ABSTRACT

HANNA, BOTROS NASEIF HANNA BISHARA. Coarse-Grid Computational Fluid Dynamics (CG-CFD) Error Prediction using Machine Learning. (Under the direction of Dr. Nam T. Dinh and Dr. Igor A. Bolotnov).

Nuclear reactor safety research requires analysis of a broad range of accident scenarios. One of the major safety barriers against nuclear fission products release is the containment structure. Modeling and simulation are essential tools to identify parameters affecting Containment Thermal Hydraulics (CTH) phenomena. The thermal-hydraulic modeling approaches used in the nuclear industry can be classified into two categories: system-level codes and Computational Fluid Dynamics (CFD) codes. System codes are not as capable as CFD of capturing and giving detailed knowledge of the multi-dimensional behavior of CTH phenomena. However, CFD computational cost is high when modeling complex accident scenarios, especially the ones which involve longtime transients. The high expense of traditional CFD is due to the need for computational grid refinement to guarantee that the solutions are grid independent.

To mitigate the computational expense, it is proposed to rely on coarse-grid CFD (CG-CFD). Coarsening the computational grid makes the computation significantly more affordable for practical applications, but it also increases the discretization (grid-induced) error which depends on the grid size and varies with space and time. Hence, a method is developed to produce a surrogate model, that predicts the distribution of the CG-CFD local error, to correct the flow variables of interest. Given high-fidelity simulations (sufficiently fine-mesh simulations), a surrogate model is trained to predict the CG-CFD local errors as a function of the coarse-grid local flow features. The surrogate model is constructed using machine learning regression algorithms (namely, artificial neural network and random forest regression).

المنارة للاستشارات

www.manaraa.com

This proposed method is applied to a three-dimensional turbulent flow inside a lid-driven cavity domain. A set of scenarios that test the proposed method are studied. These scenarios investigate the capability of the surrogate model to interpolate and extrapolate outside the training data range. These scenarios also cover a range of Reynolds number in the turbulent flow and transitional flow range plus a range of grid sizes, aspect ratios and different variables of interest. Based on the investigated cases: it was found that the random forest regression technique is predicting the grid-induced error better than the neural network although random forest technique is computationally cheaper. The proposed method maximizes the benefit of the available data and shows a potential for a good predictive capability.

The proposed CG-CFD approach is different from conventional CFD for two reasons: (1) Typically, for each new fluid flow problem, a new simulation is needed, and a grid-independent solution is required, even if the new flow problem is only slightly different from old one. In the proposed CG-CFD method, the available data are utilized to predict the variable of interest for the new flow problem (that is simulated by a coarse grid) given the available high-fidelity data. The success of this approach is based on the assumption that the available high-fidelity data and the new flow problem have similar physics. (2) In conventional turbulence modeling, turbulence models typically rely on incorporating more physics and using empirical models for some parameters based on the available validation data. For different flow conditions, new turbulence models may be constructed. The proposed method is data driven so each new experimental data or high-fidelity simulation lead to improving the method capability.

المنارة للاستشارات
www.manaraa.com

© Copyright 2018 by Botros Naseif Hanna Bishara Hanna

All Rights Reserved

المنارة للاستشارات

www.manaraa.com

Coarse-Grid Computational Fluid Dynamics (CG-CFD) Error Prediction using Machine Learning.

by
Botros Naseif Hanna Bishara Hanna

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 Co-Chair	Dr. Igor A. Bolotnov Committee Co-Chair
Dr. Robert W. Youngblood	Dr. Tiegang Fang

المنارة للاستشارات
www.manaraa.com

DEDICATION

To my Father,
Mikhaeel Hanna
Who always supported me.
Who always encouraged me to do the best I can.
Without whom none of my success would be possible.
Your guiding hand on my shoulder will remain with me.
Thank you for your endless love, sacrifices, prayers, and advice.

المنـــارة للاستشارات
www.manaraa.com

BIOGRAPHY

The author, Botros N. Hanna was born in March 1988 in Athribis in lower Egypt. His given name (Botros) means the stone. He is the first child of his father, Mikhaeel Hanna and his mother, Sonia Samaan. He has one younger brother (Pemen). Botros was married to Monika Attia in January 2016.

Botros earned the bachelor’s degree in Nuclear Engineering with a degree of honors from Alexandria University in 2010. In Fall 2012, Botros joined the Department of Nuclear Engineering in North Carolina State University as a graduate student. He started his graduate studies with Dr. Igor A. Bolotnov and Dr. Nam T. Dinh in Fall 2013 to get his Master of Science degree in 2015. His research areas include computational fluid dynamics for nuclear engineering applications plus application of machine learning algorithms to develop big-data-driven physics-informed models. He successfully defended his PhD dissertation in April 2018.

.

المنارة للاستشارات
www.manaraa.com

ACKNOWLEDGMENTS

I give my ultimate thanks to God who has given me the power and persistence to get where I am today. I cannot thank enough my beloved spouse, Monika who has been my beautiful wife, life companion and stayed by my side in good and hard times. I am incredibly thankful for my family; my mother who always believed in me, my father for his limitless love and support and my brother who is always the best friend. I also acknowledge the guidance of those who encouraged me to pursue my studies in USA (Dr. Hanaa Abou Gabal and Dr. Alya Badawi in Alexandria University, Egypt).

I would like to express my gratitude to Dr. Igor A. Bolotnov and Dr. Nam T. Dinh for their consistent support throughout my Ph.D. journey. They always provided me with their insightful discussions about the research. They encouraged me to be an independent researcher and they were always great examples to follow in academic research. I also acknowledge the advice and recommendations by Dr. Robert Youngblood and Dr. Tiegang Fang.

I also acknowledge the support of the Idaho National Laboratory through its National University Consortium and Laboratory Directed Research & Development (LDRD) Program under DOE Idaho Operations Office and the Nuclear Engineering Department at NC State University.

المنارة للاستشارات

www.manaraa.com

TABLE OF CONTENTS

المنارة للاستشارات

www.manaraa.com

المنارة للاستشارات

www.manaraa.com

المنارة للاستشارات

www.manaraa.com

LIST OF TABLES

www.manaraa.com

LIST OF FIGURES

www.manaraa.com

المنارة للاستشارات

www.manaraa.com

المنارة للاستشارات

www.manaraa.com

ABBREVIATIONS

ALES	Autonomic sub-grid Large Eddy Simulation
ANN	Artificial Neural Network
BD	Blended Difference
BWR	Boiling Water Reactor
CD	Central Difference
CFD	Computational Fluid Dynamics
CG	Coarse Grid
CG-CFD	Coarse-Grid Computational Fluid Dynamics
CTH	Containment Thermal Hydraulics
DL	Deep Learning
DNS	Direct Numerical Simulation
FDM	Finite Difference Method
FEM	Finite Element Method
FVM	Finite Volume Method
HF	High Fidelity
KDE	Kernel Density Estimation
LES	Large Eddy Simulation
LF	Low Fidelity
MATLAB	Matrix Laboratory
MD	Mahalanobis Distance
ML	Machine Learning
MSE	Mean Squared Error

المنارة للاستشارات
www.manaraa.com

NPP	Nuclear Power Plant
NS	Navier – Stokes
OOB	Out Of Bag
OpenFOAM	Open source Field Operation And Manipulation
PC	Principal Component
PCA	Principal Component Analysis
PDE	Partial Differential Equation
PIML	Physics Informed Machine Learning
PISO	Pressure Implicit Splitting of Operators
QOI	Quantity Of Interest
RANS	Reynolds Averaged Navier-Stokes
RFR	Random Forest Regression
RISMC	Risk-Informed Safety Margin Characterization
ROM	Reduced Order Modeling
SGS	Sub-Grid Scale
SVM	Support Vector Machines
TE	Truncation Error
TH	Thermal Hydraulics
TKE	Turbulence Kinetic Energy
UD	Upward Difference

المنارة للاستشارات
www.manaraa.com

NOMENCLATURE

C_s	Smagorinsky constant
$E(\kappa_l)$	The energy contained in the eddies of size l
E_{Ml}	Machine learning algorithm error
F	Surrogate function
H/W	The aspect ratio (height to width ratio)
L	Discrete operator
L_f	Discrete operator computed by a fine grid
L_s	The mixing length for sub-grid scales
L_{Δ}	Discrete operator computed by a grid, Δ
P	The rate of turbulence energy production
Q	The second invariant of the velocity gradient tensor
R	Correlation coefficient
Re	Reynolds number
Re_τ	Wall-shear stress-based Reynolds numbers
Re_t	Turbulent Reynolds number
Re_Δ	Cell Reynolds number
S_i	Silhouette value
T	Target
$\boldsymbol{U}$	Velocity vector
U_x, U_y, U_z	Velocity components in x, y, z directions

المنارة للاستشارات
www.manaraa.com

V	Volume
W	Cavity width (or cavity lid width)
X	Features (inputs) in machine learning algorithm
$\tilde{X}$	Normalized features
Y^+	$Y^+ = \frac{u_* d}{\nu}$, dimensionless wall distance
d	Distance to the nearest wall
k	Turbulent kinetic energy
l	Length scale
l_m	Mixing length
n	Control volume surface normal
p_k	Kinematic pressure
t	Time
u_*	Friction velocity at the nearest wall
x, y, z	Cartesian coordinates

Greek letters

Δ	Grid spacing or cell length
$\Delta x, \Delta y, \Delta z$	Grid spacing in the three Cartesian directions: x, y, z
δ_{ij}	Kronecker delta
ε	The rate of dissipation of the turbulence kinetic energy
$\varepsilon_{\Delta}(\varphi)$	Grid-induced error when computing a variable φ

المنارة للاستشارات
www.manaraa.com

$\tilde{\varepsilon}_{\Delta}$	Normalized grid-induced error
η	Kolmogorov length scale
κ_{l}	The wavenumber of an eddy of size is l
ν	Kinematic viscosity
ν_{sgs}	Sub-grid scale viscosity
ν_{t}	Turbulent kinematic viscosity
$\tilde{\nu}_{t}$	Spalart-Allmaras model variable (viscosity like variable)
σ	Standard deviation
τ	Reynolds stress
$\overline{\tau_{\imath\jmath}}$	Sub-grid scale stress tensor
τ_{η}	Kolmogorov time scale
φ	Flow variable
$\bar{\varphi}$	Filtered variable
$\varphi_{\boldsymbol{c}}$	Variable computed by a coarse-grid
$\varphi_{\boldsymbol{f}}$	Variable computed by a fine grid
$\varphi_{f\to\Delta}$	Flow variable computed by a fine grid and mapped onto a coarse-grid Δ
φ^{tr}	Flow variable distribution corresponding to the training data
φ^{te}	Flow variable distribution corresponding to the testing data
φ_{Δ}	Flow variable computed by a grid, Δ
ψ	A number that characterizes the flow pattern such as Reynolds number

المنارة للاستشارات

www.manaraa.com

1. INTRODUCTION

Nuclear reactor safety analysis has received more attention since the Fukushima Daiichi accident in 2011. Safety and design of Nuclear Power Plants (NPPs) imply studying Thermal Hydraulics (TH) phenomena that may evolve during normal operation, design-basis accidents (that NPP is designed to withstand) or severe accidents (that involve core melting) [1]. If an accident occurs, the role of the reactor containment is vital as the major and final safety barrier. Hence, predicting the Containment Thermal Hydraulics (CTH) phenomena impacts is needed to maintain the integrity of the containment. Experiments that mimic the potential CTH phenomena are necessary, but it is almost impossible to gain experimental full-scale data the cover all CTH phenomena. Hence, for a complex system such as NPP, modeling and simulation are necessary for performing safety analysis. There are various codes that are used to achieve this purpose: the system TH codes (see Section 1.2) and Computational Fluid Dynamics (CFD) codes (see Section 1.3).

The system TH codes can model the entire reactor and its components as they include multiple physics (fluid flow, heat transfer, neutron diffusion equations, etc.). However, system codes have some limitations as they cannot capture the local three-dimensional complex phenomena and they rely on many empirical correlations and parameters that are required due to the lack of full understanding of the complex governing physics. Hence, CFD codes are needed to capture the multi-dimensional behavior of the phenomena of interest. However, CFD approach is computationally expensive for simulating complex systems with large computational domains. The high expense of traditional CFD is due to the need for computational grid refinement to guarantee that the solutions are grid-independent. To reduce the CFD computational cost, there is a need for coarse-grid CFD.

المنارة للاستشارات
www.manaraa.com

It is proposed that the numerical error, which arises because of the coarse grids, can be reduced/recovered using data-driven models. The need of data-driven CFD models, which benefit from the available experimental and high-fidelity simulations, is emphasized in Section 1.4. Currently, there is an increasing attention to the necessity of data-driven fluid dynamics' models that can be less expensive and more adaptable to the data (see Section 1.5).

This work proposes a framework to use a data-driven CG-CFD method in CTH field (described in Section 1.6). To clarify the difference between the proposed data-driven CG-CFD method and other approaches that aim to mitigate the complexity of CFD, the scope of this work is presented in Section 1.7. Finally, the structure of the rest of the dissertation is described in Section 1.8.

1.1. Objectives

The objectives of this chapter can be summarized as follows:

- Evaluating the capabilities and limitation of potential codes in CTH fields (system codes and CFD).
- Introducing the motivation behind this research.
- Discussing the current research in data-driven CFD modeling.
- Developing a general framework to use CG-CFD simulations in CTH field.

1.2. An Overview of System Codes for Nuclear Thermal Hydraulics

System codes are designed to simulate the phenomena that are involved in design basis and severe accidents. These codes utilize control volume approach to perform affordable computations. Large control volumes are used to perform economic simulations. For instance, control volumes, that are bigger than a one-meter cube, are used in phenomena-dominating regions (reactor vessel is recently modeled by few hundreds to few thousand volumes). The control volumes are much

المنارة للاستشارات
www.manaraa.com

bigger in other less-important regions in the nuclear system [2, 3]. Within each volume, time and space dependent conservation equations for mass, momentum, and energy are solved for a homogenous two-phase (steam-water) mixture. Phenomena are modeled with different degrees of dimensionality [2]. Zero-dimensional (lumped parameter) models are used where the velocities are low (e.g. reactor lower plenum). One-dimensional models are used where the flow has a uniform direction (e.g. pipe flow). Three-dimensional models were developed for the components with three-dimensional velocity (e.g. pressure vessel) with a coarse grid. Examples of system codes include RELAP5 [3], MELCOR [4] and GOTHIC [5].

System codes are recognized in the nuclear industry as useful tools for analyzing accidents and transients on the reactor system level. They represent the currently available knowledge and can predict the figure of merit with computationally affordable simulations. However, system codes still have some limitations. Among these limitations is the coarse-space resolution so the small-scale phenomena are not resolved; Simulating highly turbulent flows requires fine-grid resolution to resolve a broad range of turbulence spatial scales (See Chapter 2). Additionally, mesh convergence to get a mesh independent solution is not expected with such a coarse resolution. A lot of closure laws, empirical equations and parameters also introduce uncertainty.

The complex geometries in the nuclear reactor system are replaced by simple models which introduce uncertainty as well. Therefore, system codes cannot properly model the complex multi-dimensional behavior of CTH phenomena (e.g. mixing and natural circulation) and therefore, cannot provide information to understand certain phenomena [6, 7]. Analyzing local phenomena in specific sceneries requires CFD simulations (Section 1.3).

المنارة للاستشارات
www.manaraa.com

1.3. An Overview of CFD Codes for Nuclear Thermal Hydraulics

The second type is Computational Fluid Dynamics (CFD) codes, such as Fluent [8], STAR-CCM+ [9] and OpenFOAM [10]. CFD simulations are based on solving Navier-Stokes (NS) equations. CFD approach is different from system codes (in Section 1.2) due to different reasons [11]: 1. CFD approach can fully predict phenomena which are multi-dimensional, local, transient, and geometry dependent (this is not possible with system codes). 2. CFD can provide insights to understand the physics (not expected with system codes). 3. It is typical, in system codes, to solve simplified balance equations for control volumes bounded by the solid walls while CFD approach is based on NS equations solved for each cell in the computational grid. 4. In CFD, mesh convergence study is performed by reducing the mesh size to a fixed value (not possible with the computational domain coarse nodalization in system codes).

The majority of flows in nuclear thermal hydraulics are turbulent. Simulation of turbulent flows with CFD requires capturing a broad range of turbulence length and time scales. Because of the complex geometry and high Reynolds number (highly turbulent) flow, it is computationally very expensive to simulate CTH flows by solving NS equations directly. Instead, the most common method is solving the Reynolds Averaged Navier-Stokes (RANS) equations. RANS modeling [12] implies solving the time-averaged NS equations while the turbulence stresses (resulting from the averaging procedures) are modeled with closure models. However, RANS approach also has its limitations. For example, in a recent study [13], OpenFOAM (CFD package) [10] was used to simulate high-pressure blowdown from the reactor cooling system and containment convective mixing. The simulation needed a week of computing time with 128 processors to simulate 10 seconds of steam blowdown. It is important to mention that a coarse-grid (cell length ranges from 3 to 14 centimeters) was used (resulted in one million computational nodes) with RANS equations.

المنارة للاستشارات
www.manaraa.com

Considering the coarseness of the grid plus the low resolution of the RANS approach results compared to solving NS equations directly (Direct Numerical Simulation (DNS)), it is clear that applying CFD would be a huge computational challenge. The challenge becomes even more formidable when modeling more complex accidents with long transient scenarios. The fine grid requirements often drive down the computational time step size, which makes the solution of long-transient problems prohibitively expensive. The need for sensitivity analysis, uncertainty quantification, and analysis of multiple scenarios required by Risk-Informed Safety Margin Characterization (RISMC) also exacerbates the issue of high computational expense [14]. CFD simulation of a full reactor core, with its fuel rods, spacer grids, and other components, is not yet practical [15].

To mitigate the computational expense, it is proposed to rely on CG-CFD models. CG-CFD simulations increase the discretization error which depends on the grid size and varies with space and time. This grid-induced error can be predicted to correct the variable of interest using data-driven models. The need for data-driven models and the progress in data-driven modeling in fluid dynamics field are discussed in Sections 1.4 and 1.5 respectively.

1.4. Motivation: The Need for Data-Driven Models

In CFD, high-fidelity results can be obtained by simulating all the turbulence length and time scales using DNS [12], which is accurate but computationally challenging for many applications (turbulent flows with high Reynolds number). A computationally cheaper alternative approach is Large Eddy Simulation (LES) [16] that requires a computational grid that is coarser than the grid needed in DNS. In LES, the large-scale motions (energy-containing eddies) are captured, while the small-scale flow motions are either modeled using an explicit sub-grid scale

المنارة للاستشارات
www.manaraa.com

model or implicitly modeled using the numerical dissipation associated with the computational scheme knows as Implicit LES (ILES) (see [17, 18]).

Despite the progress in LES methods, the Reynolds Averaged Navier-Stokes (RANS) equations [12] approach is still more popular and computationally efficient in industrial applications. RANS approach requires a coarse grid (compared to LES and DNS) as it resolves the mean flow variables only, not the detailed instantaneous flow. However, RANS models (such as $k - \varepsilon$ [19] and Spalart-Allmaras [20] model) lack adaptability, i.e. RANS models perform well only for specific flow conditions and geometries. This gives an advantage to data-driven surrogate statistical models as they could adapt via data assimilation as more data becomes available. The availability of high-fidelity simulations provides an opportunity to inform data-driven coarse-grid models. Currently, as discussed below, several groups of researchers are pursuing the development of data-driven methods to perform coarse grid simulations (see Section 1.5).

1.5. Progress in Data-Driven Models in Fluid Dynamics

Traditionally, fluid flow models (for turbulence or multiphase flow, etc.) were developed based on physical understanding of the phenomena often with certain empirical assumptions. Recently, data-driven models have been developed based on data processing and analysis (data-driven), to make more use of the available data. An overview of recent data-driven models is presented below. These data-driven models are either utilized in the context of CFD (to correct LES or RANS simulations or compute closure terms for averaged variables' equations) or in the context of Smoothed Particle Hydrodynamics (SPH) [21].

1.5.1. Autonomic sub-grid Large Eddy Simulation (ALES)

Autonomic Sub-grid Large Eddy Simulation (ALES) [22] can be regarded as a data-driven method that utilizes a grid that is deemed to be coarse compared to DNS grid requirements but

المنارة للاستشارات
www.manaraa.com

must be relatively fine to resolve turbulence kinetic energy (TKE) in the inertial range of the turbulence energy spectrum. The ALES method expresses the local sub-grid scale stress tensor as a non-linear function of the resolved variables at all locations and all times with a Volterra series (which is similar to a Taylor series). The series coefficients are computed by minimizing the error in sub-grid scale stresses at a test filter scale. Then, coefficients are mapped to the LES scale assuming scale similarity (noting that both LES scale and test scale lie in the self-similar inertial range of the turbulence energy spectrum). The ALES method is a general model-free self-optimizing approach for closure of turbulence simulations. The ALES method needs neither previous training based on DNS results nor user-specified parameters. Preliminary results with simple problems (homogenous turbulence and sheared turbulence) showed high accuracy in computing turbulent stresses compared to current turbulence models [23]. However, the scale similarity assumption, upon which ALES is based, is valid only in the inertial range (in the TKE spectrum). Therefore, applying ALES approach is still impractical in simulating large domains / long transients.

1.5.2. Autonomic sub- Closure Term in a Bubbly Flow Equation

Another example [24] from the multiphase flow field is finding the closure term in an averaged simple equation for bubbly flow using an Artificial Neural Network (ANN) [25], given accurate results obtained by DNS. In that problem, the initial vertical velocity and the average bubble density are uniform except in one of the horizontal directions. As the transient develops, the bubble density and velocity become uniform. It was assumed that the unknown closure term depends on the void fraction, the void fraction gradient, and the liquid velocity gradient. ANN learns the correlation between the closure term and these three variables from one simulation dataset. A transient of different initial conditions can then be predicted using that correlation. That

المنارة للاستشارات
www.manaraa.com

case study was quite simple (the averaged equation is one dimensional, and the boundary conditions are periodic).

1.5.3. RANS Model Discrepancy

To reduce the RANS model discrepancy by learning from data, research groups from the University of Michigan and Virginia Tech University utilized ML techniques to predict or reduce the error in RANS simulation results [26, 27]. A comparison that summarizes these efforts is presented in Table 1. Both groups made use of High-Fidelity (HF) simulation results from DNS to correct the Low-Fidelity (LF) model results from RANS. In [26], RANS results are corrected by spatially distributed multiplicative discrepancy term, β, into one of the terms of the transport equation of turbulent quantities; in [27], the Reynolds stress discrepancy is directly based on mean flow features computed by RANS.

The statistical model in both cases is trained based on the available high-fidelity data and then tested on other cases, which have a different Reynolds number or a different flow geometry. Recently, the Virginia Tech University research group suggested that [28] the group of input features could be increased to more than 50 flow features to improve the prediction capability of the statistical model. It was also shown that not only could Reynolds stress discrepancy, computed by RANS, be improved by predicting Reynolds stress discrepancy, but also the improved Reynolds stress results in a more accurate velocity field. Similarly, Barone et al. [29] used deep neural network [30] to correlate LES error with the flow variables resolved by LES.

المنارة للاستشارات
www.manaraa.com

Table 1. A Comparison between two approaches for data-driven turbulence modeling.

Authors	**Parish & Duraisamy (Journal of Computational Physics 2016) University of Michigan.**	**Wang, Wu, & Xiao (Physical Review Fluids 2017) Virginia Tech Univ.**
Method	RANS results are corrected by introducing a spatially distributed multiplicative discrepancy term β into one of the terms of the transport equation of the turbulent quantities.	Reynolds stress discrepancy (the difference between Reynolds stress computed by DNS and by RANS methods) is a function of the mean flow features computed by RANS. Reynolds stress discrepancy components and the mean flow features computed by RANS are introduced in Galilean invariant forms.
HF and LF data	HF: DNS results. LF: RANS results.	
Input and output	Input: turbulent flow features e.g. $Y^+, \frac{P}{\varepsilon,},$ etc. Output: β.	Input: turbulent flow features e.g. $Q, \frac{k}{\varepsilon,},$ etc. Output: $\tau_{DNS} - \tau_{RANS}$.
Case study	Two-dimensional domain with one dimensional fully developed channel flow.	1- The fully developed turbulent flow in a square duct. 2- Three dimensional fully developed turbulent flow in periodic hill geometry.
Training	Channel flow	Study#1: The fully developed turbulent flow in a square duct. Study#2: Three-dimensional fully developed turbulent flow in a periodic hill geometry. Study#3: Wavy channel & curved backward-facing step.

المنارة للاستشارات

www.manaraa.com

Table 2 (continued). A Comparison between two approaches for data-driven turbulence modeling.

Testing	Channel flow with a different Reynolds number.	Study#1: The fully developed turbulent flow in a square duct with a different Reynolds number. Study#2&3: Three dimensional fully developed turbulent flow in a periodic hill geometry with a different Reynolds number.
Statistical learning	Gaussian process regression [31].	Random forest regression [32].
Notes	It assumes that both the data distribution and the corrector β are Gaussian. Dependence on grid size and numerical scheme were not considered.	Dependence on grid size and numerical scheme were not considered.

Statistical data-driven models are usually trained on a set of data and then tested with another group of data that are “close” to the training data in some sense. The “closeness” between the training data and the testing data can be used to assess the prediction confidence *a priori* [33]. This closeness could be quantified by metrics like the Mahalanobis Distance [34] or the Kernel Density Estimation technique [35]. This approach was successful in most cases except when the training and testing cases are too close or too far apart [33].

ML model predictions, for RANS Reynolds stress anisotropy, for flows of similar physics, should be similar. ML model predictions should not change with changing the orientation of coordinate frame. In other words, ML predictions should be Galilean invariant. Based on this principle, a multi-layer (deep) ANN (also called “deep learning”), with invariant tensor basis, was proposed [36]. This way, Reynolds stress anisotropy tensor can be predicted with an embedded

المنارة للاستشارات
www.manaraa.com

Galilean invariance. The proposed tensor-basis ANN proved to have more accurate prediction compared to both RANS models and conventional ANN.

1.5.4. Smoothed Particle Hydrodynamics

Data-driven models, for fluid simulations, are used not only in mesh-based CFD but also in Smoothed Particle Hydrodynamics (SPH) (where Navier-Stokes equations are approximated on fluid particles instead of a computational grid [21]). Random Forest Regression (RFR) [32] was trained to predict the velocity and position of the fluid particle in the next time step based on the velocity and the position in the previous time step [37]. This approach aims to “learn” the behavior of fluid from the training examples and provides an alternative for real-time fluid simulations.

1.6. CG-CFD for Simulating CTH

Typical accident scenarios may include long transients, CTH multi-physics phenomena, and the flow regime may change over the computational domain or over time. CG-CFD approach can be used to deal with such a complex situation, as presented in Figure 1. In Figure 1, a conceptual framework is proposed to benefit from the fine-grid simulation results, to correct the CG-CFD results for every single phenomenon, while the multiple-phenomena scenario is simulated with coarse grids.

