independent steps as displayed in Figure 22. Although these steps are integrated as a modular manner, the specific assumptions, algorithms and methods applied in each step are flexible for different purposes and will not affect the execution of other steps. In order to achieve the ultimate goal: providing suggestions on the optimal mesh/model selection and error prediction on the global Quantities of Interest (QoIs), each step has its own tasks:

- How to analyze the system and specify the QoIs of simulation target? (Step 1)
- How to decompose the complex physics and identify relevant physical models applied in LF code? (Step 1)
- How to identify and define potential PFs for the involved key physics? (Step 2)
- How to construct reasonable test matrix for the target simulation? (Step 2)
- Which factors should be considered in the selection of optimal PF group, ML tool and training database for target case? (Step 3, 4 and 5)
- How to determine the contribution of each potential PF on responses? (Step 3)
- How to evaluate the predictive capability of ML algorithms? (Step 4)
- How to measure the similarity of training data and the data in target case? (Step 5)
- How to estimate the error of global QoIs and suggest on the selection of optimal mesh/model? (Step 6)

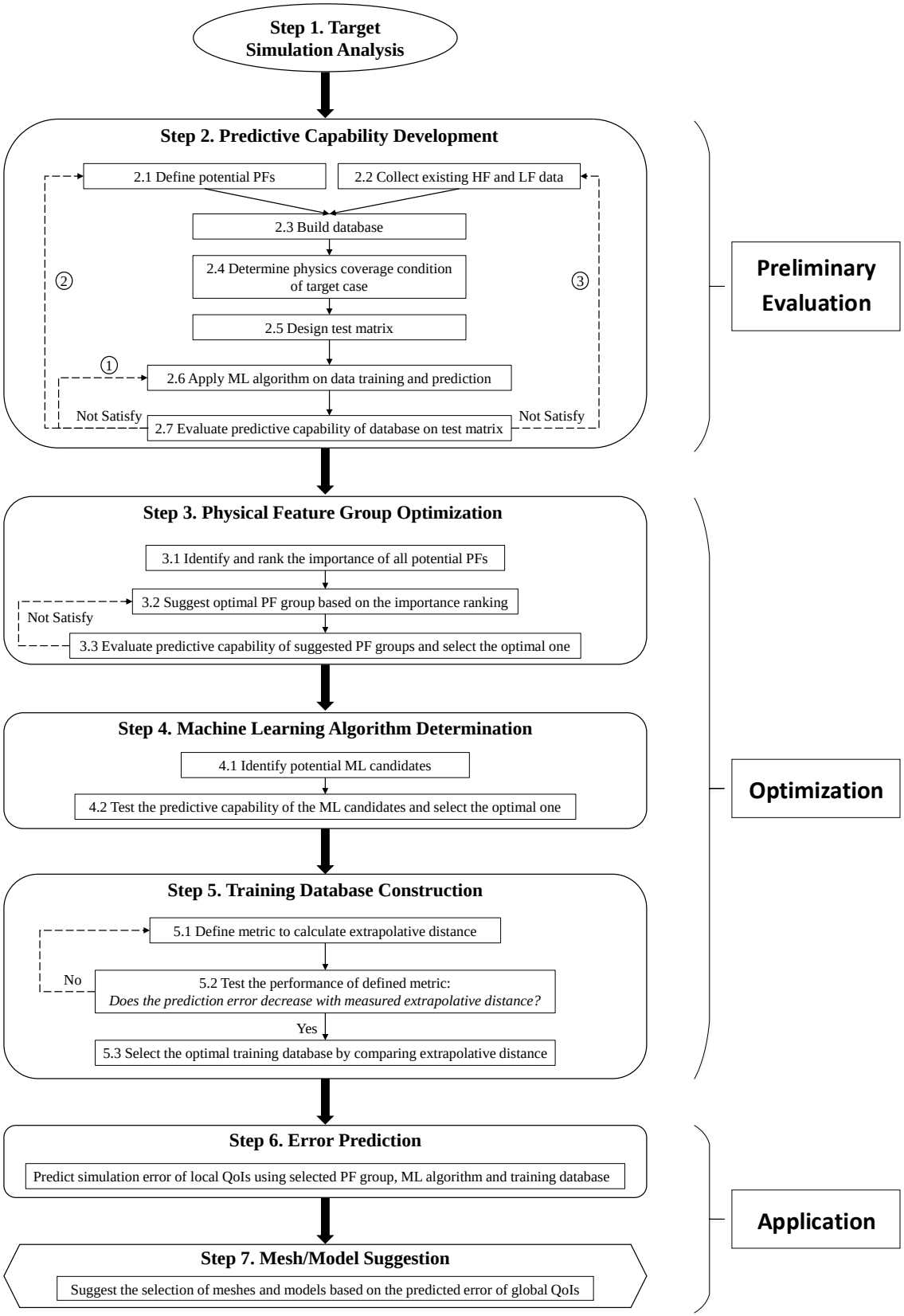


Figure 22. Diagram of the Optimal Mesh/Model Information System (OMIS) Framework

5.4.1. Step 1: Simulation Target Analysis

The first thing is to specify the key physics involved in the target simulation case. For different conditions like steady state or transient, coupled systems or single control volume, the dominating physics and relevant QoIs are also different. In the system-level thermal-hydraulic simulation of a NPP, the QoIs are normally influenced by a couple of different phenomena. A PIRT (Phenomenon Identification and Ranking Table) procedure should be executed to decompose the complex physics and identify the key phenomena. For example, a forced convection simulation using GOTHIC may imply an interaction of different physical models respectively for turbulence, wall friction and forced convection heat transfer. The respective closure models in the simulation tool should be pre-evaluated whether they are used in the applicable ranges or proper conditions.

The QoIs in system-level thermal-hydraulic simulations are normally global parameters and depending on the phenomena in the given scenario. These parameters represent the system behaviors of NPPs and provide information for the decision-makings of operators. For example, in the normal operating of Boiling Water Reactors (BWRs), the main steam line temperature and reactor vessel pressure are considered as QoIs since they indicates the performance of the heat removal of fission in the fuel bundles. The temperature/pressure in the wetwell and the hydrogen fraction in the drywell can be considered as the key QoIs if the severe events happen as in the Fukushima accident. The values of these QoIs could help the operators and decision makers estimate the benefit and risk when to inject seawater into the units. Therefore, the estimation of these key QoIs is the criteria of which mesh or model is the optimal one in the OMIS framework.

Besides, some global parameters should identified to represent the global physical condition of target case. This helps the selection of training database and also the construction of test matrix. For example, in a pipe flow, Re number should be identified as the key global physical parameter. Then the HF data, which belongs to pipe flow and has similar values of Re number should be selected into the potential training database in Step 2. According to Re number values in target case and training case, it should be addressed which “zone” the target belongs to.

Lastly, according to the geometry and structure of the control volumes in the target case, a set of potential mesh sizes should be selected for different control volumes. Based on the capability of simulation tool, these mesh sizes should be in an appropriate range in which they are neither

too fine to cost much computation nor too coarse to lose much local information. Overall, the items that should be specified in this step are (1) Key phenomena and global QoIs in the target case; (2) Applicable physical models for these key phenomena in the simulation tool; (3) Global parameters that represents the global physical condition of target case; (4) Potential mesh sizes for specific control volumes.

5.4.2. Step 2: Predictive Capability Development

The central idea has been discussed in Section 5.2, which aims to identify the relationship between simulation error and local PFs. This step is proposed to establish the predictive capability by preliminarily defining potential PFs, building database, and evaluating whether the selected database has the predictive capability on the test matrix. The procedure of Step 2 is summarized in Figure 23.

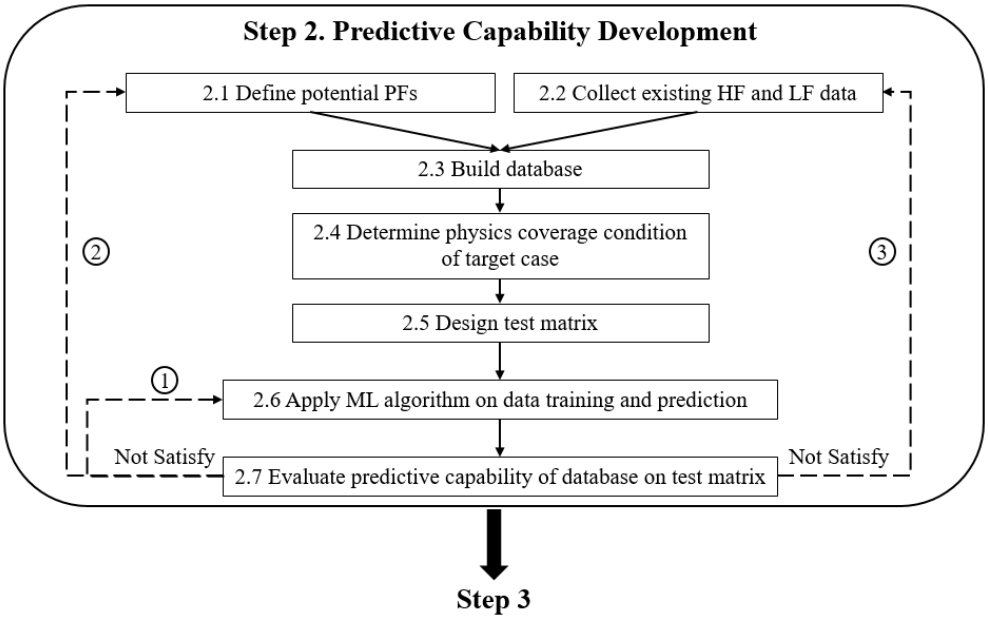


Figure 23. Diagram of Step 2: Procedure of Predictive Capability Development

- **Step 2.1: Define potential physical features**

The identification of PFs is guided by the physics decomposition and model evaluation executed in Step 1. In order to take physics scalability and regional information into consideration, the PF group should include the gradients of local variables and the local physical parameters that are able to represent the local physical behaviors or applied in crucial closure relationships. This ensures that the physical information of the physical system, model information applied and the

effect of mesh size are integrated and well represented in the PF group. As illustrated in Figure 24, the first part of local PFs are the gradients of variables including 1-order and 2-order derivatives of variables calculated using central-difference formulas. All potential PFs that satisfying the definition should be considered and included in the initial selection of PF group.

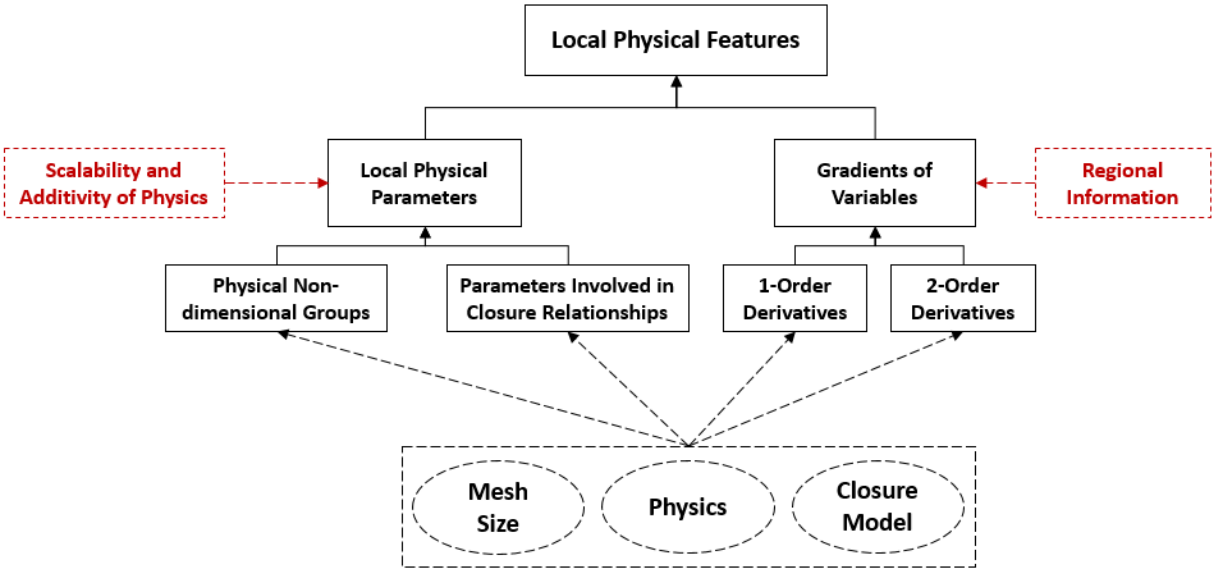


Figure 24. Identification and classification of Physical Feature

$$\left.\frac{\partial V}{\partial x_i}\right|_{(i,j)} = \frac{V_{i+1,j} - V_{i-1,j}}{2\Delta x_i} \tag{47}$$

$$\left.\frac{\partial^2 V}{\partial {x_i}^2}\right|_{(i,j)} = \frac{\left.\frac{\partial V}{\partial x_i}\right|_{(i+1,j)} - \left.\frac{\partial V}{\partial x_i}\right|_{(i-1,j)}}{2\Delta x_i} = \frac{V_{i+2,j} - 2V_{i,j} + V_{i-2,j}}{4(\Delta x_i)^2} \tag{48}$$

$$\left.\frac{\partial^2 V}{\partial x_j \partial x_i}\right|_{(i,j)} = \left.\frac{\partial^2 V}{\partial x_i \partial x_j}\right|_{(i,j)} = \frac{\left.\frac{\partial V}{\partial x_i}\right|_{(i,j+1)} - \left.\frac{\partial V}{\partial x_i}\right|_{(i,j-1)}}{2\Delta x_j} = \frac{V_{i+2,j+2} - V_{i-1,j+1} - V_{i+1,j-1} + V_{i-2,j-2}}{4\Delta x_i \Delta x_j} \tag{49}$$

The variable value in each cell is the averaged value. For the boundary cells,

$$\left.\frac{\partial V}{\partial x_i}\right|_{(1,j)} = \frac{V_{2,j} - V_{0,j}}{\frac{3}{2}\Delta x_i} \tag{50}$$

$$\left.\frac{\partial^2 V}{\partial {x_i}^2}\right|_{(1,j)} = \frac{\left.\frac{\partial V}{\partial x_i}\right|_{(2,j)} - \left.\frac{\partial V}{\partial x_i}\right|_{(0,j)}}{\frac{3}{2}\Delta x_i} = \frac{\frac{V_{3,j}-V_{1,j}}{2\Delta x_i} - \frac{V_{1,j}-V_{0,j}}{\frac{1}{2}\Delta x_i}}{\frac{3}{2}\Delta x_i} = \frac{V_{3,j}-5V_{1,j}+4V_{0,j}}{3(\Delta x_i)^2} \quad (51)$$

$$\left.\frac{\partial^2 V}{\partial x_i \partial x_j}\right|_{(1,j)} = \left.\frac{\partial^2 V}{\partial x_j \partial x_i}\right|_{(1,j)} = \frac{\left.\frac{\partial V}{\partial x_j}\right|_{(2,j)} - \left.\frac{\partial V}{\partial x_j}\right|_{(0,j)}}{\frac{3}{2}\Delta x_j} = \frac{V_{2,j+1}-V_{2,j-1}-V_{0,j+1}+V_{0,j-1}}{3\Delta x_i \Delta x_j} \quad (52)$$

The gradients of local variables imply the regional (or local surrounding) information that represents the regional physical patterns. As displayed in Figure 25, the regional information obtained from the training dataset (as Case A) can be used to teach and inform the prediction of new conditions (as Case B) in GELI condition: if the regional information in blue part of Case A is similar to the blue part of Case B. More regional information may be involved if higher order derivatives are added into the local PF group.

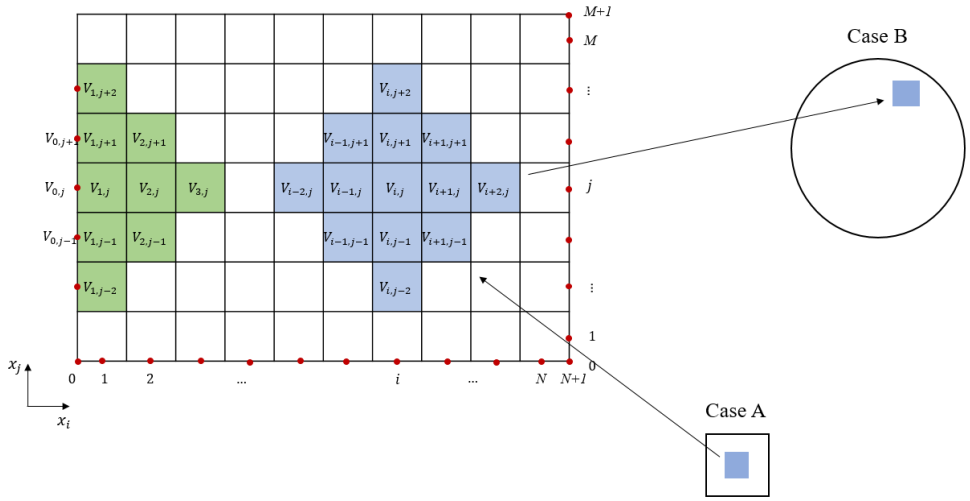


Figure 25. Illustration of How Regional Information is represented by Gradients of Variables in 2D "GELI" Problems

Second part of PF group is the local parameters that are defined to represent the local physical behaviors or applied in closure relationships. These parameters representing the local physical behaviors are supposed to provide the scalability of physics. This idea came from the early scaling, both of global approach and local approach were applied to develop non-dimensional groups based on facility dimensions and fluid conditions between full-scale facility and scaled test. The global approach was based on identifying different dimensions and fluid parameters to develop non-dimensional groups using the Buckingham Pi theorem, but these groups may not have

any physical meaning. The local approach is to non-dimensionalize the PDEs of conservation equations with reference values to obtain a set of local non-dimensional groups, such as Reynolds number, Froude number. [53] These groups do have physical meanings. Another part of local parameters as PFs is the parameters used or involved in the crucial local closure correlations for boundary layer. These parameters contain much information of length scale, model parameter and wall distance.

It should be noted that these local parameters enable PF group the scalability and additivity of physics. Scalability of physics indicates that if the PF data of existing case and target case is similar, the local physical information of target case should be covered by existing case, even if these cases are in the different length scales. Meanwhile, the PFs identified for simple single phenomenon is still usable for complex coupled physics. PF group is improvable by adding more relevant PFs if new phenomena are involved.

- **Step 2.2: Collect existing high-fidelity and low-fidelity data**

First is to collect available HF data, which is relative to the involved physics in target case. HF data includes regional data from experimental observation, DNS data, and validated high-resolution numerical results. Since the HF is normally in a limited quantity, the requirement of "HF" is flexible and determined by accuracy of expectation on target simulation. For example, if LF simulation of a NPP containment is executed by coarse-mesh GOTHIC modeling and expected to achieve the accuracy comparable to fine-mesh RANS simulation, then the results from STAR CCM+ using RANS models can be considered as HF data in this case. According to the physical conditions of limited HF data, LF data is generated using fast-running LF code with the candidates of mesh sizes and physical models.

- **Step 2.3: Build database**

Database includes PF group as input and errors of local FOMs as output. The data of PF group is calculated using LF simulation data as discussed in Step 2.1. The method applied for error calculation should be formulated. Normally, there are two methods to calculate the error between fine-mesh HF data and coarse-mesh LF data: point-to-point method and cell-to-cell method. The first one is to compare the values of FOMs at the exact locations existing in both of HF and LF data, this method can be applied if both HF and LF simulations are using Finite Element Method (FEM) or Finite Difference Method (FDM), as shown in Figure 26 (a). The second one is to

compare the values of FOMs in the coarse-mesh cell by mapping and averaging the values of FOMs in fine-mesh cells, as shown in Figure 26 (b). Here the cell-to-cell method is applied for error calculation. Errors of local FOMs in all cells should be calculated. Velocity is the main FOM for the adiabatic fluid flows, temperature should also be added in if heat transfer is involved.

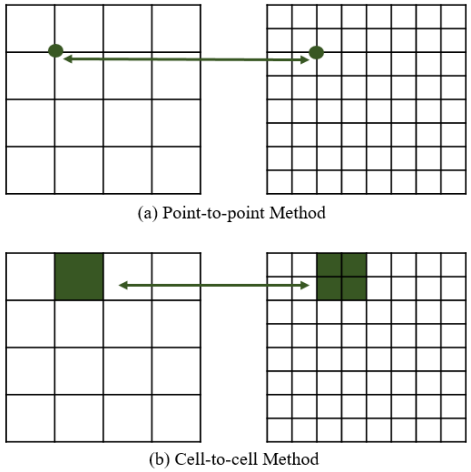


Figure 26. The Calculations of Error between Fine-Mesh Data and Coarse-Mesh Data

- **Step 2.4: Determine physics coverage condition of target case**

In the same way, PF data of target case should be calculated and compared with the collected data to determine which physics coverage condition the target case belongs to. OMIS framework is only applicable if the target case locates in GILI or GELI condition. Otherwise, the collected data is unusable to cover and inform the local physics of target case. Global parameters defined in Step 1 can be used to specify the global condition while the local condition is qualitatively identified by t-SNE (t-Distributed Stochastic Neighbor Embedding) method, which is a dimensionality reduction technique for the visualization of high-dimensional datasets. [54]

- **Step 2.5: Design test matrix**

Once the physics coverage condition of target case is determined, test matrix should be designed to investigate whether the PF group in collected database has the expected predictive capability for the determined physics coverage condition. Here extrapolative distance is defined to determine the coverage of collected data on the target data. The method applied on the calculation of extrapolative distance is described in Step 5 since it is also used to guide the construction of optimal training database for target case. The physical condition with similar extrapolative distance

should be designed for testing. According to the test matrix, the collected database is divided into training data and testing data.