In Figure 1, the boxes are numbered from 1 to 8 to refer to 8 steps. In the 1st step, the flow of every single phenomenon of interest is simulated with a grid which is coarse relative to the grid requirement of DNS. This coarse-grid simulation is not expected to obtain accurate results, but it is still expected to show a flow pattern similar to the high-fidelity simulation result. For instance, in Figure 2, the difference between the coarse-grid and the fine-grid results is clear, both results show similar flow pattern. Repeating this step with a variety of potential phenomena will result in a library of CG-CFD results (2nd step). The 3rd, 4th and 7th steps are the focus of this work

المنارة للاستشارات
www.manaraa.com

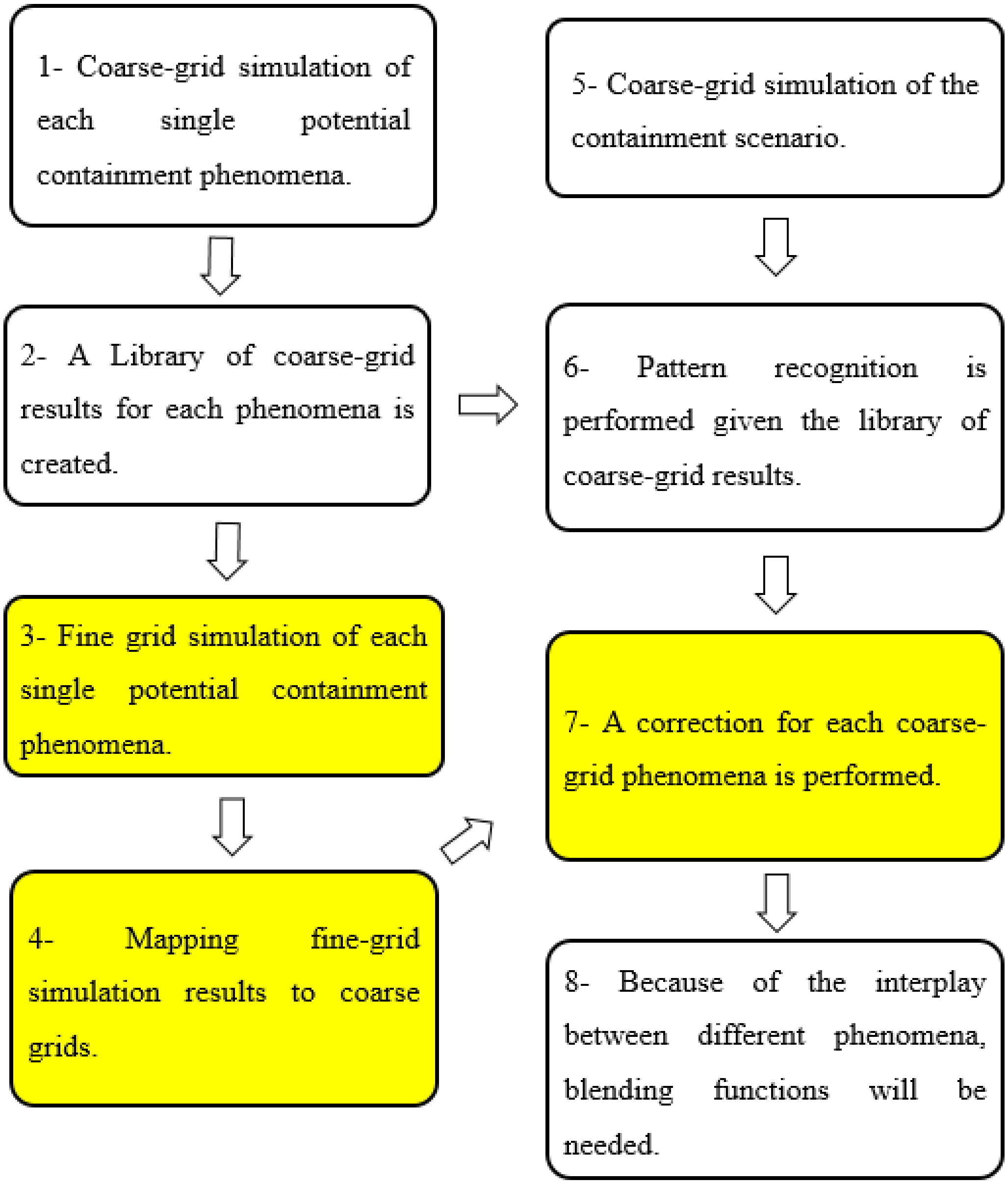

1- Coarse-grid simulation of each single potential containment phenomena.
2- A Library of coarse-grid results for each phenomena is created.
3- Fine grid simulation of each single potential containment phenomena.
4- Mapping fine-grid simulation results to coarse grids.
5- Coarse-grid simulation of the containment scenario.
6- Pattern recognition is performed given the library of coarse-grid results.
7- A correction for each coarse-grid phenomena is performed.
8- Because of the interplay between different phenomena, blending functions will be needed.

Figure 1. A conceptual framework for simulating CTH with CG-CFD. The focus of this work is highlighted in yellow.

. Since the number of nodes in a fine gird is much larger than in a coarse grid, it is hard to compare the fine and coarse grid results directly. Instead, coarse-grid results (1st step) are compared to the fine-grid results (3rd step) mapped on the coarse grid (4th step). Figure 2 gives an example of the difference between the coarse-grid, fine-grid and mapped results. While the mapped velocity

المنارة للاستشارات
www.manaraa.com

profile is not perfectly similar to the fine-grid results (especially near the corners of the cavity), the mapped profile is much more accurate than the coarse-grid one. In this work, machine learning algorithms are utilized to get a statistical data-driven model that models a correction term (7th step) that compensates for the information lost due to the coarseness of the grid.

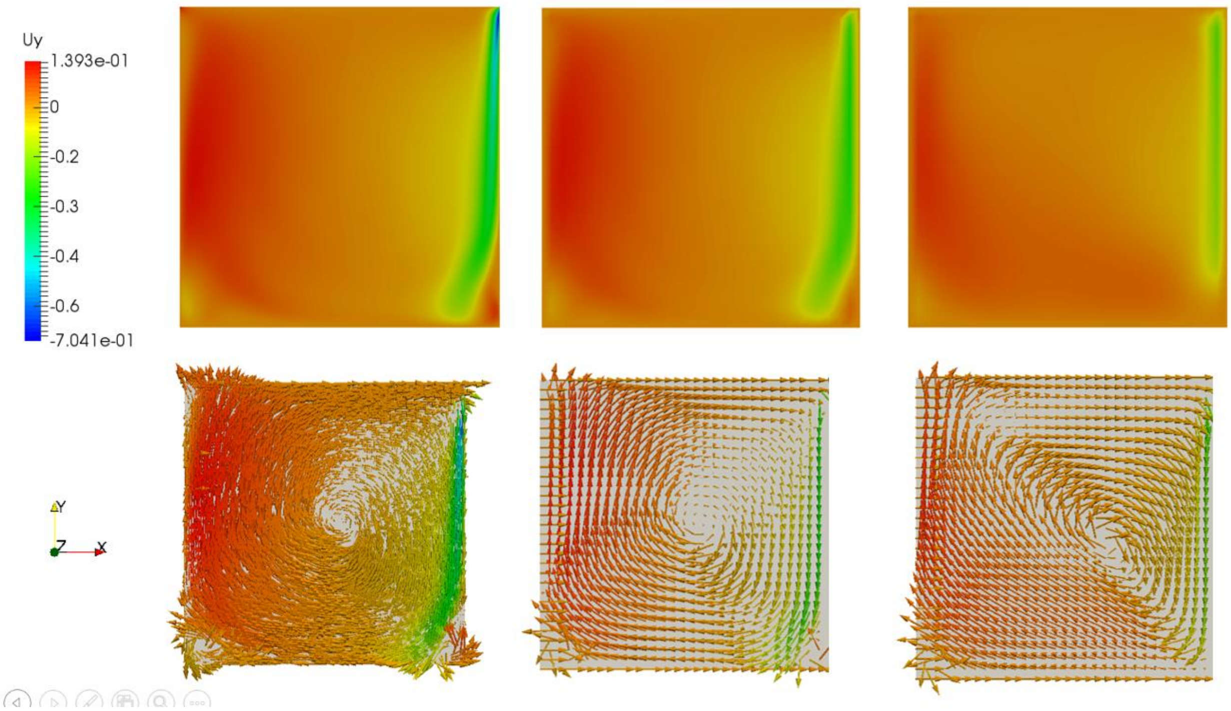


Figure 2. A snapshot of the velocity (U_y) profile in a three-dimensional quasi-steady state turbulent flow in a lid-driven cavity. The lid is moving in x-direction. The profile is computed with a fine grid (left) and a coarse-grid (right). The fine-grid result is mapped to a coarse-grid (in the middle).

Based upon the success of the first 4 steps, simulating a realistic nuclear accident scenario that involves multi-phenomena with a coarse grid can be beneficial (5th step). Given the non-accurate predictions coming out of the 5th step plus the library created in the 2nd step, different phenomena involved in a containment scenario can be identified (6th step: pattern recognition). Pattern recognition is a process of classifying the features into groups. It may help to distinguish different flow regimes (e.g. laminar vs. turbulent or shear flow vs. wall-bounded flow).

www.manaraa.com

If each phenomenon is identified correctly, it may be possible to benefit from the correction term computed in the 7th step to correct different group of results corresponding to different phenomena. It would be challenging to use correction terms computed by different data-driven models for different groups of interacting phenomena. Therefore, blending functions may be needed to deal with a mixture of models (8th step).

1.7. The Scope of This Work

Since the traditional high-resolution (mesh-independent) CFD solution is computationally expensive and using of a coarse grid would result in high grid-induced errors, the present work focuses on predicting the errors of the coarse-grid solution. While the previous research aimed largely to reduce the model form error, in this work, the sub-grid effect is compensated for by a surrogate statistical model that is trained by using high-fidelity simulation results (fine-grid CFD).

The objective of this work is to investigate the feasibility of obtaining a correction for the CG-CFD simulation results using machine learning algorithms. Numerical experiments are designed and performed to study the feasibility of utilizing machine learning tools to get a correlation between the 'correct' solution informed by fine grid simulation results and the coarse-grid variables. Among the different sources of error in CFD simulation (see Figure 3), the turbulence modeling error and the discretization error are the most challenging ones. Below, in Subsections 1.7.1, 1.7.2 and 1.7.3, the present work is compared to other CFD approaches in terms of mitigating turbulence modeling error and discretization error.

المنارة للاستشارات
www.manaraa.com

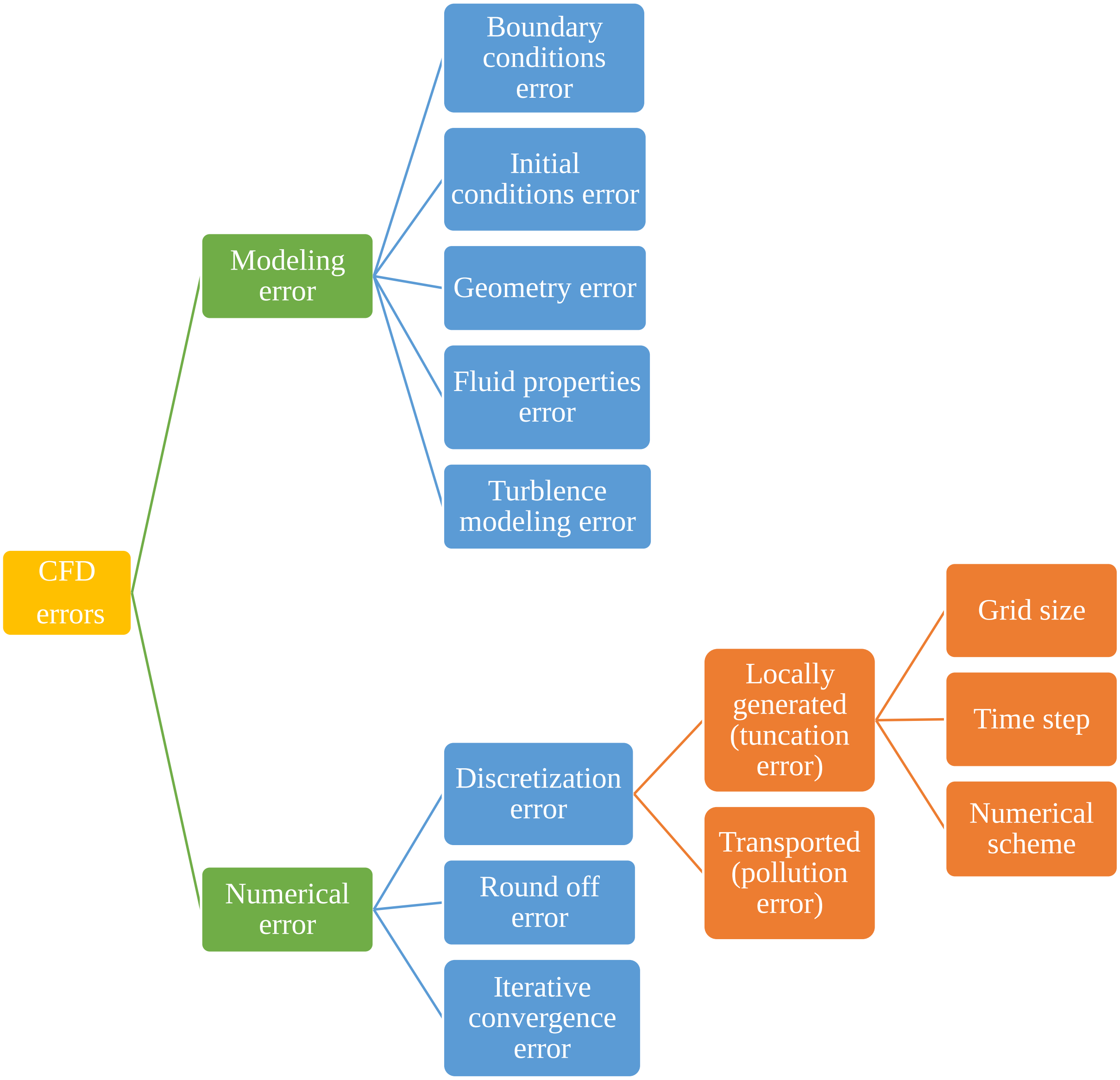


Figure 3. CFD sources of error.

1.7.1. Present Approach vs. Traditional CFD

RANS turbulence models are popular because of the reduced computational expense while representing three-dimensional flow behavior. RANS turbulence models typically rely on incorporating more physics and using empirical models for some parameters based on the available validation data. It is always required to get a grid-independent solution (with a grid convergence

www.manaraa.com

study), but this grid convergence study should be applied within the range of grids that satisfy the turbulence model grid requirements. Typically, for each new case, a new simulation is needed, even if the new case is only slightly different from old cases.

In the present work, the “No model” approach (where NS equations are solved numerically without any turbulence model) is utilized with coarse grids (discretization grids that are expected to produce non-accurate results). However, instead of investigating the turbulence modeling error, we use the same “No model” approach with different grids (fine and coarse grids) to train a surrogate model to compute this grid-induced error. The CG-CFD error is predicted by comparing the high-fidelity results (with fine grids) against CG-CFD results. With this surrogate model, the grid-induced error can be predicted for other cases that have different grid sizes, Reynolds numbers, etc. The surrogate model is adaptive with the available data, so each new set of experimental data or high-fidelity computations will be reflected in the automatically-improved model, without the need for developing other models that capture the new data.

1.7.2. Present Research vs. Turbulence Modeling Error Reduction

Recently, research has been pursued in the direction of taking advantage of the ML algorithms to develop surrogate models that can compute the turbulence modeling error (for instance, [26, 27]). This approach is known as Physics Informed Machine Learning (PIML) approach. Both the present work and PIML method benefit from ML algorithms to produce data-driven statistical models. PIML recent research assumes that one of the RANS models is used and the grid convergence criterion for this model is satisfied. Through PIML, the difference between RANS flow variables profile and DNS profile is computed. On the other side, the present work depends on the same NS equations used with both fine and coarse grids. After that, the error resulting from the grid coarseness is computed with the ML surrogate model.

المنارة للاستشارات
www.manaraa.com

1.7.3. This Work vs. Reduced Order Modeling (ROM)

ROM techniques can be categorized in two classes: dimensionality reduction and surrogate modeling [38]. The first class is the dimensionality reduction techniques which aim to reduce the number of inputs or (dimensions) of the model without changing the original function. CG-CFD approach discussed in this work does not involve any dimensionality reduction because the same flow variables are included in both fine and coarse-grid simulations.

The second class is the surrogate modeling which leads to substituting the original model with another approximate more simplified function. This simplification is either based on physics or based on statistics (data). For instance, LES and RANS method can be regarded as physics-based approaches because both methods were developed based on some physical assumptions and are considered as approximations of the original NS equation. On the other side, the statistics-based surrogate functions are selected heuristically by regression methods or ML algorithms. To classify the current work, the proposed CG-CFD approach is described symbolically as follows:

The discretized NS equations solved on a sufficiently fine grid is written as: $L_f(\varphi) = 0$. φ_f is the flow variables computed by the fine grid and L_f is the discrete operator computed by a fine grid. The discretized CG-CFD equations can be written as $L_\Delta(\varphi_\Delta) = 0$. Δ is the cell length of the coarse grid. The grid-induced error, when computing a variable φ, is $\varepsilon_\Delta(\varphi) = \varphi_{f\to\Delta} - \varphi_\Delta$, while $\varphi_{f\to\Delta}$ is the φ_f field mapped on a coarse grid. Using ML, a surrogate function F is computed:

$$\varepsilon_\Delta(\varphi) = F(X(\varphi_\Delta)) \qquad (1\text{-}1)$$

where $\mathrm{X}(\varphi_\Delta)$ is the features set (new variables) computed given φ_Δ. The process of obtaining $\varepsilon_\Delta(\varphi)$, as a function of $X(\varphi_\Delta)$, can be classified as a statistics-based surrogate model construction because it is mainly dependent on the available data.

www.manaraa.com

1.8. The Structure of This Dissertation

Sources of uncertainty in CFD are mainly turbulence modeling error and numerical error (see Chapters 2 and 3, respectively). The numerical error (in particular, the grid-induced error) is recovered using data-driven techniques based on ML. Different ML algorithms are listed in Chapter 4. These ML algorithms are utilized to produce a statistical model that can predict the CFD coarse-grid-induced error, given the flow features computed by a coarse grid (see Chapter 5). The capability of the statistical model to predict the grid-induced error in different cases is illustrated and assessed in Chapter 6. The conclusion of this dissertation and the future work plan are provided in Chapter 7. The following section (Section 1.9) is presented to clarify the meaning of some terms used in the context of this work.

1.9. Definitions

- Training data: A set of data needed to discover the relationship between different variables. For instance, this group of data is used to find the regression function which relates the output and input.
- Testing data: After training the statistical model given the “training data”, this model is assessed using the testing data. Testing data are totally independent of the training data.
- Features: (or attributes or inputs) are the input variables which characterize the problem. For instance, fluid flow features can be the velocity or the pressure and so on.
- Discretization error: the difference between the exact solution of the discretized equation and the exact solution of the original Partial Differential Equations (PDEs). It includes both truncation error (locally generated) and the pollution error (transported component).
- Truncation error: when approximating a function by a summation (for instance, Taylor series), the infinite sum is truncated and approximated by a finite sum. This is called truncation error

المنـــارة للاستشارات
www.manaraa.com

and it occurs when truncating the infinite series used to approximate a PDE (when discretizing PDEs). This error is eliminated by refining the grid. Truncation error is the local source of the discretization error.

- Pollution error: It is another component of the discretization error. It is not produced locally but transports from elsewhere in the computational domain. This error is transported through the domain similar to the solution transport (the error can be convected or diffused).
- Round-off error: Difference between the actual number and its approximation (rounding) stored on the computer. This error is assumed to be small compared to other errors.
- Iterative convergence error: it is an error that results from the iterative algorithm that is used during the simulation. Typically, this iteration is stopped based on a criterion (typically the residual). It is the difference between the solution of the discretized equation with the stopping criterion and the exact solution of the discretized equation with a zero residual.

1.10. Chapter Summary

Modeling and simulation of Containment Thermal Hydraulics (CTH) phenomena are achieved either by system codes or Computational Fluid Dynamics (CFD). Affordable computations on the reactor-system level are performed by system codes using large control volumes (larger than one cubic meter). Using System codes, CTH phenomena are modeled with different degrees of dimensionality. Although system codes represent the currently available knowledge in the nuclear industry, they cannot properly model the complex multi-dimensional behavior of CTH phenomena because of the uncertainty related to the used empirical equations, empirical parameters, simplified geometries and coarse-mesh resolution. On the other side, CFD codes can model phenomena which are multi-dimensional, local, transient, and geometry dependent plus providing insights to understand the physics. However,

المنارة للاستشارات
www.manaraa.com

applying CFD would be a huge computational challenge when modeling complex accidents with long transient scenarios.

This work is motivated by the need to mitigate CFD computational expense and the current progress in data-driven modeling using machine learning algorithms that can benefit from the available high-fidelity simulations and experimental data. Hence, it is proposed to rely on the coarse-grid CFD simulations (which are affordable) while the grid-induced error can be predicted using data-driven models.

Currently, data-driven models (trained by Direct Numerical Simulations (DNS) data) are utilized in the context of CFD to correct Large Eddy Simulations (LES), Reynolds Averaged Navier-Stokes (RANS) simulations or to compute closure terms for averaged variables' equations. On the other side, the present work investigates the feasibility of correcting coarse-grid solutions computed by Navier-Stokes equations, without turbulence modeling, using data-driven models that are trained with sufficiently fine-grid simulations.

A method that predicts the grid-induced error given the coarse-grid solution features is proposed. Currently, this method is applied to a turbulent three-dimensional flow inside a cavity. A General framework that benefits from the proposed method to simulate CTH phenomena using coarse-grid CFD is also proposed in this work.

المنارة للاستشارات
www.manaraa.com

2. TURBULENCE MODELING

CFD approach is used in CTH analyses to capture the relevant multi-dimensional phenomena in detail. CFD implies solving Navier-Stokes equations numerically and this requires finer grid resolution especially if the flow is highly turbulent. CTH phenomena typically include turbulent flows, therefore, solving NS equation directly will be computationally expensive due to fine-resolution requirements. After clarifying the objectives of this chapter (Section 2.1), basic turbulence theory is introduced briefly (Section 2.2). The methods which are typically used to model turbulence are presented (namely, LES (Section 2.3) and RANS (Section 2.4)). This chapter gives the reader an overview of the traditional mechanistic turbulence model used to mitigate the need for high-resolution grids. The confidence in either LES or RANS simulations is related to reducing CFD sources of error (See Figure 3). Among these sources of error, grid induced-error is the focus of this work as we propose to rely on coarse-grid simulations. Hence, the grid-induced error, in the simulations performed by LES, RANS and the proposed CG-CFD method, is discussed in Section 2.5 and the whole chapter is summarized in Section 2.6.

2.1. Objectives

The objectives of this chapter can be summarized as follows:

- Discussing the traditional mechanistic turbulence models.
- Providing an overview of the grid-induced error treatment in the proposed CG-CFD method vs. the corresponding error when applying traditional turbulence modeling.

2.2. Basics of the Turbulence Theory

Turbulence is unsteady and chaotic fluid motion in which the three velocity components fluctuate mixing matter, momentum, and energy. Turbulent flows always occur at higher Reynolds numbers when inertia forces overcome viscous forces. Turbulence can be imagined as a set of

المنارة للاستشارات

www.manaraa.com

eddies of different sizes. There are large scales where the energy enters the flow through mean shear; there is an inertial range (intermediate scales) where energy flows to the smaller scales of the dissipation range where the energy dissipates into heat due to molecular viscosity effects [12]. A typical example of the distribution of turbulence energy contained in the eddies of size l (corresponding to wavenumber $\kappa_l = \frac{2\pi}{l}$) is illustrated in Figure 4.

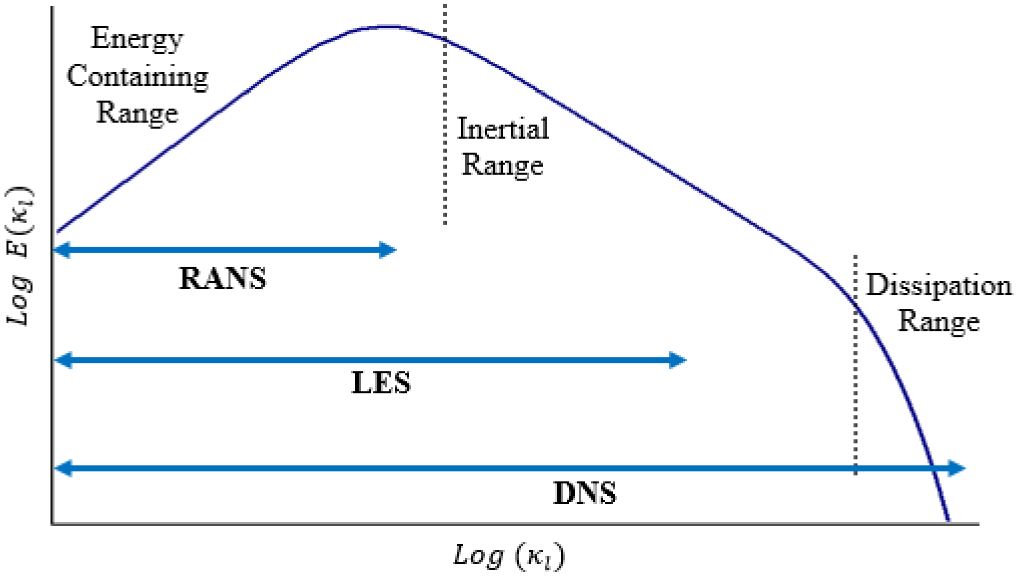


Figure 4. Distribution of turbulent energy in wavenumber space.

In order to represent the eddies belonging to various energy ranges, NS equations can be solved directly on a sufficiently fine grid using a small-time step to capture length scales down to Kolmogorov length scale (η) and time scales down to τ_η:

$$\eta = (\nu^3/\varepsilon)^{1/4} \tag{2-1}$$

$$\tau_\eta = (\nu/\varepsilon)^{1/2} \tag{2-2}$$

www.manaraa.com

This method (Direct Numerical Simulation, DNS) is the most accurate method of solving turbulent flows. It results in accurate three-dimensional instantaneous velocity and pressure numerical data. It is accurate for simple flows, but it is computationally challenging for real-world applications (turbulent flows with higher Reynolds number, Re). The number of grid points needed for DNS simulation is proportional to Re$^{9/4}$ [12] (see Figure 5). For most applications, it is unnecessary to resolve all the turbulence scales. It suffices to predict the effect of turbulence on mean flow behavior. Therefore, turbulence models are developed to close the system of the mean flow equations. Different turbulence modeling methods are described in Sections 2.3 and 2.4. It is worth noting that the averaged flow pattern is a statistical flow property (cannot be directly observed in nature for turbulent flows).

2.3. Large Eddy Simulation (LES)

LES method was proposed by Smagorinsky [16]. In this method, turbulent flow large eddies are resolved directly while the sub-grid scale eddies (the ones which cannot be resolved using the coarser grid) are modeled. This results in the reduced computational cost of LES compared to DNS. LES requires the smallest resolved spatial and temporal scales to lie in the inertial subrange of turbulence (see Figure 4). Large eddies contain most of the turbulent kinetic energy and are responsible for turbulent mixing and momentum transfer while the small eddies are isotropic (universal and less dependent on boundary conditions) which allows developing a nearly universal sub-grid model for them. The computational cost of LES (for free shear flow simulations) varies weakly with Reynolds number and is proportional to $Re^{0.4}$ [39]. For wall-bounded flows, LES computational cost (number of grid points) scales with $Re^{1.76}$ [39] (this computational cost is comparable to DNS as illustrated in Figure 5) . Thus, LES method allows to perform simulations with Reynolds number higher than those simulated by DNS using the same computational power.

المنارة للاستشارات

www.manaraa.com

In LES, any variable in NS equation, φ is spatially filtered to resolve the large turbulent eddies whose scales range from domain size to filter width while the eddies of scales between filter size and grid width are modeled. The filters are localized functions that include Gaussian filtering, box filtering (simple average over volume) and cutoff filtering (Fourier coefficients belonging to wavenumbers above the cutoff are eliminated). For example, using a box filter, the filtered variable $\bar{\varphi}$ is given by:

$$\bar{\varphi}(x) = \frac{1}{V}\int_{v}\varphi(x')dx', \qquad x' \in v \tag{2-3}$$

where V is the volume of a computational cell. The filtered NS equation (for incompressible flow) is:

$$\frac{\partial \overline{u_\imath}}{\partial t} + \frac{\partial \overline{u_\imath}\,\overline{u_\jmath}}{\partial x_j} = \frac{\partial}{\partial x_j}\left(\frac{\partial \overline{\sigma_{\imath\jmath}}}{\partial x_j}\right) - \frac{\partial \overline{p_k}}{\partial x_i} - \frac{\partial \overline{\tau_{\imath\jmath}}}{\partial x_j} \tag{2-4}$$

$$\overline{\sigma_{\imath\jmath}} = \nu\left(\frac{\partial \overline{u_\imath}}{\partial x_j} + \frac{\partial \overline{u_\jmath}}{\partial x_i}\right) - \frac{2}{3}\nu\frac{\partial \overline{u_\imath}}{\partial x_l}\delta_{ij} \tag{2-5}$$

$$\overline{\tau_{\imath\jmath}} = \overline{u_\imath u_\jmath} - \overline{u_\imath}\,\overline{u_\jmath} \tag{2-6}$$

where p_k and ν are the kinematic pressure and kinematic viscosity. $\overline{\tau_{\imath\jmath}}$ is the Sub-Grid Scale (SGS) stress tensor. One of the earliest models to solve $\overline{\tau_{\imath\jmath}}$ is the Smagorinsky turbulence model [16]. Similar to the dissipation due to dynamic fluid viscosity in a laminar flow, the concept of SGS viscosity, ν_{sgs}, is introduced to represent the dissipation effect of small scales as following:

$$\overline{\tau_{\imath\jmath}} - \frac{1}{3}\overline{\tau_{kk}}\delta_{ij} = -2\nu_{sgs}\overline{S_{\imath\jmath}} \tag{2-7}$$

www.manaraa.com

$$\overline{S_{ij}} = 0.5\left(\frac{\partial \overline{u_i}}{\partial x_j} + \frac{\partial \overline{u_j}}{\partial x_i}\right) \tag{2-8}$$

$$\nu_{sgs} = L_s^2 C_s^2 |S| \tag{2-9}$$

$$S = \sqrt{2\overline{S_{ij}S_{ij}}} \tag{2-10}$$

$$L_s = min\left[\frac{\kappa y}{C_\Delta}\left(1 - e^{-\frac{y^+}{A^+}}\right), V^{1/3}\right] \tag{2-11}$$

where $\overline{S_{ij}}$ is the strain rate of the resolved velocity field, L_s is the mixing length for sub-grid scales and C_s is a model constant (proposed to be 0.18) [40]. The SGS viscosity value is large near the wall because of the high velocity gradient at the wall. Near wall turbulent fluctuations go to zero so ν_{sgs} must go to zero as well. Therefore, Van Driest damping [41] is added (Eq. (2-11)). In this equation, κ is the von Karman constant, y is the closest distance to the wall, C_Δ and A^+ are model constants, V is the computational cell volume. This model has some limitations because C_s value needs to be changed for different flow patterns. It may take values from 0.065 to 0.25 [42].