- **Step 2.6: Apply machine learning algorithm on data training and prediction**

In this step, ML algorithm is applied to train the error database and obtain the regression function whose input are PFs and output are the errors of FOMs. As one of supervised learning methods, FNN works well for high dimensionality problems with large datasets while little knowledge about the underlying process or suitable physical features exist. FNN can be very efficiently done on GPU platforms that accelerate the learning process quite a lot and also has the capability of deep learning in which low-level features can be combined and transformed into high-level features. This capability allows it to learn meta-properties like symmetry or invariance more easily, however, a network with more hidden layers can raise the risk of overfitting the training data. [48] After evaluating these existing supervised learning methods, multi-layer FNN is identified as the currently efficient ML algorithm for OMIS application. After trained by training data, the FNN with adjusted hyper-parameters is able to give error prediction on the testing case.

- **Step 2.7: Evaluate predictive capability on test matrix**

In this step, the original values of FOMs from LF simulations are modified by predicted errors from previous step. Then modified FOMs are compared with the FOMs in testing HF data. The prediction uncertainty mainly comes from the identification of PF group, data quality and quantity, and ML algorithm itself. This step is just a preliminary evaluation and following steps are trying to reduce the uncertainty. Here Mean Squared Error (MSE) is used to quantitatively evaluate the predictive capability, which is defined as below,

$$MSE_{prediction} = \frac{1}{n}\sum\left(QoI_{HF,i} - QoI_{predicted,i}\right)^2 \quad (53)$$

The modified FOMs are also compared with the original LF results to investigate how much improvement is obtained. Once the comparison with HF data satisfies the expected accuracy, Step 2 is finished. Otherwise, there are three ways to improve the predictive capability, which are denoted as dash lines in Figure 23: (1) improving FNN structure, (2) defining new PFs and (3) collecting more data. Considering their workloads, these three amendments should be performed in the notated order.

The information flow of Step 2 is described in Figure 27. After identifying PFs and building database, training flow and testing flow are divided based on test matrix. In training flow, LF simulations with different mesh sizes and models are performed and then compared with HF data. The local errors (ε_i) of variables between mapped HF data ($V_{HF,i}$) and LF simulations ($V_{LF,i}$) are calculated and collected to obtain the error training database. The PF values (PF_i) of training flow are obtained using LF simulation results. The regression error function $\varepsilon = f(PF)$ is obtained based on the training database using multi-layer FNN. Then by inserting the new PF values (PF_j) of testing flow into the error function, the respective errors (ε_j) can be predicted to modify the LF simulation results ($V_{LF,j}$). The modified variable ($V_{m,j}$) are compared with the ones from HF data ($V_{HF,j}$). The predictive capability is tested via validation metric (MSE) to check whether the prediction satisfies the expected accuracy. The determination of expected accuracy is based on the simulation purpose and limited knowledge on the true physics.

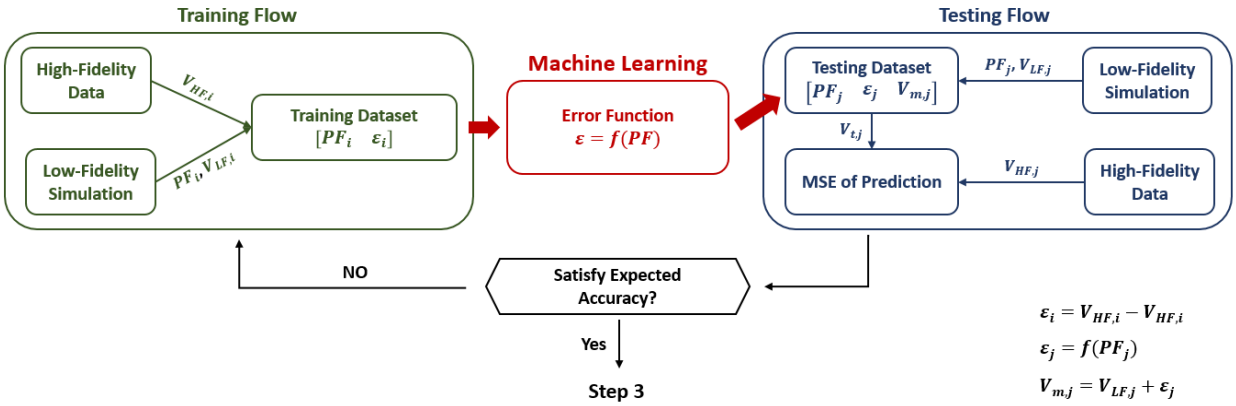


Figure 27. Schematic of OMIS Approach: Training Flow and Testing Flow

5.4.3. Step 3: Physical Feature Group Optimization

This step is trying to answer one question: which factors should be considered in the selection of optimal PF group? According to the definition and classification of PFs discussed in Step 2.1, there can be a number of potential PFs in multi-physics condition. These PFs have different impacts on the responses (errors of local FOMs). Since training a multi-layer FNN with a huge number of PFs is computationally expensive, it is necessary to identify and rank the importance of each potential PF and select optimal PF group with respect to both of PF importance ranking and computation cost for data training. The procedure of PF group optimization is summarized in Figure 28.

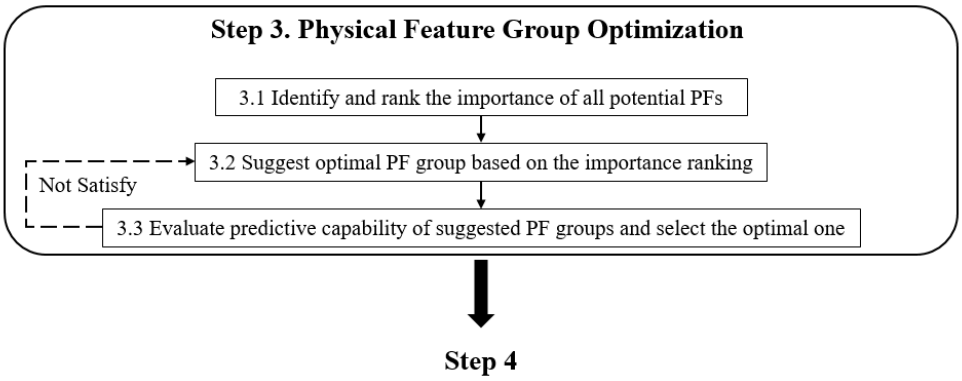


Figure 28. Diagram of Step 3: Procedure of PF Group Optimization

- **Step 3.1: Identify and rank the importance of all potential physical features**

In the past decades, researchers have put many efforts on variable importance analysis based on computational codes and measured data in almost all fields of engineering and science. Summarily, importance analysis aims to quantify [55]

1. The change of model output value with respect to the variation of input variables, or
2. The contribution of the uncertainties of input variables to the uncertainty of model output variable, or
3. The strength of dependence between the model output variable and input variables.

Currently, the popular importance analysis methods can be divided into two groups: mathematical techniques and statistical techniques. The mathematical techniques include the difference-based methods such as Morris' screening [56], variance-based methods [57] and momentum-based methods [58]. These methods are developed to measure the importance of input variables of models and most of them need to compute the model response function at prescribed or well-designed points. [59] This feature makes them not suitable for the situation where only data not model is available. It should be noted that the concept of sensitivity analysis mostly used for computational models is similar to the first two definitions of importance analysis. These sensitivity analysis methods mostly belong to the group of mathematical techniques. There are some rigorous requirements to apply these methods. For example, difference-based methods are based on the computation of partial derivatives of model output to input variables, they are not applicable for the models with non-smooth response functions. Variance-based methods require the input variables to be independent, the correlated effects between input variables are not

considered. In this work, only data not model is available to generate input variable (PFs) information, and the inherent correlations between these variables are the key mutual property. Therefore, these traditional sensitivity analysis methods or mathematical techniques are not suitable to identify the PF importance.

Another group, specified as statistical techniques, are designed to explore the variable importance based on data including parametric regression and non-parametric regression techniques. These methods are applicable for both computational model and pure data since the data of input variables can be generated by calling the response function or sampling from prepared database. Compared to parametric regression methods, non-parametric regression methods do not require a fixed regression model form or an uncorrelated relationship between the input variables. The relationship between PFs and errors of FOMs is highly non-linear and affected by the integration of several physical models, PF identification, data collection and numerical solvers. There are several popular non-parametric regression techniques such as Gaussian Process Regression [60] and Random Forest Regression (RF regression, or RFR) [46]. In this work, RFR is applied to quantify and rank the PF importance. As a supervised learning algorithm, RFR is much computationally efficient than multi-layer FNN. Compared to traditional statistical methods or other non-parametric regression methods, RFR do not need to assume any formal distributions for the data and can fast fit highly non-linear interactions even for large problems.

RFR is an ensemble learning technique by constructing a forest of uncorrelated regression trees at training time and outputting mean prediction from these individual trees. The training algorithm for random forests applies the general technique of bootstrap aggregating (or bagging). Given a training set $D_N = \{(X_{N\times F}, Y_N)\}$, bagging repeatedly (M times) selects a random subsample ($\Theta_m, m = 1,2, \dots M$) with replacement of the training set and fits trees to these subsamples. N is the number of data points, F is the number of input variables. M regression trees ($\{h(\Theta_m), m = 1,2, \dots M\}$) are trained and built for M times samplings, then provide M times of prediction ($p_{m(x_p)}$) for a new unseen sample (x_p). Then final prediction is made by averaging the M predictions:

$$P = \frac{1}{M}\sum_{m=1}^{M} p_{m(x_p)} \qquad (54)$$

This bootstrapping procedure leads to a good predictive performance since it de-correlates these regression trees and decreases the bias of ensemble prediction by providing different training datasets. Besides, the prediction uncertainty can be estimated as the standard deviation of the predictions from all the individual regression trees:

$$\sigma = \sqrt{\frac{1}{M-1}\sum_{m=1}^{M}\left(p_{m(x_p)} - P\right)^2} \qquad (55)$$

Normally, M is assigned as a few hundred to several thousand depending on the size of training dataset. Once the regression trees have been built, the importance of variables can be measured by observing the Out-Of-Bag (OOB) error, which is called Permutation Variable Importance Measure (PVIM) [46]. A set of OOB datasets can be generated as $B_m = D_N - \Theta_m$. The following process describes the estimation of variable importance values by PVIM. Suppose the OOB data can be expressed as $B_m = \{(y_j^m, x_j^m), m = 1,2,\dots M \text{ and } j = 1,2,\dots,S\}$, S is number of sample points.

1. For the *m*th tree, the prediction errors on the OOB data before and after randomly permuting the values of the input variable X_f $(f = 1,2,\dots,F)$ are calculated using,

$$MSE_m = \frac{1}{S}\sum_{j=1}^{S}\left(y_j^m - \hat{y}_j^m\right)^2 \text{ and } MSE_{m,f} = \frac{1}{S}\sum_{j=1}^{S}\left(y_j^m - \hat{y}_{j,f}^m\right)^2 \qquad (56)$$

where $\hat{y}_j^m$ and $\hat{y}_{j,f}^m$ are the prediction from the *m*th tree respectively before and after permutation.

2. The difference between two predictions are defined as the value of PVIM for the *m*th tree:

$$PVIM_{m,f} = MSE_{m,f} - MSE_m \qquad (57)$$

3. The overall PVIM of X_f in the OOB data is then calculated as,

$$PVIM_f = \frac{\frac{1}{M}\sum_{m=1}^{M} PVIM_{m,f}}{\sigma_f} \qquad (58)$$

where σ_f is the standard deviation of the differences over the total OOB data. The value of $PVIM_f$ indicates the OOB importance of X_f on the response. In this way, the OOB importance can be measured for each input variable. In the *m*th tree, if X_f is not selected as the splitting variable

then $PVIM_f = 0$. It implies that the interactions between X_f and other variables are considered to measure its contribution on the prediction accuracy. The importance of a variable increases with the value of PVIM.

- **Step 3.2: Suggest optimal physical feature group based on the importance ranking**

By performing importance analysis via RFR, the importance of each PF can be quantified and scored by the value of PVIM. Normally, the scores are in the range from 0 to 10. Based on the scores, the importance of PF is ranked in three levels: High, Middle and Low (H, M, and L). Different PF groups can be divided respectively including PFs in H level, H+M level and H+M+L level.

- **Step 3.3: Evaluate predictive capability of suggested physical feature group on test matrix**

After the importance identification and ranking, computational cost is saved in the data training for the PF groups only with H level or H+M level. However, uncertainty is also introduced due to reduction of PF dimensionality. The selected PF may be not sufficient to represent the underlying physics. Therefore, it is necessary to go back to Step 2.7 and re-test the predictive capability, as the dash line 1 shown in Figure 22. The selection of optimal PF group should well balance the accuracy and computation cost. If the predictive accuracy does not satisfy, the importance analysis should be executed by using better approaches. Two metrics should be considered here to finalize the optimal PF group:

1. MSE of prediction: whether the reduced PF group keeps the underlying physics;
2. Computational cost for data training: how much computation is saved using the reduced PF group.

5.4.4. Step 4: Machine Learning Algorithm Determination

After the optimization of PF group, the ML algorithm for data training and prediction is optimized and determined in this step. FNN is applied considering its deep-learning capability to explore the highly non-linear relationship between PFs and simulation errors. Therefore, the procedure of ML algorithm determination contains two parts: identify potential FNN candidates, test their predictive capability and select the optimal one with the consideration of accuracy and computation cost, as shown in Figure 29.

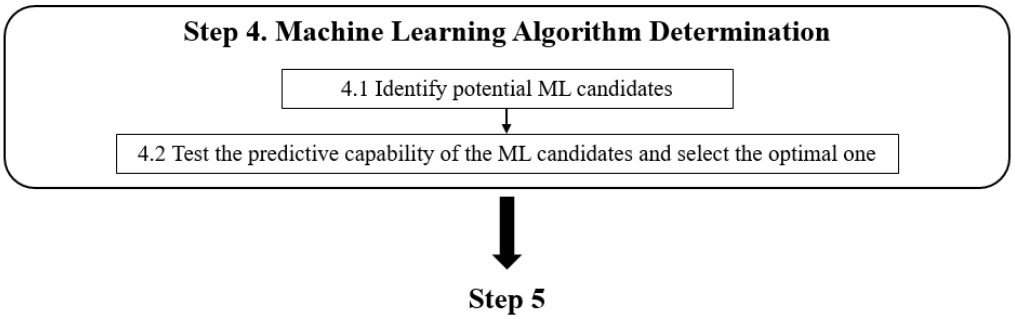


Figure 29. Diagram of Step 4: Procedure of ML Algorithm Determination

There are three main hyper-parameters: the number of hidden layers, the number of nodes per hidden layer and the learning rate in the gradient descent training algorithm. All three of these parameters can have a significant effect on model performance. Therefore, several FNN structures with different HLs and neuron numbers can be constructed as the potential ML method for following data training and prediction. Larger networks (with more hidden layers and more neurons per layer) can fit more complex data, but are more prone to overfitting the data and more computationally expensive. Therefore, these FNN candidates should be tested by the test matrix built in Step 2.7 and the optimal FNN structure can be selected based on the MSE of prediction and the computational cost for data training. The dash line 2 shown in Figure 22 represents the re-test of these potential FNN structures.

5.4.5. Step 5: Training Database Construction

This step focuses on how to select sufficient and necessary data for training and prediction. The training data is assumed abundant to "cover" the physics in the target case; however, some existing data may be not similar or even relevant to the target case. It is necessary to select the sufficient datasets as the final training database to avoid the huge computational cost on data training. Therefore, we need to answer a question: how to quantitatively measure the similarity of the data in the target case and training data? It is obvious that if target data is more covered or similar to the training data, the prediction error on the target case is smaller. In this step, a concept of extrapolative distance is defined to, (1) determine the coverage (or similarity) of training data on the target data and (2) guide the selection/generation of training data source. Then the performance of defined metric should be tested that whether the prediction error decrease with the measured extrapolative distance. Lastly, determine the optimal training database by comparing extrapolative distances between target data and the candidates of training database: the one with

smallest extrapolative distance is the optimal training database for final data training and prediction. The three-step procedure of training database construction is illustrated in Figure 30.

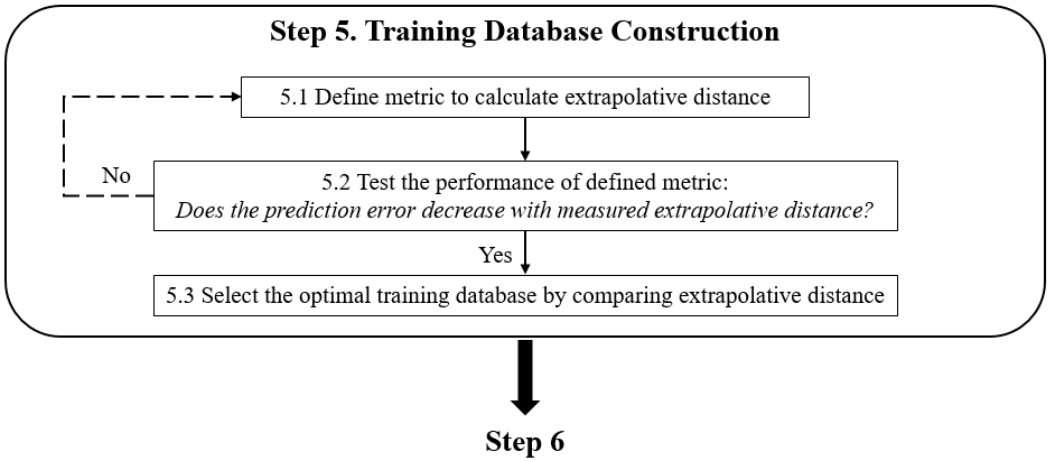


Figure 30. Diagram of Step 5: Procedure of Training Database Construction

- **Step 5.1: Define metric to calculate extrapolative distance**

The goal of extrapolative distance is to measure how far the target point (x_{ta}, y_{ta}) is from the training dataset $D_{tr} = \{(x_i, y_i), i = 1,2, \dots, tr\}$. Several approaches have been applied to quantify the distance. As a basic distance metric, Euclidean distance between the target point and the training dataset can be expressed as,

$$d_{Eu} = \frac{1}{tr}\sum_{i=1}^{tr}\sqrt{(x_{ta} - x_i)^2 + (y_{ta} - y_i)^2} \qquad (59)$$

Another similar metric, the nearest neighbor distance represents the Euclidean distance between the target point and its nearest point in the training dataset. These metrics based on Euclidean distance are easy to compute but very susceptible to noise and memory-consuming since all the points in training dataset are used. Besides, these metrics treat the training data as uncorrelated points and ignore their underlying interactions. There are some promising metrics which are designed memory-efficient by considering the distribution of the training dataset. Mahalanobis distance is defined as the distance between a point ($\boldsymbol{q}$) and the mean of training data ($\boldsymbol{\mu}$) with the covariance matrix ($\boldsymbol{\Sigma}$), which can be expressed as,

$$d_{Ma} = \sqrt{(\boldsymbol{q} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{q} - \boldsymbol{\mu})} \qquad (60)$$

Mahalanobis distance only considers the statistical parameters like mean and covariance instead of the entire raw data, this makes it more memory efficient. However, the drawback of Mahalanobis distance is its stringent assumption that the training data points yield a multivariate Gaussian distribution ($\boldsymbol{\mu}, \boldsymbol{\Sigma}$). This is a weak assumption to deal with the data from thermal hydraulic simulations, especially for turbulent flows where multi-mode distributions may be common. To overcome this problem, a method called Kernel Density Estimation (KDE) is introduced in this step. KDE is a non-parametric way to estimate the probability density function, which assumes the training data distribution can be approximated as a sum of multivariate Gaussians. One can use a kernel distribution when a parametric distribution cannot properly describe the data, or when one wants to avoid making assumptions about the distribution of the data. KDE can be considered as the probability that the point ($\boldsymbol{q}$) locates in the distribution of training data ($\boldsymbol{p_i}, i = 1,2, \dots, n$). It is expressed as, [61]

$$p_{KDE} = \frac{1}{n \cdot h_1 h_2 \dots h_d} \sum_{i=1}^{n} \prod_{j=1}^{d} k\left(\frac{q_j - p_{i,j}}{h_j}\right) \qquad (61)$$

Where d is the number of variables in $\boldsymbol{q}$ and $\boldsymbol{p_i}$. k is the kernel smoothing function. h_j is the bandwidth for each variable. A multivariate kernel distribution is defined by a smoothing function (k) and a bandwidth matrix defined by $H = h_1, h_2, \dots, h_d$, which control the smoothness of the resulting density curve. Therefore, KDE can be used to measure the distance by estimating the probability of a given point locating in a set of training data points. In this step, the KDE distance is standardized as,

$$d_{KDE} = 1 - \frac{p_{KDE}}{p_{KDE} + 0.1} \qquad (62)$$

Before the calculation of KDE distance, the data of PFs should be normalized into the range [0, 1]. Then the normalized KDE distance will locate from 0 to 1. Higher value of KDE distance means higher level of extrapolation.

- **Step 5.2: Test the performance of defined metric**

This step is proposed to whether the defined metric for extrapolative distance can represent the coverage of training data on target data. In other word, does the prediction error decrease with

the measured extrapolative distance? A test matrix can be built with same training database and different testing data sets. The mean of KDE distance for each test case can be calculated as,

$$D_{KDE} = \frac{1}{n}\sum_{i=1}^{n} d_{KDE,i} \tag{63}$$

$d_{KDE,i}$ represents the KDE distance in each local cell of the target case. Then check whether the prediction errors of responses increase monotonically with the value of D_{KDE}. If yes, go to Step 5.3. Otherwise, a better metric should be explored to measure the coverage.

- **Step 5.3: Select the optimal training database by comparing extrapolative distance**

By comparing the extrapolative distance of each candidate of training database, the optimal one can be selected with the smallest value of KDE distance.

5.4.6. Step 6: Mesh/Model Suggestion

After establishing the predictive capability (Step 2) and selecting the optimal PF group (Step 3), ML algorithm (Step 4) and training database (Step 5), the error prediction of local FOMs can be performed for the target case.

Since the global QoIs normally have the most concerns on simulation analysis, the criterion of optimal mesh/model combination is whether this combination can lead to the least prediction error of the global QoIs for the target simulation case. The prediction accuracy of global QoIs depends on the accuracy of local predictions. The estimated error of global QoIs (ε_{global}) for different combinations of mesh size candidates and model candidates can be expressed as the average of estimated local error,

$$\varepsilon_{global} = \frac{1}{n}\sum \varepsilon_{local,i} \tag{64}$$

Select the one with least estimated error of global QoIs as the "optimal" mesh size and model for the target simulation using the LF code. The estimation on the error of LF simulation results are provided.

5.5. Chapter Summary

In this chapter, a data-driven framework for mesh-model optimization in system-level thermal-hydraulic modeling and simulation is proposed. The relevant mathematical basis, practical

consideration, basic assumptions and hypotheses are described before the framework formulation. The central idea is to develop a surrogate model to explore the relationship between local simulation error and specific local Physical Features (PFs). The identification of PFs integrates the physical information of the system of interest, model information and the effect of mesh size.

The main outcomes of OMIS framework are the error prediction and suggestion on the optimal mesh and model selection using machine learning algorithms. OMIS framework is accomplished via a systematic procedure, the sub-outcomes include: (1) PF group is identified based on knowledge basis and has the extendibility from single phenomenon to complex physics; (2) scalability of identified PF group is pre-evaluated via test matrix and optimized by importance study before application; (3) different DNN structures are tested and compared to balance the prediction accuracy and computational cost; (4) data similarity of training data and testing data is measured using KDE distance and visualized in Physical Feature Coverage (PFC) using dimensionality reduction techniques, this provides a guide on the selection of training datasets. These outcomes not only serve on the error prediction and mesh/model selection, but also provide an insight on how to develop, evaluate and optimize a data-driven surrogate model in thermal-hydraulic modeling and simulation.

There are several other advantages of the proposed framework. Firstly, this modularized process has the extendibility to modeling and simulation using other coarse-mesh codes where mesh size is one of the key model parameters. For example, the error prediction on two-phase flow simulation using coarse-mesh RANS model can be performed using OMIS framework since mesh size is involved as a model parameter in the wall functions and closure models for interfacial forces. Here mesh convergence is not achievable and discretization error and model error are tightly connected. Besides, OMIS framework is applicable with limited available data due to the usage of advanced statistical and machine learning whose regression capability is trustworthy, and able to provide better predictions when more relevant data is provided. Finally, the most important benefit from this data-driven framework is its scalability achieved by exploring local physics instead of global physics. It is expected to have the scalability to improve the scale-distorted approaches that connect scaled data to the real full-scale applications and reduce the uncertainty of scaling.

The limitations of the framework also exist. Currently, the framework is proposed for steady-state modeling and simulation, how to apply the framework in a scenario simulation is a challenge, especially when the system condition changes frequently. Moreover, the uncertainty introduced by statistical and machine learning algorithms are not quantified, the relevant uncertainty propagation needs more analysis. Lastly, due to its data-driven property, the framework performance greatly relies the quality and quantity of available data. The uncertainty from the insufficiency of data also requires further studies.

CHAPTER 6. CASE STUDIES OF THE PROPOSED DATA-DRIVEN FRAMEWORK

6.1. Introduction

This chapter illustrates OMIS framework with the case study on mixed convection. Targeting on the "GELI" condition, OMIS framework is developed as a TDMI approach that deals with data, physical model and coarse-mesh simulation in an integrated manner using ML algorithms. By concentrating on the similarity of local physics, OMIS framework has a potential scalability to the globally extrapolative conditions. The underlying local physics of one specific physic condition is assumed to be represented by a set of Physical Features (PFs).

The outcomes of OMIS approach are (1) quantitatively measuring the PF similarity of existing data and target data, and (2) identifying the relationship between these local PFs and local simulation error for future predictions. Therefore, the case study has been designed and executed to demonstrate the scalability and predictive performance of OMIS framework in the GELI condition.

A 2D cavity with hot air injection on bottom of one sidewall, a vent on the other sidewall and a cold top wall has been modeled to simulate the mixed convection considering turbulence.

To evaluate the proposed framework in different GELI conditions, different global extrapolations are designed in Section 6.2 and Section 6.3. Section 6.2 illustrates the entire OMIS framework based on the extrapolation of global parameter case study.

Section 6.3 respectively discusses the OMIS application in GELI condition in extrapolation of geometry (aspect ratio), boundary condition and dimension. Global parameters are defined based on the injection rate and temperature, geometry represents the aspect ratio of the cavity, while boundary condition refers to the heat removal condition: a cold top wall with a fixed temperature or a fixed heat flux.

These three conditions all exceeds the application domain since the global physics condition changes with an unknown uncertainty. The overall objective is to provide error estimation and suggestion on optimal mesh/model selections, while the sub-objectives in each step include (1) establish the predictive capability, (2) identify optimal PF group, (3) determine optimal DNN structure, and (4) construct optimal training database. Lessons learned from each case study are also recorded for the improvement of the framework in the future.

6.2. Case Study: Extrapolation of Global Parameter

6.2.1. Formulation

The mixed convection case with hot air injection on bottom of one side wall and a vent on the other side wall was simulated using a GOTHIC 2D model, as shown in Figure 31. The height and length of this cavity are both 1m, while the height of the inlet and vent are both 0.2 m. The target case and the data warehouse are listed in Table 3. The global parameters are defined as below,

$$Gr_i = \frac{g(\rho_w - \rho_i)\rho_f H^3}{\mu^2} \quad (65)$$

$$Re_i = \frac{U_i \rho_i H}{\mu} \quad (66)$$

It is obvious that Gr_i of the target case is extrapolative of the cases in the data warehouse while Re_i of the target case is interpolative. Therefore, this case study is to investigate the performance of OMIS framework in the extrapolation of high Gr_i. By training a DNN using the data in data warehouse, the simulation error and optimal mesh/model selection of the target case will be provided.

Table 3. Target Case and Data Warehouse of Case Study

Case NO.		T_i (℃)	U_i (m/s)	Gr_i	Re_i
Data Warehouse	1	30	0.1	1.124E+09	5.863E+03
	2	33	0.2	1.414E+09	1.159E+04
	3	36	0.3	1.695E+09	1.717E+04
	4	39	0.4	1.967E+09	2.262E+04
	5	42	0.1	2.231E+09	5.585E+03
	6	45	0.2	2.486E+09	1.103E+04
	7	48	0.3	2.733E+09	1.634E+04
	8	51	0.4	2.971E+09	2.152E+04
	9	54	0.1	3.201E+09	5.312E+03
	10	57	0.2	3.424E+09	1.049E+04
	11	60	0.3	3.638E+09	1.554E+04
Target case		63	0.4	3.845E+09	2.045E+04

* For each case, one HF simulation is performed by Star CCM+, four LF simulations are performed by GOTHIC with different coarse meshes (1/10, 1/15, 1/25, 1/30 m). Each case generates 1850 data points.

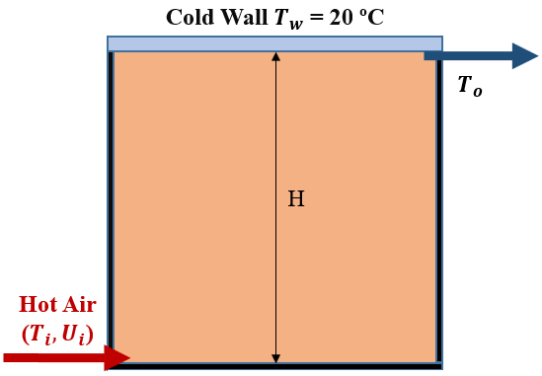


Figure 31. The Illustration of GOTHIC 2D Model for Mixed Convection Case Study

6.2.2. Implementation

- **Step 1. Simulation Target Analysis:**

The physics investigated in this case is mixed convection with hot fluid injection and top heat removal. By performing PIRT process, mixed convection is mainly dominated by wall friction modeling, turbulence modeling and convection heat transfer modeling in GOTHIC, as illustrated in Figure 32.

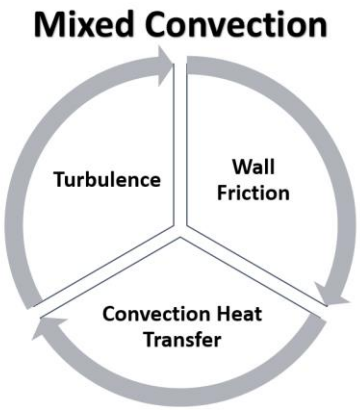


Figure 32. The Illustration of Physics Decomposition for Mixed Convection in Case Study

The respective closure models in GOTHIC for these three phenomena are reviewed and qualitatively assessed in Chapter 3. The wall friction model applied in this case is Equation (9), which is completed by Equation (10) to Equation (24). The turbulence model is the standard two-equation k–e model with the near-wall treatment in GOTHIC, as reviewed in Figure 9. The convection heat transfer model is Equation (6), which considers both of natural convection and forced convection. The models are listed as below,

$$H_{nc} = \frac{k}{l} Max\left(0.54Ra^{0.25}, 0.14Ra^{1/3}\right) \quad (67)$$

$$H_{fc} = \frac{k}{l} 0.023Re^{0.8}Pr^{0.3} \quad (68)$$

The natural convection model shown in Equation (67) is a mix of two different convection models, which are developed to define a view of flat horizontal surface that is facing down, such as a ceiling in this case study. The forced convection model shown in Equation (68) is developed for pipe flow, which is the only well-defined forced convection model in GOTHIC. The real heat transfer coefficient is the maximum one of H_{nc} and H_{fc}. In this case study, the physical models applied in Low-Fidelity (LF) simulation are fixed, the goal is simplified to predict the simulation error and suggest the optimal mesh size for the target case. It should be noted that mesh size is one of the key parameters of these applied closure models. Four different mesh sizes are applied for LF modeling and simulation: 1/10 m, 1/15 m, 1/25 m, and 1/30 m.

The global QoI in this case is set as the outlet temperature T_o via the vent, which is directly affected by these key physics. T_o represents the heat removal capability of this square cavity. These parameters represent the system behaviors of NPPs and provide information for the decision-makings of operators. For example, in the normal operating of BWRs, the temperature/pressure in the wetwell and the hydrogen fraction in the drywell can be considered as the key QoIs if the severe events happen as in the Fukushima accident. The values of these QoIs could help the operators and decision makers estimate the benefit and risk when to inject seawater into the units. Therefore, the estimation of these key QoIs is the criteria of which mesh or model is the optimal one in the OMIS framework. The global parameters have been identified as Gr_i and Re_i, which includes the global information such as injection condition, geometry condition and boundary condition. This helps the selection of training database and also the construction of test matrix.

- **Step 2. Predictive Capability Development:**

This step establishes the predictive capability of the data-driven model by preliminarily defining potential Physical Features (PFs), building database, and evaluating whether the selected database has the predictive capability on the test matrix.

- Step 2.1: Define potential physical features

The identification of PFs is guided by physics decomposition in Step 1.

The identified PFs in this case study are marked in red, as displayed in Figure 33. In addition to variable gradients, other part of PF group are the local parameters that are defined to represent the local physical behaviors or applied in closure relationships. Five different non-dimensional parameters are defined in this case study: $\boldsymbol{R}$ includes the turbulent information; $\boldsymbol{Re}$ is defined with the consideration of both $\boldsymbol{Re}$ in free cells and $\boldsymbol{Re}$ in near-wall cells; $\boldsymbol{Gr}$ approximates the ratio of the buoyancy to viscous force acting on a fluid by considering local density change; $\boldsymbol{Ri}$ expresses the ratio of the buoyancy term to the flow shear term, which represents the importance of natural convection relative to the forced convection; $\boldsymbol{Pr}$ reflects the ratio of momentum diffusivity to thermal diffusivity, which depends only on the fluid property and state.

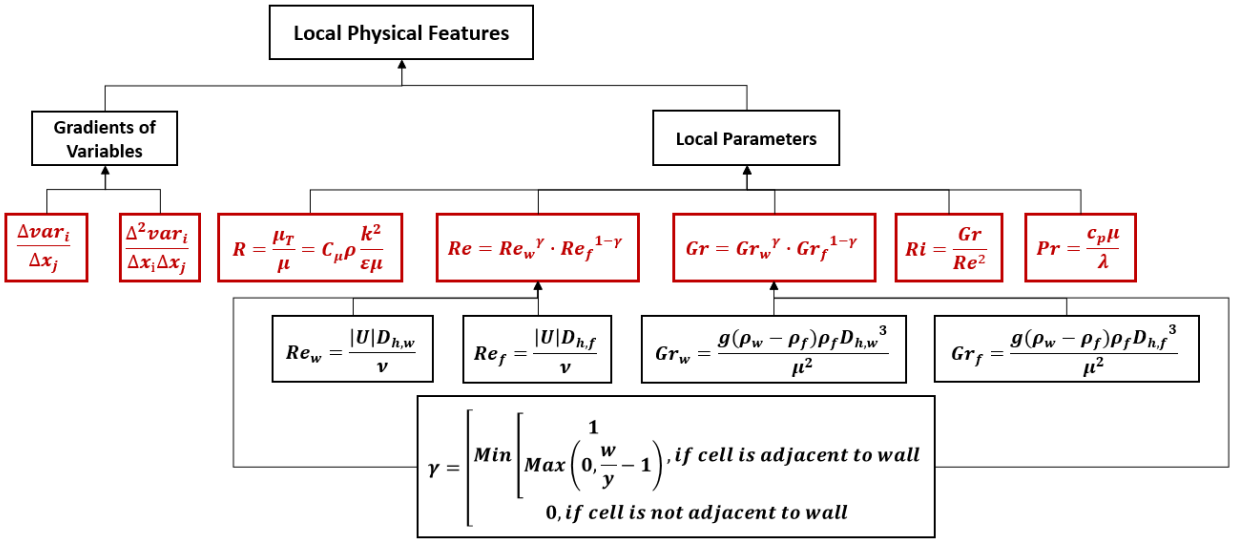


Figure 33. Identification and classification of Physical Features in Case Study

It should be noted that these local parameters enable PF group the scalability and additivity of physics. Scalability of physics indicates that if the PF data of existing case and target case is similar, the local physical information of target case should be covered by existing case, even if these cases are in the different length scales. Meanwhile, the PFs identified for simple single phenomenon is still usable for complex coupled physics.

PF group is improvable by adding more relevant PFs if new phenomena are involved. For example, in a pre-test before this case study where only wall friction and turbulence are considered in an adiabatic condition, only $\boldsymbol{R}$ and $\boldsymbol{Re}$ are used. $\boldsymbol{Gr}$, $\boldsymbol{Ri}$ and $\boldsymbol{Pr}$ were added in when convection heat transfer should be considered.

- Step 2.2: Collect existing high-fidelity and low-fidelity data

High-Fidelity (HF) and Low-Fidelity (LF) data are respectively generated by Star CCM+ with fine mesh and GOTHIC with coarse mesh, as displayed in Figure 34. In this case study, HF data is generated using 2D RANS model in Star CCM+ with a nodalization of 150*150 in bulk and 600 refinement on top and bottom layer. The refinement on top and bottom is designed to capture the detailed information from injection in bottom part, heat removal and vent in top part. Standard *k*–ε low-Re model is applied with all y+ wall treatment since (1) this model is robust and easy to implement in small pressure gradient and good for mixing simulation; (2) low Re number approach provides identical coefficients to standard *k*–ε model and damping functions; (3) all y+ wall treatment is a hybrid treatment that emulates the low y+ wall treatment for fine meshes and the high y+ wall treatment for coarse meshes.

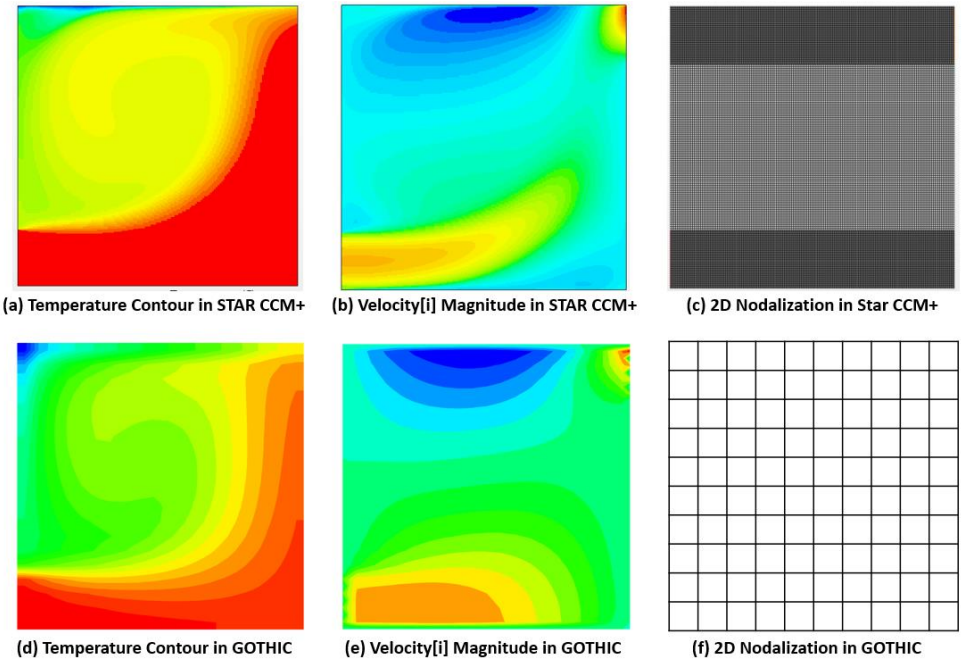


Figure 34. Illustration of 2D GOTHIC Model with Coarse Meshes and 2D Star CCM+ Model with Fine mesh

LF data is generated by GOTHIC in four groups with same closure models and different uniform mesh sizes: 1/10 m, 1/15 m, 1/25 m, and 1/30 m. In Figure 34, the 2D nodalizations in Star CCM+ and GOTHIC (10*10) are displayed with the temperature distribution and horizontal velocity magnitude. It is obvious that the variables in fine-mesh modeling are much more smoothly simulated and distributed.

- Step 2.3: Build database

The inputs of database are made of the PFs defined in Step 2.1, and the outputs of database are the errors of the FOMs (u, v, T). As described in Chapter 5, the cell-to-cell method is applied to calculate the errors in this case study. The inputs and outputs of database are listed in Table 4, where variables are u, v, T, P, k. The database includes the data from case 1 to 11 in Table 3.

Table 4. Database Inputs and Outputs of Case Study

	Physical Feature	Number
Inputs	$\frac{\Delta var_i}{\Delta x_j} + \frac{\Delta^2 var_i}{\Delta x_j \Delta x_i}$	10 +15 (2D)
	Re, Gr, Ri, Pr, R	5
Outputs	$\Delta FOM_i = FOM_{i_{HF}} - FOM_{i_{LF}}$	3 (2D)

- Step 2.4: Determine physics coverage condition of target case

Considering that OMIS framework is only applicable when the target case locates in GILI or GELI condition, PF data of target case should be calculated and compared with the collected data to determine which physics coverage condition the target case belongs to. Otherwise, the collected data is unusable to cover and inform the local physics of target case.

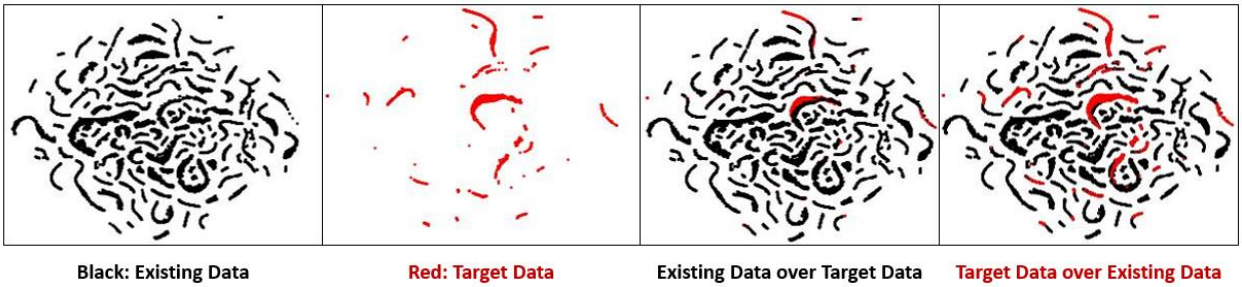


Figure 35. Physical Feature Coverage of Target Case in Case Study

By using the dimensionality reduction technique t-SNE (t-Distributed Stochastic Neighbor Embedding) method, the physics coverage condition of the target case can be visualized, as shown in Figure 35. It is obvious that most of the data points of target case (red points) are covered or overlapped by the training data points (black points) in case 1-11, even though globally, target case is an extrapolation of training case. Considering the dataset is reduced from high dimensionality (30 D) to low dimensionality (2D), only coverage or overlapping represents the strong similarity. The relative distances among the points are stored and reflected from high dimensionality to low

dimensionality. For example, if the distance between point A and point B is further than the distance between point A and point C in original dimensionality, the distance between point A and point B is still further than the distance between point A and point C in new dimensionality. The physics coverage condition of target case is determined as GELI condition.