To eliminate the need for tuning Smagorinsky constant, a dynamic model is developed by Germano and Lilly to [40] compute the Smagorinsky constant dynamically. In this model, the flow variables are filtered by 2 filters (grid filter and test filter) while the test filter is larger than the grid filter width. Filtering with a grid filter results in SGS tensor, $\overline{\tau_{ij}}$ (see Equation (2-6)). Computing the SGS tensor based upon test filter gives ${\tau_{ij}}^t$. Applying both filters (grid filter and test filter) yields another SGS tensor, T_{ij}:

$$T_{ij} = \overline{\overline{u_i u_j}} - \bar{\bar{u}}_i \bar{\bar{u}}_j \tag{2-12}$$

www.manaraa.com

$$L_{ij} = T_{ij} - {\tau_{ij}}^{t} \tag{2-13}$$

$$M_{ij} = -2\Delta^2 \left(\alpha^2 \overline{\overline{S_{\imath\jmath}}}\,\overline{\overline{|S|}} - \overline{\overline{S_{\imath\jmath}}\,\overline{|S|}}\right) \tag{2-14}$$

$$C_s^2 = \frac{\langle L_{ij}\, M_{ij} \rangle}{\langle M_{ij} M_{ij} \rangle} \tag{2-15}$$

where Δ the test is scale and $\alpha\Delta$ is the resulted scale from applying the two filters. This model assumes that Smagorinsky constant value at Δ and $\alpha\Delta$ is the same (scale invariance). In this way, C_s can be computed for each grid cell and each time step. However, this may cause ν_{sgs} value to be negative in some locations [43]. In general, LES is capable of simulating separated flow but it is not affordable when simulating near-wall flow at high Reynolds numbers. The filtered variables are not averaged over time, so the flow variables are functions of time. Therefore, LES can compute transient flows. LES is still much more expensive than RANS which is the most used approach in industrial CFD (see Section 2.4). As depicted in Figure 5, the number of grid points needed to simulate wall-bounded flows, using RANS approach, is proportional to $ln(Re)$ [12].

المنارة للاستشارات

www.manaraa.com

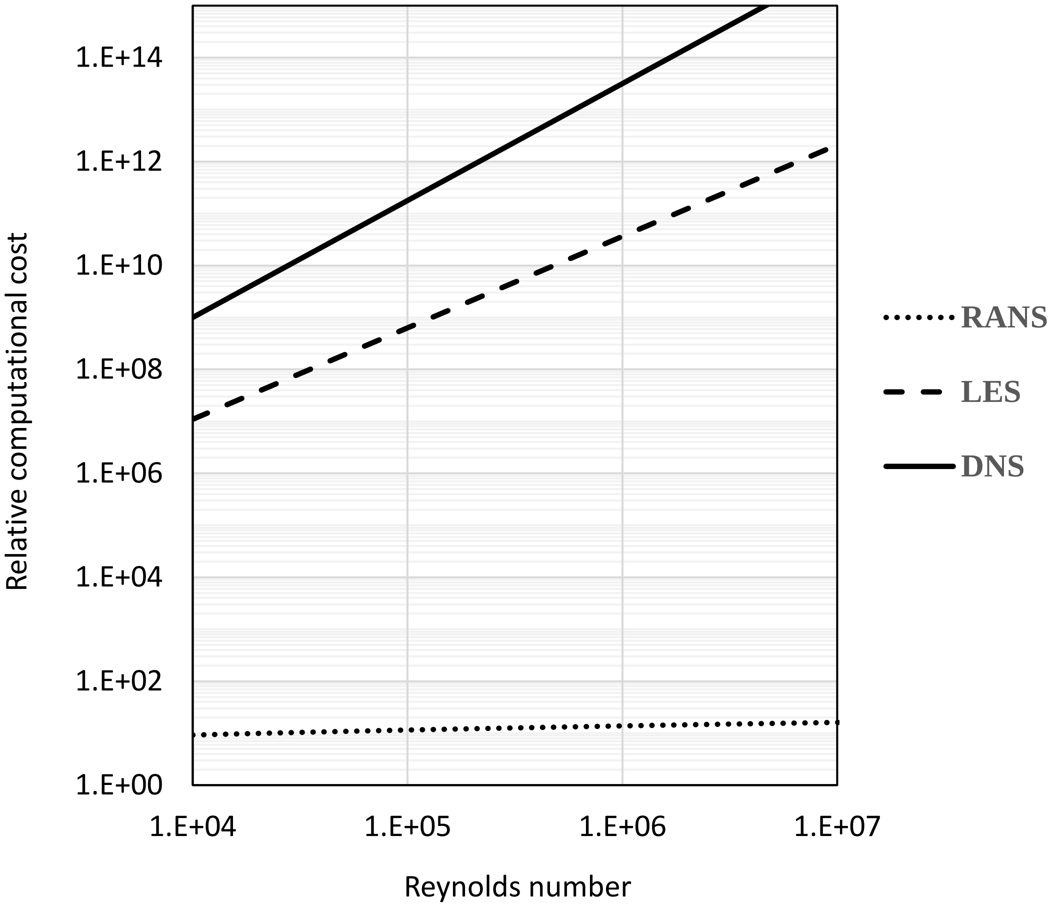


Figure 5. Computational expense (of DNS, LES, and RANS) as a function of Reynolds number.

2.4. Reynolds Averaged Navier-Stokes (RANS) Equations

RANS is the workhorse tool in turbulence modeling in industry. Reynolds decomposition of the velocity into a mean velocity component and a turbulent fluctuating component is written as:

$$u = \overline{u} + u' \tag{2-16}$$

The same way of decomposition is applied to pressure. Applying Reynolds decomposition on NS Equation (2-17) results in RANS Equation (2-18).

المنارة للاستشارات
www.manaraa.com

$$\frac{\partial u_i}{\partial t} = -u_j \frac{\partial u_i}{\partial x_j} - \frac{1}{\rho}\frac{\partial p_k}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j \partial x_j} \tag{2-17}$$

$$\frac{\partial \overline{u_i}}{\partial t} = -\overline{u_j}\frac{\partial \overline{u_i}}{\partial x_j} - \frac{1}{\rho}\frac{\partial \overline{p_k}}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j \partial x_j} - \frac{\overline{\partial u_i' u_j'}}{\partial x_j} \tag{2-18}$$

NS Equation (2-17) is expressed in tensor notation (for incompressible Newtonian fluid). The right side of the equation is the momentum time rate of change while the right-hand side is the summation of the inertial force term (momentum transport because of the bulk fluid motion), the pressure force and the viscous force. In RANS Equation (2-18), the last term of the right-hand side is called Reynolds stresses [12]. This term is modeled according to Boussinesq approximation that Reynolds stresses are proportional to the velocity gradient of the mean flow:

$$\overline{u_i' u_j'} = \nu_t \left(\frac{\partial \overline{u_i}}{\partial x_j} + \frac{\partial \overline{u_j}}{\partial x_i} \right) - \frac{2}{3} k \delta_{ij} \tag{2-19}$$

$$\delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \tag{2-20}$$

where k is the turbulent kinetic energy and ν_t is the turbulent viscosity. The closure problem is to model ν_t. An overview of the common RANS turbulence models that predict ν_t are listed in Table 3. All these models (LES or RANS) still lack accuracy or/and affordability when simulating complex long transient flows similar to CTH flows. These models are also calibrated on limited data. Hence, data-driven statistical models developed by ML algorithms (see Chapter 4) are suggested in this work.

www.manaraa.com

Table 3. Different RANS turbulence models.

Model	Equation	Applications
Zero equation Model [44]	$\nu_t = l_m^2 \left(\frac{du}{dy}\right)$ (2-21) This equation is based on dimensional analysis. Velocity scale is proportional to the velocity gradient, (du/dy) and mixing length, l_m. l_m is introduced (the length over which there is a high interaction of vortices in a turbulent flow field). It is a problem-specific parameter.	It works well only for flows that are characterized by a single l_m.
One equation models	$\nu_t \approx l_m \sqrt{k}$ (2-22) Velocity scale is proportional to the square root of the kinetic energy, k. One differential transport equation is developed for TKE [12]: $\frac{\partial k}{\partial t} + U_i \frac{\partial k}{\partial x_i} = \frac{1}{\sigma} \frac{\partial}{\partial x_j} \left[(\nu + \nu_t) \frac{\partial k}{\partial x_i} \right] + P(k) - D(k)$ (2-23) Another model is the Spalart-Allmaras model where a transport equation is written for ν_t	Performs poorly when l_m changes rapidly from wall to free shear region. It does not work for free jets, adverse pressure gradient and separated flows.
Two-equation model: *($k - \varepsilon$)* [19]	$\nu_t = C_\mu (k^2/\varepsilon)$ (2-24) Two differential transport equations are developed for TKE, k (like equation (2-23)) and the turbulent dissipation rate, ε: $\frac{\partial \varepsilon}{\partial t} + U_i \frac{\partial \varepsilon}{\partial x_i} = \frac{\partial}{\partial x_i} \left[\left(\nu + \frac{\nu_t}{\sigma_\varepsilon} \right) \frac{\partial \varepsilon}{\partial x_i} \right] + \frac{C_{\varepsilon 1} f_{\varepsilon 1} \varepsilon}{k} P_k + \frac{C_{\varepsilon 2} f_{\varepsilon 2} \varepsilon^2}{k}$ (2-25)	More accurate in free shear flows. Poor in rotating flows. Parameters need tuning.

المنارة للاستشارات
www.manaraa.com

Table 4 (continued). Different RANS turbulence models.

$k-\omega$ [45]	A transport equation for ω is developed similar to ε transport Equation, (2-25) given that $\approx \frac{\varepsilon}{k}$.	Derived for wall-bounded flows. Sensitive to inlet boundary conditions
Shear Stress Transport (k − ω) [46]	Behaves like $k-\omega$ near the wall and switches to $k-\varepsilon$ model away from the wall. It is called Shear Stress Transport (SST) model because it is based upon the shear stress relationship that the shear stress is proportional to the turbulent kinetic energy.	Works for separation flow and adverse pressure gradient flows.

2.5. Grid-Induced Error in LES, RANS, and CG-CFD

The confidence in CFD results utilizing either RANS or LES is determined by two major factors: the physical modeling error and the numerical error. The physical modeling error is attributed to the model formulation, simplifications or assumptions made when developing the model. The numerical error includes round-off error, iterative error and discretization error. Minimizing the discretization error brings attention to the grid size. In this section, the grid refinement, required for traditional approaches (LES and RANS) and for the proposed approach (CG-CFD), is discussed.

2.5.1. LES

Given sufficiently fine grids, LES approach is capable of producing accurate results but the challenge here is the difficulty of determining the sufficient resolution without some knowledge about the flow behavior [47]. Attempting to perform successive grid refinement to get grid-independent solution will lead to resolving more small-scale turbulent eddies which will

المنارة للاستشارات

www.manaraa.com

finally transition to DNS as stated by Celik [48]: "A good LES is that which tends to DNS as the grid resolution tends to Kolmogorov scales. Therefore, there is no such thing as grid-independent LES in theory, because a grid-independent LES is essentially DNS, and the philosophy of LES loses its meaning if it is grid independent;".

Practically, LES is done based on the following criteria: (1) the computational resource limit, (2) the scale of the phenomena of interest (based on prior knowledge of physics), (3) dimensionless wall spacing, Y^+(see NOMENCLATURE section for Y^+ definition). For instance, for wall resolved LES, the grid requirements, in a flat plate domain, are: $\Delta x^+ < 150, \Delta y^+ < 1$, $\Delta z^+ < 40$ which are significantly less demanding than DNS whose criteria are: $\Delta x^+ < 20, \Delta y^+ < 1,\ \Delta z^+ < 10$. Here x, y, z, directions are referring to the stream-wise, wall normal and span-wise homogenous directions [49]. It is also necessary to assure that the cutoff wavenumber is in the inertial sub-region (at least in the regions of interest in the simulated domain).

2.5.2. RANS

To estimate the discretization error, it is needed to find the asymptotic range where the solution is approximately independent. The existence of the asymptotic range, for quasi-steady state turbulent flow simulated by RANS with eddy viscosity models, was demonstrated [50]. Therefore, it is often required to demonstrate grid convergence (grid-independent solution) with RANS approach contrary to LES where grid convergence is not applicable [49]. When using wall functions with RANS, the first node from the wall should lie in the log region of the boundary layer ($200 > y^+ > 30$) otherwise y^+ is adjusted to be 1. If $y^+ < 30$, when using wall functions, wall shear stress and/or wall heat transfer will be impacted. Hence, grid convergence study, in the presence of wall functions, fails) [47]. In general, given enough grid, LES gives more accurate

المنارة للاستشارات
www.manaraa.com

results compared to RANS. In some cases, physical error and numerical error compensate for each other so better performance is obtained at coarser grids [47].

2.5.3. CG-CFD

In the present work (CG-CFD), the goal is not eliminating discretization error and performing grid convergence study to obtain a grid-independent solution. Instead, the simulation is performed with a coarse grid which is expected to result in inaccurate result that is corrected by a data-driven model. In the proposed CG-CFD method, we are trying to answer these questions:

- Is it possible to predict the coarse-grid induced error given some “local features” of the coarse-grid results?
- Which local features will be selected (velocity, pressure or other features)?
- Is it feasible to use the computed grid-induced error to obtain sufficiently accurate solution?

In this dissertation, we address and discuss these questions. In CG-CFD, it is not expected to capture the small-scale phenomena that are resolved by DNS. The goal is to predict the grid-induced error in the large-scale phenomena. It is expected that the large-scale phenomena are usually the phenomena we are interested in. Hence, the coarseness of the grids utilized in CG-CFD is limited by the scale of the phenomena of interest so the utilized grids maybe coarser than those corresponding to RANS. CG-CFD framework is presented in Chapter 5. Numerical results that illustrate the feasibility of the proposed CG-CFD method are presented in Chapter 6.

2.6. Chapter Summary

Turbulence effects are typically modeled by Large Eddy Simulation (LES) or Reynolds Averaged Navier-Stokes (RANS) equations. In LES, turbulent flow large eddies are resolved directly while the sub-grid scale eddies (the ones which cannot be resolved using the coarser grid) are modeled. In general, LES is capable of simulating separated flow but it is not affordable when

المنارة للاستشارات
www.manaraa.com

simulating near wall flow at high Reynolds numbers. LES is still much more expensive than RANS which is the most used approach in industrial CFD. Each RANS turbulence models is calibrated for specific flow conditions (given specific experimental data). Hence, there is a need for data-driven models that can be adaptive to different flow conditions.

The grid-induced error is a major source of uncertainty which is traditionally reduced by performing grid convergence studies. For LES approach, attempting to perform successive grid refinement to get grid-independent solution will lead to resolving more small-scale turbulent eddies which will finally transition to DNS. Thus, LES computational cost may be comparable to DNS cost. For RANS approach, grid convergence study has some limitations because of the presence of the wall functions. For the proposed approach, in this work, the goal is not eliminating discretization error or performing grid convergence study to obtain a grid-independent solution. Instead, the simulation is performed with a coarse grid which is expected to result in inaccurate result. Next, the grid-induced error is predicted using a trained data-driven model.

المنارة للاستشارات
www.manaraa.com

3. NUMERICAL SOLUTION ERROR

There are various sources of CFD errors (as was illustrated in Chapter 1 (in Figure 3)). The focus of this work is predicting the discretization error which arises when solving the governing equations numerically.

Once the physical problem of interest is defined, the Partial Differential Equation (PDE) is created with well-defined initial and boundary conditions. The PDE in flow dynamics' problems is represented by the NS equations (mass and momentum conservation laws) which include nonlinear terms and cannot be solved analytically for most practical cases. That is why NS equations are solved numerically. The numerical solution approach includes discretizing the problem domain by generating a computational grid and the discrete model is obtained and solved.

After clarifying the objective of this chapter (Section 3.1), the methods, used to discretize the NS equations, are presented in Section 3.2. The discretized equations are solved using iterative methods (see Section 3.3). Discretization error and iterative convergence error are discussed in Sections 3.4 and 3.5, respectively.

3.1. Objectives

The objectives of this chapter can be summarized as follows:

- Studying the major steps for solving Navier-Stokes equations numerically.
- Investigating the issues that arise due to the discretization error (mainly, the grid-induced error).

3.2. Discretization Methods

In this section, the methods, which are utilized to approximate the PDEs by a system of linear algebraic equations, are presented briefly. The presented methods include the Finite Difference Method (FDM), the Finite Element Method (FEM) and the Finite Volume Method

المنارة للاستشارات
www.manaraa.com

(FVM). FDM and FEM are presented briefly in Sections 3.2.1 and 3.2.2, respectively. Because FVM is the method which is used in this work (embedded in CFD software, OpenFOAM [10]), FVM is explained in more detail in Section 3.2.3.

3.2.1. Finite Difference Method (FDM)

In FDM, the flow variables and their derivatives are approximated at each point through the Taylor expansion of each variable around a point. For instance, if we consider a partial derivative $\frac{\partial \varphi}{\partial x}$, the Taylor-series expansion of φ around a point x_i can be written as:

$$\varphi(x) = \varphi(x_i) + \Delta x \left(\frac{\partial \varphi}{\partial x}\right)_i + \frac{(\Delta x)^2}{2}\left(\frac{\partial^2 \varphi}{\partial x^2}\right)_i + \cdots \tag{3-1}$$

The higher order terms in the expansion can be neglected so the derivative of φ is given by:

$$\frac{\partial \varphi}{\partial x} = \frac{\varphi(x_i + \Delta x) - \varphi(x_i)}{\Delta x} + \cdots \tag{3-2}$$

Calculating the first order derivative by neglecting (truncating) the higher order terms is what causes the truncation error. Higher-order derivatives can be derived in a similar way. Estimating the derivatives based on the Taylor series can be done with different schemes. The scheme in Equation (3-2) is called forward finite difference scheme. Examples of the other schemes include the backward difference and the central difference schemes.

After the derivatives are approximated using the present approach, the PDEs can be represented in the form of the system of linear equations. This system of algebraic equations is solved at each grid point. This system can be written in a matrix form as:

$$\mathrm{A}\varphi = Q \tag{3-3}$$

المنارة للاستشارات
www.manaraa.com

This system is solved with a matrix solving scheme given the initial conditions and the equations written for the boundary nodes. The FDM requires structured grids and cannot be applied with curvilinear coordinates (must be converted to Cartesian coordinates first). Because FDM has difficulties with irregular geometry, it is mainly used with simple geometry problems (e.g. box-type domains) and it is utilized sometimes with DNS. Besides, FDM is not an inherently conservative method (does not conserve mass or temperature) and that is why it is rarely used in CFD industrial applications [51].

3.2.2. Finite Element Method (FEM)

FEM is a weighted residual method: If a general PDE is written as: $L(\varphi) = 0$, it is assumed that the equation has an approximate solution, $\bar{\varphi}$ which has a form:

$$\bar{\varphi} = a_0 + a_1 x + a_2 x^2 + \cdots .. \tag{3-4}$$

where a_0, a_1, a_2,.. are unknown coefficients that need to be computed. Using initial coefficients, $\bar{\varphi}$ will not satisfy the original PDE therefore the residual, R is defined as: $L(\bar{\varphi}) = R$. In FEM, the residual is driven to equal zero, using weight functions W as follows:

$$\int WRdx = 0 \tag{3-5}$$

In this way, PDEs are integrated over an element after being multiplied by a weight function. Because the continuous PDE is divided into a finite number of elements, the method is called the Finite element method. FEM can be used with complex geometries [52]. The solution is obtained at element nodes, and typically linearly approximated within each element to obtain the continuous distribution of state variables in the domain.

المنارة للاستشارات
www.manaraa.com

3.2.3. Finite Volume Method (FVM)

Unlike FEM and FDM, the variables' values are computed at the centers of the grid cells on the meshed domain and the equation is integrated over the volume of the grid cell. Applying FVM method with NS equation, NS equation is integrated over the computational cell with the assumption that the time-dependent term in NS equation is constant over the cell volume as follows:

$$\iiint_V dV \left[\frac{\partial u_i}{\partial t} + \frac{\partial u_i u_j}{\partial x_j}\right] = \iiint_V dV \left[-\frac{\partial p_k}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j \partial x_j}\right] \tag{3-6}$$

V is the cell volume. The divergence theorem is applied so the volume integrals are replaced by surface integrals:

$$V\frac{\partial u_i}{\partial t} + \iint_A dA u_i u_j n_j = -\iint_A dA p_k n_i + \iint_A dA \nu \frac{\partial u_i}{\partial x_j} n_j \tag{3-7}$$

A is the surface area of the cell, n is the cell's surface normal vector. The integrated NS equation consists of four terms which need to be linearized. The four terms are (from left to right): the time dependent term, the convection term, the pressure term and the diffusion term. These four terms are linearized with different schemes which are explained in the following subsections [53]:

<u>The time-dependent term</u>

The time change of the velocity u_i or any variable φ can be discretized by the implicit Euler scheme:

$$\frac{\partial \varphi}{\partial t} = \frac{\varphi_P^n - \varphi_P^o}{\Delta t} \tag{3-8}$$

The superscripts o and n refer to the variable values which are stored from the previous time step and the time step we solve for respectively. The subscript P is referring to the center of the cell of

المنارة للاستشارات
www.manaraa.com

interest. The Euler time integration scheme is a first-order accurate scheme (numerical scheme is nth order accurate if the error is proportional to $(\Delta t)^n$). For getting higher-order accuracy (second order), the backward difference scheme can be used:

$$\frac{\partial \varphi}{\partial t} = \frac{3\varphi_P^n - 4\varphi_P^o + \varphi_P^{oo}}{2\Delta t} \tag{3-9}$$

Where the superscript oo refer to the variable value stored from a time step previous to the last.

The convective term

The convective term of the velocity component u_i or any convected variable φ is discretized as follows:

$$\iint_A dA\varphi u_j n_j = \sum_f (A_f . U_f)\varphi_f \tag{3-10}$$

where the subscript f refer to the computational cell faces (6 faces for a cubic cell). $(A_f \cdot U_f)$ is the dot product of the velocity vector and the face area at each face. The face field φ_f can be computed by various schemes such as Central Differencing (CD):

$$\varphi_f = f_x\varphi_P + (1 - f_x)\varphi_N \tag{3-11}$$

P and N are the centers of the cell of interest and the neighboring cell. f_x is the distance between the face f and cell center N divided by the distance between cell centers P and N. CD is a second order accurate scheme, but it is unbounded. Being unbounded means that it may result in a solution which does not lie in the physical bounds. Before convergence is reached, CD scheme may result in a solution which does not satisfy the continuity equation (unbounded). Hence, the convergence rate and the stability of the solution can be impacted.

المنارة للاستشارات
www.manaraa.com

Another alternative is the Upwind Differencing (UD) which is bounded but not very accurate. It determines φ_f based on flow direction as follows:

$$\varphi_f = \begin{cases} \varphi_P \; for \; (A_f.U_f) \geq 0 \\ \varphi_N \; or \; (A_f.U_f) < 0 \end{cases} \tag{3-12}$$

It is also possible to mix both CD and UD in what is called the Blended Differencing (BD). In this way, boundedness and reasonable accuracy are maintained as follows:

$$\varphi_f = (1-\gamma){\varphi_f}^{UD} + \gamma{\varphi_f}^{CD} \tag{3-13}$$

where the value of the blending coefficient, γ is specified by the user $(0 > \gamma > 1)$.

<u>The pressure term</u>

The pressure term is linearized as follows:

$$-\iint_A dAp_k n_i = -\sum_f {p_k}_f . A_f \tag{3-14}$$

<u>The diffusion term</u>

The diffusion term (Laplacian term) of the velocity component u_i or any variable φ is linearized as follows:

$$\iint_A dA\nu \frac{\partial\varphi}{\partial x_j} n_j = \nu \iint_A dA.(\nabla\varphi) = \nu \sum_f A_f.(\nabla\varphi)_f \tag{3-15}$$

$$A_f.(\nabla\varphi)_f = |A_f| \frac{\varphi_N - \varphi_P}{|d|} \tag{3-16}$$

where d is the distance between the cell centers N and P.

www.manaraa.com

3.3. Iterative Methods

Solving incompressible NS equations, including the continuity equation ($\nabla \cdot U = 0$), is not a straightforward process because there is no explicit equation for the pressure. A pressure equation can be derived by taking the divergence of the momentum equation and substituting in the continuity equation as follows.

The NS equation is written in a semi-discretized form (using the numerical schemes presented in section 3.2):

$$a_P U_P = H(U) - \nabla p_k \tag{3-17}$$

$$H(U) = -\sum_n a_N U_N + \frac{U^o}{\Delta t} \tag{3-18}$$

where a_P and a_N are the matrices of the coefficients of the cell of interest P and the neighboring cells, N. The momentum equation is interpolated for the face velocity, U_f:

$$(a_P)_f U_f = (H(U))_f - (\nabla p_k)_f \tag{3-19}$$

Substituting Equation (3-19) into discretized continuity equation (Equation (3-20)), the pressure equation is obtained (Equation (3-21)).

$$\sum_f A_f . U_f \tag{3-20}$$

$$\nabla . \left(\frac{1}{(a_p)_f} (\nabla p_k)_f \right) = \nabla . \left(\frac{(H(U))_f}{(a_p)_f} \right) \tag{3-21}$$

Iterative procedures are necessary to solve the discretized pressure equation and the momentum equation together. There are various algorithms to do these procedures. Here, Pressure

المنارة للاستشارات
www.manaraa.com

Implicit Splitting of Operators (PISO algorithm) is presented as an example of how the pressure-velocity coupling is treated.

The PISO algorithm

PISO algorithm [54] is described as follows:

- Momentum predictor: Initially the pressure field is unknown so the pressure field from the previous time step is used to solve the momentum equation to get an estimation of the velocity fields.
- Pressure solution: Given the estimated velocity field, the pressure equation is solved to obtain a new pressure field.
- Velocity correction: Based on the new pressure field, the velocity field is corrected using the pressure equation.
- This loop (momentum predictor, pressure correction, and velocity correction) is repeated until convergence is achieved.

3.4. Discretization Error

Representing the flow PDE equations in a discrete form results in the discretization error which includes the errors due to the spatial and temporal approximations. The discretization error is eliminated when the grid spacing and the time step tend to go to zero (which would make the computational cost infinite). To determine the discretization error in a particular solution, a grid convergence study is performed (a temporal convergence study is done as well).

Grid convergence study

Grid convergence examination requires performing the simulation with a set of successively finer grids. As the grid is refined, the discretization error approaches zero asymptotically. The behavior of the solution error is characterized by the grid convergence order.

المنارة للاستشارات
www.manaraa.com

The discretization error, ε_Δ, is related to the grid spacing, Δ, and the convergence order, p, as follows [55]:

$$\varepsilon_\Delta \approx \Delta^p \qquad (3\text{-}22)$$

For instance, the solution is a second order correct if $p = 2$. The value of p can be estimated theoretically based on the used numerical schemes. However, the actual order of convergence is usually lower that the theoretical one because of the boundary conditions. When determining the value, p, the ratio $(\varepsilon_\Delta/\Delta^p)$ is an indication of the asymptotic range. Asymptotic range is reached when the ratio $(\varepsilon_\Delta/\Delta^p)$ gets constant.

Discretization error estimation

Usually the discretization error is difficult to estimate. Various approaches have been proposed to perform this task. The discretization error estimators include Richardson extrapolation [56], adjoint methods, and discretization error transport equation [55]. Reliability of the error estimator demands to have numerical results that lie in the asymptotic range. Reaching the asymptotic range is computationally expensive for the three-dimensional NS equations.

3.5. Iterative Convergence Error

As explained in Section 3.3, the momentum and pressure equations are solved by iterative algorithms. The iteration loop continues until a convergence criterion is achieved. The convergence criterion is typically one the following criteria:

- Residual value: The residual is computed by substituting with the numerical result in the numerical equations at each grid cell. Theoretically, the residual should equal zero but in reality, it never gets to zero. The value of the acceptable maximum residual is typically specified by the user.

المنارة للاستشارات
www.manaraa.com

- The quantity of interest convergence: The user may be interested only in one quantity (for instance, the maximum temperature over the domain). Therefore, the Quantity Of Interest (QOI) can be monitored to assure that that the change in QOI with iteration is smaller than a maximum allowed value.

The iterative convergence error cannot be completely removed because a stopping criterion must exist. Although it is possible to reduce the iterative convergence error by reducing the residual criterion (for instance), the number of iterations will increase with the reduction in the residual criterion value. This results in a high computational cost.

3.6. Chapter Summary

The Navier-Stokes equations are approximated by a system of linear algebraic equations. Discretization methods which are utilized to approximate the Navier-Stokes equations include Finite Difference Method (FDM), the Finite Element Method (FEM) and the Finite Volume Method. FDM method is rarely used in CFD industrial applications. FEM and FVM can be used with complex geometries.

Solving the discretized Navier-Stokes equation requires iterative procedures. Both discretization methods and iterative algorithms result in numerical errors. Reducing the discretization error (by refining the grid and/or the time step) or reducing the iterative convergence error (by achieving convergence criteria) leads to a higher computational cost.

The focus of this research is the discretization error. Typical discretization error estimators (like Richardson extrapolation) demands to have numerical results that lie in the asymptotic range. Reaching the asymptotic range is computationally expensive for three-dimensional Navier-Stokes equations. Hence, in this work, a data-driven method is proposed to predict the discretization error

المنارة للاستشارات
www.manaraa.com

(grid-induced error) given the results computed by the grids which are away from the asymptotic convergence range.

المنارة للاستشارات
www.manaraa.com

4. MACHINE LEARNING

Traditionally, phenomena in a nuclear reactor and containment thermal hydraulics have been investigated by means of scaled experiments, including separate-effect tests and integral-effect tests. More recently, due to the increasingly affordable computing power and the advances in numerical solution methods and tools, there is a growing interest and application of CFD for modeling and simulation of separate thermal hydraulic processes and components of nuclear power plants. Examples of experimental studies of thermal-hydraulic effects, such as BWR suppression pool thermal stratification and hydrogen mixing and distribution can be found in [57, 58]. Also, examples of CFD studies of buoyant convection and suppression pool direct contact condensation can be found in [59, 60].

The experimental studies and high-fidelity simulations result in a large amount of data (a lot of flow variables distributed in space and time) which is useful but not optimally usable. It is hard to explore/ analyze/ model/ exploit the “big data” generated by high-resolution validated numerical simulations and physical experiments. ML algorithms are used to find patterns in data and build new models that predict new outcomes based on historical data. In general, ML algorithms are classified depending on the available information and the user purpose in two types: supervised learning and unsupervised learning. Supervised learning demands having a set of input variables (also called features) and their corresponding outputs. These inputs and outputs help to train a model that could predict the output for a new input. Supervised learning includes applications like regression and classification. On the other side, unsupervised learning is more challenging because the observations (raw data) are just given without identifying inputs or outputs. Its function is to extract the internal relationships in the dataset. For instance: grouping

المنارة للاستشارات

www.manaraa.com

the data (clustering) or recognizing the anomalies in the dataset or discovering the association between some variables.

ML algorithms are also classified as categorical or continuous depending on the type of data. Numerical data like velocity or pressure represent continuous data while the type of the flow (laminar/turbulent) or (bubbly/slug/annular) represent categorical data. A general taxonomy of ML applications, with examples of some algorithms, is illustrated in Table 5 [61]. In this chapter, an overview is presented for the different ML algorithms. Among these algorithms, Artificial Neural Network (ANN) and Random Forest Regression (RFR) techniques are used in this work for regression. Therefore, ANN and RFR are explained in more detail.

Table 5. Taxonomy of ML algorithms.

	Continuous	**Categorical**
Supervised	Regression (Section 4.5) • Artificial Neural Network (ANN). • Deep Learning (DL). • Random Forest Regression (RFR).	Classification (Section 4.4) • Support Vector Machines (SVM).
Unsupervised	Clustering (Section 4.2) • K-means clustering. Dimensionality reduction (Section 4.3) • Principal Component analysis (PCA).	Association analysis[1] • Apriori algorithm.

4.1. Objectives

The objectives of this chapter can be summarized as follows:

[1] Association analysis algorithms are not presented in this dissertation as they have no applications in fluid dynamics (to the extent of our knowledge).

المنارة للاستشارات
www.manaraa.com

- Providing an overview of the machine learning algorithms that can be used for CFD applications.
- Discussing the machine learning regression algorithms that are used in this work (artificial neural network and random forest regression).

4.2. Clustering

Clustering is a process of grouping the data in a way that each group (cluster) of data includes data points that are closer to each other. It is one of the data mining techniques that may precede the *regression* process because sometimes the entire data set cannot be represented by a single model. Therefore, the data are clustered into different groups and each group is represented by a different correlation.

One of the clustering techniques is the k-means clustering, or Lloyd's algorithm [62]. The k-means clustering is an iterative algorithm which assigns n data points to one of the k clusters. Each cluster is defined by a centroid and k is the number of the clusters (chosen before the algorithm starts). The algorithm can be summarized in the following steps:

1. Initial k centroids are selected.

2. Point to cluster centroid distance for each data point is computed.

3. Each observation is assigned to the cluster with the closest centroid.

4. The average of observations in each cluster is computed to gain new centroids.

5. Steps 2, 3 and 4 are repeated until cluster assignments do not change.

To determine how well the groups are separated, a *silhouette* plot is used. The silhouette value for any point is a metric of the point similarity to other points in the same group, compared to points in other groups. The silhouette value S_i for point i is:

المنارة للاستشارات
www.manaraa.com

$$S_i = \frac{B_i - A_i}{max(B_i, a_i)} \tag{4-1}$$

where A_i is the average distance from the ith point to other points in the same cluster. B_i is the minimum average distance from ith point to points in a different cluster (averaging is performed for each single cluster and minimization is done over clusters). S_i value ranges from -1 to 1. If S_i is closer to 1, this indicates that this ith point is well matching to its cluster. Negative silhouette values indicate that the point was assigned to the wrong cluster.

4.3. Dimensionality Reduction

Given a high-dimensional dataset, it would be useful to reduce the dimensionality of the data by extracting the features which are more important. This will allow faster computations and easier data regression (if needed). It also helps to visualize the data and reduce the storage space. It may even give insights into physics by investigating why some features impact the figure of merit more than other features.

One of the popular techniques for dimensionality reduction is the Principal Component Analysis (PCA) [63]. The Principal Components (PCs) are the new set of features which are generated to replace the original features. Each PC is a linear combination of the original features. The first PC is the axis of the highest variability of data while the second PC is the next orthogonal (uncorrelated) direction with the highest variability and so on.

PCA is performed through these procedures:

1. Given a data matrix $X_{k\times N}$ (k features and N samples), a covariance matrix, $S_{k\times k}$ for the data, X is computed.

2. An equation for the eigenvectors e and the eigen-functions λ is solved as follows:

المنارة للاستشارات
www.manaraa.com

$$Se = \lambda e \tag{4-2}$$

$$P = Xe \tag{4-3}$$

where e is the eigenvectors matrix (also named empirical orthogonal function). The eigenvectors (orthogonal functions) are the new system of coordinates in which the data can be described instead of the original variables. The eigenvalues, λ represent the importance of each orthogonal functions so the functions with smaller eigen-values can be ignored (that is how dimensionality is reduced). The principal component values, P at any point represent how much each orthogonal function is more dominant at this point. PCA approach was used in the field of fluid dynamics to produce simplified (reduced order) models given high-fidelity CFD results (see [64]).

4.4. Classification

A classification problem is a problem in which all the available observations can be categorized in some distinct classes (for instance: incompressible vs. compressible flow or laminar vs. turbulent flow) and the function of the classification algorithm is to find the distinction or the separating line between the different classes. Given this separating line, the classification algorithm predicts the category to which a new data point belongs. One of the most successful ML algorithms for classification is Support Vector Machine (SVM) [65] which is explained here as an example of the classification algorithm.

المنارة للاستشارات
www.manaraa.com

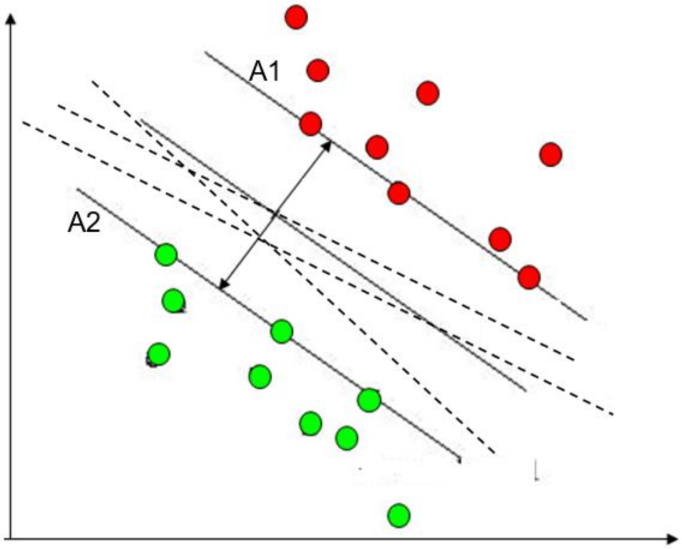


Figure 6. Illustration of the classification problem: various dashed lines separating two groups of data.

As depicted in Figure 6, for two groups of data points (red and green points), there is a lot of possible hyper-planes (or dashed lines for two-dimensional space) separating the two groups. SVM algorithm works to find the optimal separating plane. Given input feature, x_i, the output is taken to be $y_i \in \{-1, +1\}$ while 1 and -1 refer to the green points and red points respectively. Any hyper-plane separating the two groups of points is described by an equation:

$$\vec{w}^T \cdot \vec{x} + b = 0 \tag{4-4}$$

while the vector $\vec{w}$ and the constant b characterize the hyper-plane. The aim is finding a hyper-plane which can classify the data as follows:

$$\vec{w}^T \cdot \vec{x_i} + b > 1 \text{ when } y_i = 1 \tag{4-5}$$

$$\vec{w}^T \cdot \vec{x_i} + b < -1 \text{ when } y_i = -1 \tag{4-6}$$

The two lines (or hyper-planes), that are represented by solid lines and named $A1$ and $A2$ in Figure 6, are identified respectively by the equations:

المنارة للاستشارات
www.manaraa.com

$$\vec{w}^T \cdot \overrightarrow{x_\iota} + b = 1 \tag{4-7}$$

$$\vec{w}^T \cdot \overrightarrow{x_\iota} + b = -1 \tag{4-8}$$

The red or green points on planes $A1$ and $A2$ are the support vectors. The support vectors are the points that matter for the training process and the other points can be ignored. The perpendicular distance between $A1$ and $A2$ is called the margin. The bigger the margin, the better the classification process. The margin, D can be given by:

$$D = \frac{2}{||\vec{w}||} \tag{4-9}$$

Therefore, the margin is maximized by minimizing the magnitude of the vector $\vec{w}$ while satisfying the two conditions (Equations (4-7), (4-8)). This becomes an optimization problem to find the optimum $\vec{w}$ based on the given training data. Thus, the optimal separating hyper-plane is found. SVM is an accurate classification tool that can deal with the high-dimensional dataset (even when the number of samples is fewer than the number of features) but it is sensitive to noisy data. SVM can be used as a classification tool in CFD applications. For instance, in [66], the classification ability of SVM is used for identification of two-phase flow regimes.

4.5. Regression

Regression is one of the most common uses of machine learning. Examples of the regression algorithms include Artificial Neural Network (ANN) (Section 4.5.1), Deep Learning (DL) (Section 4.5.2) and Random Forest Regression (RFR) (Section 4.5.3).

4.5.1. Artificial Neural Network (ANN)

Artificial Neural Network (ANN) is a well-known non-linear adaptive regression tool that identifies a function that relates the given inputs and outputs [25]. ANN structure was originally

المنارة للاستشارات

www.manaraa.com

intended to mimic that of a human brain. ANN consists of computational units that are named artificial neurons. The biological neuron receives signals and activates if the signal is strong (exceeds a threshold), and then sends the signal to another neuron, and so on. The artificial neuron executes a similar process: it receives input variables which are multiplied by weights (which modify the signal strength) and are then processed by a mathematical activation function (like neuron activation) to produce the output.

<u>ANN structure</u>

ANN structure can be represented mathematically as follows [25]:

$$\underbrace{\boldsymbol{a}}_{S\times 1} = f(\underbrace{\boldsymbol{w}}_{S\times R}\underbrace{\boldsymbol{X}}_{R\times 1} + \underbrace{\boldsymbol{b}}_{S\times 1}) \tag{4-10}$$

$$\underbrace{\boldsymbol{Y}}_{\mathbf{1\times 1}} = \underbrace{\boldsymbol{\alpha}}_{\mathbf{1\times}\boldsymbol{S}}\underbrace{\boldsymbol{a}}_{\boldsymbol{S}\mathbf{\times 1}} + \underbrace{\boldsymbol{\lambda}}_{\mathbf{1\times 1}} \tag{4-11}$$

where $\boldsymbol{X}$ is the vector of features (inputs) and Y is the scalar output. The number of inputs is R and the number of neurons is S. f is the activation function, which is most often chosen to be a hyperbolic tangent function. For instance, in our case, $\boldsymbol{X}$ is a vector of flow features computed at any point on the coarse grid of the simulation domain and the output variable Y is the local coarse grid-induced error. Given some number of samples, the weights, w and α, and the biases, b and λ, are adjusted to optimize the performance of the ANN by minimizing the Mean Squared Error (MSE). The MSE at each sample (data point) is defined as follows:

$$MSE = \frac{1}{N}\sum_{k=1}^{N}(T_k - Y_k)^2 \tag{4-12}$$

where N is the number of samples, and T_k and Y_k are the target and its corresponding output (prediction by ANN) respectively at each point k.

المنارة للاستشارات

www.manaraa.com

ANN training and testing

To adjust the weights and biases - that is, to train the ANN - there are many different algorithms. The algorithm that was used in this work is the Levenberg-Marquardt algorithm [67, 68]. For a vector of weights and biases, $\boldsymbol{Z}_k$ (at iteration k), the iterative algorithm to update $\boldsymbol{X}_k$ is:

$$\boldsymbol{Z}_{k+1} = \boldsymbol{Z}_k - [J_k^T \boldsymbol{J}_k + \mu \boldsymbol{I}]^{-1} \boldsymbol{J}_k \boldsymbol{e}_k \tag{4-13}$$

where the vector of network error (the difference between target and output) is $\boldsymbol{e}$ whose first derivatives with respect to weights and biases are contained in the Jacobian matrix, $\boldsymbol{J}$. $\boldsymbol{I}$ is the identity matrix, and μ is a positive number called the combination effect. $\mu \boldsymbol{I}$ is added to avoid having a non-invertible matrix, $\boldsymbol{J}_k^T \boldsymbol{J}_k$. μ is called the combination effect because if μI is large, this method is called the gradient descent method, and if $\mu \boldsymbol{I}$ is small (near zero), it becomes Newton's method. The training process is started with a relatively large value of μ because the initial weights are far from the optimum weights. Large value of μ leads to a small weights update (i.e. $(\boldsymbol{Z}_{k+1} - \boldsymbol{Z}_k)$ is small). If this update of weights leads to an improvement (MSE decreases so the output gets closer to the target), the μ value is decreased so the weights are allowed to change faster and the ANN training process gets faster.

To test the efficiency of ANN, the samples are divided into three separate groups:

(1) The training data (most of the data, typically 70% of the whole dataset): these samples are utilized by the network to adjust the weights and biases.

(2) The validation data (typically 15% of the whole data): these samples are used to measure the network generalization (if MSE gets reduced with iterations for training data but it increases for the validation data group, which means the generalization deteriorates). Therefore, the training process stops when generalization stops improving.

المنارة للاستشارات
www.manaraa.com

(3) The testing data (typically 15% of the data): these samples do not influence the training process and they give an independent measure of the ANN predictive capability with a new dataset (one that was not used in the training process).

ANN complexity

It is worth noting that the number of coefficients which are tuned to minimize MSE is not small. The number of adjusted coefficients equals $S[R+2]+1$. For example, with 3 input variables and 10 neurons, the number of adjusted coefficients is 51. In general, ANN is used in cases where many data points are included. Therefore, the number of coefficients is relatively insignificant compared to hundreds or thousands of data points.

4.5.2. Deep Learning

Deep learning means using ANN with several layers between the input and output. The output of each layer is the input of the next one [30]. In this way, high order attributes (inputs) are extracted from the initial low order simple attributes in the first input layer. DL proved to be successful in dealing with many complex applications including speech recognition, object recognition and automatic game playing. However, DL is still computationally expensive compared to other ML algorithms such as Support Vector Machines (SVM) [65] or decision trees [69]. DL capability to obtain high accuracy is dependent on having a lot of data and adjusting a lot of coefficients during the training process. Thus, it takes much time. Another issue about DL that it is susceptible to overfitting because the number of weights and biases, that are adjusted through training the multi-layer ANN, is enormous (maybe comparable to the number of samples). Hence, the deep ANN may memorize the data instead of producing a more generally applicable model. Another factor to consider when dealing with DL is the variation of the conditions that need to be

www.manaraa.com

set (hyperparameters) before getting satisfactory results including the number of layers, number of neurons per each layer and the initial weights.

4.5.3. Random Forest Regression

Random forest regression is composed of a group of regression trees. A regression tree [69] predicts responses to input variables following the decisions in the tree from the root node down to a leaf node (see Figure 7). Each leaf node contains a possible response, Y. Each step in the prediction involves checking the value of one predictor (input). On each predictor, all the possible binary splits are examined. The split that leads to the best result is selected; the optimization criterion is typically minimizing the MSE. The split may result in a child node that has very few observations (data); this is why a minimum leaf size (minimum observations per node) is specified *a priori*. For the new child nodes (new leaves), the same procedure is taken to split the new nodes. This process continues again and again. Therefore, the regression tree always has a stopping rule. The stopping rule means that the splitting stops when the number of the observations per leaf is equal to the minimum one.

The process of splitting is illustrated in Figure 7, which represents a sample tree with x_1, x_2 and x_3 as inputs (predictors) with 4 leaves (4 responses). Regression trees are computationally cheap, but they over-fit the data; this is why many trees are used together (tree bagging).

In the tree bagging approach, a random sample of the training data is given to each regression tree so each tree is trained based on a different dataset. Hence, for a new sample (unseen data), each tree will make a different prediction. The prediction of the whole tree bag is the average of all the individual trees' predictions [32].

المنارة للاستشارات
www.manaraa.com

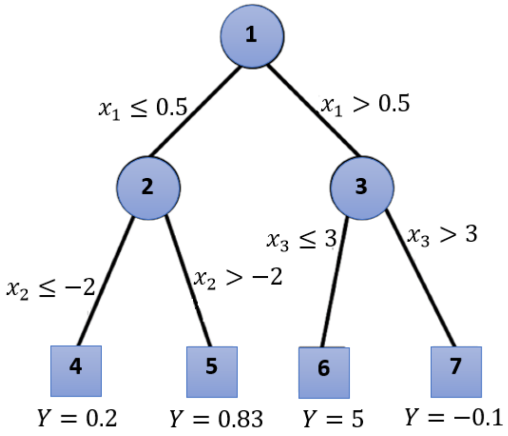


Figure 7. Regression tree.

This method is called the random forest because a random subset of input variables (features) are utilized at each split in each tree (in the tree bag). If the number of the features is R, the number of features at each split is typically $R/3$ [70].

The sampling technique used in random forest algorithm is the bootstrap sampling [71]. In this technique, the sample size (for each individual tree) is equal to the original dataset size but the data are chosen with a replacement which means that some data points (rows) may be selected more than once and other rows are not chosen at all. This way, the data given to each tree is different so the trees' predictions will be different and the over-fitting is eliminated when combining trees' results together. On the other side, bootstrapping leads to a sample per tree which looks like the original sample of the original dataset.

The optimum number of trees is determined based on the "Out Of Bag error" (OOB error). OOB error is the average prediction error on a training data point if we used the trees that did not have this data point when bootstrap sampling. It is always desired to achieve minimum OOB error; therefore, the number of trees is chosen based on this criterion. The typical number of trees in random forest algorithms is 100.

المنارة للاستشارات
www.manaraa.com

4.6. Chapter Summary

Given a large amount of data (experiments and simulations) and many variables (fluid flow features distributed in space and/or time), machine learning algorithms can be used to explore data patterns and build regression models that predict new outcomes. Machine learning algorithms can be used for: (1) Classification, (2) Clustering, (3) Dimensionality reduction, and (4) Regression:

1. A classification problem is a problem in which all the available observations can be categorized in some distinct classes (for instance laminar vs. turbulent flow) and the function of the classification algorithm is to find the separating line between the different classes. One of the classification algorithms is the Support Vector Machine (SVM). For instance, the classification ability of SVM is used for identification of two-phase flow regimes.
2. Clustering is a process of grouping the data in a way that each group of data includes data that are closer to each other. It is may precede the regression process because sometimes the entire data set cannot be represented by a single model. Therefore, the data are clustered into different groups and each group is represented by a different *correlation*. An example of clustering algorithms is the *k*-means algorithm.
3. Given a high-dimensional dataset, it would be useful to reduce the dimensionality of the data by extracting the features which are more important (Principal component analysis is an example of dimensionality reduction techniques).
4. Regression is one of the most common uses of machine learning. Examples of the regression algorithms include Artificial Neural Network (ANN) and Random Forest Regression (RFR). Both ANN and RFR are used in this work.

www.manaraa.com

- Artificial Neural Network (ANN) is a well-known non-linear adaptive regression tool that identifies a function that relates the given inputs and outputs. ANN consists of computational units that are named artificial neurons. The artificial neuron receives input variables which are multiplied by weights and are then processed by a mathematical activation function to produce the output.
- Random forest regression is composed of a group of regression trees. A regression tree predicts responses to input variables following the decisions in the tree from the root node down to a leaf node. Each leaf node contains a possible response. Each step in the prediction involves checking the value of one (input). On each predictor, all the possible binary splits are examined. The split that leads to the best result is selected; the prediction of the whole tree bag is the average of all the individual trees' predictions.

A method, that shows how machine learning regression algorithms (ANN and RFR), is used for correlating the CG-CFD error with the coarse-grid flow features and presented in the next chapter (Chapter 5).

المنارة للاستشارات
www.manaraa.com

5. COARSE-GRID ERROR PREDICTION METHOD

This work addresses the problem of the CG-CFD grid-induced error so the CG-CFD flow solution can be corrected to compensate for this error. This problem is presented in Section 5.2. The CG-CFD error can be predicted under certain hypothesis (see Section 5.3). The elements of this novel method, proposed for computing CG-CFD grid-induced error, are explained in Section 5.4. The capability of the proposed approach is assessed by applying it on a turbulent flow inside a lid-driven 3D cavity (Section 5.5).

5.1. Objectives

The objectives of this chapter can be summarized as follows:

- Developing a novel method for predicting the CG-CFD error, using machine learning algorithms, given the flow features computed by solution on coarse grids.
- Discussing the hypothesis on which the proposed CG-CFD method is based.
- Producing a set of scenarios that test the proposed CG-CFD method.

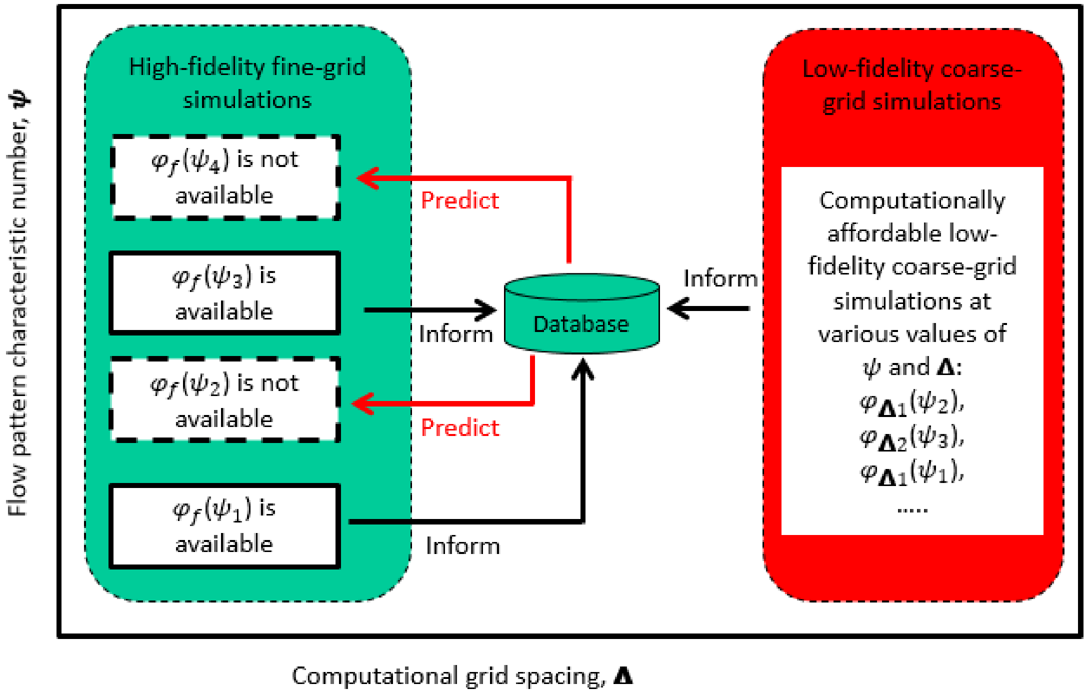
5.2. Problem Statement

The general problem to be addressed in this work can be formulated as illustrated in Figure 8. In Figure 8, the vertical axis, ψ, is a number or a set of numbers that characterize the flow pattern (such as Reynolds number, Rayleigh number, domain aspect ratio, etc.), while the horizontal axis is the computational grid spacing, Δ.

For a specific problem of interest (in this work it is the flow inside a lid-driven cavity), the green-colored area in Figure 8 represents the high-fidelity data computed by sufficiently fine grids (Δ approaching zero). The subscript f refers to a fine grid and φ_f is the flow variable of interest (e.g. velocity component or pressure) computed by a sufficiently fine grid. The red-colored area refers to low-fidelity computations, of φ, performed by coarse grids, that may or may not have

المنارة للاستشارات
www.manaraa.com

corresponding available high-fidelity computations, φ_f. The problem here is how to use the low fidelity and corresponding high-fidelity data to construct a function that can predict the correct value of a new (not available) φ, given only the corresponding low-fidelity computation only.



High-fidelity fine-grid simulations
$\varphi_f(\psi_4)$ is not available
$\varphi_f(\psi_3)$ is available
$\varphi_f(\psi_2)$ is not available
$\varphi_f(\psi_1)$ is available
Predict
Inform
Predict
Inform
Database
Inform
Low-fidelity coarse-grid simulations
Computationally affordable low-fidelity coarse-grid simulations at various values of ψ and $\mathbf{\Delta}$:
$\varphi_{\mathbf{\Delta}1}(\psi_2)$,
$\varphi_{\mathbf{\Delta}2}(\psi_3)$,
$\varphi_{\mathbf{\Delta}1}(\psi_1)$,
.....
Flow pattern characteristic number, $\boldsymbol{\psi}$
Computational grid spacing, $\mathbf{\Delta}$

Figure 8. Workflow for data generation in the CG-CFD problem.

Traditionally, a new high-fidelity computationally expensive simulation is performed for each new ψ. In this work, the benefit of the data is maximized using the available high-fidelity data to capture the relation between coarse and fine grid data to predict φ for a new case without having to run the expensive simulation.