- Step 2.5: Design test matrix

Once the physics coverage condition of target case is determined as GELI, test matrix should be designed to investigate whether the PF group in collected database has the expected predictive capability for the determined physics coverage condition. Here extrapolative distance is defined to determine the coverage of collected data on the target data. The metric applied in this step is the mean of KDE distance, which is defined in Equation (63). The physical condition with similar mean of KDE distance should be designed for testing. Different conditions are designed and compared as listed in Table 5. Higher mean of KDE distance implies less coverage and similarity. The mean of KDE distance of target case from case 1-11 is a little smaller than the mean of KDE distance of case 11 from case 1-7, therefore, here Condition 1 in test matrix is selected as the test case. If the prediction on case 11 by using case 1-7 as training data is within an acceptable accuracy, the prediction on target case by using case 1-11 is trustworthy since they have similar mean of KDE distance. The Probability Density Functions (PDF) of KDE distance for four conditions in test matrix are displayed in Figure 36. Although they have similar distribution, Condition 1 has a "worse" coverage than others. Therefore, Condition 1 is a conservative option as the test case.

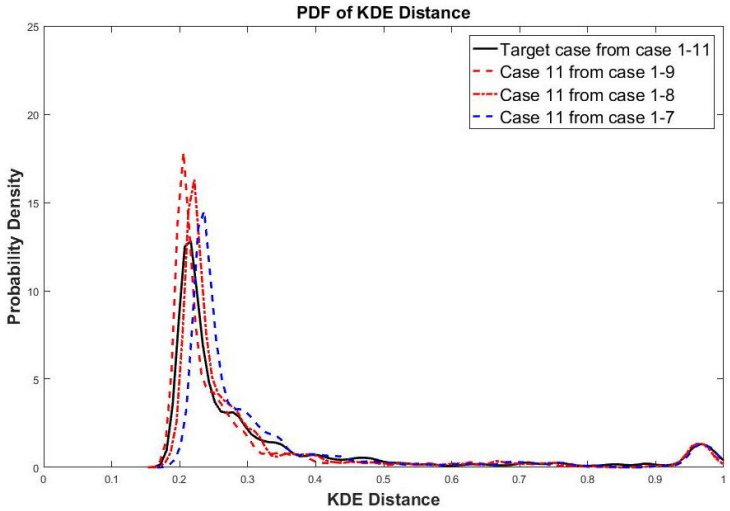


Figure 36. PDFs of KDE Distance for Different Conditions in Test Matrix for Case Study

Table 5. Test Matrix with Different Training Case and Testing Case for Case Study

Test Matrix	Testing Case	Training Case	Mean of KDE distance
-	Target case	1-11	0.3354
Condition 1	11	1-7	0.3389
Condition 2	11	1-8	0.3216
Condition 3	11	1-9	0.3067
Condition 4	11	1-10	0.2940

- Step 2.6: Apply machine learning algorithm on data training and prediction

In this step, a FNN with 3 Hidden Layers (HLs) and 20 neurons in each HL is applied for data training and prediction on the test case. This 3-HL 20-neuron FNN is just used for testing, not the FNN structure for final training and prediction.

- Step 2.7: Evaluate predictive capability on test matrix

Here the prediction performed in previous step is evaluated using Mean Squared Error (MSE) as the metric. The original GOTHIC simulation results are compared with modified values by ML prediction, as shown in Figure 37. The vertical axis is the HF data averaged from Star CCM+. The values of predicted variables (red circles) are quite close to the values from HF data with small values of MSE. Blue points are the comparison between LF results and HF data. The proposed data-driven approach represents good predictive capability and scalability on estimating the local simulation error within an acceptable uncertainty even for the extrapolation of global physics. The MSEs of predictions are listed in Table 6. The results have been greatly improved by error prediction using ML. Better performance can be achieved if better DNN structure is applied for data training and prediction.

Table 6. MSEs of Predictions for the Test Case in Case Study

Testing Case	Training Case	MSE (u)	MSE (v)	MSE (T)
11	1-7	1.0e-3	9.0e-4	2.65
Original GOTHIC Simulation		9.3e-3	9.0e-3	24.3

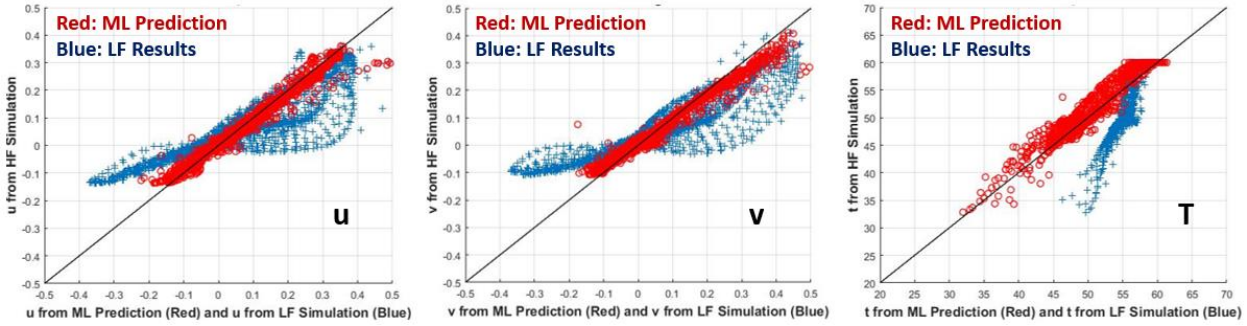


Figure 37. Comparisons between Original GOTHIC Simulation Results and Modified Results by ML Prediction

- **Step 3. Physical Feature Group Optimization:**

In this step the importance analysis has been performed on these PFs to select optimal PF group with respect to both of PF importance ranking and computation cost for data training.

- Step 3.1: Identify and rank the importance of all potential physical features

By applying Permutation Variable Importance Measure (PVIM) based on Random Forest Regression (RFR), the importance of all potential PFs are identified, quantified and ranked, as shown in Figure 38. Higher value implies higher importance.

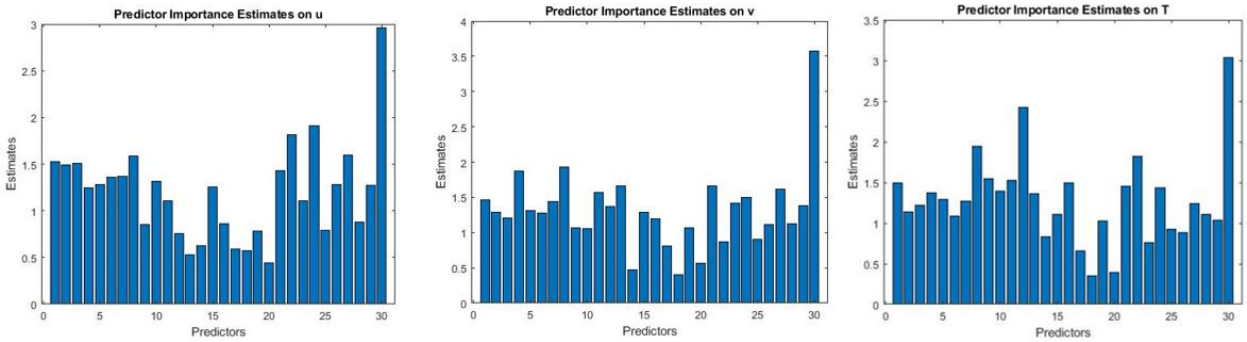


Figure 38. Importance Estimation of PFs on Different Local FOMs

The gradients of velocity, temperature and kinetic energy presents more importance than the ones of pressure since pressure does not change much in the entire cavity. The gradients of pressure are relatively implicit compared with others. All the local physical parameters shows great importance, especially Pr number which is defined as the ratio of momentum diffusivity to thermal diffusivity. It should be noted that PVIM is a non-parametric method and purely relying on data.

- Step 3.2: Suggest optimal physical feature group based on the importance ranking

According to the importance score of each PF in Step 3.1, the importance of PF can be manually classified into three levels: High, Middle and Low (H, M, and L). Each PF has 3 importance scores depending on the number of FOMs. High level means all these three scores of this PF are higher than 1; Low level means all the scores are less than 1; Middle level represents the rest conditions. The importance classification of each PF is listed in Table 7. Therefore, three different PF groups can be generated respectively including PFs in H level, H+M level and H+M+L level. The classification of PF importance and group may have different criteria, which makes it feasible to select optimal PF group to represent the characteristics of the underlying physics in data.

Table 7. Importance Classification of Each PF in Case Study

NO.	PF	Importance Level	NO.	PF	Importance Level	NO.	PF	Importance Level
1	$\frac{\Delta u}{\Delta x}$	H	11	$\frac{\Delta t}{\Delta x}$	H	21	$\frac{\Delta k}{\Delta x}$	H
2	$\frac{\Delta u}{\Delta y}$	H	12	$\frac{\Delta t}{\Delta y}$	M	22	$\frac{\Delta k}{\Delta y}$	M
3	$\frac{\Delta v}{\Delta x}$	H	13	$\frac{\Delta p}{\Delta x}$	M	23	$\frac{\Delta^2 k}{\Delta x \Delta x}$	M
4	$\frac{\Delta v}{\Delta y}$	H	14	$\frac{\Delta p}{\Delta y}$	L	24	$\frac{\Delta^2 k}{\Delta y \Delta y}$	H
5	$\frac{\Delta^2 u}{\Delta x \Delta x}$	H	15	$\frac{\Delta^2 t}{\Delta x \Delta x}$	H	25	$\frac{\Delta^2 k}{\Delta x \Delta y}$	L
6	$\frac{\Delta^2 u}{\Delta y \Delta y}$	H	16	$\frac{\Delta^2 t}{\Delta y \Delta y}$	M	26	Re	M
7	$\frac{\Delta^2 v}{\Delta x \Delta x}$	H	17	$\frac{\Delta^2 p}{\Delta x \Delta x}$	L	27	R	H
8	$\frac{\Delta^2 v}{\Delta y \Delta y}$	H	18	$\frac{\Delta^2 p}{\Delta y \Delta y}$	L	28	Gr	M
9	$\frac{\Delta^2 u}{\Delta x \Delta y}$	M	19	$\frac{\Delta^2 t}{\Delta x \Delta y}$	M	29	Ri	H
10	$\frac{\Delta^2 v}{\Delta x \Delta y}$	H	20	$\frac{\Delta^2 p}{\Delta x \Delta y}$	L	30	Pr	H

* H: all scores > 1.0, L: all scores < 1.0, M: others.

- Step 3.3: Evaluate predictive capability of suggested physical feature group on test matrix

In order to make a balance between prediction accuracy and computational efficiency, not all the potential PFs are needed for FNN training and error prediction. Three PF groups with different importance levels are applied and investigated in this step. Here MSEs of prediction are considered as the metrics to evaluate the prediction accuracy and training time of FNN represents the computational cost in FNN training. It shows that G2 with the PFs in H and M levels still well captures the underlying physics of data and also saves computation for FNN training. In Table 8, MSEs of G2 PF group is very close to the values of G3 PF group, while the training time is less than the half. Therefore, G2 PF group with PFs in H and M levels is used as the optimal PF group in this case study for the following steps.

Table 8. Predictive Capability of Different PF Groups on Test Case

PF Group	NO. of PF	MSE (u)	MSE (v)	MSE (T)	Training Time	Testing Case	Training Case	FNN Structure
G1 (H)	16	6.0e-3	4.2e-3	3.73	1 h	11	1-7	3-HL 20-Neuron FNN
G2 (H+M)	25	1.1e-3	1.1e-3	2.69	1.5 h			
G3 (All)	30	1.0e-3	9.0e-4	2.65	3.5 h			
Original GOTHIC Simulation		9.3e-3	9.0e-3	24.3				

The PFs in G2 PF group are listed in Table 9, the number of PF is reduced from 30 to 25. It is expected to reduce more for more complex conditions. It should be noted that more groups can be generated if needed, and predictive capability can be improved using complex FNNs. No matter which PF group is selected, original GOTHIC simulation is greatly improved.

Table 9. The Optimal PF Group for Case Study

Optimal Physical Features		Number
Variable Gradients	$\frac{\Delta u}{\Delta x}, \frac{\Delta u}{\Delta y}, \frac{\Delta v}{\Delta x}, \frac{\Delta v}{\Delta y}, \frac{\Delta t}{\Delta x}, \frac{\Delta t}{\Delta y}, \frac{\Delta p}{\Delta x}, \frac{\Delta k}{\Delta x}, \frac{\Delta k}{\Delta y}, \frac{\Delta^2 u}{\Delta x \Delta x}, \frac{\Delta^2 u}{\Delta y \Delta y}, \frac{\Delta^2 v}{\Delta x \Delta x}, \frac{\Delta^2 v}{\Delta y \Delta y}, \frac{\Delta^2 u}{\Delta x \Delta y}, \frac{\Delta^2 v}{\Delta x \Delta y}, \frac{\Delta^2 t}{\Delta x \Delta x}, \frac{\Delta^2 t}{\Delta y \Delta y}, \frac{\Delta^2 t}{\Delta x \Delta y}, \frac{\Delta^2 k}{\Delta x \Delta x}, \frac{\Delta^2 k}{\Delta y \Delta y}$	20
Local Parameters	Re, Gr, Ri, Pr, R	5

- **Step 4. Machine Learning Algorithm Determination:**

After the optimization of PF group, the ML algorithm for data training and prediction is optimized and determined in this step. Multi-layer FNN is selected as the ML algorithm in this work considering its deep-learning capability to explore the highly non-linear relationship between PFs and simulation errors. Therefore, the procedure of ML algorithm determination contains two parts: identify potential ML candidates, test their predictive capability and select the optimal one with the consideration of accuracy and computation cost. Same as the previous step, MSEs and training time are identified as the evaluation metrics. In this step, different multi-layer FNN structures are investigated for the test case. The performance of each FNN structure can be found in Table 10. It shows that the 4-HL 20-neuron FNN has the most promising performance: higher accuracy and less computational cost. Therefore, the 4-HL 20-neuron FNN is selected as the optimal FNN structure and ML algorithm in following steps.

Table 10. Performance of FNN Candidates for Test Case

Hidden Layer NO.	Neuron NO.	Training Time	MSE (u)	MSE (v)	MSE (T)	Testing Case	Training Case
1	20	0.5 h	3.2e-3	4.4e-3	9.13	11	1-7
2	20	1 h	2.4e-3	1.3e-3	3.02		
3	20	1.5 h	1.1e-3	1.1e-3	2.69		
3	30	9 h	1.2e-3	7.2e-4	1.57		
4	20	6 h	8.2e-4	7.1e-4	1.07		
Original GOTHIC Simulation			9.3e-3	9.0e-3	24.3		

The predictive performance using 4-HL 20-neuron FNN for training and prediction is shown in Figure 39. Compared with the original GOTHIC simulation results (blue points), the values of predicted variables (red circles) are much close to the values mapped from HF data with small MSE. And it is obvious that the prediction using 4-HL 20-neuron FNN is much better than the prediction using 3-HL 20-neuron FNN.

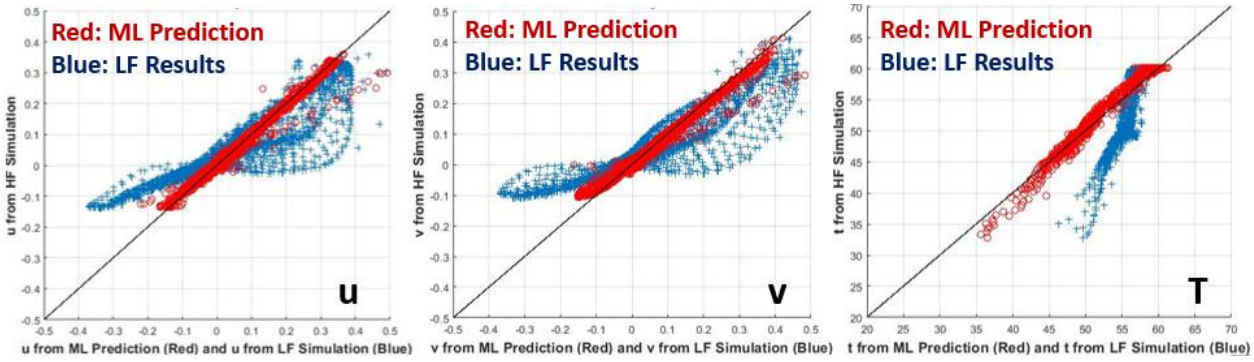


Figure 39. Predictive Performance using 4-HL 20-neuron FNN

- **Step 5. Training Database Construction:**

After the optimization of PF and FNN selection, this step focuses on how to select sufficient and necessary data for training.

- Step 5.1: Define metric to calculate extrapolative distance

In this work, KDE distance is applied as the extrapolative distance. If target data is more covered or similar to the training data, the prediction error on the target case is smaller. KDE is a non-parametric way to estimate the probability density function, which assumes the training data distribution can be approximated as a sum of multivariate Gaussians. One can use a kernel distribution when a parametric distribution cannot properly describe the data, or when one wants to avoid making assumptions about the distribution of the data. KDE can be used to measure the distance by estimating the probability of a given point locating in a set of training data points. In this step, the KDE distance is standardized as in Equation (62). Before the calculation of KDE distance, the data of PFs should be normalized into the range [0, 1]. The normalized KDE distance locates from 0 to 1. Higher value of KDE distance means less similarity.

- Step 5.2: Test the performance of defined metric

In this step, the capability of KDE distance to represent the coverage of training data on target data is evaluated. In other word, does the prediction error decrease as the KDE distance increases? A test matrix can be built with same training database and different testing data sets. The mean of KDE distance for each test case can be calculated in Equation (63). Several tests are performed to explore the relationship between mean of KDE distance and MSEs of prediction, as displayed in Table 11. It seems that there is a nearly positive relationship between mean of KDE

distance and MSEs of prediction, as displayed in Figure 40. With higher mean of KDE distance, the MSEs of prediction tends to increase. This conclusion is instructive for the selection of optimal training database and also the development of validation experiments. This relationship should be more distinct when more data are included.

Table 11. Mean of KDE distance and MSEs of Prediction of Tests

Training Cases	Testing Case	FNN Structure	Mean of KDE distance	MSE (u)	MSE (v)	MSE (T)
1-7	8	4-HL 20-Neuron	0.2282	8.9e-5	8.54e-5	0.20
	9		0.2493	8.5e-4	8.6e-4	2.01
	10		0.2834	1.0e-3	9.0e-4	2.11
	11		0.3450	1.1e-3	1.1e-3	2.69
1-8	9		0.2442	5.2e-4	6.5e-4	1.46
	10		0.2732	9.36e-4	7.53e-4	1.75
	11		0.3269	1.20e-3	1.03e-3	1.72
1-9	10		0.2622	8.7e-4	9.0e-4	2.34
	11		0.3112	1.14e-3	9.5e-4	1.79

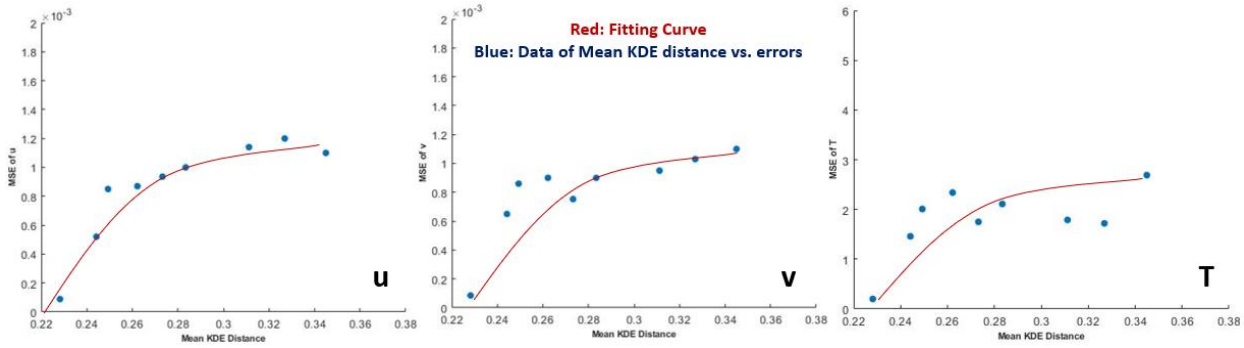


Figure 40. Relationship between Mean of KDE distance and MSEs of Prediction

- Step 5.3: Select the optimal training database by comparing extrapolative distance

By comparing the mean of KDE distance of each candidate of training database, the optimal one can be selected with the smallest value of KDE distance, as shown in Table 12. According to the values of mean of KDE distance, it is proved that case 11 has very similar data as the target case, which is obvious. Case 1 seems has the largest difference with the target case

since when it is not included in the training database, the mean of KDE distance decreases a lot. Although the training database with case 3-11 has smaller mean of KDE distance than the one of training database with case 2-11, the latter one is selected as the optimal training database since the prediction error does not change much when mean of KDE distance exceeds 0.3, according to Figure 40. And the performance of multi-layer FNN relies on the size of training database, it tends to include more data to fully capture the underlying information. By considering FNN performance and computational cost in FNN training, here the training database with case 2-11 is selected as the optimal training database, as displayed in

Table 13. The Probability Density Functions (PDFs) of KDE distance for the candidates of training database are displayed in Figure 41.