5.3. Research Hypothesis

The present hypothesis is that the CG-CFD error is predictable, using data-driven surrogate model, under the following assumptions:

www.manaraa.com

- The training flows (to train the surrogate model) and the testing flows (to test the trained surrogate model) have similar physics with similar complexity. To correct the grid-induced error over the whole domain (for the training flows), high-fidelity data over the whole domain are needed.
- When comparing the fine and coarse-grid data, the number of the cells in both grids is not the same. Thus, to compute the local grid-induced error, it is necessary to perform mapping of the fine grid data onto the coarse grid. In other words, φ_f, is replaced by $\varphi_{f\to\Delta}$ (the fine grid field of φ mapped on a grid whose cell length is Δ). This mapping (averaging) constitutes a source of error because of losing some details of the fine-grid flow field profile. However, this mapped field is much more accurate than the field computed by a coarse grid (for instance, see Figure 2).
- Using a coarse grid in the CG-CFD framework is limited by the scale of the phenomenon of interest. If the phenomenon of interest has a length scale, l, the coarse grids should have a grid spacing, $\Delta < l$. It is assumed that the phenomenon of interest has a scale that is large enough that a data-driven surrogate model can be used to predict the grid-induced error occurred due to the inability of the coarse grid to resolve sub-grid length-scales.
- Sub-grid flow features, computed by a coarse grid, are expected to be inaccurate. However, it is hypothesized that the grid-induced error is a function of the inaccurate coarse-grid flow features.
- For most of the similar efforts in the literature (e.g. [26, 27]) the aim was to get a data-driven surrogate model that compensates for sub-grid effects; the ultimate goal was predicting a quantity of interest (such as the lift force or Nusselt number) or specific linear profile (such as the axial velocity profiles). On the other hand, in the present work, it was

المنارة للاستشارات

assumed that we are interested in *all* the data through the whole domain (flow variable value at each grid cell). Thus, the proposed approach makes more use of the full field data.

5.4. Proposed Approach

This section presents the proposed method for the CG-CFD error prediction using ML algorithms. In the following paragraphs, we will discuss the error prediction method, feature selection, and ML error evaluation.

5.4.1. CG-CFD Error Prediction Method

The proposed method illustrated in Figure 9 consists of 2 sets of flows: training flows and testing flows. Training flows are utilized to construct a surrogate model of the grid-induced error as a function of the coarse-grid flow features. High-fidelity simulations of the training flows are performed with sufficiently fine grids and represented by φ_f^{tr}(tr refers to training flows). Next, the same simulation is performed again with a coarse grid producing φ_Δ^{tr}. The fine-grid training flow solution, φ_f^{tr}, is mapped onto the coarse grid, Δ, to obtain $\varphi_{f\rightarrow\Delta}^{tr}$. Based on the coarse-grid simulations of the training flows, flow "features", $\boldsymbol{X}(\varphi_\Delta^{tr})$, are computed. These features are the inputs to a surrogate model whose output is the local grid induced error, $\varepsilon_\Delta(\varphi^{tr})$:

$$\varepsilon_\Delta(\varphi^{tr}) = \varphi_{f\rightarrow\Delta}^{tr} - \varphi_\Delta^{tr} = F\big(\boldsymbol{X}(\varphi_\Delta)\big) \qquad (5\text{-}1)$$

Selection of the features is discussed in the next section. The function F is obtained from ML algorithms (ANN and RFR). The capability of the surrogate model, obtained from Equation (5-1), is assessed by applying the model on testing flows as illustrated in Figure 9.

www.manaraa.com

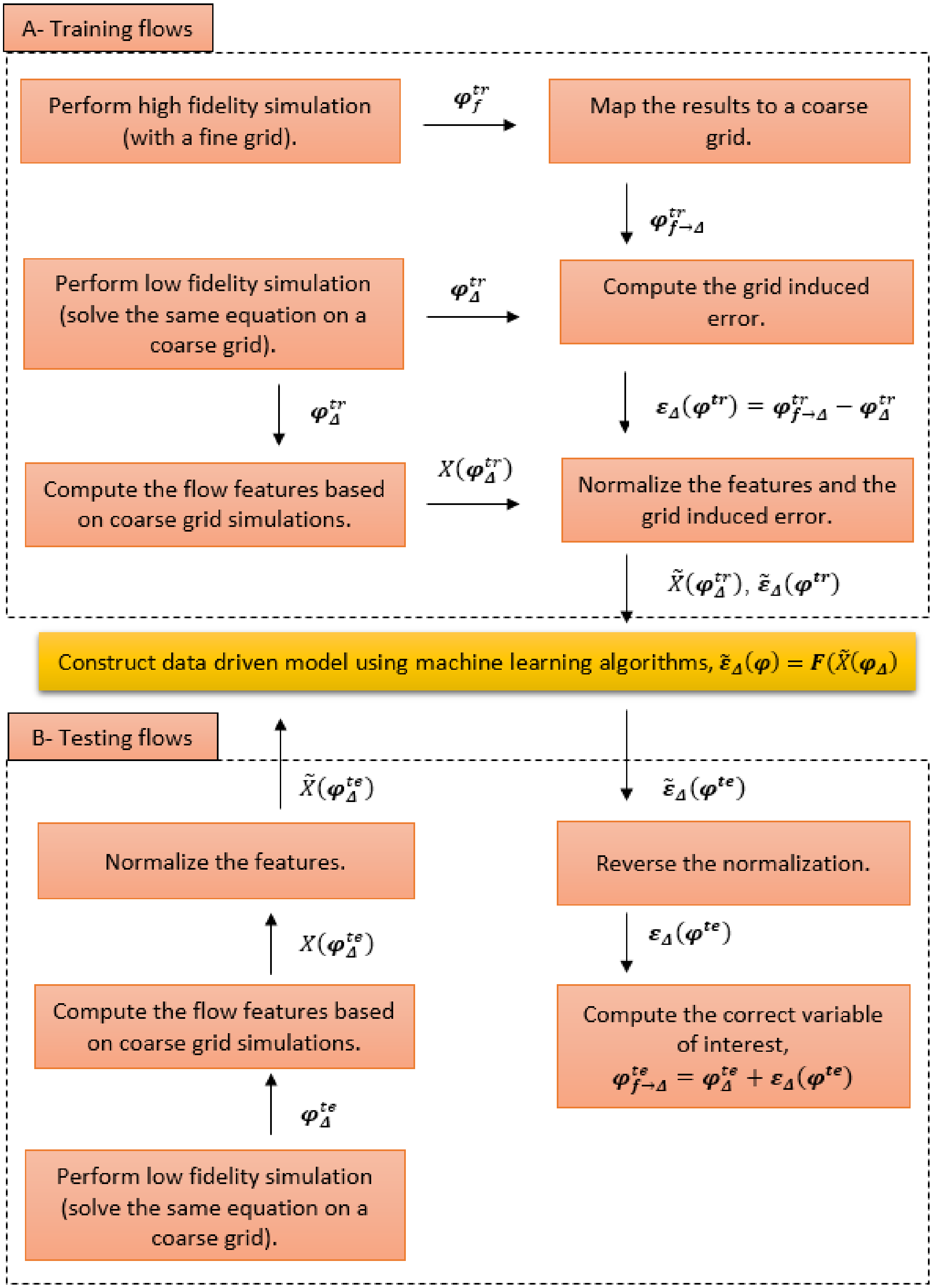

A- Training flows
Perform high fidelity simulation (with a fine grid).
φ_f^{tr}
Map the results to a coarse grid.
$\varphi_{f\to\Delta}^{tr}$
Perform low fidelity simulation (solve the same equation on a coarse grid).
φ_Δ^{tr}
Compute the grid induced error.
$\varepsilon_\Delta(\varphi^{tr}) = \varphi_{f\to\Delta}^{tr} - \varphi_\Delta^{tr}$
φ_Δ^{tr}
Compute the flow features based on coarse grid simulations.
$X(\varphi_\Delta^{tr})$
Normalize the features and the grid induced error.
$\tilde{X}(\varphi_\Delta^{tr}), \tilde{\varepsilon}_\Delta(\varphi^{tr})$
Construct data driven model using machine learning algorithms, $\tilde{\varepsilon}_\Delta(\varphi) = F(\tilde{X}(\varphi_\Delta))$
B- Testing flows
$\tilde{X}(\varphi_\Delta^{te})$
$\tilde{\varepsilon}_\Delta(\varphi^{te})$
Normalize the features.
Reverse the normalization.
$X(\varphi_\Delta^{te})$
$\varepsilon_\Delta(\varphi^{te})$
Compute the flow features based on coarse grid simulations.
Compute the correct variable of interest, $\varphi_{f\to\Delta}^{te} = \varphi_\Delta^{te} + \varepsilon_\Delta(\varphi^{te})$
φ_Δ^{te}
Perform low fidelity simulation (solve the same equation on a coarse grid).

Figure 9. The proposed method to predict CG-CFD error using ML algorithms.

المنارة للاستشارات
www.manaraa.com

5.4.2. Features Selection

Assuming a smooth profile for the flow variable, φ, consider a Taylor series expansion for a flow field variable φ along the x-direction in a grid with a cell length of Δx:

$$\varphi = \varphi_0 + \Delta x \left.\frac{d\varphi}{dx}\right|_{\Delta x} + \frac{(\Delta x)^2}{2} \left.\frac{d^2\varphi}{dx^2}\right|_{\Delta x} + \cdots \tag{5-2}$$

For a linear state variable profile, the grid-induced error will be of the order $(\Delta x)^2 \left.\frac{d^2\varphi}{dx^2}\right|_{\Delta x}$ and the effect of $\Delta x \left.\frac{d^2\varphi}{dx^2}\right|_{\Delta x}$ may be considered for non-linear profiles. The flow state variables' derivatives were utilized before as indicators of the need for high local grid refinement in the adaptive-grid-refinement literature [72-74]. This was inspired by the general observation that a finer grid is needed near a discontinuity or a large gradient of the solution. Another idea to be considered is the fact that the value of the variable φ and its corresponding error at any grid cell is dependent on the value of φ and the corresponding error at the neighboring cells. Hence, the derivatives $\frac{d\varphi}{dx}$ and $\frac{d^2\varphi}{dx^2}$ are carrying the effect of the neighboring cells.

Additionally, the fluid flow is typically characterized by global numbers like Reynolds number, Rayleigh number, etc. Thus, we propose to utilize a local number (as flow feature) which substitutes for the global number. In the present work, the quantity of interest is the velocity. Therefore, we selected the cell Reynolds number as the local quantity of interest. The cell Reynolds number substitutes for the global velocity and length scales with local velocity and cell length, Δ. The cell Reynolds number, Re_Δ, is defined in terms of the local velocity magnitude, $|U|$ and the kinematic viscosity, ν:

$$Re_\Delta = |U|\Delta/\nu \tag{5-3}$$

www.manaraa.com

Based on the previous discussions, the proposed features vector, $X(\varphi_\Delta)$, becomes:

$$\boldsymbol{X}(\varphi_\Delta) = \left(Re_\Delta, (\Delta x_{\bar{j}}) \left.\frac{du_i}{dx_{\bar{j}}}\right|_{\Delta x_{\bar{j}}}, (\Delta x_{\bar{j}})^2 \left.\frac{d^2u_i}{{dx_{\bar{j}}}^2}\right|_{\Delta x_{\bar{j}}} \right) \tag{5-4}$$

The index $\bar{j}$ is used to indicate no summation. Actually, the term $(\Delta x_{\bar{j}}) \left.\frac{du_i}{dx_{\bar{j}}}\right|_{\Delta x_{\bar{j}}}$ represents 9 features because the velocity has 3 components: $U_{x,}U_y$ and U_z. The derivative is performed with respect to 3 directions: x, y and z (for instance: $\frac{dU_{x,}}{dx}, \frac{dU_{x,}}{dy}, etc.$). Similarly, the term $(\Delta x_{\bar{j}})^2 \left.\frac{d^2u_i}{{dx_{\bar{j}}}^2}\right|_{\Delta x_{\bar{j}}}$ represents 27 features (accounting for all the second derivatives for the 3 velocity components).

5.4.3. Data Comparison (Mapping)

To compute the grid-induced error, it is necessary to compare the flow field computed by the fine and coarse grids. This comparison can be performed with different methods as illustrated in Figure 10. The first method is comparing each point on the coarse grid to the corresponding point on the fine grid. If there is no corresponding point on the fine grid, flow variable is interpolated on the fine grid.

The second method is comparing each coarse-grid cell to the averaged field over the corresponding set of cells on the fine grid. Both methods result in information loss because of averaging. In this work, the second method (comparing cells) is used because, through the averaging process, more information from the fine grid is utilized compared to the first method.

www.manaraa.com

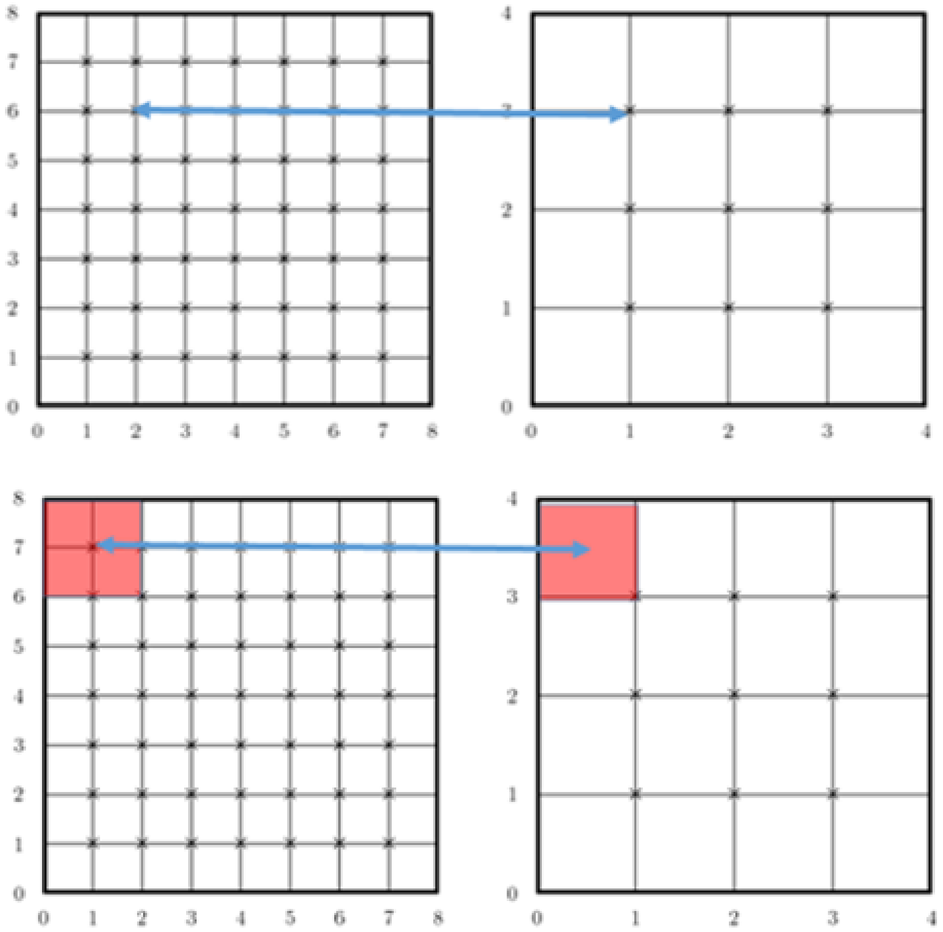


Figure 10. Mapping flow variable, computed by a fine grid onto a coarse grid by one of two approaches: point to point (top) or cell to cell (bottom).

5.4.4. Data Normalization

When comparing different sets of CFD data, it is desired to use dimensionless data for performing comparisons between different geometries; hence, the features are normalized using the length scale and the velocity scale (see Table 6). The normalized features, $\tilde{X}$, and the normalized grid induced-error, $\tilde{\varepsilon}_{\Delta}^{tr}$, are incorporated into Equation (5-1) as:

$$\tilde{\varepsilon}_{\Delta}(\varphi^{tr}) = F\big(\widetilde{\boldsymbol{X}}(\varphi_{\Delta})\big) \tag{5-5}$$

المنارة للاستشارات
www.manaraa.com

In this work (a lid-driven cavity flow), the length scale, W, is the lid width, and the velocity scale is the cavity lid velocity, U_{lid}.

Table 6. Normalization factors for different variables.

Feature	**Normalization factor**
Δ	W
$\left.\frac{du_{\bar{\imath}}}{dx_{\bar{\jmath}}}\right\|_{\Delta x_{\bar{\jmath}}}$	W/U_{lid}
$\left.\frac{d^2u_{\bar{\imath}}}{dx_{\bar{\jmath}}{}^2}\right\|_{\Delta x_{\bar{\jmath}}}$	W^2/U_{lid}
$\varepsilon_\Delta(U_x)$, $\varepsilon_\Delta(U_y)$, $\varepsilon_\Delta(U_z)$	U_{lid}

5.4.5. *Machine Learning Error Assessment*

To assess the accuracy of the surrogate model produced by ML algorithm, a metric, E_{ML} is proposed. This metric (ML error) is calculated as:

$$E_{ML} = \frac{|\tilde{\varepsilon}_\Delta^{act}(\varphi) - \tilde{\varepsilon}_\Delta^{ML}(\varphi)|}{\sigma_{\tilde{\varepsilon}_\Delta^{act}(\varphi)}} \tag{5-6}$$

where $\tilde{\varepsilon}_\Delta^{act}(\varphi)$ is the actual grid-induced error when computing the variable φ and $\tilde{\varepsilon}_\Delta^{ML}(\varphi)$ is the estimated error. The difference between the actual and ML prediction of the grid-induced error is normalized by the standard deviation of the actual grid-induced error, $\sigma_{\tilde{\varepsilon}_\Delta^{act}(\varphi)}$. Typically, when computing the surrogate model error, the difference between the actual and the estimated value is

المنارة للاستشارات
www.manaraa.com

divided by the actual value to get a relative error. However, here, we divide by the standard deviation of the actual value to avoid having zeros in the denominator. Additionally, computing ML error in Equation (5-6) gives the ML error in terms of the standard deviation of the grid-induced error.

5.5. Case Study

The lid-driven cavity flow is one of the most studied problems in CFD [75, 76]. Although the problem can be stated relatively simply, the flow in a cavity features interesting flow physics with counter-rotating vortices appearing at the corners of the cavity [76], and it serves as a benchmark problem for numerical methods [77, 78].

The case studied here is a fluid contained in a cube whose height is one meter, with five fixed walls and one lid, moving with velocity, $U_{lid} = 1\ \mathrm{m/s}$ (see Figure 11). The physics of this problem is related to the global Reynolds number, Re, which is defined in terms of the velocity magnitude, $|U|$, the fluid kinematic viscosity, ν, and the cavity characteristic length (lid width), W.

$$Re = \frac{|U|W}{\nu} \tag{5-7}$$

At a Reynolds number between 2000 and 3000, an instability appears in the vicinity of the downstream corner [79]. As expected, the turbulence level near the wall increases with the increasing Reynolds number. The flow becomes fully turbulent near the downstream corner eddy at Reynolds number higher than 10,000 [79].

المنارة للاستشارات
www.manaraa.com

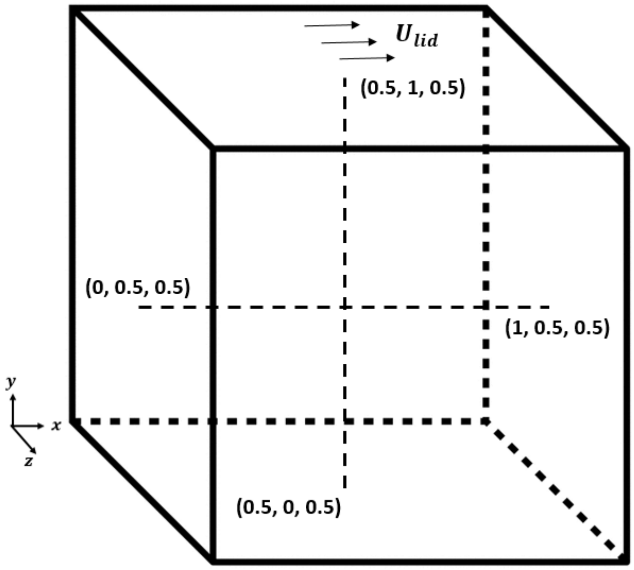


Figure 11. Lid-driven cubic cavity. The lid velocity is parallel to the x-axis.

5.5.1. Numerical Simulations

In this work, CFD simulations were performed with one of the available open source CFD software packages, OpenFOAM [10]. All the simulations were performed with three-dimensional NS equations (incompressible flow of a viscous Newtonian fluid) [12]. NS equations are discretized in space using finite volume method (see the equations in Section 3.2.3) on meshes composed of hexahedral computational cells. For the spatial derivatives, the central difference scheme (second-order accurate) has been used [53].

High-fidelity simulations were performed on a $120 \times 120 \times 120$ grid, with additional mesh refinement near the walls. The length of the cells touching the walls is 0.0014 meters so the total number of cells in the fine grid is 2×10^6 cells.

In OpenFOAM, a transient solver (named *pisoFoam*) for incompressible, turbulent flow, using the PISO algorithm [54], was used. Using *pisoFoam*, unsteady simulation can be performed, but the focus of the present work is the quasi-steady-state three-dimensional flow inside the cavity.

المنارة للاستشارات
www.manaraa.com

5.5.2. *Training and Testing Cases*

When training a surrogate data-driven model, the predictive capability of the model is impacted by the similarity or the difference between the training flows and the testing flow. Thus, to assess the surrogate model, different scenarios of the available training data and the targeted testing case are studied (see the list of scenarios in Table 7).

For instance, in the first scenario, lid-driven cavity flows with Reynolds numbers ($Re \in (6000\ ,8000\ ,12000)$) and simulated with the same grid size, $\Delta = 1/30\ m$, are used to train the model, in Equation (5-5). Next, this function, F, is used to predict $\varepsilon_{\Delta}(\varphi)$ given the features computed by a coarse grid with $\Delta = 1/30\ m$ for a new Reynolds number, $Re = 10000$. The 1st scenario is not very challenging because the testing case Reynolds number lies in the training cases' Reynolds numbers' range (interpolation mode). The 2nd scenario gets more challenging by extrapolation in the direction of the global Reynolds number. In the next scenarios, the ML surrogate model is tested against more challenging scenarios by interpolating and extrapolating in terms of Re and Δ. In all the scenarios in Table 7 (except the last one), coarse-grid simulations are performed with uniform grids (the same grid spacing, Δ, in the three Cartesian coordinates). In the scenario X, testing flow simulation is performed with a grid that has a different grid spacing in different directions ($\Delta x \neq \Delta y \neq \Delta z$).

All the scenarios in Table 7 have the same geometry (cubic cavity) and the lid velocity (one meter per second). Reynolds number changes only by changing the kinematic viscosity. To test the capability of the surrogate model to predict the flow variables for different aspect ratios, additional scenarios are studied (see Table 8). The aspect ratio is the cavity height to width ratio ($\boldsymbol{H/W}$). The height is the cavity side perpendicular to the lid (in **y** direction) and the width is the other side (in $\boldsymbol{x}$ or $\boldsymbol{z}$ direction).

المنارة للاستشارات
www.manaraa.com

Table 7. Training and testing data (first set of scenarios in terms of Re and Δ). Re and $\mathit{\Delta}$ are the Reynolds number and the grid spacing, respectively.

Scenario	Training flows		Testing flow	
	Re	$\mathit{\Delta}\ (m)$	Re	$\mathit{\Delta}\ (m)$
I: Re interpolation.	$\{6000, 8000, 12000\}$	$\frac{1}{30}$	10000	$\frac{1}{30}$
II: Re extrapolation (higher Re).	$\{6000, 8000, 10000\}$	$\frac{1}{30}$	12000	$\frac{1}{30}$
III: Re extrapolation (lower Re).	$\{ 8000, 10000, 12000\}$	$\frac{1}{30}$	6000	$\frac{1}{30}$
IV: Δ interpolation.	12000	$\left\{\frac{1}{40}, \frac{1}{20}\right\}$	12000	$\frac{1}{30}$
V: Δ extrapolation (finer grid).	12000	$\left\{\frac{1}{30}, \frac{1}{20}\right\}$	12000	$\frac{1}{40}$
VI: Δ extrapolation (coarser grid).	12000	$\left\{\frac{1}{40}, \frac{1}{30}\right\}$	12000	$\frac{1}{20}$
VII: Re and Δ interpolation.	$\{ 8000, 12000\}$	$\left\{\frac{1}{40}, \frac{1}{20}\right\}$	10000	$\frac{1}{30}$
VIII: Re and Δ extrapolation (higher Re and finer grid)	$\{ 8000, 10000\}$	$\left\{\frac{1}{30}, \frac{1}{20}\right\}$	12000	$\frac{1}{40}$
IX: Re and Δ extrapolation (lower and coarser grid).	$\{ 10000, 12000\}$	$\left\{\frac{1}{40}, \frac{1}{30}\right\}$	8000	$\frac{1}{20}$
X: Different grid spacing in different directions.	12000	$\left\{\frac{1}{40}, \frac{1}{30}, \frac{1}{20}\right\}$	12000	$\mathit{\Delta}x = \frac{1}{40}$ $\mathit{\Delta}y = \frac{1}{30}$ $\mathit{\Delta}z = \frac{1}{20}$

المنارة للاستشارات

www.manaraa.com

Table 8. Training and testing data (second set of scenarios in terms of Re, Δ, and H/W). Re, Δ, and (H/W) are the Reynolds number, grid spacing and aspect ratio, respectively

Scenario	Training flows			Testing flow		
	Re	$\Delta\ (m)$	(H/W)	Re	$\Delta\ (m)$	(H/W)
XI: (H/W) extrapolation.	6000	$\frac{1}{30}$	1	6000	$\frac{1}{30}$	1.2
XII: (H/W) and Re extrapolation.	8000	$\frac{1}{30}$	1	6000	$\frac{1}{30}$	1.2
XIII: (H/W) , Re and Δ extrapolation.	8000	$\frac{1}{40}$	1	6000	$\frac{1}{30}$	1.2
Increasing the aspect ratio difference between training and testing data.						
XIV: (H/W) interpolation.	6000	$\frac{1}{30}$	$\{1, 4\}$	6000	$\frac{1}{30}$	2
XV: (H/W) extrapolation to a lower ratio.	6000	$\frac{1}{30}$	$\{2, 4\}$	6000	$\frac{1}{30}$	1
XVI: (H/W) extrapolation to a higher ratio.	6000	$\frac{1}{30}$	$\{1, 2\}$	6000	$\frac{1}{30}$	4
Increasing the Reynolds number.						
XVII: (H/W) extrapolation to a higher ratio.	12000	$\left\{\frac{1}{30}, \frac{1}{20}\right\}$	$\{1, 2\}$	12000	$\frac{1}{30}$	4
XVIII: (H/W) extrapolation to a higher ratio and finer grid size.	12000	$\left\{\frac{1}{30}, \frac{1}{20}\right\}$	$\{1, 2\}$	12000	$\frac{1}{40}$	4

المنارة للاستشارات
www.manaraa.com

5.6. Summary

The problem of interest here is how to use the coarse-grid simulations' numerical data and the corresponding validated fine-mesh simulations numerical data to train a statistical model that can predict the correct value of the flow variable, we are interested in, given the corresponding coarse-grid simulations' data only.

The statistical model is *trained* using machine learning algorithms (e.g. artificial neural network and random forest regression) and *tested* using a new set of numerical data from coarse-grid simulations. The training data and the testing data are similar in boundary conditions and physics (incompressible isothermal turbulent flow inside a lid-driven cavity) and different in Reynolds number (viscosity), grid size and aspect ratio.

Local fluid flow features (based on the coarse-grid solution) are proposed based on Taylor series expansion and local cell Reynolds number. The local grid-induced error is hypothesized to be a function of those inaccurate flow features computed by coarse grids.

A metric to quantify the machine learning error, when predicting the flow variable of interest, was proposed instead of inspecting the machine learning predictions qualitatively.

A set of scenarios that test the proposed method are studied. These scenarios investigate the capability of the statistical model to interpolate and extrapolate outside the training data range. These scenarios also cover a range of Reynolds numbers in the turbulent flow and transitional flow range plus a range of grid sizes, aspect ratios and different variables of interest.

المنارة للاستشارات
www.manaraa.com

6. RESULTS AND DISCUSSIONS

This chapter presents the results of applying the proposed grid-induced error prediction method on the flow inside a lid-driven cavity. We start by validating the high-fidelity fine grid simulations that are utilized to compute the grid-induced error (see Section 6.2). Next, the capability of the proposed CG-CFD approach, to predict the grid-induced error, is assessed given a variation of training and testing data (see Section 6.3). A set of numerical experiments are performed to predict the grid-induced error corresponding to the different velocity components given the scenarios listed in Section 5.5.2.

6.1. Objectives

The objectives of this chapter can be summarized as follows:

- Assessing the capability of the proposed CG-CFD error prediction method by applying it on a three-dimensional turbulent flow numerical data.
- Utilizing the computed grid-induced error to obtain the spatial distribution of the variable of interest.
- Comparing artificial neural network and random forest regression in terms of accuracy and computational cost.

6.2. Validation

In this work, simulations are performed for a lid-driven cavity flow with Reynolds numbers ranging from 6,000 to 12,000. Because the grid size requirements increase with increasing Reynolds number, we started by validating OpenFOAM simulation results for a flow whose Reynolds number equals 12,000. Next, the grid, that was used with $Re = 12{,}000$, is also used for the lower Reynolds-number simulations. This ensures more than adequate resolution for lower Reynolds number simulations. Figure 12 compares the velocity profile captured with experiment

المنارة للاستشارات
www.manaraa.com

[80, 81] with the OpenFOAM simulations, performed as a part of the present work. Figure 12 depicts the velocity profiles computed with different grids: $\Delta = 0.0083\mathrm{m}$ (corresponding to 120×120×120 grid), $\Delta = 0.0167m$ (corresponding to 60×60×60 grid), and $\Delta = 0.033m$ (corresponding to 30×30×30 grid).

It is shown that the computed profile agrees well with the experimental one when adding wall refinement cells. The finest grid ($\Delta = 0.0083\mathrm{m}$ with wall refinement near the wall) is utilized to gain high-fidelity data for different Reynolds numbers. This emphasizes the importance of resolving the wall flow region in the lid-driven cavity flow. Using coarse grids, without wall region refinement, results in inaccurate results. In the next section, uniform coarse grids ($\Delta = 0.05\mathrm{m}, \Delta = 0.033\mathrm{m}, \Delta = 0.025\mathrm{m}$) inaccurate results are corrected with the help of a data-driven model.

6.3. Training and Testing Cases

This section goes through the scenarios listed in Section 5.5.2 (Table 7 and Table 8) to capture the grid-induced error for the testing flow if the surrogate model is trained with training flows' data. The variable of interest, in the first 9 scenarios, was taken to be the spatial distribution of U_x. For the first scenario, both ANN (with 20 neurons) and RFR (with 100 regression trees) were used to predict $\varepsilon_\Delta(U_x)$.

www.manaraa.com

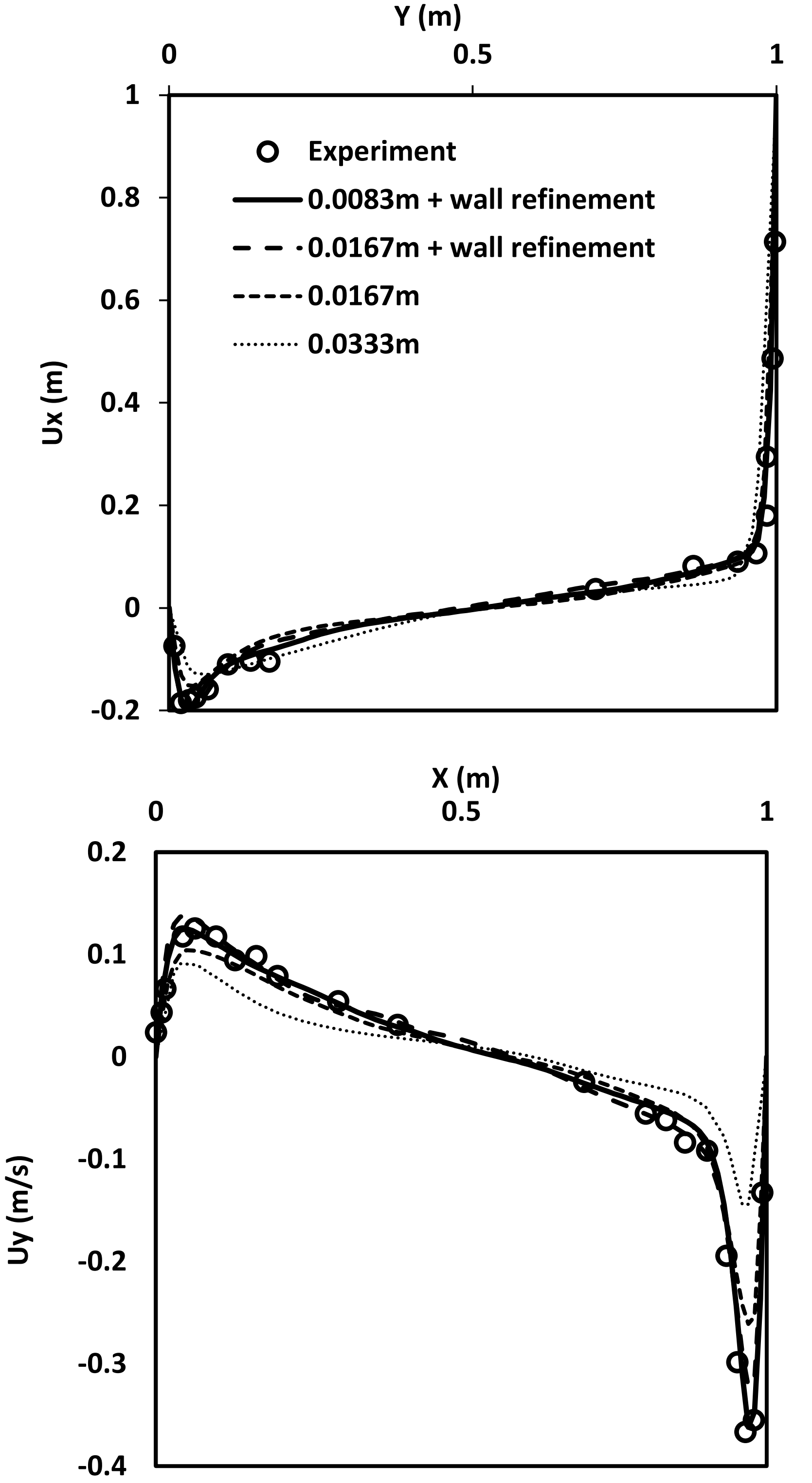


Figure 12. The axial profile (y direction) for velocity, U_x (top) and the axial profile (x direction for the velocity, U_y (bottom) at $Re = 12000$.

المنارة للاستشارات
www.manaraa.com

6.3.1. *Scenario I: Reynolds Number Interpolation*

Starting with one-layer ANN (represented by Figure 13), Figure 14 depicts the reduction in MSE with increasing the number of training iterations (epochs). In each iteration, ANN network weights and biases are re-adjusted to reduce the MSE given the target (actual $\varepsilon_{\Delta}(U_x)$) and the output (ANN predicted $\varepsilon_{\Delta}(U_x)$). It is shown that the MSE gets saturated at around 0.0001 after 78 epochs. Ideally, MSE should be equal to zero. In Figure 14, a scatter plot, of the ANN-predicted $\varepsilon_{\Delta}(U_x)$ vs. the actual $\varepsilon_{\Delta}(U_x)$, is presented. The training flows, given to ANN, are divided into three groups (training, testing and validating data), as explained in Section 4.5.1. The dotted diagonal line plot represents a perfect agreement between the targeted and the predicted $\varepsilon_{\Delta}(U_x)$ while the solid line is the best fit for the available data. The correlation coefficient, R is close to unity which indicates a very good agreement between the ANN prediction and the target value for all the data groups (training, testing and validating data groups).

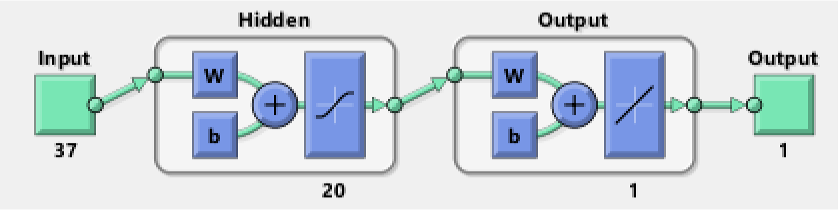


Figure 13. A representation of one-hidden-layer ANN with 37 inputs, 20 neurons and one output as represented by MATLAB [25]. *w* and *b* represent the weights and biases as presented in Section 4.5.1.

المنارة للاستشارات
www.manaraa.com

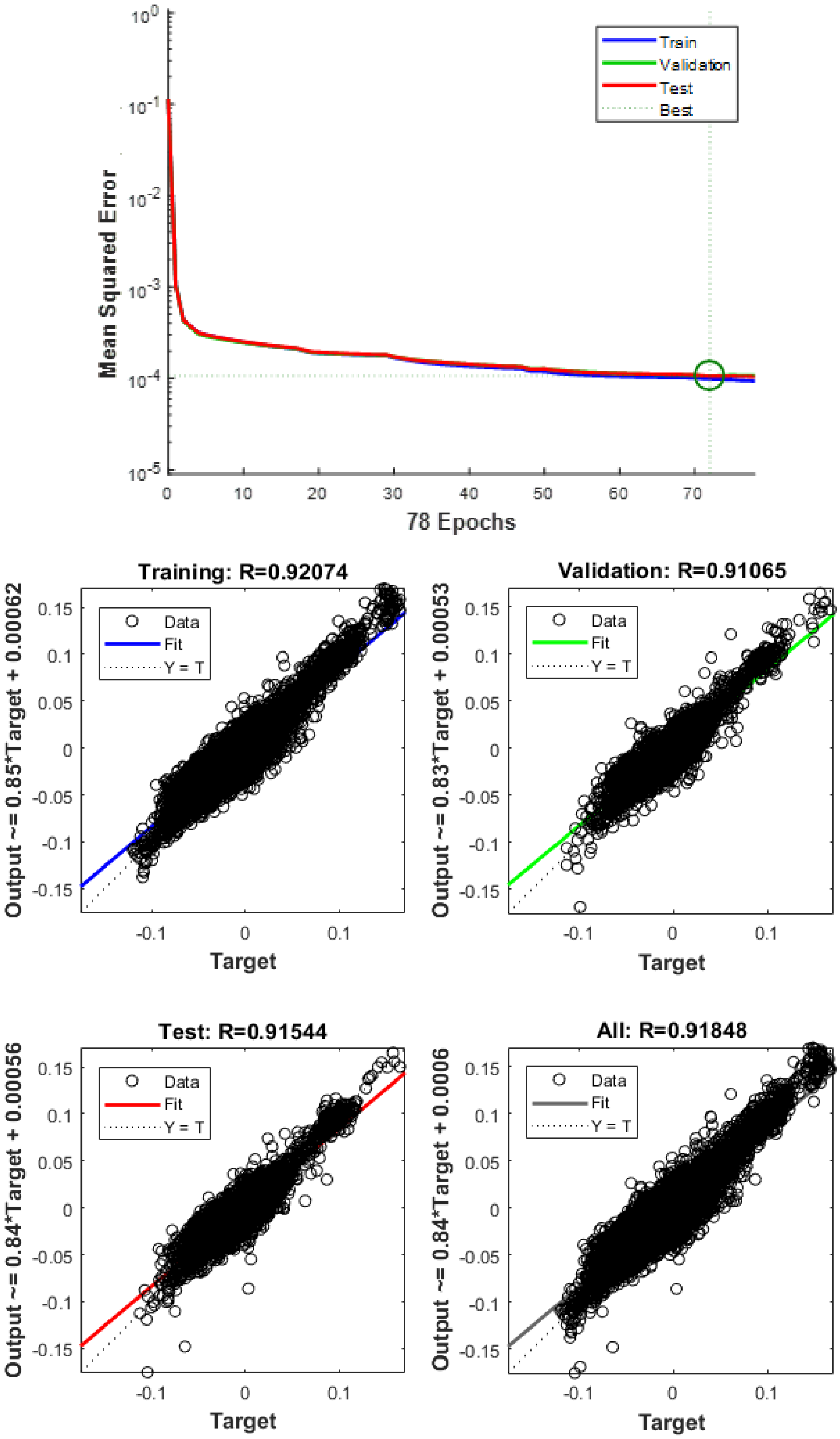


Figure 14. ANN performance in scenario I (training flows). MSE vs. the number of iterations (epochs) (above). $\varepsilon_{\Delta}(U_x)$ expected by ANN vs. the actual $\varepsilon_{\Delta}(U_x)$ (below).

المنارة للاستشارات
www.manaraa.com

ANN predicts $\varepsilon_{\Delta}(U_x)$ but the variable of interest is U_x (in particular: $U_{x_{\boldsymbol{f\rightarrow\Delta}}} = U_{x_{\boldsymbol{\Delta}}} + \varepsilon_{\Delta}(U_x)$). A comparison between the actual $U_{x_{\boldsymbol{f\rightarrow\Delta}}}$ and the ANN prediction of $U_{x_{\boldsymbol{f\rightarrow\Delta}}}$ is illustrated in Figure 15. The vertical axis refers to the actual $U_{x_{\boldsymbol{f\rightarrow\Delta}}}$ (u_f in Figure 15). The horizontal axis refers to one of three variables: (1) $U_{x_{\boldsymbol{f\rightarrow\Delta}}}$ (u_f), which corresponds to the solid black line, referring to the perfect case when the ANN-predicted $U_{x_{\boldsymbol{f\rightarrow\Delta}}} = U_{x_{\boldsymbol{\Delta}}} + \varepsilon_{\Delta}(U_x)$ is exactly equal to the actual one; (2) $U_{x_{\boldsymbol{\Delta}}}(u_c$ in Figure 15) which corresponds to the crossed points referring to the coarse grid results compared to the fine grid results; (3) ML prediction (circled points), which computes $U_{x_{\boldsymbol{f\rightarrow\Delta}}}$ given coarse grid results, $U_{x_{\boldsymbol{\Delta}}}$,and ANN prediction of $\varepsilon_{\Delta}(U_x)$.

It is shown in Figure 15 that ML predictions for both training and testing cases are close to the ideal case. Some circled points deviated from the “ideal” solid line but most ML predictions (circled points) give visible improvements over the coarse-grid results (crossed points). Note that Figure 15 shows a regression of “big data”. For instance, training flows include data from 3 grids for training: each grid has 27,000 cells with 37 features computed at each cell.

المنارة للاستشارات
www.manaraa.com

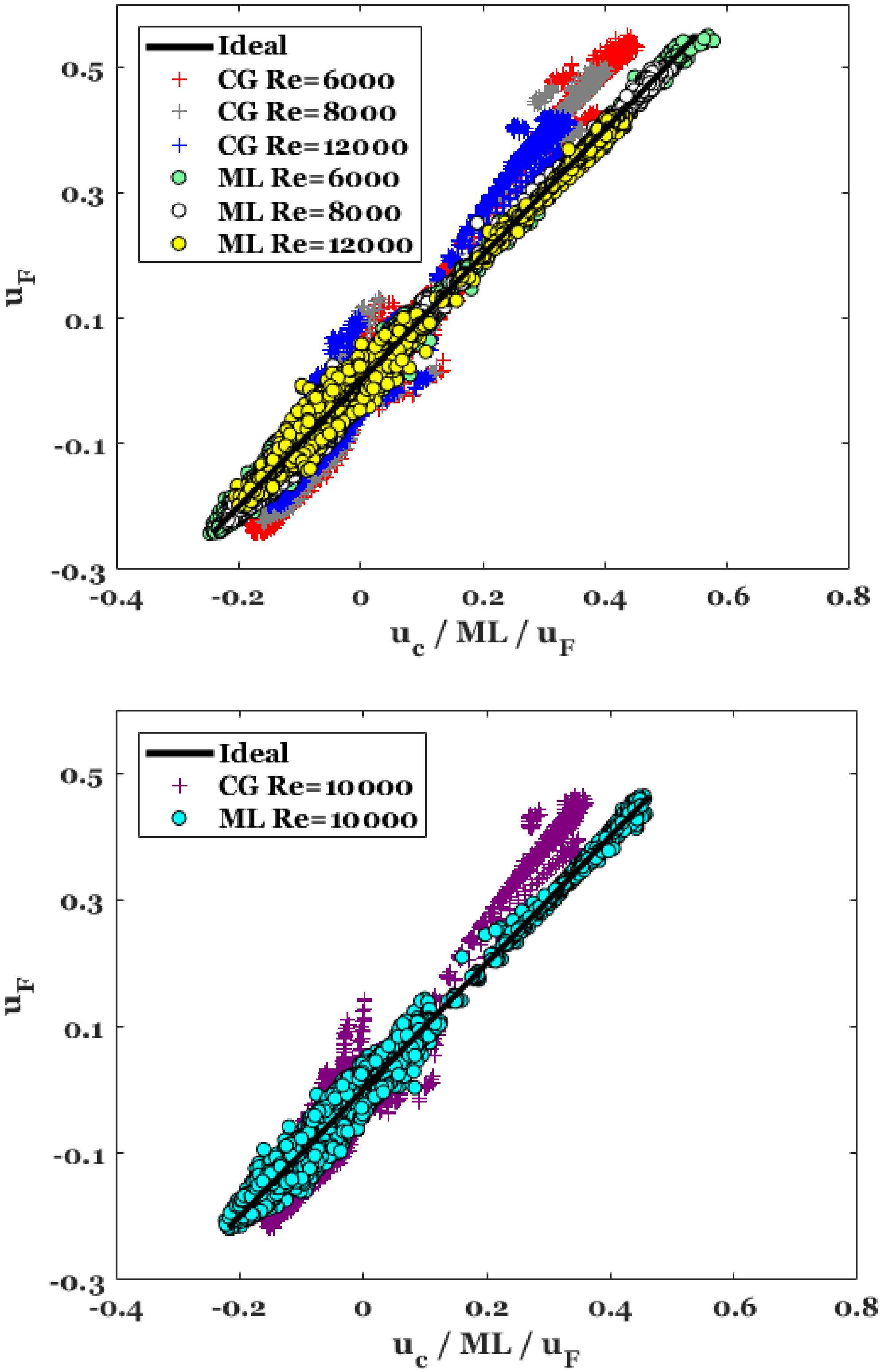


Figure 15. Scenario I (Reynolds number interpolation) for U_x (by ANN). $\Delta = 0.033m$. Training data (above) and testing data (below). CG: Coarse grid predictions. ML: Machine learning predictions.

www.manaraa.com

For scenario I, RFR is also utilized for predicting $\varepsilon_{\Delta}(U_x)$. To optimize the number of trees in RFR, OOB error (explained in Section 4.5.3) is minimized. As illustrated in Figure 16, the OOB error gets saturated at a minimum value when using 100 regression trees. As was shown for the ANN, Figure 17 illustrates the comparison between RFR predictions and the coarse-grid predictions. RFR training is better than ANN without a single point deviating from the ideal solid line.

6.3.2. Scenarios II and III: Reynolds Number Extrapolation.

In the second scenario, the testing flow case gets more challenging because the testing case is outside of the training flows' range. Prediction of the grid-induced error using ANN and RFR are presented in Figure 18 and Figure 19. ML predictive capability in scenario II is not as good as the scenario I. However, ML predictions are still better than coarse-grid results. RFR predictions do not deviate much away from the ideal line. In Table 9, a comparison between ANN and RFR efficiency, in different scenarios, is presented. The comparison is presented in terms of ML error, E_{ML}, defined by Equation (5-6) and the computational cost. In both scenarios (I and II), RFR leads to smaller mean and maximum E_{ML} compared to ANN. RFR is also computationally cheaper.

المنارة للاستشارات
www.manaraa.com

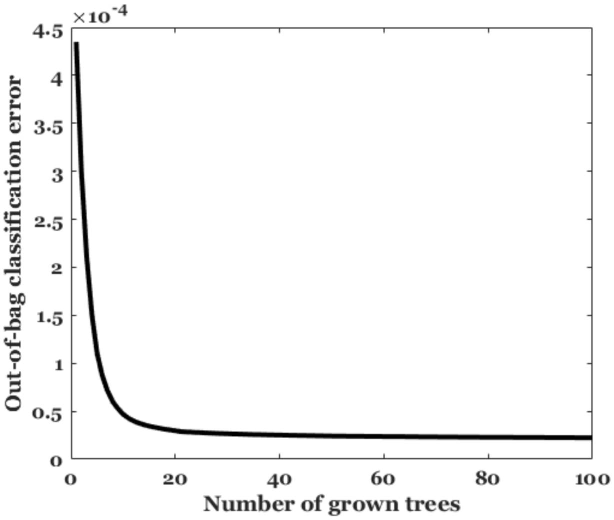


Figure 16. RFR OOB error decreases with increasing number of trees.

المنارة للاستشارات

www.manaraa.com

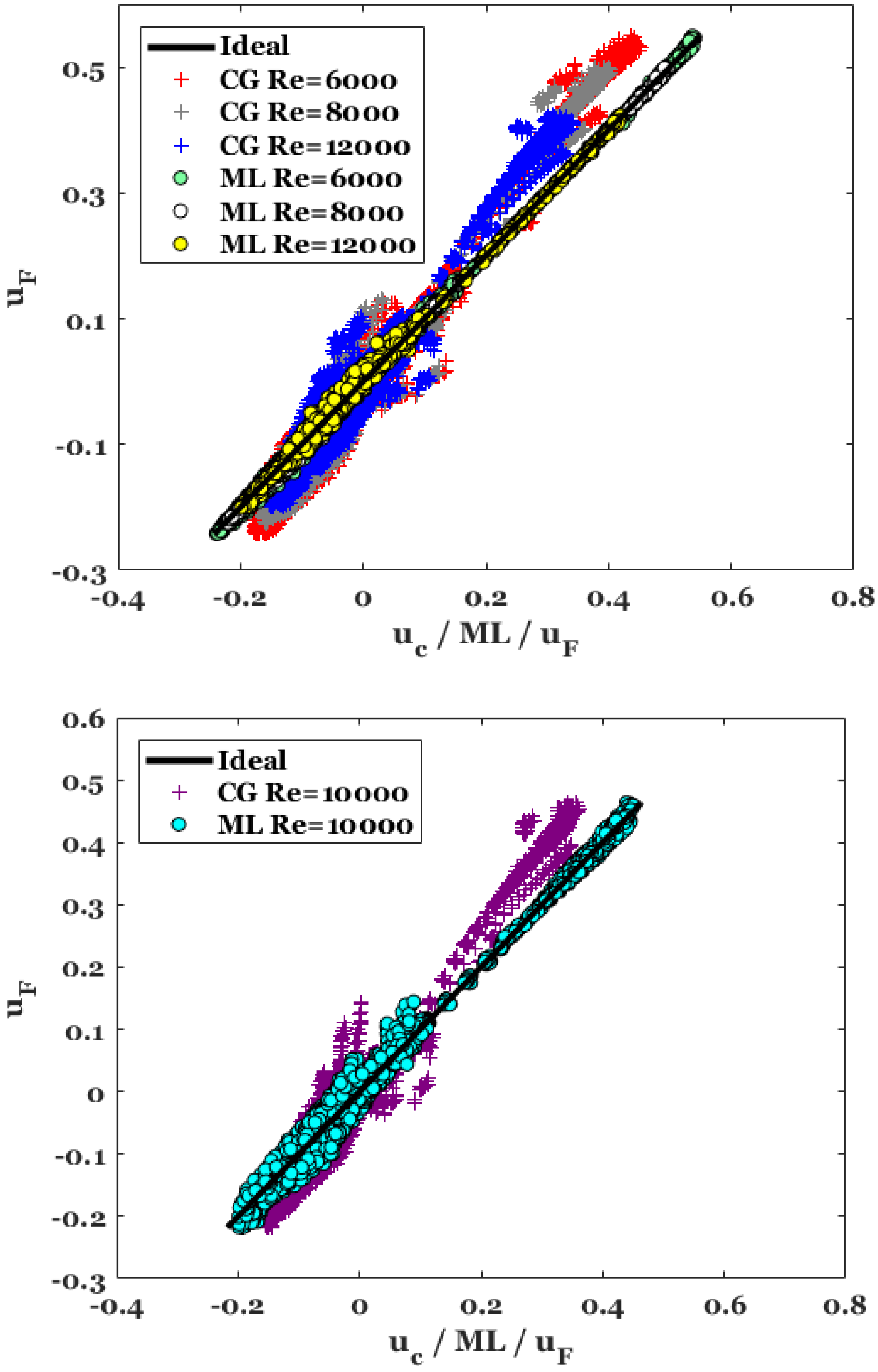
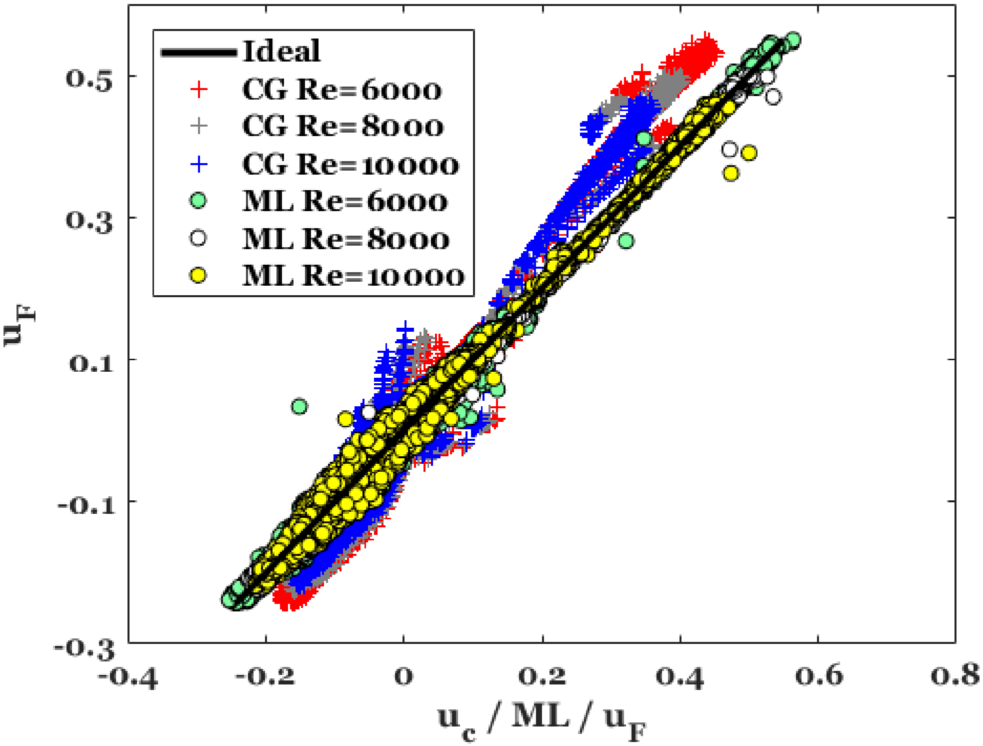


Figure 17. Scenario I (Reynolds number interpolation) for U_x (by RFR). $\Delta = 0.033m$. Training data (above) and testing data (below).

www.manaraa.com



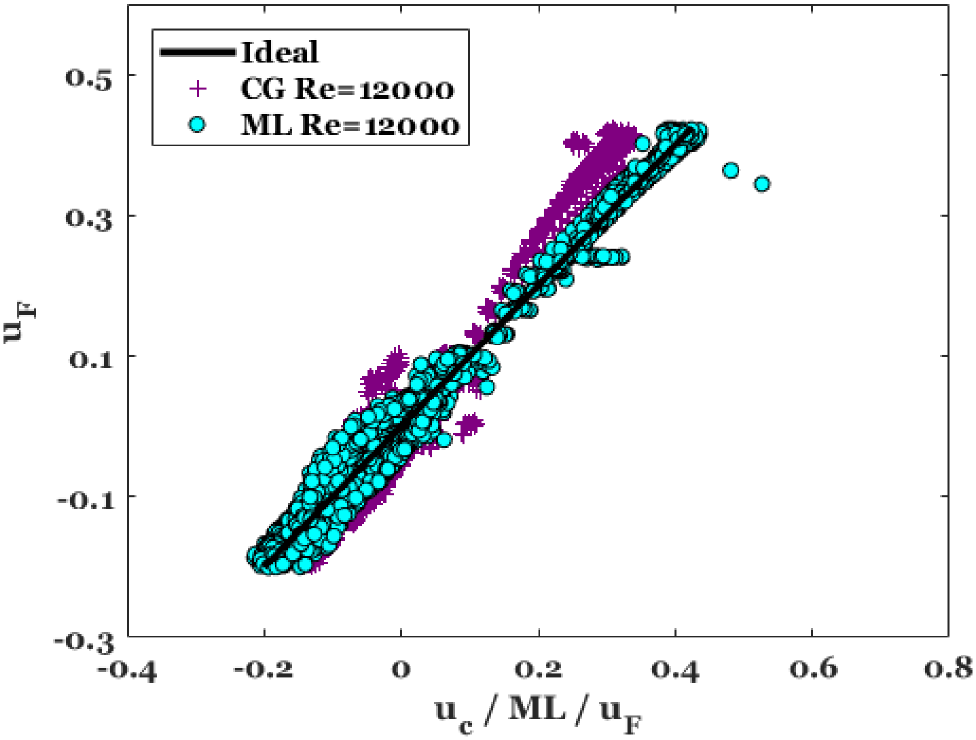


Figure 18. Scenario II (Reynolds number extrapolation to a higher value) for U_x (by ANN). $\Delta = 0.033m$. Training data (above) and testing data (below).