Table 12. Mean of KDE distance of Training Database Candidates

Testing Case	Training Cases	Mean of KDE distance
Target case	1-11 (all)	0.3388
	1-10	0.3542
	2-11	0.3253
	3-11	0.3126

Table 13. Optimal Training Database and Target Case

Case NO.		T (℃)	U(m/s)	Gr_i	Re_i
	1	30	0.1	1.124E+09	5.863E+03
Training Database	2	33	0.2	1.414E+09	1.159E+04
	3	36	0.3	1.695E+09	1.717E+04
	4	39	0.4	1.967E+09	2.262E+04
	5	42	0.1	2.231E+09	5.585E+03
	6	45	0.2	2.486E+09	1.103E+04
	7	48	0.3	2.733E+09	1.634E+04
	8	51	0.4	2.971E+09	2.152E+04
	9	54	0.1	3.201E+09	5.312E+03
	10	57	0.2	3.424E+09	1.049E+04
	11	60	0.3	3.638E+09	1.554E+04
Target case		63	0.4	3.845E+09	2.045E+04

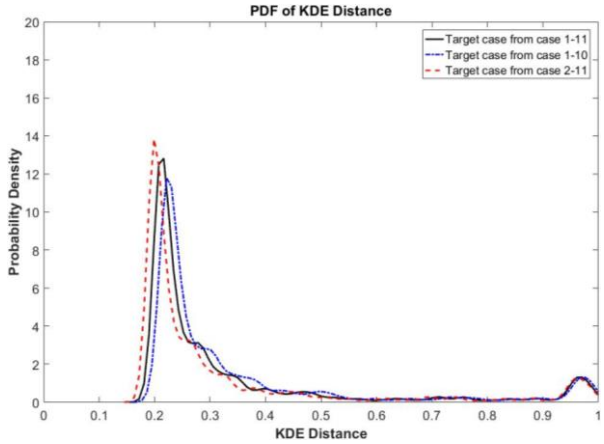


Figure 41. PDFs of KDE Distance for Different Conditions in Test Matrix

- **Step 6. Mesh/model Suggestion:**

The error prediction of local FOMs can be performed for the target case by using optimal PF group in Step 3, optimal NN structure in Step 4, and optimal training database in Step 5.

Table 14. Optimal PH Group, FNN structure and Training Database

Optimal Physical Features	
Variable Gradients	$\frac{\Delta u}{\Delta x}, \frac{\Delta u}{\Delta y}, \frac{\Delta v}{\Delta x}, \frac{\Delta v}{\Delta y}, \frac{\Delta t}{\Delta x}, \frac{\Delta t}{\Delta y}, \frac{\Delta p}{\Delta x}, \frac{\Delta k}{\Delta x}, \frac{\Delta k}{\Delta y}, \frac{\Delta^2 u}{\Delta x \Delta x}, \frac{\Delta^2 u}{\Delta y \Delta y}, \frac{\Delta^2 v}{\Delta x \Delta x}, \frac{\Delta^2 v}{\Delta y \Delta y}, \frac{\Delta^2 u}{\Delta x \Delta y}, \frac{\Delta^2 v}{\Delta x \Delta y}, \frac{\Delta^2 t}{\Delta x \Delta x}, \frac{\Delta^2 t}{\Delta y \Delta y}, \frac{\Delta^2 t}{\Delta x \Delta y}, \frac{\Delta^2 k}{\Delta x \Delta x}, \frac{\Delta^2 k}{\Delta y \Delta y}$
Local Parameters	Re, Gr, Ri, Pr, R
Optimal Neural Network Structure	
4-HL 20-neuron FNN	
Optimal Training Database	
Case 2 - 11	

In this case study, the global QoI is defined as the outlet temperature. The criterion of optimal mesh/model combination is whether this combination can lead to the least prediction error of the global QoIs for the target simulation case. The prediction accuracy of global QoIs depends on the accuracy of local predictions. The estimated error of global QoIs (ε_{global}) for different combinations of mesh size candidates and model candidates can be expressed as the average of

estimated local error, as in Equation (64). Considering there are four different mesh sizes for selection, the outlet temperature is calculated as the average shown in Figure 42. The predicted errors of outlet temperature with different mesh sizes are listed in Table 15. Considering the HF data for the target case is assumed not available, one cannot compare GOTHIC results and determine which mesh produces the least simulation error. According to the error prediction using optimal FNN, PF group and training database, GOTHIC simulation with the mesh size as 1/30 m has the least predicted error of outlet temperature considering the closure model selection is fixed in this case study. Therefore, 1/30 m is the optimal mesh size for this target case, and the predicted error of outlet temperature is 0.89.

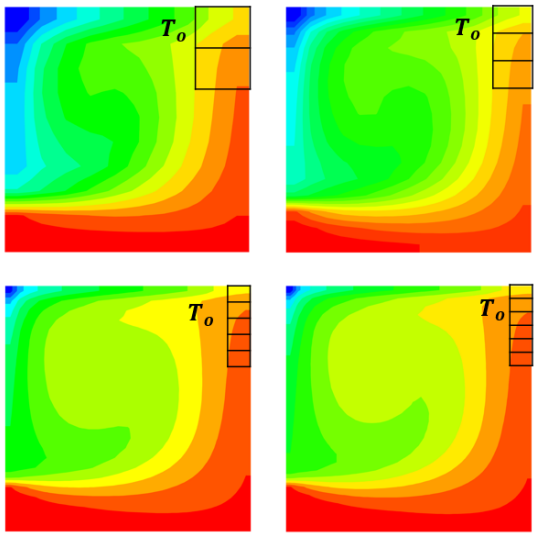


Figure 42. Illustration of Outlet Temperature Calculation in Each Coarse-mesh Simulation

Table 15. Predicted Error of Outlet Temperature with Different Mesh Sizes

Mesh and Model Candidates		Predicted Error of T_o
Model is fixed in this case	10*10	1.88
	15*15	0.96
	25*25	1.74
	30*30	0.89

6.2.3. Lessons Learned

After the application of the framework on this case study, we can evaluate,

1. Whether 1/30 m is the optimal mesh size for the target case when the physical model selection is fixed?

The comparison between original GOTHIC simulation error and predicted error from OMIS are displayed in Table 16. When 1/30 m is used as the mesh size, LF simulation using GOTHIC has the least simulation error on the prediction of outlet temperature. HF data are mapped from fine-mesh simulations using Star CCM+. It is proved that 1/30 is the optimal mesh size for this case study.

2. Whether the error prediction on outlet temperature is accepted?

By comparing LF simulation error and predicted error from OMIS in Table 16, the error of prediction from OMIS is calculated and much smaller than 1%. It is sufficiently accurate and well acceptable.

Table 16. Comparison of Original GOTHIC Simulation Error, Predicted Error by OMIS and Prediction Error of Outlet Temperature

Mesh Size	$T_{o_{LF}}$	$T_{o_{HF}}$	$T_{o_{predicted}}$	LF Simulation Error	Predicted Error	Relative Prediction Error
10*10	59.11	61.08	60.99	1.97	1.88	0.15%
15*15	60.33	61.43	61.29	1.1	0.96	0.22%
25*25	60.01	61.64	61.75	1.63	1.74	0.17%
30*30	60.74	61.68	61.63	0.94	0.89	0.08%

3. Whether the LF simulation using GOTHIC can be well corrected by OMIS framework?

The corrected results by ML training are compared with the original GOTHIC simulation as displayed in Figure 43. LF simulation is greatly improved by applying OMIS framework.

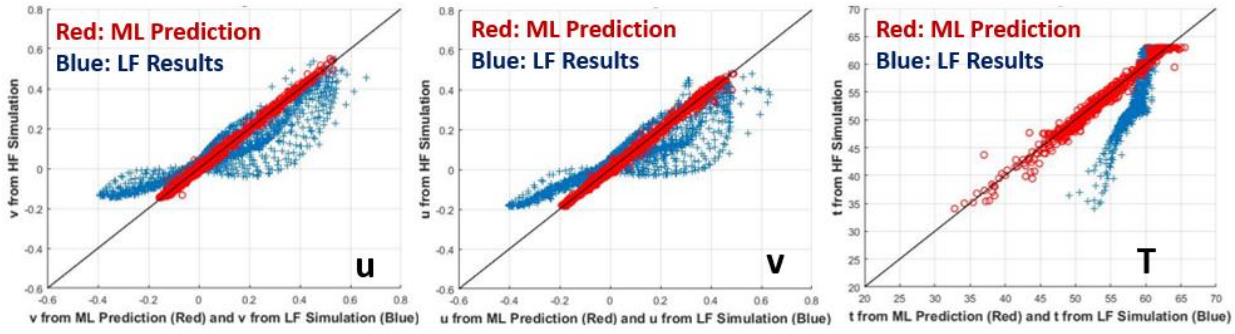


Figure 43. Comparisons between GOTHIC Simulation Results and Corrected Results by OMIS framework for the Simulation of Target Case with 1/30 m as Mesh Size

This case study denotes that data similarity between training data and target data is directly affecting the NN training and prediction. The metric to measure extrapolative distance is set as mean of KDE distance in this work. The relationship between mean of KDE distance and prediction error (MSEs of local QoIs) needs more investigation and analysis. Meanwhile, the construction of training database should consider both of the quantity and similarity of training data. The balance between data quantity and data similarity is discussed in Section 6.3.

6.3. Discussion on Application

In this section, three extrapolative situations in GELI condition are proposed, analyzed and evaluated. The mixed convection simulation is still used as the case study. Section 6.3.1 discusses the extrapolation of geometry, training cases and testing cases have different aspect ratios. Section 6.3.2 demonstrates the extrapolation of boundary condition, training cases have fixed top wall temperature while the testing cases have fixed top heat flux. Section 6.3.3 investigates the extrapolation of dimension, testing cases have larger height and length than the training cases, so it can be considered as the extrapolation of global length scale. The objective of the efforts in this section is to investigate the predictive capability of proposed OMIS approach in GELI condition. Besides, these extrapolation case studies also:

1. Explore the importance of training data size and similarity. The similarity is quantified using KDE distance between training data and target data. (**Extrapolation of geometry**)

2. Explore the relationship between extrapolative distance and prediction error. The metrics for extrapolative distance and prediction error are respectively mean of KDE distance and Normalized Root Mean Squared Errors (NRMSEs) of variables. NRMSE is calculated using Equation (69). (**Extrapolation of boundary condition, extrapolation of dimension)**

$$NRMSE_{prediction} = \frac{\sqrt{\frac{1}{n}\sum\left(QoI_{HF,i} - QoI_{predicted,i}\right)^2}}{\frac{1}{n}\sum QoI_{HF,i}} \quad (69)$$

6.3.1. Extrapolation of Geometry (Aspect Ratio)

In this case, three cavities with different aspect ratios are modeled, as shown in Figure 44. The injection condition and geometry parameters are listed in Table 17. Dataset A contains the case 1-12 in the previous case study discussed in Section 6.2. Dataset B and C respectively contain the rectangular modeling cases with aspect ratios as 1/0.8 and 0.8/1.

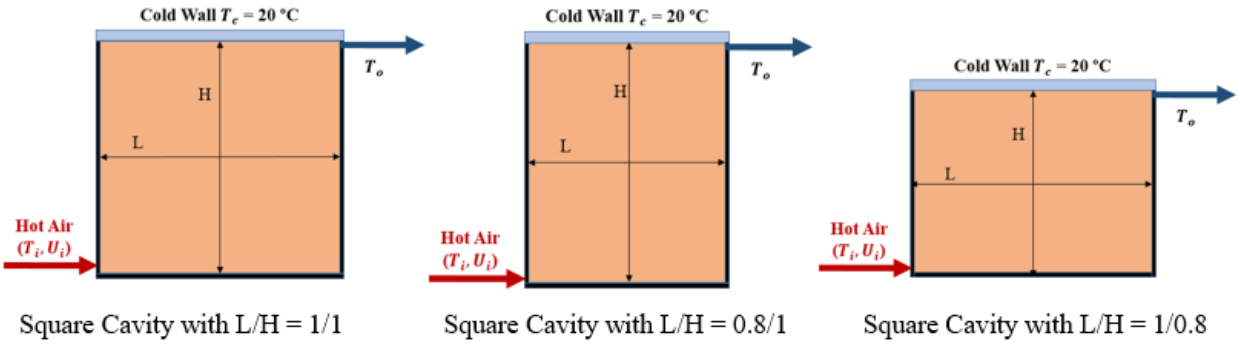


Figure 44. Three Cavity Models with Different Aspect Ratios

Table 17. Geometry and Injection Conditions of Datasets in Extrapolation of Geometry

Dataset		Geometry	Aspect Ratio	Injection Temperature	Injection Rate	Data Size
A	1	Square	L/H = 1/1	30 C	0.1 m/s	12*1850
	2			33 C	0.2 m/s	
	3			36 C	0.3 m/s	
	4			39 C	0.4 m/s	
	5			42 C	0.1 m/s	
	6			45 C	0.2 m/s	
	7			48 C	0.3 m/s	
	8			51 C	0.4 m/s	
	9			54 C	0.1 m/s	
	10			57 C	0.2 m/s	
	11			60 C	0.3 m/s	
	12			63 C	0.4 m/s	
B		Rectangular	L/H = 1/0.8	45 C	0.2 m/s	3*1480
				48 C	0.3 m/s	
				51 C	0.4 m/s	
C		Rectangular	L/H = 0.8/1	45 C	0.2 m/s	3*1480
				48 C	0.3 m/s	
				51 C	0.4 m/s	

* For each case, one HF simulation is performed by Star CCM+, four LF simulations are performed by GOTHIC with different coarse meshes (1/10, 1/15, 1/25, 1/30 m). Each case in A generates 1850 data points, while each case in B and C generates 1480 data points.

As shown in Table 18, test 1~3 are designed to investigate how the training data size affects the predictive capability of this data-driven approach. By using the dimensionality reduction technique t-SNE (t-Distributed Stochastic Neighbor Embedding) method, the physics coverage condition of the target case can be visualized, as shown in Figure 45. It is obvious that the data

points of "rectangular" case are covered or overlapped by the training data points in "square" cases, even though globally, testing dataset is an extrapolation of geometry to training dataset. The physics coverage condition of target case is determined as GELI condition.

Table 18. Physics Coverage Conditions in Extrapolation of Geometry Case Study

Test NO.	Training Dataset	Testing Dataset	Physics Coverage Condition	Global Physics	Local Physics
1	A (1 ~ 12)	B+C	GELI	Geometry (Aspect Ratio)	Physical Feature Group
2	A (4 ~ 10)	B+C			
3	A (6 ~ 8)	B+C			
4	B+C	A (6 ~ 8)	GILI		

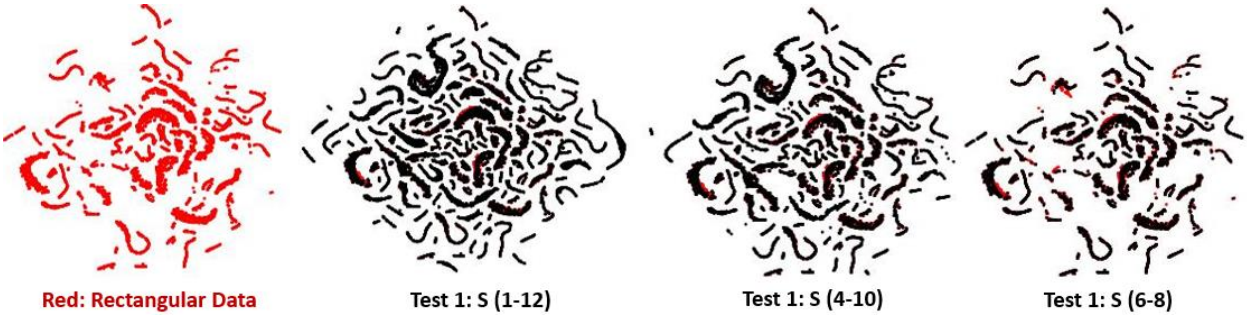
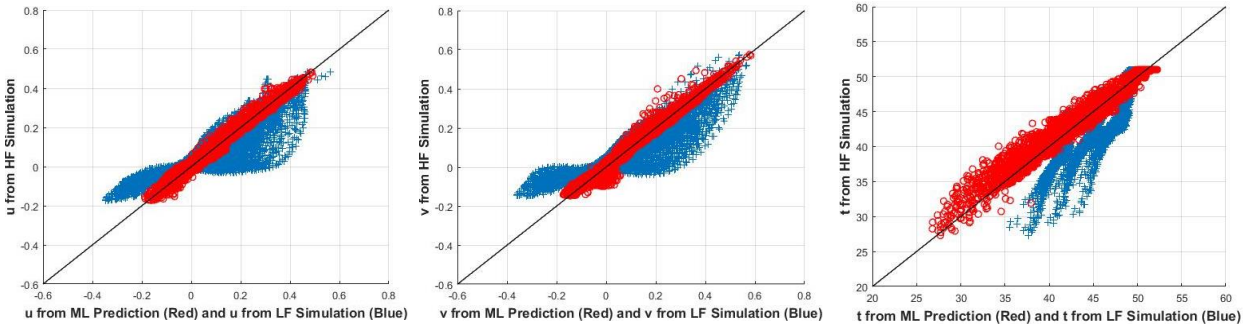


Figure 45. Physical Feature Coverage of "Rectangular" Cases in "Square" Cases

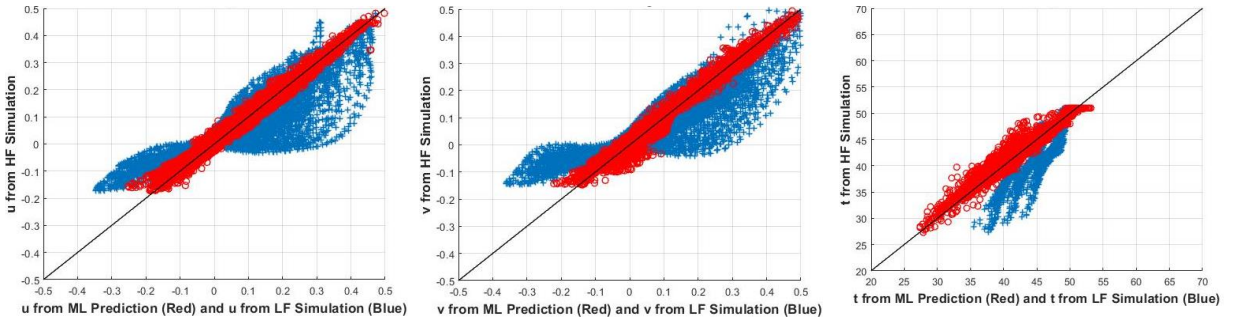
Table 19. Tests in Extrapolation of Geometry Case Study

Test NO.	Training Dataset	Testing Dataset	Physics Coverage Condition	NRMSE (u)	NRMSE (v)	NRMSE (T)	Mean of KDE Distance
1	A (1 ~ 12)	B+C	GELI	0.2712	0.3558	0.0223	0.3190
2	A (4 ~ 10)	B+C		0.2640	0.3514	0.0243	0.2894
3	A (6 ~ 8)	B+C		0.0278	0.0287	0.0022	0.2773
4	B+C	A (6 ~ 8)	GILI	0.0151	0.0140	0.0009	0.2687



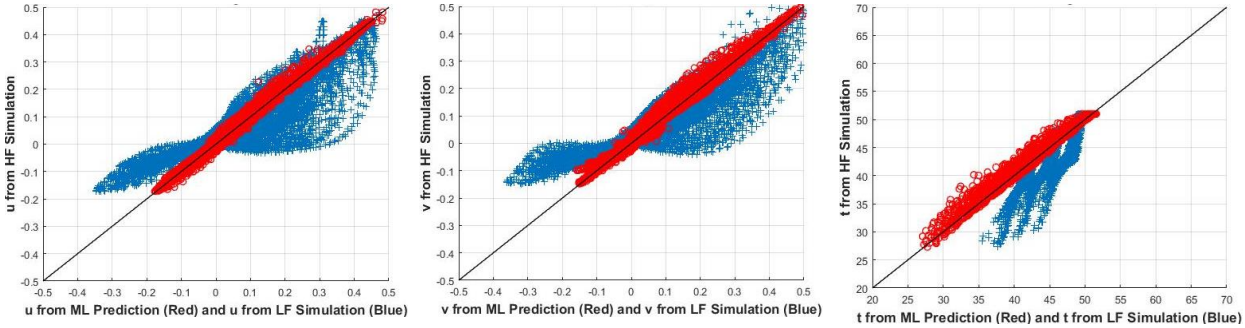
(a) Horizontal Velocity (b) Vertical Velocity (c) Temperature

Test 1: Training Data A (1 ~ 12)



(d) Horizontal Velocity (e) Vertical Velocity (f) Temperature

Test 2: Training Data A (4 ~ 10)



(g) Horizontal Velocity (h) Vertical Velocity (i) Temperature

Test 3: Training Data A (6 ~ 8)

Figure 46. Comparisons between Original GOTHIC Simulation Results and Corrected Results based on OMIS Prediction of Test 1 to Test 3

The values of mean of KDE distance are also listed in Table 19. It shows that when the mean of KDE distance decreases, the prediction accuracy increases even if the training data size decreases. Higher mean of KDE distance represents less similarity of training data and testing data. It implies that the similarity between training data and testing data should be considered in the first place to construct the training database. The application of ML algorithm requires the training

database should include data points at a certain scale, however, the similarity or relevance of data should not be ignored. Adding too much dissimilar or irrelevant data may "hurt" the training and predictive capability of ML algorithms. The comparisons between original GOTHIC simulation results and corrected results based on OMIS prediction of test 1 to test 3 are shown in Figure 46, where blue points represents original GOTHIC simulation results and red circles are corrected results based on OMIS prediction. The corrected results based on OMIS prediction tends to be closer to HF data as the similarity of training data and testing data increases.