www.manaraa.com

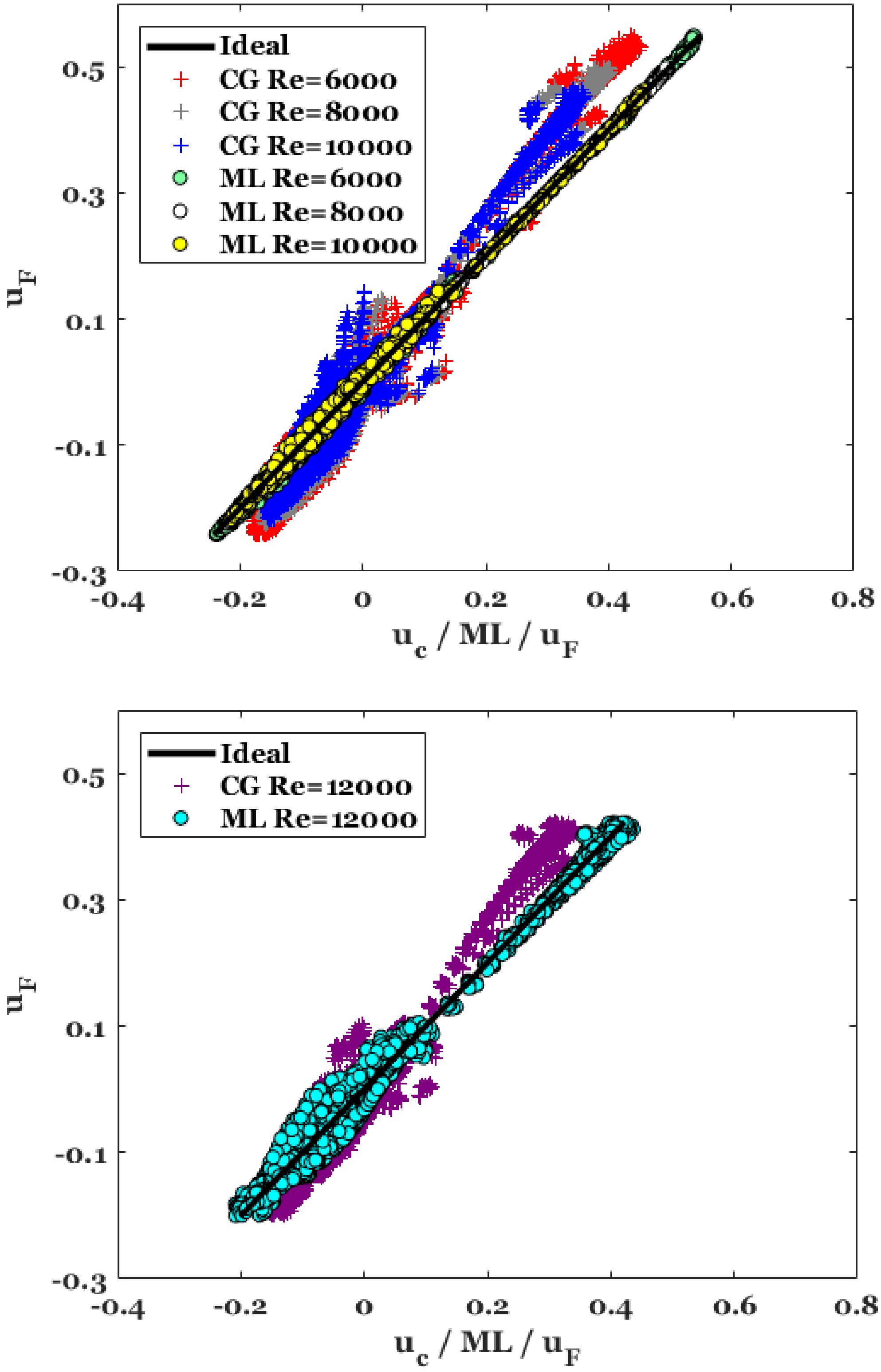


Figure 19. Scenario II (Reynolds number extrapolation to a higher value) for U_x (by RFR). $\Delta = 0.033m$. Training data (above) and testing data (below).

www.manaraa.com

Table 9. A comparison between different scenarios in terms of ML error and computational time.

Scenario	ML	Training data			Testing data		Variable of interest
		Mean E_{ML}	Max E_{ML}	Computational Time (min)	Mean E_{ML}	Max E_{ML}	
I: Re interpolation	ANN	0.3	3.5	12.9	0.4	3.4	U_x
	RFR	0.06	1.5	11.1	0.3	2.8	
II: Extrapolation to a higher Re	ANN	0.3	7.6	13	0.4	7	
	RFR	0.06	1	11.2	0.4	2.7	
III: Extrapolation to a lower Re		0.07	1.3	11.3	0.4	2.9	
IV: Δ interpolation		0.05	2	6.4	0.3	3	
V: Extrapolation to a finer Δ		0.05	1.1	3	0.4	4	
VI: Extrapolation to a coarser Δ		0.05	1.9	8.2	0.4	2	
VII: Re and Δ interpolation.		0.06	1.5	22	0.3	4.6	
VIII: Extrapolation to a higher Re and finer Δ.		0.06	1.2	13.3	0.3	3.3	
IX: Extrapolation to a lower Re and coarser Δ.		0.07	1.8	32.3	0.4	2.3	
VII: Re and Δ interpolation.		0.06	2.2	14.9	0.3	4.9	U_y
VIII: Extrapolation to a higher Re and finer Δ.		0.07	1.9	6.3	0.3	4.6	

www.manaraa.com

Table 10 (continued). A comparison between different scenarios in terms of ML error and computational time.

VII: Re and Δ interpolation.	RFR	0.1	5.3	17.2	0.6	10	U_z
VIII: Extrapolation to a higher Re and finer Δ.		0.1	4.6	6	0.7	13.1	
X: Different Δ in different directions.		0.06	1.7	6.5	0.4	4.3	U_x

Adding more layers to ANN

Statistical learning by ANN can become "deeper" by adding more hidden layers to the ANN as explained in Sections 4.5.1 and 4.5.2 and represented by Figure 20. Increasing the number of layers leads to a better data training for scenarios I and II (smaller ML mean and maximum ML error for training and testing data) as seen in Figure 21. However, RFR still gives better predictions than two or three-layer ANN (compare the ML error values in Table 9 with ML error in Figure 21). There is a chance that multi-layer ANN may give better predictions by adding more layers, but the computational cost of this deep ANN is too high compared to RFR as illustrated in Figure 22. Training deep ANN may take hours compared to minutes when training RFR although RFR still shows better predictive capability. Thus, for the next scenarios, only RFR is utilized.

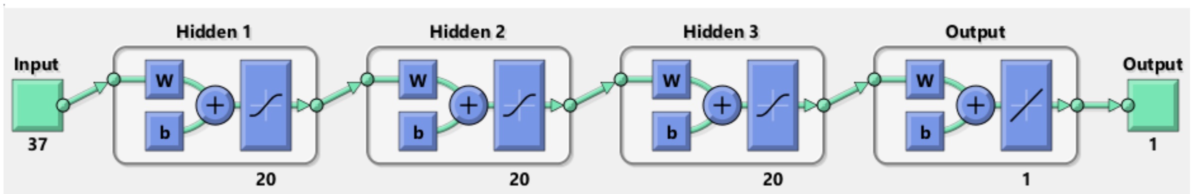


Figure 20. A representation of Three-hidden-layer ANN with 37 inputs, 20 neurons for each layer and one output as represented by MATLAB [25].

المنارة للاستشارات
www.manaraa.com

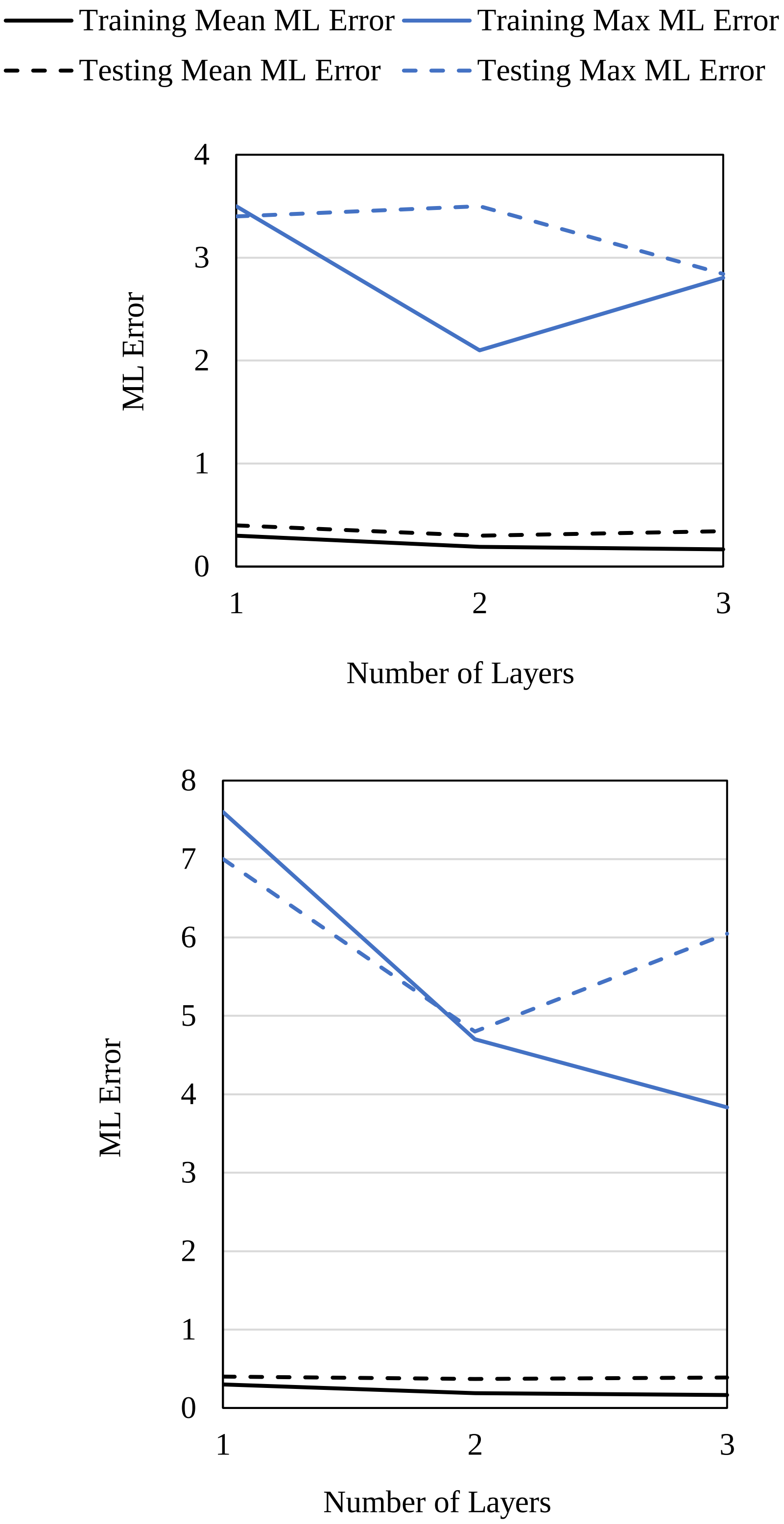


Figure 21. Effect of increasing the number of neural network layers on ML error for the scenario I (above) and the scenario II (below).

www.manaraa.com

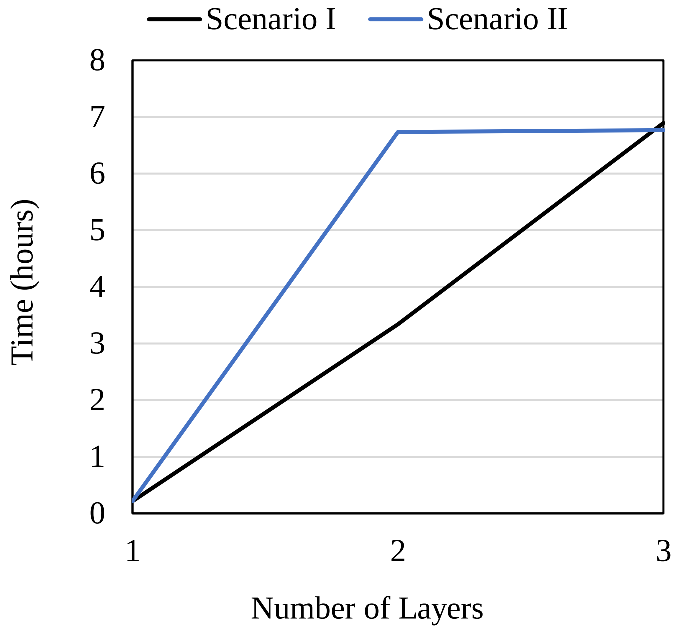


Figure 22. Effect of increasing the number of neural network layers on the computational time.

Why RFR may or may not perform better than deep ANN?

- Deep ANN typically requires a lot of data for training. Although deep ANN was trained with thousands of data points, these points still represent few CFD simulations (e.g. few global Reynolds numbers).
- Deep ANN may work even better given more layers and computational power.

In scenario III (extrapolation to a lower Reynolds number, plotted in Figure 23), RFR still results in good predictions.

www.manaraa.com

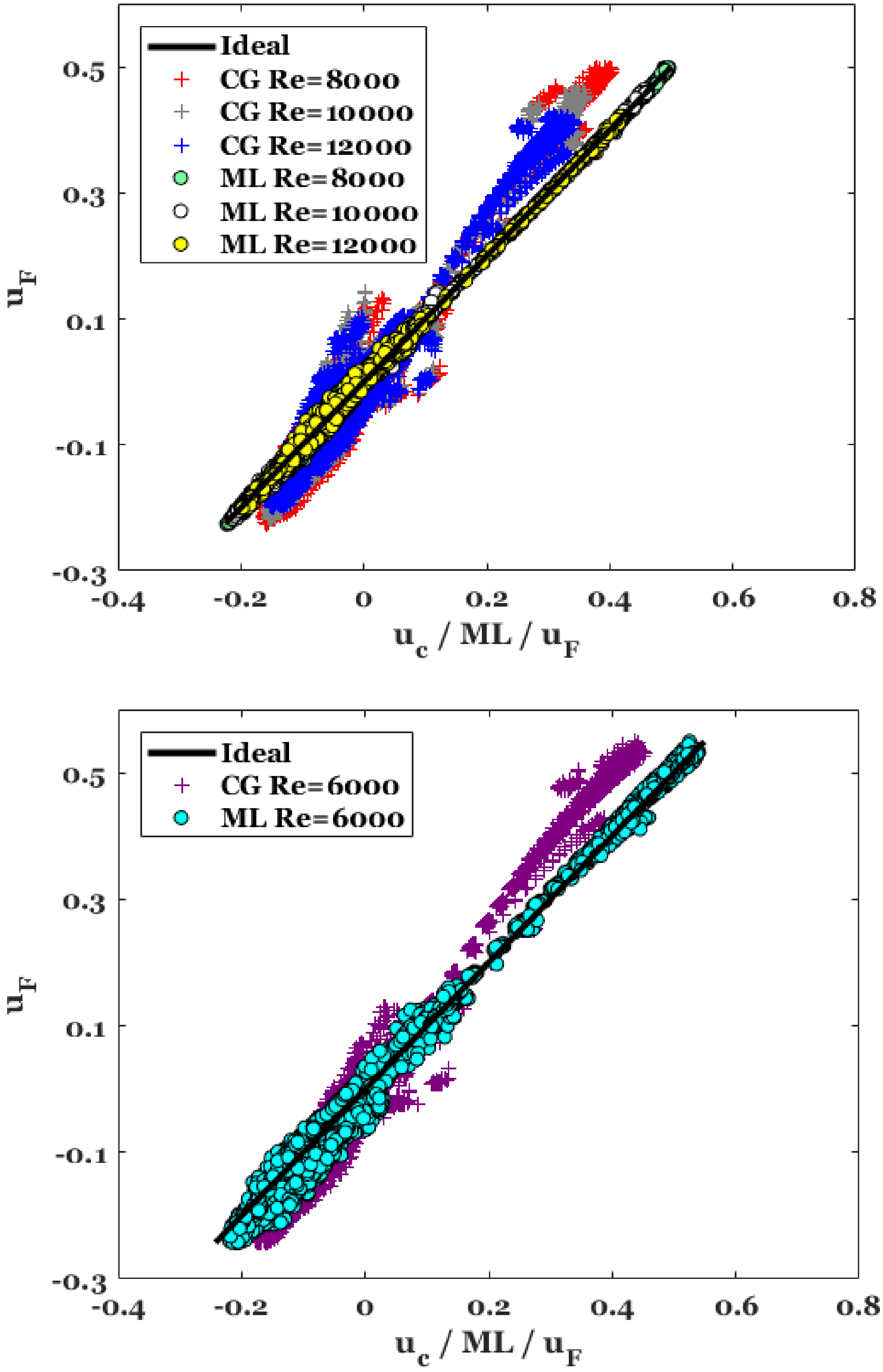


Figure 23. Scenario III (Reynolds number extrapolation to a lower value) for U_x (by RFR). $\Delta = 0.033m$. Training data (above) and testing data (below).

المنارة للاستشارات
www.manaraa.com

6.3.3. Scenarios IV, V and VI: Grid Size Interpolation and Extrapolation.

The focus of scenarios IV, V and VI is the grid size. RFR predictions for these scenarios are illustrated in Figure 24, Figure 25, and Figure 26. RFR performs very good data training for the three scenarios. RFR regression for the testing case is not as good as the training flows' data. RFR predictions for the testing cases (interpolation or extrapolation to a finer or a coarser grid) are still close to the ideal solid line. In Table 9, the mean E_{ML} is around 0.4 for the three scenarios while the maximum E_{ML} increases when extrapolation to a finer grid ($E_{ML} = 4$) and decreases in extrapolation to a coarser grid ($E_{ML} = 2$).

6.3.4. Scenarios VII, VIII and IX: Reynolds Number and Grid Size Interpolation and Extrapolation.

Scenarios VII, VIII and IX are the most challenging ones because the testing flows differ from the training flows in both Reynolds number and grid size. As illustrated in Figure 27, Figure 28 and Figure 29, most RFR predictions are very close the ideal solid line. According to Table 9, for the scenarios VII, VIII, and IX, the mean E_{ML} is 0.4 or lower, and the maximum E_{ML} is not high compared to the range of E_{ML} for previous scenarios which are less demanding.

6.3.5. Other Variables $\boldsymbol{U_y}$ and $\boldsymbol{U_z}$

All the previous numerical experiments were performed assuming that the variable of interest is U_x (which corresponds to the velocity direction of the cavity lid motion). In this section, the variables U_y and U_z are considered with scenarios VII and VIII (in Table 7). Both scenarios VII and VIII were selected as they represent the more general scenarios (with testing flows' grid size and Reynolds number different from the training flows). Note that the features' vector defined in Equation (5-4) remains unchanged while the output of interest ($\varepsilon_{\Delta}(U_x)$) is replaced by $\varepsilon_{\Delta}\left(U_y\right)$ or $\varepsilon_{\Delta}(U_z)$.

www.manaraa.com

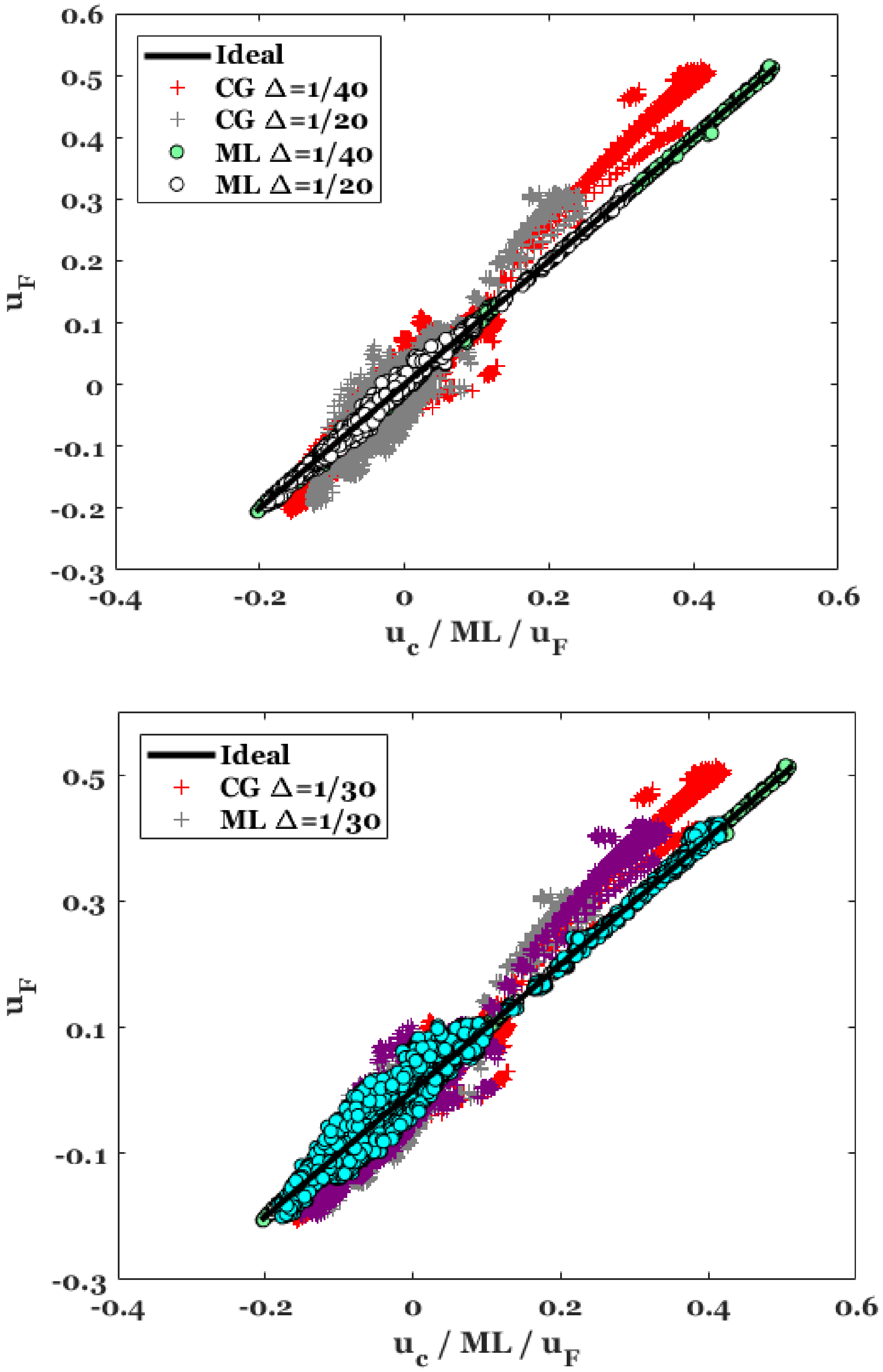


Figure 24. Scenario IV (grid size interpolation) for U_x (by RFR). $Re = 12000$. Training data (above) and testing data (below).

المنارة للاستشارات
www.manaraa.com

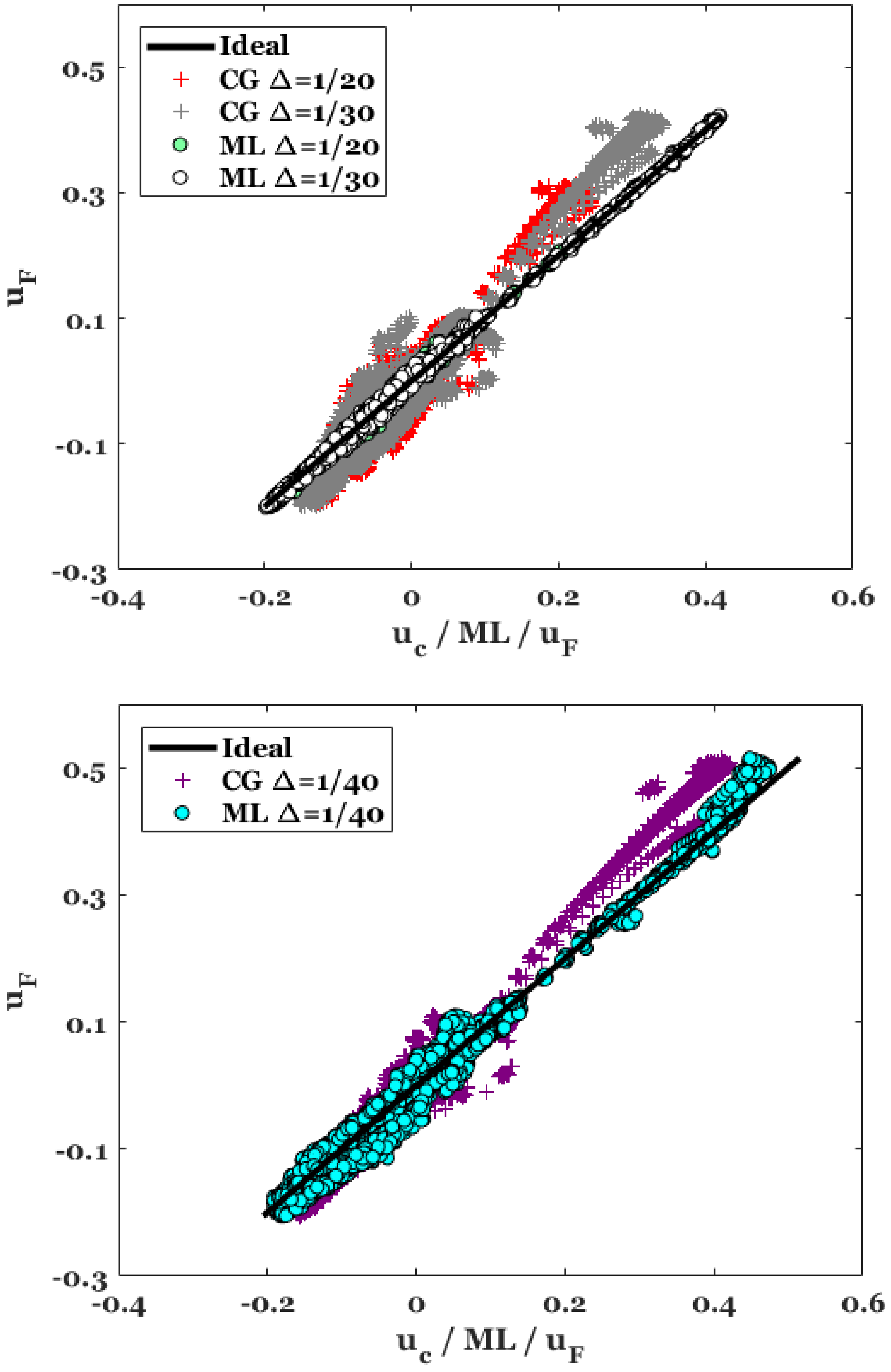


Figure 25. Scenario V (grid size extrapolation to a finer grid) for U_x (by RFR). $Re = 12000$. Training data (above) and testing data (below).