Test 4 is identified as a situation in GILI condition since the square cavity can be considered as an interpolation of these two rectangular cavities with aspect ratio respectively equaling to 1/0.8 and 0.8/1. By performing data training using same FNN structure and initial hyper parameters, the prediction errors (NRMSEs) are listed in Table 19, which is much smaller than the tests in GELI condition. The comparisons between original GOTHIC simulation results and corrected results based on OMIS prediction of test 4 are shown in Figure 47, where blue points represent original GOTHIC simulation results and red circles are corrected results based on OMIS prediction. The corrected results based on OMIS prediction tends to be closer to HF data than the tests in GELI condition. It meets the expectation that OMIS approach has good predictive capability in GILI condition.

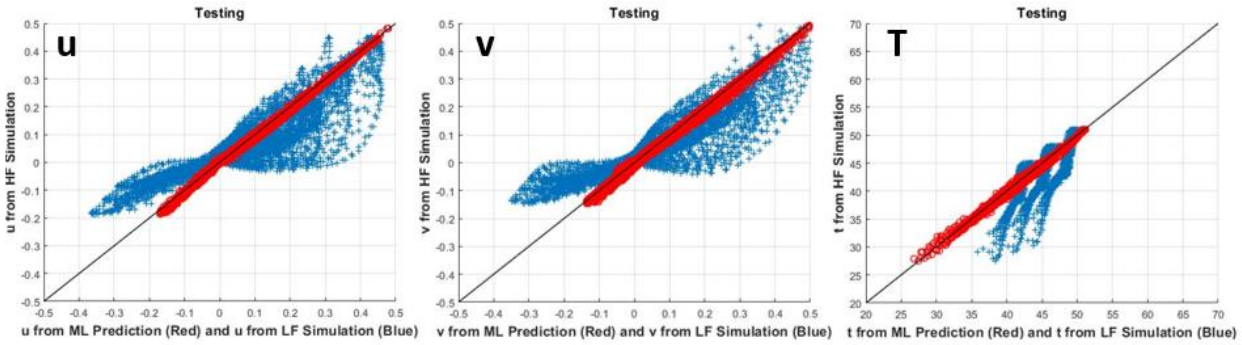
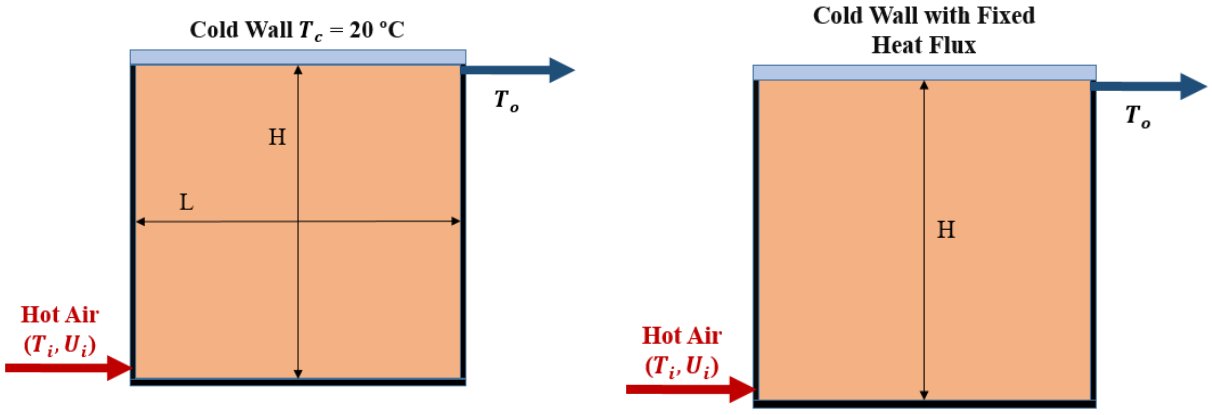


Figure 47. Comparisons between Original GOTHIC Simulation Results and Corrected Results based on OMIS Prediction of Test 4

6.3.2. Extrapolation of Boundary Condition

In this case, two cavities with different boundary conditions are modeled, as shown in Figure 48. The boundary and injection conditions are listed in Table 20. Dataset A contains the case 1-12 in the previous case study discussed in Section 6.2. Dataset D, E, F and G respectively

contain the cases with fixed uniform heat flux 100, 120, 150 and 200 W/m^2 on top wall. The averaged heat removal flux from top wall in the cases of Dataset A are listed in Table 20.



(a) Cold Top Wall with Fixed T = 20 C (b) Cold Top Wall with Fixed Heat Flux

Figure 48. Two Cavity Models with Different Boundary Conditions

Table 20. Boundary and Injection Conditions of Datasets in Extrapolation of Boundary Condition Case Study

Dataset		Boundary Condition		Injection Temperature	Injection Rate	Data Size
A	1	Cold Top Wall with Fixed T = 20 C	26.6 W/m^2	30 C	0.1 m/s	12*1850
	2		55.0 W/m^2	33 C	0.2 m/s	
	3		88.7 W/m^2	36 C	0.3 m/s	
	4		128.9 W/m^2	39 C	0.4 m/s	
	5		57.6 W/m^2	42 C	0.1 m/s	
	6		104.6 W/m^2	45 C	0.2 m/s	
	7		153.4 W/m^2	48 C	0.3 m/s	
	8		207.8 W/m^2	51 C	0.4 m/s	
	9		87.8 W/m^2	54 C	0.1 m/s	
	10		153.1 W/m^2	57 C	0.2 m/s	
	11		216.7 W/m^2	60 C	0.3 m/s	
	12		284.8 W/m^2	63 C	0.4 m/s	
D		Cold Top Wall with Fixed Heat Flux	100 W/m^2	48 C	0.3 m/s	4*1850
E			120 W/m^2			
F			150 W/m^2			
G			200 W/m^2			

* For each case, one HF simulation is performed by Star CCM+, four LF simulations are performed by GOTHIC with coarse meshes (1/10, 1/15, 1/25, 1/30 m). Each case generates 1850 data points.

The comparisons between original GOTHIC simulation results and corrected results based on OMIS prediction with Dataset E as testing case are shown in Figure 49, where blue points represents original GOTHIC simulation results and red circles are corrected results based on OMIS prediction. OMIS approach presents great predictive capability on velocities, but some predictions on temperature are not as good as others. Predicted temperatures in LF simulation and ML prediction are both higher than the values in HF data.

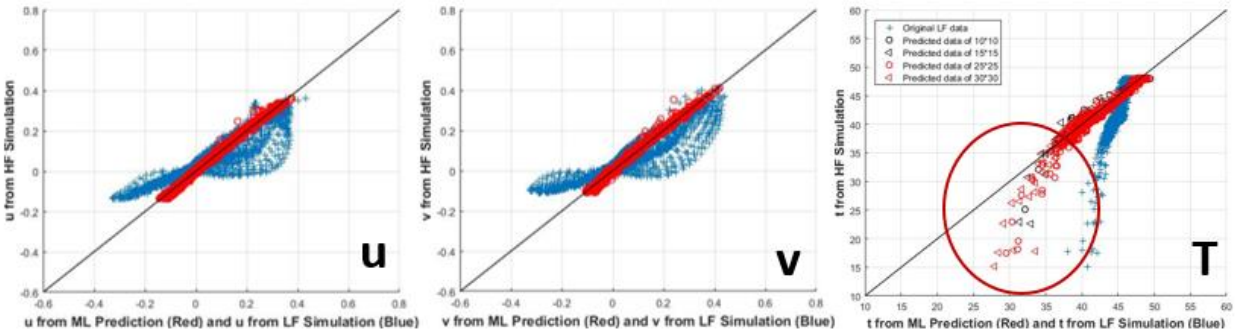


Figure 49. Comparisons between Original GOTHIC Simulation Results and Corrected Results based on OMIS Prediction with Dataset E as Testing Case

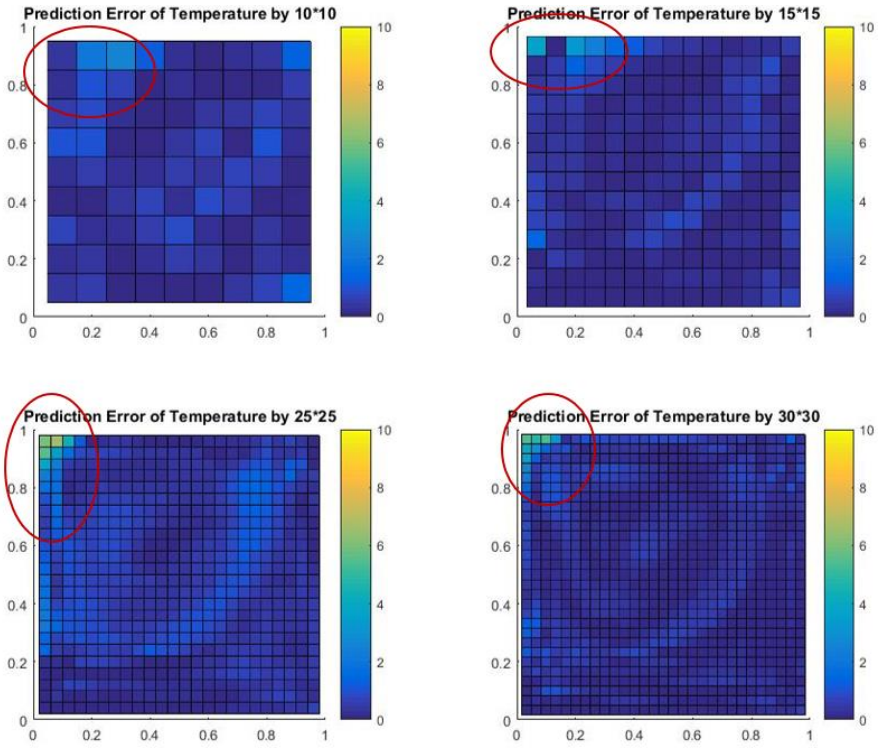
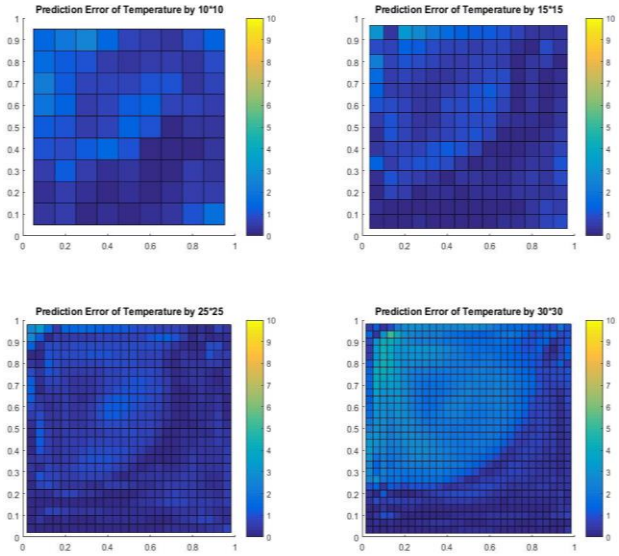
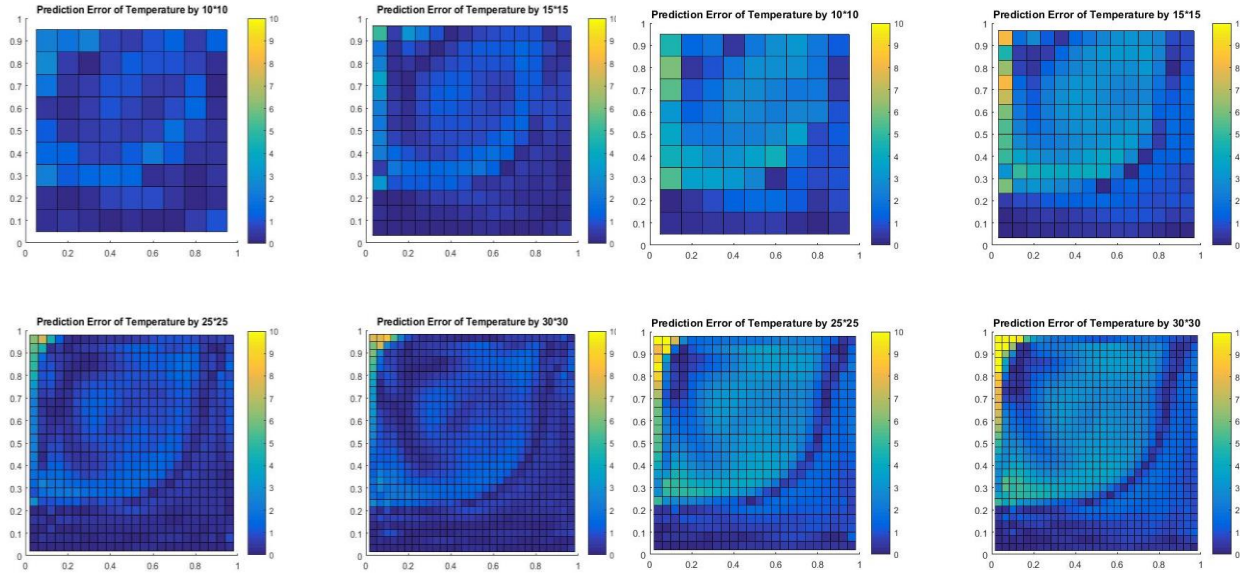


Figure 50. Distribution of Prediction Error of Temperature using Different Mesh Sizes with Heat Flux Equal to 120 W/m^2



(a) 100 W/m^2



(b) 150 W/m^2

(c) 200 W/m^2

Figure 51. Distribution of Prediction Error of Temperature using Different Mesh Sizes with Heat Flux Equal to (a) 100 W/m^2 (b) 150 W/m^2 (c) 200 W/m^2

To find out the locations of these "bad" prediction, the distribution of prediction error of temperature is plotted in Figure 50. It shows that these "bad" predictions mainly locate on the left top part of the cavity, which have been marked in red circles, no matter which mesh size is applied. Heat transfer is a little bit underestimated in LF simulation and ML prediction. Compared with training cases, the heat removal in this region is higher in testing cases because a fixed heat flux is

required. For example in case A4, although the averaged heat flux is 128.9 W/m^2 and similar to case E, the real heat flux at the left top part is much smaller than 128.9 W/m^2. The heat flux is not uniform along the top wall. This underlying physics is learned by the well-trained FNN and reflected on the prediction. Therefore, this well-trained FNN estimates a relatively high temperature for the case where the real temperature is low since the heat flux is higher than what FNN expected. This sort of "wrong" learning in the left top part also reflects in the predictions on Dataset D, F and G, which are shown in Figure 51. The NRMSE of predictions are compared for these four tests, as listed in Table 21. Smaller mean of KDE distance implies smaller NRMSEs and better predictions.

Table 21. Prediction Results of the Extrapolation of Boundary Condition Case Study

Test NO.	Training Dataset	Testing Dataset	Physics Coverage Condition	NRMSE (u)	NRMSE (v)	NRMSE (T)	Mean of KDE Distance
1	A	D (100W/m^2)	GELI	0.254	0.297	0.034	0.2466
2		E (120W/m^2)		0.156	0.202	0.026	0.2444
3		F (150W/m^2)		0.280	0.305	0.043	0.2493
4		G (200W/m^2)		0.623	0.658	0.087	0.2566

6.3.3. Extrapolation of Dimension

In this case, cavities with different dimensions are modeled. The boundary and injection conditions are listed in

Table 22. Dataset A contains the case 1-12 in the previous case study discussed in Section 6.2. Dataset H, I, J and K respectively contain the cases with length equal to 1.2m, 1.5m, 2m and 5m. Same nodalization are applied for all the simulations: 10*10, 15*15, 25*25 and 30*30. Dataset A is applied as training data for all the tests in this section, and other datasets are set as testing data.

The comparisons between original GOTHIC simulation results and corrected results based on OMIS prediction with Dataset E as testing case are shown in Figure 52, where blue points represents original GOTHIC simulation results and red circles are corrected results based on OMIS prediction. OMIS approach presents good predictive capability on velocities and temperature compared with LF simulations.

Table 22. Boundary and Injection Conditions of Datasets in Extrapolation of Boundary Condition Case Study

Dataset		Dimension (Height & Length)	Injection Temperature	Injection Rate	Data Size
A	1	1m*1m	30 C	0.1 m/s	12*1850
	2		33 C	0.2 m/s	
	3		36 C	0.3 m/s	
	4		39 C	0.4 m/s	
	5		42 C	0.1 m/s	
	6		45 C	0.2 m/s	
	7		48 C	0.3 m/s	
	8		51 C	0.4 m/s	
	9		54 C	0.1 m/s	
	10		57 C	0.2 m/s	
	11		60 C	0.3 m/s	
	12		63 C	0.4 m/s	
H		1.2m*1.2m	48 C	0.3 m/s	4*900
I		1.5m*1.5m			
J		2m*2m			
K		5m*5m			

* For each case in A, one HF simulation is performed by Star CCM+, four LF simulations are performed by GOTHIC with different coarse meshes (1/10, 1/15, 1/25, 1/30 m). For each case in H~K, one HF simulation is performed by Star CCM+, one LF simulation is performed by GOTHIC with same nodalization (30*30). The coarse mesh size used in case H~K are respectively 1.2/30, 1.5/30, 2/30 and 5/30 m.

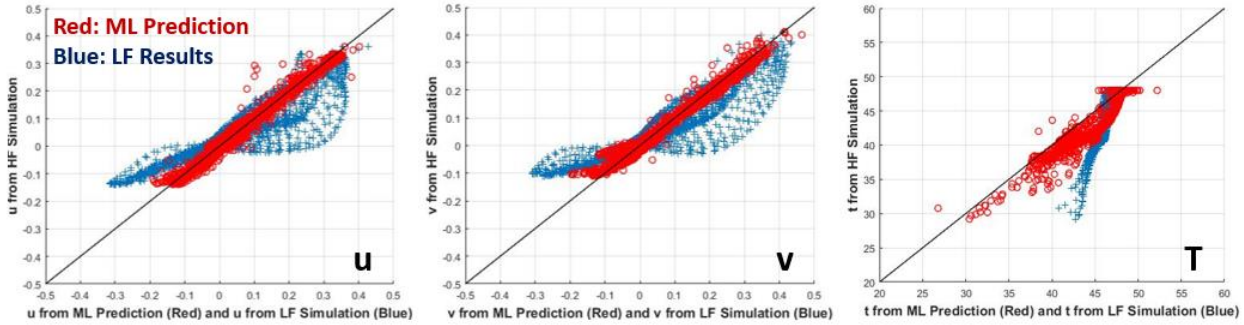


Figure 52. Comparisons between Original GOTHIC Simulation Results and Corrected Results based on OMIS Prediction with Dataset H as Testing Case

The NRMSEs of predictions are compared for these four tests, as listed in Table 23. Smaller mean of KDE distance implies smaller NRMSEs and better predictions. This feature is also

illustrated in the comparison of Physical Feature Coverages (PFCs) in Figure 53, where red points represent testing data and black points represent training data. The values of NRMSEs are much higher than other case studies since the physics coverage is much less, which can be observed in Figure 53. For the test with Dataset H as testing data, testing data is almost fully covered by training data. For other tests, testing data is rarely covered, which can be considered as GELE condition. Both of global physics (dimension) and local physics (physical features) are extrapolative. This explains why OMIS approach does not present good predictive capability in this case study.

Table 23. Prediction Results of the Extrapolation of Boundary Condition Case Study

Test NO.	Training Dataset	Testing Dataset	Physics Coverage Condition	NRMSE (u)	NRMSE (v)	NRMSE (T)	Mean of KDE Distance
1	A	H (1.2 m)	GELI	0.291	0.378	0.043	0.2039
2		I (1.5 m)	GELE	0.552	0.785	0.083	0.2819
3		J (2 m)		0.803	1.163	0.123	0.3059
4		K (5 m)		1.106	2.852	0.106	0.3373

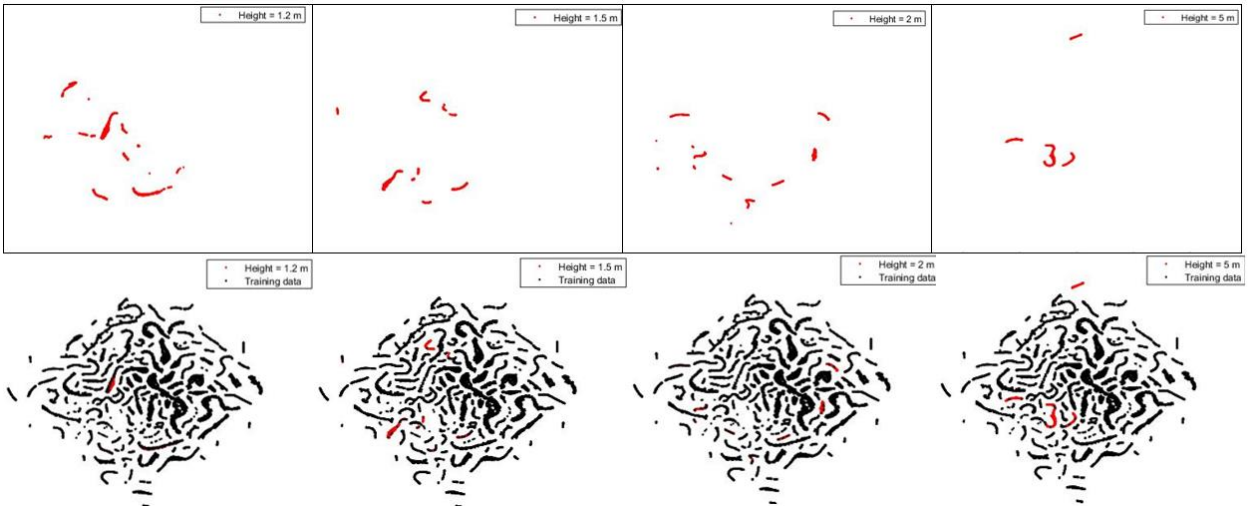


Figure 53. Physical Feature Coverages in Extrapolation of Boundary Condition Case Study

6.3.4. Lessons Learned

This section has discussed the applications of OMIS approach in GELI, GILI and GELE conditions. Several lessons can be learned from this section:

(Section 6.3.1) **In GELI condition, smaller KDE distance means higher similarity, and smaller prediction error**. The corrected results based on OMIS prediction tends to be closer to HF data as the similarity of training data and testing data increases. Similarity of training data

should be considered in the first place to construct the training database. The application of ML algorithm requires the training database should include data points at a certain scale, however, the similarity or relevance of data should not be ignored. Adding too much dissimilar or irrelevant data may "hurt" the training and predictive capability of ML algorithms.

(Section 6.3.2) Essential difference in physics between training cases and testing cases should be sufficiently considered and reflected in the identification of physical features. For example in the extrapolation of boundary condition case study, the feature of non-uniform heat flux distribution is learned by the well-trained FNN and reflected on the prediction. This leads to a "bad" prediction when this "uniform" feature is totally "wrong" in the testing case. Actually, the reason can be considered as the insufficiency of training database. The feature in testing case is an unknown to the training cases. No matter how much new data we collect from these training cases, the prediction cannot be improved.

(Section 6.3.3) OMIS approach shows great predictive capability in GILI and GELI conditions, but not in GELE condition. This satisfies the hypothesis proposed in framework formulation. In the extrapolation of dimension case study, Dataset H is almost fully covered by training data. For other tests, testing data is rarely covered, which can be considered as GELE condition. Both of global physics (dimension) and local physics (physical features) are extrapolative. This explains why OMIS approach does not present good predictive capability in this case study. The reason may be that the mesh size differs too much, which greatly affects the values and coverage of physical features.

6.4. Chapter Summary

This chapter illustrates OMIS framework with the case study on mixed convection. Targeting on the "GELI" condition, OMIS framework is developed as a TDMI approach that deals with data, physical model and coarse-mesh simulation in an integrated manner using machine learning algorithms. By concentrating on the similarity of local physics, OMIS framework has a potential scalability to the globally extrapolative conditions. The underlying local physics of one specific physic condition is assumed to be represented by a set of Physical Features (PFs). Section 6.2 has illustrated the OMIS framework on the mixed convection case study. The two outcomes of the proposed framework, error prediction and optimal mesh suggestion, present the good

predictive performance of this data-driven framework. Several lessons are learned from this case study:

1. The proposed framework can be divided into three parts: preliminary evaluation, optimization and application. The first part makes efforts on the development of predictive capability: identifying PFs, build database and evaluate predictive capability on test matrix. The second part focuses on the optimization of the predictive capability by optimizing PF group, FNN structure and training database. After the execution of each optimization step, the predictive capability should be assessed again on the test matrix. The third part is to apply this optimized data-driven predictive capability on the target case to provide error prediction and optimal mesh/model suggestion. All the three parts are tightly organized but the techniques applied in each step are independent and non-instructive to other steps. This makes OMIS framework improvable when more advanced techniques or algorithms are involved and also feasible for other codes.

2. The step of PIRT and physics decomposition before building database and performing data training is indispensable since the identification of PFs only depends on these involved physics, respective closure models and mesh sizes. It makes this data-driven framework informed by the knowledge base of physics. This case study shows that the local physical parameters has the equivalent importance as the variable gradients.

3. The objective of optimization steps is to make a balance between accuracy and efficiency for the system-level thermal hydraulic modeling and simulation of multi-component, multi-physics and multi-scale nuclear power plants. The evaluation metrics used in these steps are MSE of variables and computational cost for FNN training. By reducing the dimensionality of PF group, investigating FNN performance and similarity of training database, computation cost is saved while the predictive accuracy is still maintained in an acceptable level.

Section 6.3 respectively discusses the OMIS application in GELI condition in extrapolation of geometry (aspect ratio), boundary condition and dimension. Smaller mean of KDE distance implies smaller Normalized Root Mean Squared Errors (NRMSEs) and better predictions. There is a positive relationship between extrapolative distance and prediction error. The metrics for extrapolative distance and prediction error are respectively mean of KDE distance and NRMSEs of the efforts in this section.

Similarity of training data should be considered in the first place to construct the training database. The application of ML algorithm requires the training database should include data points at a certain scale, however, the similarity or relevance of data should not be ignored. Adding too much dissimilar or irrelevant data may "hurt" the training and predictive capability of ML algorithms. These case studies also prove the importance of data sufficiency in the application of OMIS approach: sufficient data are required for the data-driven model to collect enough information and capture the underlying physics. Finally, OMIS approach shows great predictive capability in GILI and GELI conditions, but not in GELE condition.

CHAPTER 7. INTEGRATION OF PROPOSED DATA-DRIVEN FRAMEWORK WITH EVALUATION MODEL DEVELOPMENT AND ASSESSMENT PROCESS

7.1. Introduction

This chapter represents the effort to integrate the proposed OMIS (Optimal Mesh/Model Information System) framework and EMDAP (Evaluation Model Development and Assessment Process). In 2005, USNRC issued an important document regulatory guide 1.203 to provide an acceptable Evaluation Model Development and Assessment Process (EMDAP) for the best estimate calculations of NPP transient and accident analysis. [10] EMDAP aimed to evaluate the adequacy of the applied codes and provide guidance for the following experiment and analytical tool development. However, the system analysis and scaling analysis in EMDAP are highly heuristic and difficult to implement on codes, and the mesh effect on code/model scalability was not fully considered. Based on the concept of Total Data-Model Integration (TDMI), the proposed OMIS framework treats multi-scale data, key closure models and numerical simulation in an integrated manner to develop an integrated data-driven model that bridge the scale gap. The data-driven OMIS framework has a potential to be a supplement to make the implement of EMDAP feasible and practical. The elements of EMDAP and relevant scaling analysis are reviewed in Section 7.2, and the role of OMIS framework in EMDAP architecture is discussed in Section 7.3.

7.2. Overview of Evaluation Model Development and Assessment Process (EMDAP)

The procedure of EMDAP are illustrated in the left part of Figure 54. [10] The basic principles of EMDAP were developed based on the Code Scaling and Applicability Uncertainty (CSAU) methodology, while EMDAP has formal and explicit descriptions for the concepts, definitions and processes, including the PIRT (Phenomena Identification and Ranking Table), assessment base, evaluation model, scaling analysis.

EMDAP includes four major elements. Element 1 aims to establish requirements for evaluation model capability. The exact application envelope for the Evaluation Model (EM) and constituent phenomena, process and key parameters within this envelope are determined and identified at the beginning. Specifying analysis purpose and target is important since the statement of purpose influences the entire process of development, assessment and analysis. Figures of Merit (FOMs) are defined as those quantitative standards of acceptance that are used to define acceptable

answers for a safety analysis. Systems, components, phases, geometries, fields, and processes that much be modeled are identified for the development of PIRT. It should be noted that a figure of merit other than the applicable acceptance criterion is more appropriate as a standard for identifying and ranking phenomena. The development of PIRT heavily relies on expert opinion which can be subjective and expensive. Therefore, it is important to validate the PIRT using experimentation and analysis. For the application of OMIS framework, the content in Element 1 of EMDAP is also necessary and considered as the input for Step 1 of OMIS framework.

The purpose of Element 2 is to provide the basis for development and assessment of EM, especially the experimental database and its scale-up capability. One distinctive feature of EMDAP is the high attention on scalability analysis on data. Because all experiments are compromised with full-scale plant systems, scaling analysis should be conducted to ensure the applicability of the data and models on analysis of full-scale plant transient. In EMDAP, scaling analysis employ both top-down and bottom-up approaches. The top-down approach evaluates the global system behavior and interactions from Integral Effect Test (IET) facilities. By deriving non-dimensional groups that govern similarity between facilities in different scales, the top-down scaling approach assumes that these groups have the scalability on the results among these facilities. The bottom-up scaling approach mainly relies on Separate Effect Tests (SETs) or small IETs and addresses the scaling issues in localized plant behavior or processes.

However, in most applications where a large number of processes and phenomena are involved, it is difficult to design test facilities that preserve sufficient similarity between experiment and full-scale plant. These physics-based non-dimensional groups are not able to fully represent the underlying similarity. No matter in Element 2 or Element 4, EMDAP does not provide a detailed guide to help on the development these non-dimensional groups and identification of the optimum similarity criteria. Therefore, although huge amounts of experimental data have been generated and collected, the ghost of scaling distortion still haunts around. Here is where OMIS framework can be applied as a supplement to bridge the scale gap.

Element 3 focuses on the development of EM. EM is a collection of calculation tools (codes and procedures) developed and organized to meet the requirements established in Element 1. Information from the scaling activity (in Element 2) is fed into the EM development activity: scaling analysis are performed to demonstrate the relevancy and sufficiency of the collective

database for representing the expected behaviors and to investigate the scalability of EM for representing the important phenomena. The scaling uncertainty due to the extrapolation of non-dimensional group (which may locate in GELI or GELE condition) is difficult to quantify, even nominally full-scale experiments do not ensure a high similarity between the experiments and plant. The performance of EM in this element highly relies on the scalability of the collected data in Element 2.

EM adequacy is assessed in Element 4. Similar to the scaling approaches applied in Element 2, the EM assessment is divided into two parts. The first part (Step 13 ~ 15) focuses on the bottom-up evaluation of closure models and correlations by considering their applicability, fidelity, and scalability. The second part (Step 16 ~ 19) contributes to the top-down evaluation of governing equations, the integrated performance of each code and the integrated performance of the entire EM based on data from IETs. Compared to the first part, the second part mainly focuses on the integrated capability and performance of the EM. The first part is clearly described to implement since the target closure models mostly contribute to single phenomenon or process. In contrast, the second part is difficult to perform due to the complexity of involved physics and lack of sufficient validation data. In Step 19, the need to assess the scalability of integrated calculations and data for scaling distortion is proposed, however, it is not clearly explained how to implement this scalability assessment.

Besides, the importance of nodalization and determination of mesh was not fully considered in the preparation of calculation input in Step 18. As discussed before, as one of the key model parameters, the effect of mesh size on the model/code performance should be fully considered since it also greatly affects the scalability assessment in Step 19 and uncertainty analysis in Step 20. Furthermore, the extrapolation application of model/code is not mentioned or well guided in the scalability assessment. The classification of Physics Coverage Condition (PMC) and definition of Physical Feature Coverage (PFC) may open a door for these extrapolation conditions. It also should be noted that the model consistency may occur when both of these two part are performed. The integration or interactions among the well-evaluated closure models in first part may lead to different results with the data from IETs. A data-driven Validation and Uncertainty Quantification (VUQ) framework [16] has been proposed to identify the model consistency by introducing the concept of Total Data-Model Integration (TDMI).

After the logical and comprehensive validation, the decision process was executed to evaluate whether the code meets the adequacy standard and can be used for plant scenario analysis. However, the acceptance criteria were not explicitly defined.

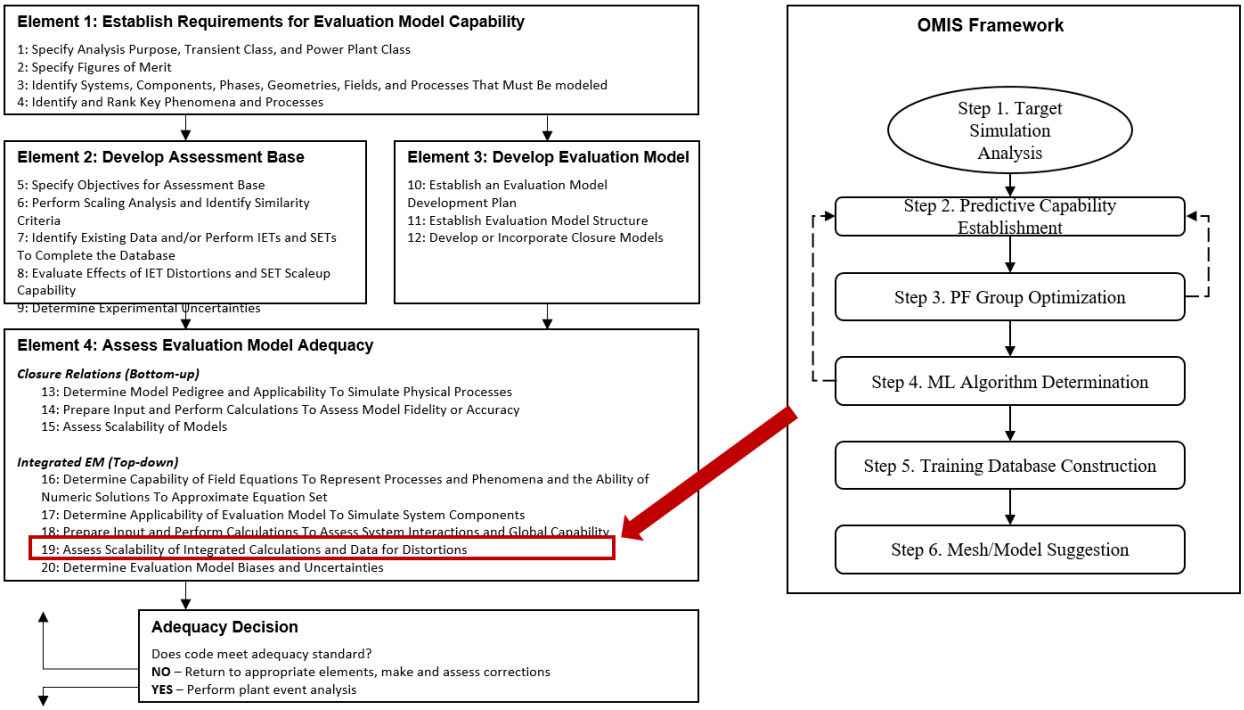


Figure 54. Where OMIS Framework Supplements EMDAP

In a word, EMDAP is heuristic and difficult to implement even if it has formal and explicit descriptions for the concepts, definitions and processes. Especially the assessment on scaling approaches and scalability assessment are not distinctly defined and explained. Besides, the mesh effect on code/model scalability and uncertainty analysis was not fully considered. Finally, the acceptance criteria were not clearly defined. The execution of EMDAP needs more the state of the art techniques, scaling approaches and frameworks, and decision-making systems to supplement and support. By treating mesh error and model error together and introducing machine learning algorithms to explore the local physics, OMIS framework has the potential to bridge the scale gap and work as a supplement to the implementation of EMDAP Step 19 in the assessment of integrated scalability, as shown in Figure 54.

7.3. OMIS: A Potential Data-driven Supplement for Scalability Assessment in EMDAP

This section proposes some thoughts on the integration of OMIS framework and EMDAP. As discussed in previous section, there are two main cons of EMDAP where OMIS framework

can support and supplement: (1) insufficient quantification of mesh effects on model/code scalability and uncertainty analysis; (2) obscure assessment on the scalability of model/code in integrated calculations.

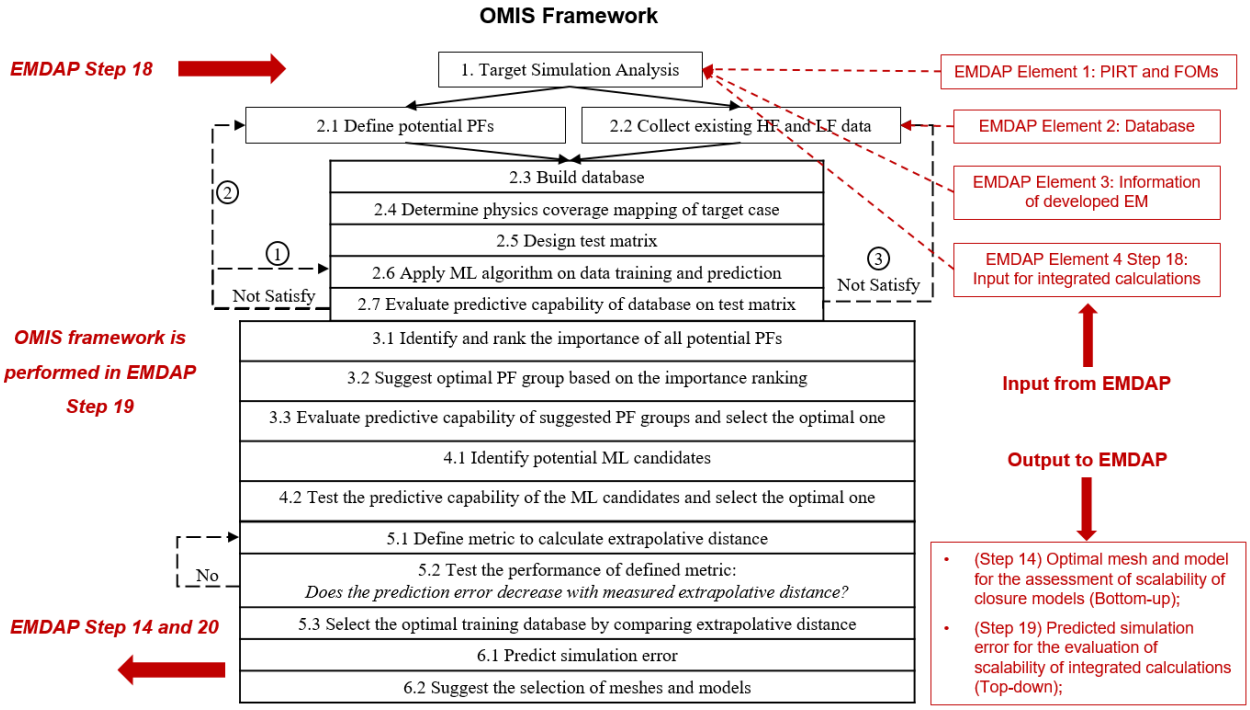


Figure 55. Illustration of the Integration of OMIS Framework and EMDAP

The integration, information exchange and main outcomes between OMIS framework and EMDAP are illustrated in Figure 55. The OMIS framework follows Step 18 of EMDAP, where the input for integrated calculations have been prepared. Then OMIS framework obtains the information from four sources in EMDAP:

1. OMIS Step 1 requires to identify the key phenomena, global QoIs and FOMs, which are provided by the PIRT process in EMDAP Element 1;

2. OMIS Step 1 requires to evaluate the applicability of closure models for the key phenomena in the simulation tool. The model/code information can be provided by EMDAP Element 3, where EM is developed;

3. OMIS Step 1 requires input information for the integrated calculations, which can be provided by Element 4 Step 18;

4. OMIS Step 2 requires to collect HF and LF data, which can be provided by EMDAP Element 2 where database is built.

The OMIS framework provides two main outcomes back to EMDAP:

1. (To EMDAP Step 14) OMIS framework provides the suggestion on optimal mesh and model selections. The scalability of these closure models need to be evaluated using bottom-up approach, where EMDAP Step 14-15 should be repeated using the suggested mesh;

2. (To EMDAP Step 19) OMIS framework provides prediction on the simulation error of QoIs, which can be used to assess the scalability of the model/code in the integrated calculations.

Besides, the biases and uncertainty" in EMDAP Step 20 (determine EM biases and uncertainties) includes model uncertainty/error, mesh error and other numerical uncertainties with the consideration of scaling effect. This "uncertainty" has the same content as the concept of "simulation error" in OMIS framework.

7.4. Chapter Summary

In this chapter, the integration of the proposed OMIS framework and EMDAP has been described and explained. EMDAP aimed to evaluate the adequacy of the applied codes and provide guidance for the following experiment and analytical tool development. However, EMDAP is heuristic and difficult to implement even if it has formal and explicit descriptions for the concepts, definitions and processes. Especially the assessment on scaling approaches and scalability assessment are not distinctly defined and explained. Besides, the mesh effect on code/model scalability and uncertainty analysis was not fully considered. Finally, the acceptance criteria were not clearly defined. The execution of EMDAP needs more the state of the art techniques, scaling approaches and frameworks, and decision-making systems to supplement and support. By treating mesh error and model error together and introducing machine learning algorithms to explore the local physics, OMIS framework has the potential to bridge the scale gap and work as a supplement to the implementation of EMDAP in the assessment of integrated scalability.

The main inputs from EMDAP to OMIS framework are (1) information of the key phenomena, global QoIs and FOMs, which are provided by the PIRT process in EMDAP Element 1; (2) information of closure models for the key phenomena in the simulation tool, which are

provided by EMDAP Element 3; (3) input information for the integrated calculations, which are provided by Element 4 Step 18; (4) HF and LF data, which are provided by EMDAP Element 2.

The main outcomes from OMIS framework to EMDAP are (1) suggestion on optimal mesh and model selections. The scalability of these closure models need to be evaluated using bottom-up approach, where EMDAP Step 14-15 should be repeated using the suggested mesh; (2) prediction on the simulation error of QoIs, which can be used to assess the scalability of the model/code in the integrated calculations in EMDAP Step 19 and also the uncertainty analysis in Step 20.

CHAPTER 8. CONCLUSIONS

This work is initially motivated by the high demand on fast-running and sufficiently accurate simulation tools for system-level thermal-hydraulic modeling and simulation. A data-driven framework is developed and demonstrated to improve the coarse-mesh CFD-like codes by predicting the simulation error and suggesting the optimal mesh size and closure models. Since mesh size is one of the key model parameters in the simplified boundary-layer closure models of these CFD-like codes, the two main error sources, model error and mesh error cannot be quantified separately. This makes it difficult to perform traditional Verification and Validation (V&V) on these coarse-mesh codes. Meanwhile, a huge amount of simulation data has been generated using these fast-running codes, which makes it possible to apply advanced statistical techniques and Machine Learning (ML) algorithms to learn from multiscale data and explore the underlying physics. This proposed data-driven methodology takes benefits from the increasing computational power and rapid development of ML techniques to integrate the data, physical models and numerical simulation together. Another motivation is the issues raised from scaling distortion. The scaling-induced uncertainty and lack of validation data hamper the credibility of system-level simulation that supports risk-informed analysis of safety transient scenarios in novel reactor systems. In addition to improve the coarse-mesh simulations, this work also provides an insight on the development of a data-driven scale-invariant approach to deal with scaling issues and provide evidence for the generation of validation data.

In this chapter, the proposed framework and case studies are summarized in Section 8.1. Section 8.2 highlights the contributions and Section 8.3 outlines the future works.

8.1. Summary Remarks

In this work, a data-driven framework was proposed, developed and demonstrated to improve the coarse-mesh CFD-like codes by predicting the simulation error and suggesting the optimal mesh size and closure models.

Firstly, the pros and cons of current traditional and data-driven V&V frameworks have been discussed if applied to the coarse-mesh CFD-like codes for system thermal-hydraulic modeling and simulation. Considering it is a cross-disciplinary work, the required knowledge and efforts from multidisciplinary fields including system thermal-hydraulic modeling and simulation, V&V, machine learning are also reviewed and analyzed. The scope of this work is described from

three aspects: (1) provide a potential data-driven approach for the validation of these CFD-like codes in system-level thermal-hydraulic modeling and simulations; (2) develop a data-driven scale-invariant approach to deal with scaling issues and provide evidence for the generation of validation data; (3) provide a supplement for the execution of Evaluation Model Development and Assessment Process (EMDAP).

Secondly, this work investigates sufficient technical capabilities and proposes a data-driven optimization framework. The central idea is to develop a surrogate model to represent the relationship between local simulation error and specific local Physical Features (PFs). The identification of PFs integrates the physical information of the system of interest, model information and the effect of mesh size. The main outcomes of OMIS framework are error prediction and suggestion on the optimal mesh and model selection using machine learning algorithms. OMIS framework is accomplished via a systematic procedure, the sub-outcomes include: (1) PF group is identified based on knowledge basis and has the extendibility from single phenomenon to complex physics; (2) scalability of identified PF group is pre-evaluated via test matrix and optimized by importance study before application; (3) different DNN structures are tested and compared to balance the prediction accuracy and computational cost; (4) data similarity of training data and testing data is measured using KDE distance and visualized in Physical Feature Coverage (PFC) using dimensionality reduction techniques, this provides a guide on the selection of training datasets. These outcomes not only serve on the error prediction and mesh/model selection, but also provide an insight on how to develop, evaluate and optimize a data-driven surrogate model in thermal-hydraulic modeling and simulation.

The proposed framework can be divided into three parts: preliminary evaluation, optimization and application. The first part makes efforts on the development of predictive capability: identifying PFs, build database and evaluate predictive capability on test matrix. The second part focuses on the optimization of the predictive capability by optimizing PF group, FNN structure and training database. After the execution of each optimization step, the predictive capability should be assessed again on the test matrix. The third part is to apply this optimized data-driven predictive capability on the target case to provide error prediction and optimal mesh/model suggestion. All the three parts are tightly organized but the techniques applied in each step are independent and non-instructive to other steps. This makes OMIS framework improvable when more advanced techniques or algorithms are involved and also feasible for other codes.

Furthermore, the proposed framework has been illustrated based on the mixed convection case study. The entire framework including preliminary evaluation, optimization and application is followed and executed. Test matrix is designed and conducted to demonstrate how to apply the framework to a system thermal-hydraulic simulation. After the application of the framework on this case study, two objectives: error prediction and optimal mesh/model suggestion are successfully achieved. Besides, the OMIS application is discussed in different GELI conditions: extrapolation of geometry (aspect ratio), boundary condition and dimension. Smaller mean of KDE distance implies better predictions. It is found that there is a positive relationship between extrapolative distance and prediction error. It should be noted that the application of ML algorithm and advanced statistical techniques also introduces the uncertainty which is difficult to be quantified, although the techniques in the state of the art have been well evaluated. The uncertainty from the identification of PFs and insufficiency of training database are also introduced. The uncertainty propagation from one step to the entire framework process should be investigated in future.

Lastly, the integration of OMIS framework and EMDAP is discussed in Chapter 7. The proposed framework provides a supplement to the implementation of Evaluation Model Development and Assessment Process (EMDAP) in depth. The main outcomes from OMIS framework to EMDAP are (1) suggestion on optimal mesh and model selections. The scalability of these closure models need to be evaluated using bottom-up approach in EMDAP; (2) prediction on the simulation error of QoIs, which can be used to assess the scalability of the model/code in the integrated calculations and also the uncertainty analysis in EMDAP.

8.2. Contributions

The contributions of this dissertation focus on:

1. The development and demonstration of a data-driven framework to guide simulation error prediction and optimal mesh/model selection in system-level thermal-hydraulic modeling and simulations. This data-driven framework makes benefits from rapid development of ML techniques, fast-running feature of coarse-mesh CFD-like code and increasing computational power. Traditionally, the simulation error prediction highly relies on verification, validation and uncertainty quantification, which are not suitable for these coarse-mesh CFD-like codes. The selection of optimal mesh and models is mainly determined by expert opinion, which

is not trustworthy if the models or codes are not in the application domain. By learning from massive data instead of human experience, OMIS framework provides a smart guide for user to improve the modeling and simulation. The application of ML algorithms realizes the data-driven concept of OMIS to "learn" information directly from data without assuming a predetermined equation as a model. It implies a tendency that data science and analysis plays an important role in the future analysis of nuclear thermal-hydraulic and safety systems.

2. The development of a potential data-driven framework for the validation of the CFD-like codes in the system-level thermal-hydraulic modeling and simulations. Traditional V&V frameworks analyze model error and mesh error separately with another fixed, the logic of which is impractical to the coarse-mesh CFD-like codes since the mesh size is treated as one of key model parameters and mesh convergence is not expected. To overcome this difficulty in the V&V of CFD-like codes, OMIS framework considers these two main error sources together. OMIS framework treats physical models, coarse mesh sizes and numerical solvers as an integrated model, which can be considered as a surrogate of governing equations and closure correlations. The development of the integrated surrogate model does not need relevant prior knowledge, and purely depends on existing data. In some respects, OMIS framework is expected to provide a potential data-driven approach for the validation of these CFD-like codes in the system-level thermal-hydraulic modeling and simulations. As the response of trained data-driven model, simulation error in each cell is estimated according to the Physical Features (PFs) in each local cell. By introducing the concept of TDMI and various PFs, the prediction of simulation error takes all the error sources into accounts and has a promising accuracy even for extrapolative conditions where validation data is not available. OMIS framework also has a strong flexibility to extend to other codes (e.g., coarse-mesh CFD simulations) where mesh size is treated as one of the key model parameters of closure laws.

3. Providing an insight on the development of a data-driven scale-invariant approach to deal with scaling issues and provide guidance for the generation of validation data. This work proposes a concept of Physics Coverage Condition (PCC), which is trying to classify the physics condition into four different parts. One part called GELI (Global Extrapolation through Local Interpolation) condition indicates the situation that the global physical condition of new case is identified as an extrapolation of existing cases, but the local physics are similar. The underlying local physics is assumed to be represented by a set of Physical Features (PFs). This makes it

possible to bridge the scale gap by exploring the local physics instead of global physics with the usage of advanced deep learning techniques and statistical approaches. The similarity or difference between the training data and testing data are quantified and visualized by defined extrapolative distance and Physical Feature Coverage (PFC). The data similarity is depending on the identification of PFs, data quality and quantity. OMIS framework is proposed to seek a technical basis for the preliminary development and proof-of-concept of the scale-invariant approach for the modeling and simulation of system thermal hydraulics in advanced nuclear reactors.

4. The application of evaluation metrics to quantitatively measure the similarity between training data and testing data. The outcomes of OMIS framework include (1) quantitatively measuring the PF similarity of training data and testing data, and (2) identifying the relationship between these local PFs and local simulation error for future predictions. KDE distance is used as a metric to measure the similarity of training data and testing data and has a positive relationship with prediction error of machine learning. The development of validation data plan can also informed by considering to make up the "uncovered" part of Physical Feature Coverage (PFC) in testing cases. It is expected that the prediction by well-trained data-driven model has higher accuracy as the similarity of training data and testing data increases.

5. A supplement to the execution of Evaluation Model Development and Assessment Process (EMDAP) in depth. EMDAP aimed to evaluate the adequacy of the applied codes and provide guidance for the following experiment and analytical tool development. However, EMDAP is heuristic and difficult to implement even if it has formal and explicit descriptions for the concepts, definitions and processes. Especially the assessment on scaling approaches and scalability assessment are not distinctly defined and explained. Besides, the mesh effect on code/model scalability and uncertainty analysis was not fully considered. By treating mesh error and model error together and introducing machine learning algorithms to explore the local physics, OMIS framework has a potential to bridge the scale gap and work as a supplement to the implementation of EMDAP in the assessment of integrated scalability.

8.3. Future Works

The limitations of the proposed framework exist.

It should be noted that the proposed framework is only demonstrated on a synthetic example in steady state, the involved processes and phenomena are still far from the practical

application in real modeling and simulation of multi-component and multi-physics plant transient. The scalability and predictive capability of OMIS framework still need to be investigated for other physics (e.g., two-phase flow and boiling) with complex geometries (structure, volume size, location/size of injection/vent). Therefore, the first future work is focused on (1) how to extend the current framework to more complex physics and geometry, and (2) how to improve OMIS framework for plant scenario simulation.

Another future work is the uncertainty quantification of OMIS framework. The main uncertainty sources of OMIS framework are mainly (1) ML uncertainty, which depends on the applied ML algorithm itself; (2) insufficiency of training data, which includes the data size, similarity and identification of Physical Features (PFs). The uncertainty introduced by statistical and ML algorithms are not quantified, the relevant uncertainty propagation needs more analysis. Currently, multi-layer FNN is used as the main ML algorithm. The impact of ML uncertainty may decrease when a FNN with more hidden layers and neurons is applied in the future, but the computational cost also increases. Due to its data-driven nature, performance of OMIS framework greatly relies the size and similarity of available data, and the identification of PFs. In current work, the similarity of data is quantitatively measured using KDE distance. Higher KDE distance implies less similarity, which means more data should be added to increase the similarity of training data and testing data. However, more data may not lead to better prediction since the similarity may reduce if irrelevant data is involved. Meanwhile, the selection of PFs is another challenge. The identification of PFs depends on the understanding of local physics, respective models applied in the computational code, and geometry conditions. These factors make it difficult to quantify the uncertainty from the PF selection. Technically, more PFs can capture more local behaviors to inform the OMIS model for further predictions. In the future, the requirements of PF identification for different physics and geometries should be specified. Sensitivity/importance study should be performed to rank the importance of selected PFs to quantify how much they influence the regression response and decide which PFs should be added to rise scalability or be ignored to reduce the dimensionality.

REFERENCES

[1]. Smith C, Schwieder D, Phelan C, Bui A, Bayless P. Light water reactor sustainability program: Risk informed safety margin characterization (RISMC) advanced test reactor demonstration case study. 2012;INL/EXT-12-27015.

[2]. Van Dorsselaere J, Lamy J, Schumm A, Birchley J. Chapter 8 - integral codes for severe accident analyses. In: Sehgal BR, ed. *Nuclear safety in light water reactors.* Boston: Academic Press; 2012:625-656.

[3]. EPRI. GOTHIC thermal hydraulic analysis package, version 8.2(QA). 2014.

[4]. Bao H, Zhao H, Zhang H, Zou L, Sharpe P, Dinh N. Safe reactor depressurization windows for BWR mark I station blackout accident management strategy. *Annals of Nuclear Energy*. 2018;114:518-529.

[5]. Chen Y, Yuann Y. Negative pressure difference evaluation of lungmen ABWR containment by using GOTHIC. *Annals of Nuclear Energy*. 2015;75(Supplement C):672-681.

[6]. Chen Y, Yuann Y, Dai L, Lin Y. Pressure and temperature analyses using GOTHIC for mark I containment of the chinshan nuclear power plant. *Nuclear Engineering and Design*. 2011;241(5):1548-1558.

[7]. D'Auria F, Galassi G. Code validation and uncertainties in system thermal hydraulics. *Progress in Nuclear Energy*. 1998;33(1):175-216.

[8]. Prosek A, Mavko B. Review of best estimate plus uncertainty methods of thermal-hydraulic safety analysis. *Proceedings of the International Conference Nuclear Energy for New Europe 2003*. 2003:827.

[9]. Boyack B, Catton I, Duffey R, et al. Quantifying reactor safety margins part 1: An overview of the code scaling, applicability, and uncertainty evaluation methodology. *Nuclear Engineering and Design*. 1990;119(1):1-15.

[10]. USNRC. Transient and accident analysis methods. 2005; REGULATORY GUIDE 1.203.

[11]. Oberkampf W, Pilch M, Trucano T. Predictive capability maturity model for computational modeling and simulation. October 2007; SAND2007-5948.

[12]. Athe P. *A framework for predictive capability maturity assessment of computer simulation codes.* [Doctor of Philosophy]. Department of Nuclear Engineering, North Carolina State University; 2018.

[13]. Lin L, Dinh N. Predictive capability and maturity assessment with bayesian network. *2018 ANS Annual Meeting*. June 17–21, 2018;118.

[14]. Dinh N, Nougaliev R, Bui A, Lee H. Perspectives on nuclear reactor thermal hydraulics. 2013.

[15]. Chang C, Dinh N. Classification of machine learning frameworks for data-driven thermal fluid models. *ArXiv e-prints*. 2018.

[16]. Liu Y, Dinh N, Smith R. A validation and uncertainty quantification framework for eulerian-eulerian two-fluid model based multiphase-CFD solver. Part I: Methodology. *ArXiv e-prints*. 2018.

[17]. Lin L, Dinh N. Formulation of data-driven methodology for validation of RISMC models. 2018; M2NU-16-NC-NCSU-030401-153.

[18]. Skelton P, Lane J. Milestone 2 deliverable to support NEUP 16-10918 - input on validation requirements for data-driven analysis. 2018; NAI-2018-002.

[19]. Oberkampf W, Roy C. *Verification and validation in scientific computing.* 1st ed. New York, NY, USA: Cambridge University Press; 2010.

[20]. Feng J, Bolotnov I. Evaluation of bubble-induced turbulence using direct numerical simulation. *International Journal of Multiphase Flow*. 2017;93:92-107.

[21]. Oberkampf W, DeLand S, Rutherford B, Diegert K, Alvin K. Error and uncertainty in modeling and simulation. *Reliability Engineering & System Safety*. 2002;75(3):333-357.

[22]. Aksan S, D'Auria F, Städtke H. User effects on the thermal-hydraulic transient system code calculations. *Nuclear Engineering and Design*. 1993;145(1):159-174.

[23]. Bestion D, D'Auria F, Lien P, Nakamura H. A state-of-the-art report on scaling in system thermal hydraulics applications to nuclear reactor safety and design. 2016; NEA/CSNI/R(2016)14.

[24]. Milano M, Koumoutsakos P. Neural network modeling for near wall turbulent flow. *Journal of Computational Physics*. 2002; 182(1):1-26.

[25]. Tracey B, Duraisamy K, Alonso J. A machine learning strategy to assist turbulence model development. *53rd AIAA aerospace sciences meeting.* American Institute of Aeronautics and Astronautics; 2015.

[26]. Parish E, Duraisamy K. A paradigm for data-driven predictive modeling using field inversion and machine learning. *Journal of Computational Physics*. 2016; 305(Supplement C):758-774.

[27]. Zhang Z, Duraisamy K. Machine learning methods for data-driven turbulence modeling. *22nd AIAA Computational Fluid Dynamics Conference*. 2015.

[28]. Ling J, Ryan K, Bodart J, Eaton J. Analysis of turbulent scalar flux models for a discrete hole film cooling flow. *ASME. Journal of Turbomachinery.* 2015.

[29]. Ling J, Jones R, Templeton J. Machine learning strategies for systems with invariance properties. *Journal of Computational Physics*. 2016; 318(Supplement C):22-35.

[30]. Ling J, Kurzawski A, Templeton J. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *Journal of Fluid Mechanics*. 2016;807:155-166.

[31]. Wang J, Wu J, Xiao H. A physics-informed machine learning approach of improving RANS predicted reynolds stresses. *55th AIAA aerospace sciences meeting.* American Institute of Aeronautics and Astronautics; 2017.

[32]. Weatheritt J, Sandberg R. A novel evolutionary algorithm applied to algebraic modifications of the RANS stress–strain relationship. *Journal of Computational Physics*. 2016;325(Supplement C):22-37.

[33]. Zhu Y, Dinh N. A data-driven approach for turbulence modeling. *17th International Topical Meeting on Nuclear Reactor Thermal Hydraulics*. 2017.

[34]. Hanna B, Dinh N, Youngblood R, Bolotnov I. Coarse-grid computational fluid dynamic (CG-CFD) error prediction using machine learning. *ArXiv e-prints*. 2017.

[35]. Bao H, Zhao H, Zhang H, Zou L, Szilard R, Dinh N. Simulation of BWR mark I station black-out accident using GOTHIC: An initial demonstration. *2015 ANS Winter Meeting.* 2015.

[36]. Bao H, Dinh N, Omotowa O, et al. A study of BWR mark I station blackout accident with GOTHIC modeling. *International Congress on Advances in Nuclear Power Plants.* 2016.

[37]. Zenke D. Handbook of multiphase systems. *Acta Polymerica*. 1983;34(6):380-380.

[38]. Govier GW, Aziz K. The flow of complex mixtures in pipes. 2nd ed. Society of Petroleum Engineers; 2008.

[39]. Wilcox DC. *Turbulence modeling for CFD.* California ed. DCW Industries, Inc.; 1993.

[40]. Rodi W. Turbulence models for environmental problems. In: *PMTF zhurnal prikladnoi mekhaniki i tekhnicheskoi fiziki.* ; 1980:259-349.

[41]. Tran C, Dinh N. The effective convectivity model for simulation of melt pool heat transfer in a light water reactor pressure vessel lower head. part I: Physical processes, modeling and model implementation. *Progress in Nuclear Energy.* 2009;51(8):849-859.

[42]. Hanna B, Dinh N, Bolotnov I. High-fidelity simulation-driven model development for coarse-grained computational fluid dynamics. *2016 Advanced Thermal Hydraulics.* 2016.

[43]. Bishop C. Neural networks for pattern recognition. New York, NY, USA: Oxford University Press, Inc; 1995.

[44]. Gurney K. *An introduction to neural networks.* Bristol, PA, USA: Taylor \& Francis, Inc; 1997.

[45]. Rasmussen C, Williams C. *Gaussian processes for machine learning (adaptive computation and machine learning).* The MIT Press; 2005.

[46]. Breiman L. Random forests. *Machine Learning*. 2001;45(1):5-32.

[47]. Ferreira C. Gene expression programming: A new adaptive algorithm for solving problems. *CoRR*. 2001;cs.AI/0102027.

[48]. Gallant SI. *Neural network learning and expert systems.* Cambridge, MA, USA: MIT Press; 1993.

[49]. Demuth H, Beale M. Neural network toolbox user's guide. 2002.

[50]. Hagan M, Menhaj M. Training feedforward networks with the marquardt algorithm. *IEEE Trans Neural Networks*. 1994;5(6):989-993.

[51]. Burden F, Winkler D. Bayesian regularization of neural networks. In: Livingstone DJ, ed. *Artificial neural networks: Methods and applications.* Totowa, NJ: Humana Press; 2009:23-42.

[52]. Møller M. A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks*. 1993;6(4):525-533.

[53]. Hall E. *Review: theory of thermodynamics*, by edgar buckingham. *Bulletin of the American Mathematical Society.* 1902;9(3):173-175.

[54]. Laurent L, Le Riche R, Soulier B, Boucard P. An overview of gradient-enhanced metamodels with applications. *Archives of Computational Methods in Engineering*. 2017.

[55]. Wei P, Lu Z, Song J. Variable importance analysis: A comprehensive review. *Reliability Engineering & System Safety*. 2015;142:399-432.

[56]. Morris M. Factorial sampling plans for preliminary computational experiments. *Technometrics*. 1991;33(2):161-174.

[57]. Sobol′ I. Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. *Mathematics and Computers in Simulation*. 2001;55(1):271-280.

[58]. Borgonovo E. A new uncertainty importance measure. *Reliability Engineering & System Safety*. 2007;92(6):771-784.

[59]. Helton J, Sallaberry C. Sampling-based methods for uncertainty and sensitivity analysis. 2009.

[60]. Wu X, Kozlowski T, Meidani H, Shirvan K. Inverse uncertainty quantification using the modular bayesian approach based on gaussian process, part 1: Theory. *Nuclear Engineering and Design*. 2018;335:339-355.

[61]. Scott D. *Multivariate density estimation: Theory, practice, and visualization.* Wiley; 2015.