المنارة للاستشارات

www.manaraa.com

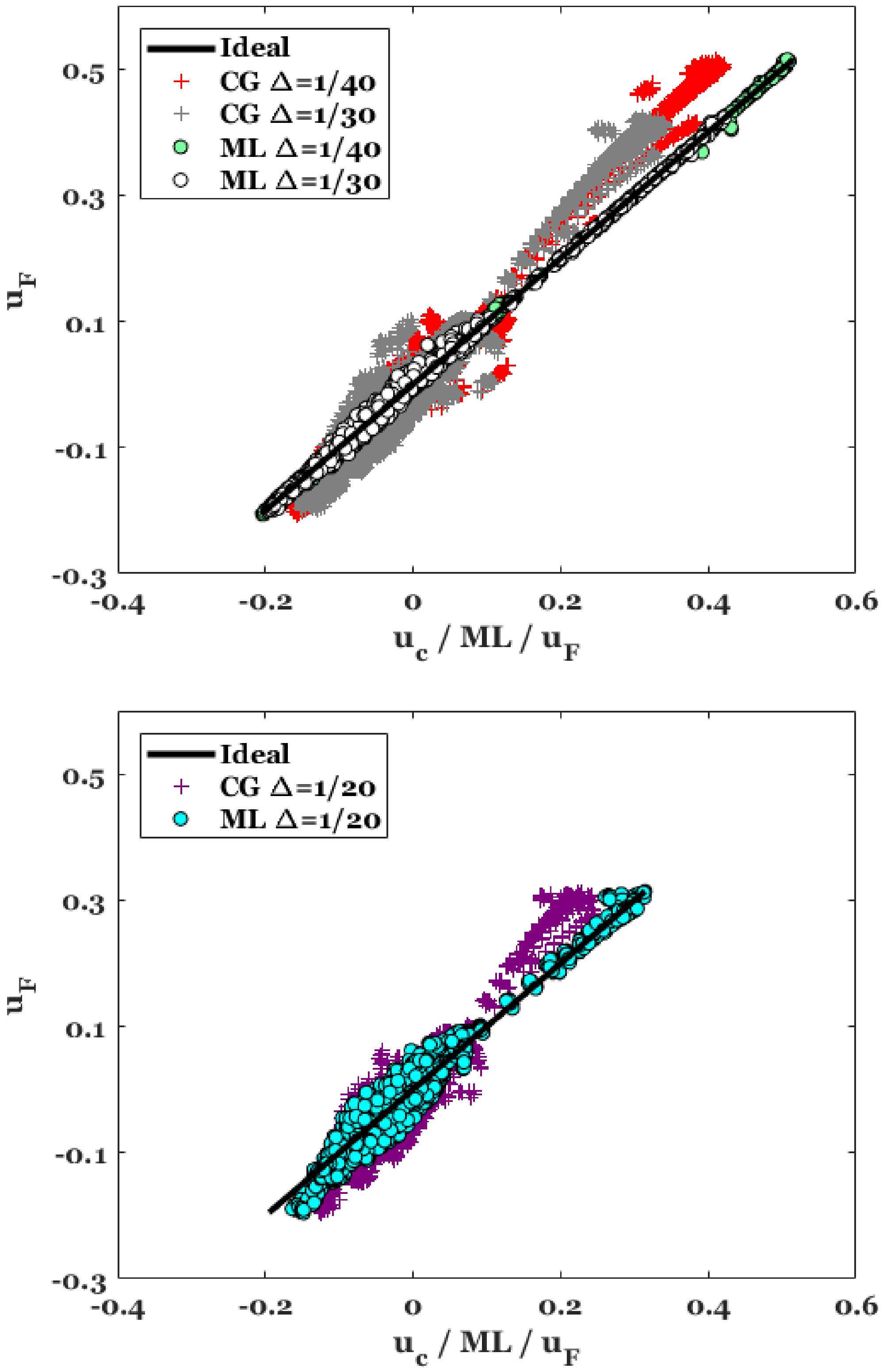


Figure 26. Scenario VI (grid size extrapolation to a coarser grid) for U_x (by RFR). $Re = 12000$. Training data (above) and testing data (below).

المنارة للاستشارات
www.manaraa.com

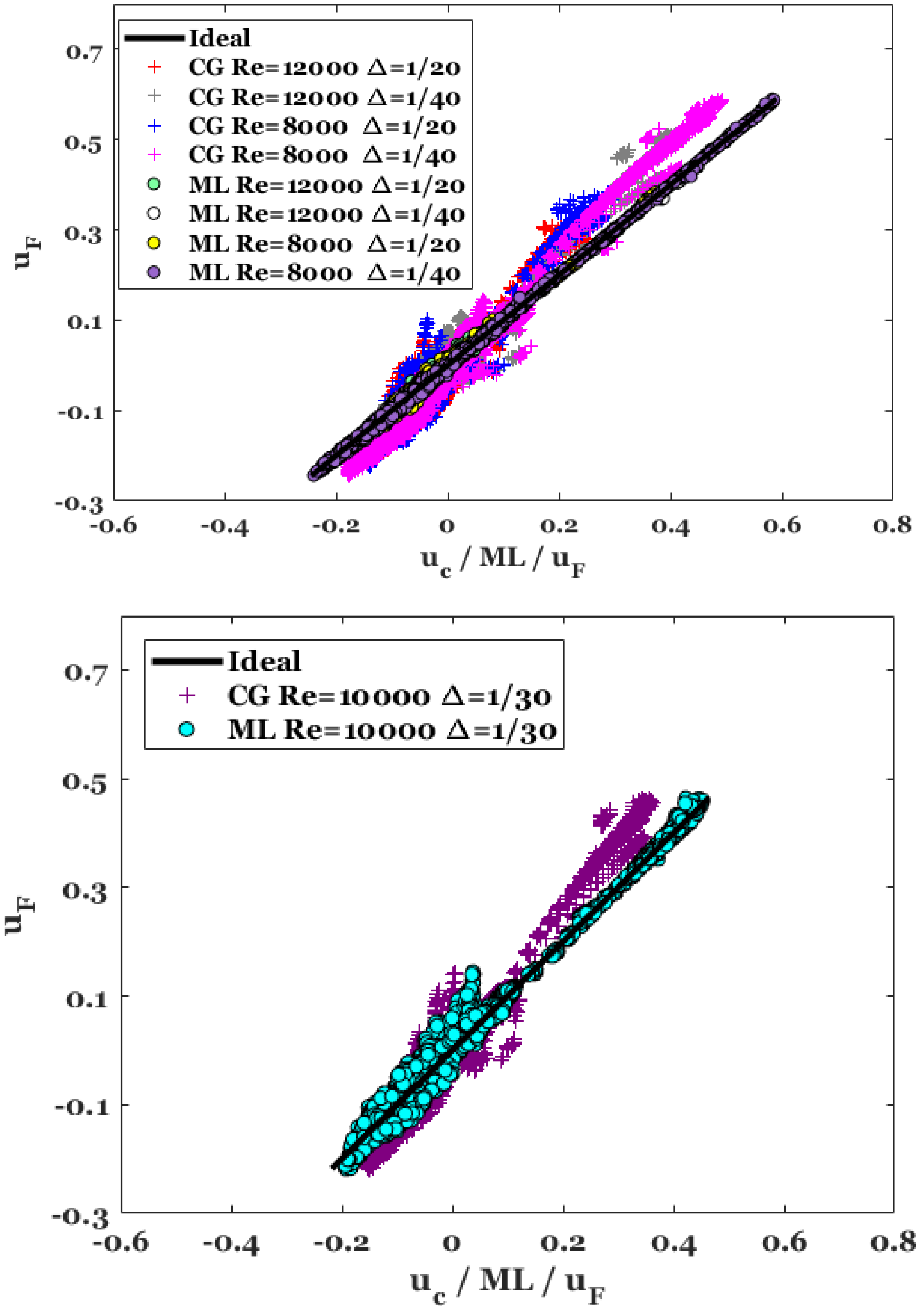


Figure 27. Scenario VII (Reynolds number and grid size interpolation) for U_x (by RFR). Training data (above) and testing data (below).

www.manaraa.com

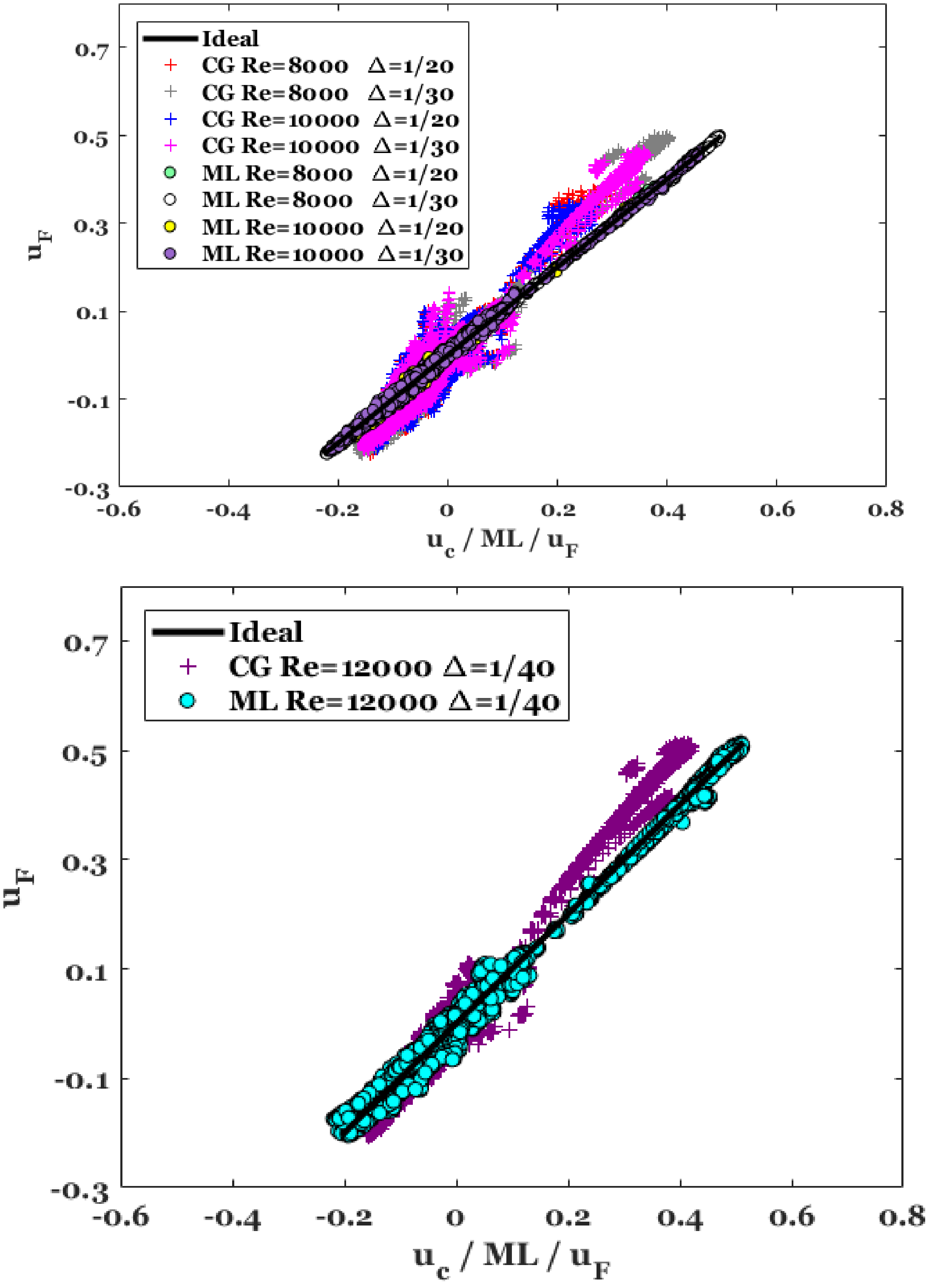


Figure 28. Scenario VIII (extrapolation to a higher Reynolds number and a finer grid) for U_x (by RFR). Training data (above) and testing data (below).

www.manaraa.com

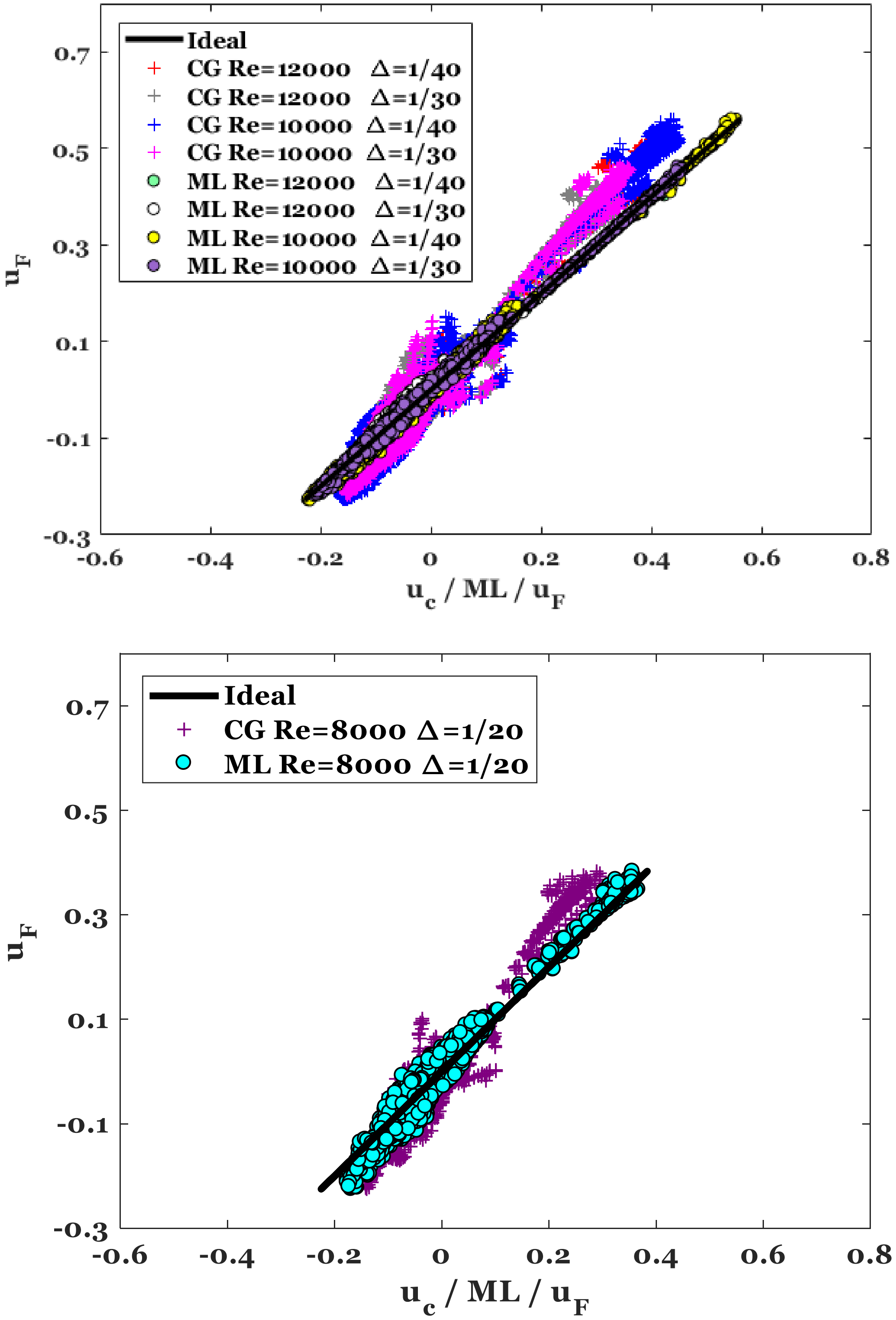


Figure 29. Scenario IX (extrapolation to a lower Reynolds number and a coarser grid) for U_x (by RFR). Training data (above) and testing data (below).

المنارة للاستشارات
www.manaraa.com

In Figure 30 and Figure 31, RFR predictions, for the velocity U_y, are compared to the true values. In both cases, RFR prediction gives an improvement over the coarse-grid predictions for most points. On the other hand, RFR performance is poor when predicting U_z in Figure 32 and Figure 33. RFR predictions for U_z give no improvement over the coarse-grid results.

The performance of RFR in predicting U_y and U_z is assessed in Table 9 in terms of E_{ML}. For the velocity component, U_y, E_{ML} is comparable to (a little bit higher than) E_{ML} when computing U_x. On the other hand, for the velocity U_z, ML errors for the testing data are very high compared to U_x and U_y. For U_z, the correlation between the fine-grid velocity and the coarse-grid velocity is very poor. U_z is very small (and less significant) compared to the other velocity components, so it is sensitive to any ML error.

6.3.6. Computational Time for Big Data

The capability of the trained model using Random Forest is expected to increase when the model is trained using a large dataset (more simulation cases). This large dataset leads to a higher computational time when training RFR. The effect of the data amount of the computational cost is depicted in Figure 34 using different scenarios in Table 7. The computational cost seems to increase almost linearly with the amount of training data points.

6.3.7. Data Convergence

In the proposed approach, the flow of interest (testing flow) simulation grid-induced error is predicted given a surrogate model based on data from the available training flows. It is necessary to explore how this surrogate model accuracy will be impacted by adding or removing some data. The surrogate model should give better performance with more training flows. This is called “data convergence”.

www.manaraa.com

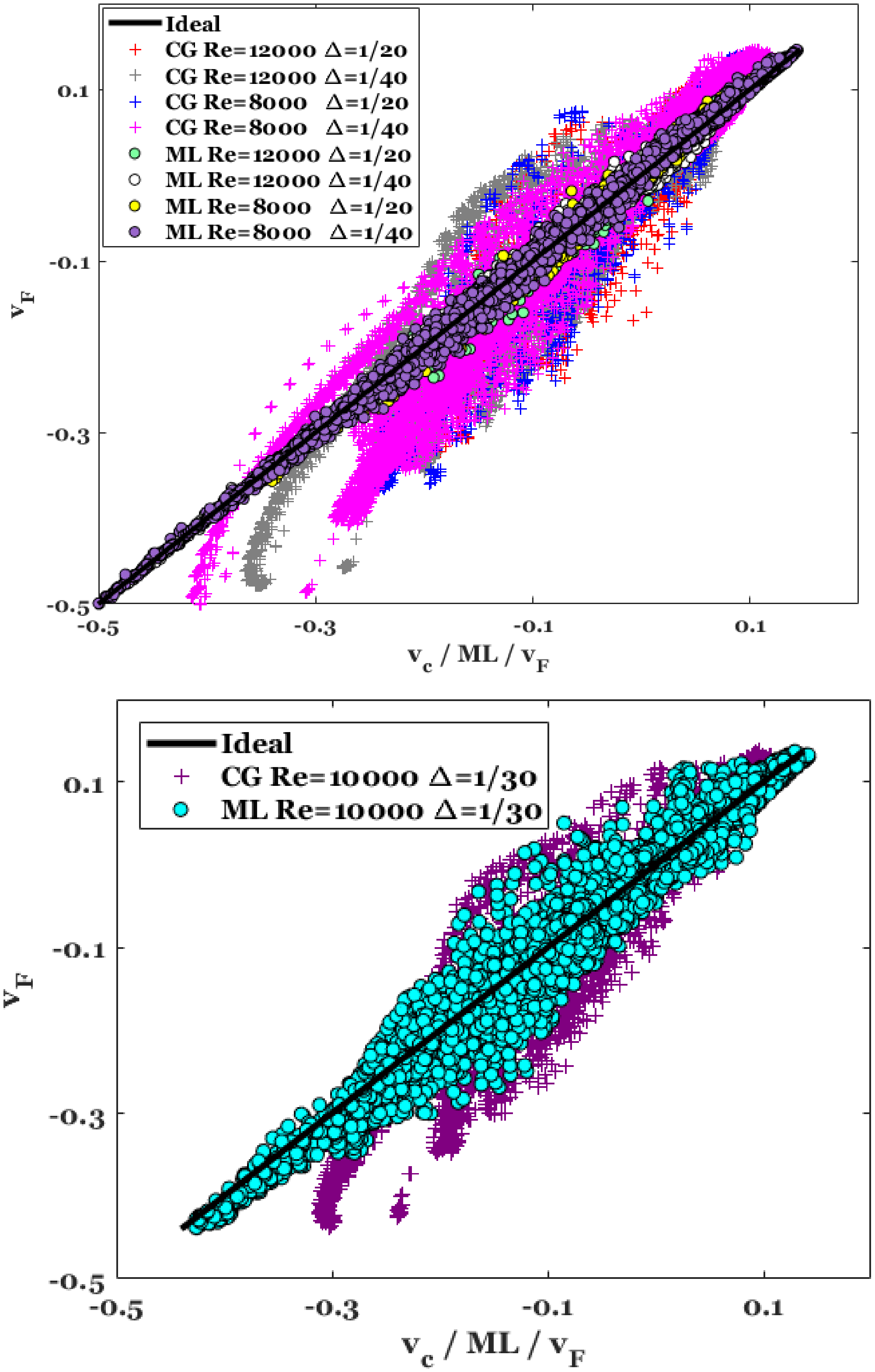


Figure 30. Scenario VII (Reynolds number and grid size interpolation) for U_y (by RFR). Training data (above) and testing data (below).

المنارة للاستشارات
www.manaraa.com

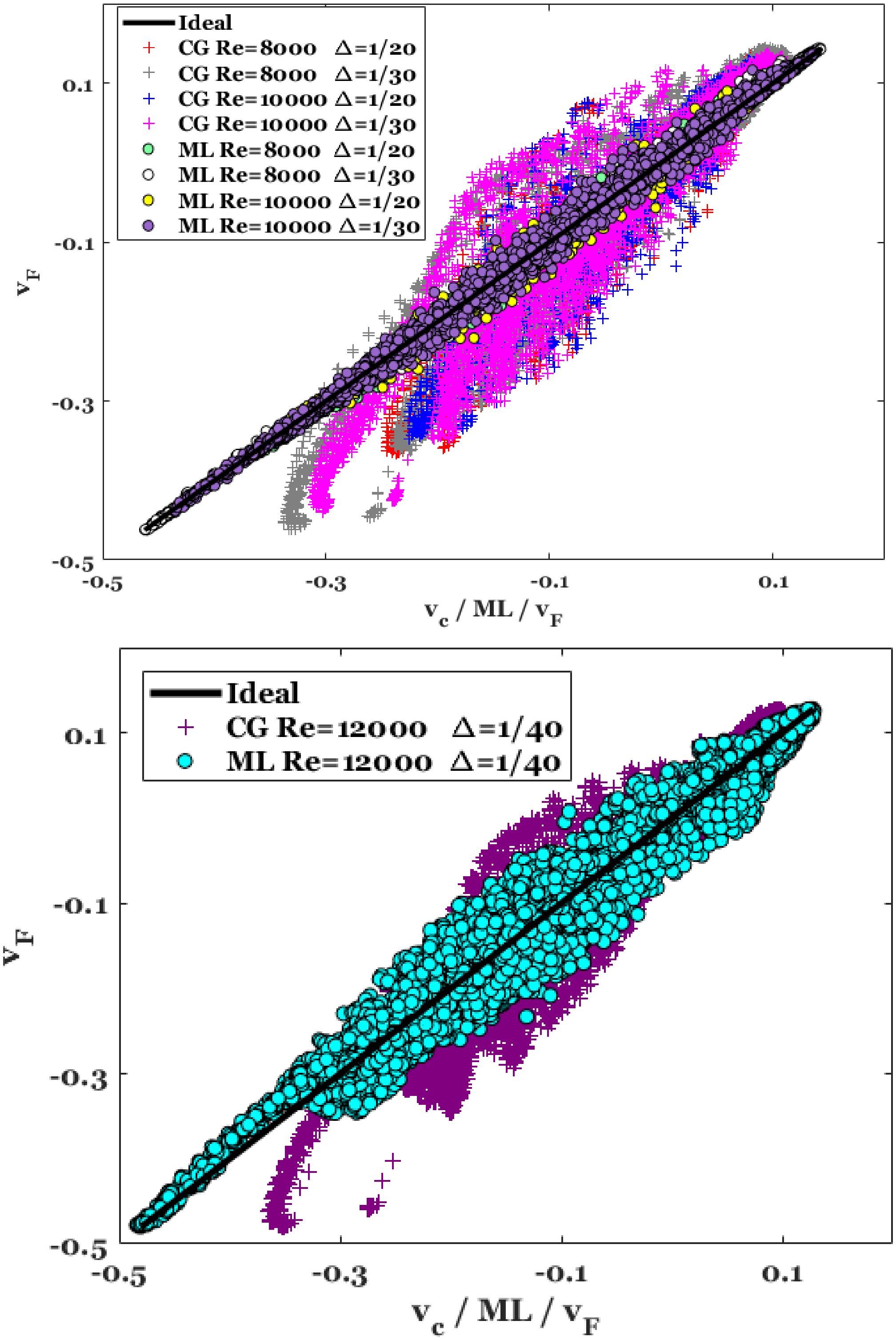


Figure 31. Scenario VIII (extrapolation to a higher Reynolds number and a finer grid) for U_y (by RFR). Training data (above) and testing data (below).

المنارة للاستشارات
www.manaraa.com

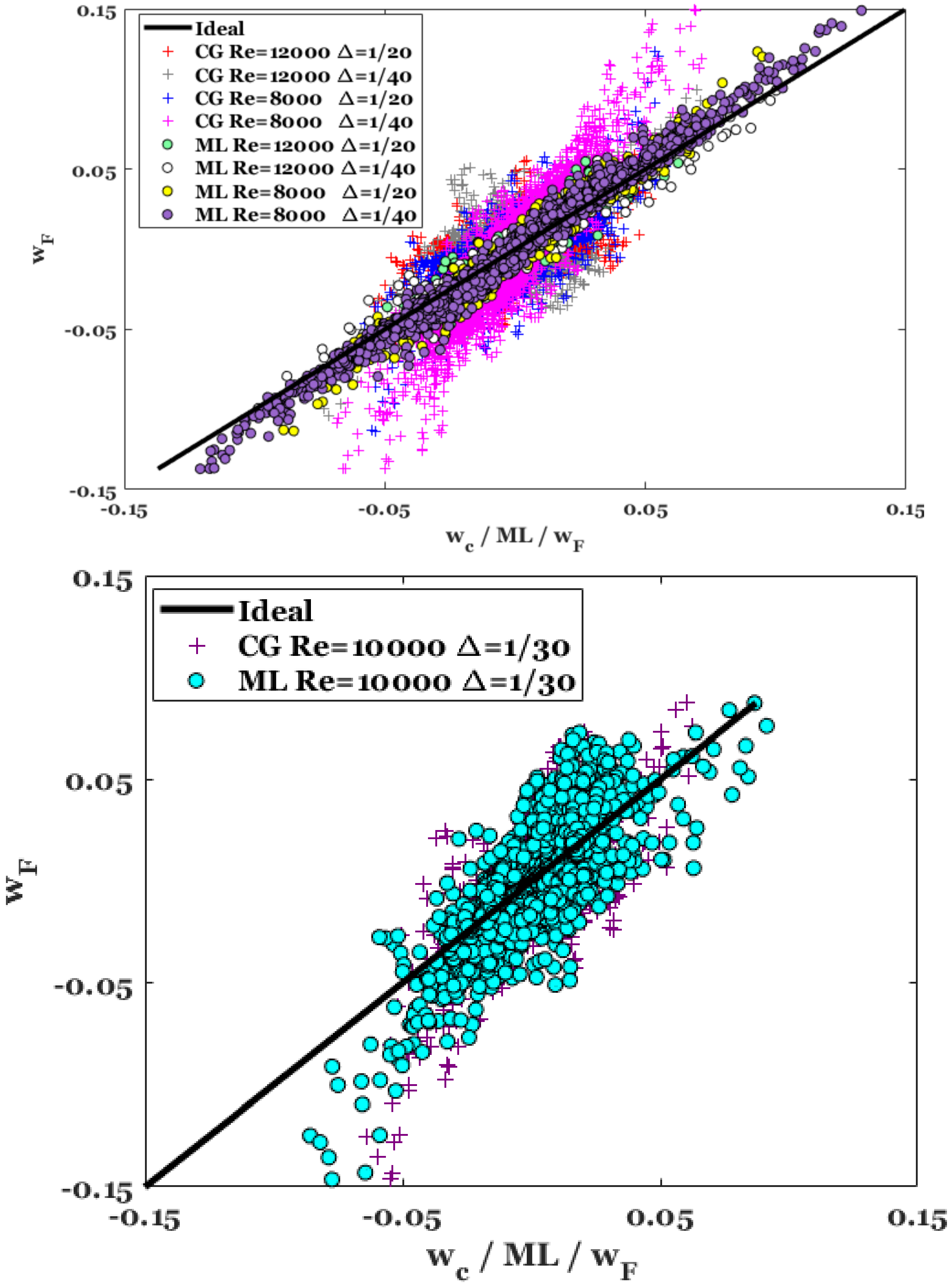


Figure 32. Scenario VII (Reynolds number and grid size interpolation) for U_z (by RFR). Training data (above) and testing data (below).

www.manaraa.com

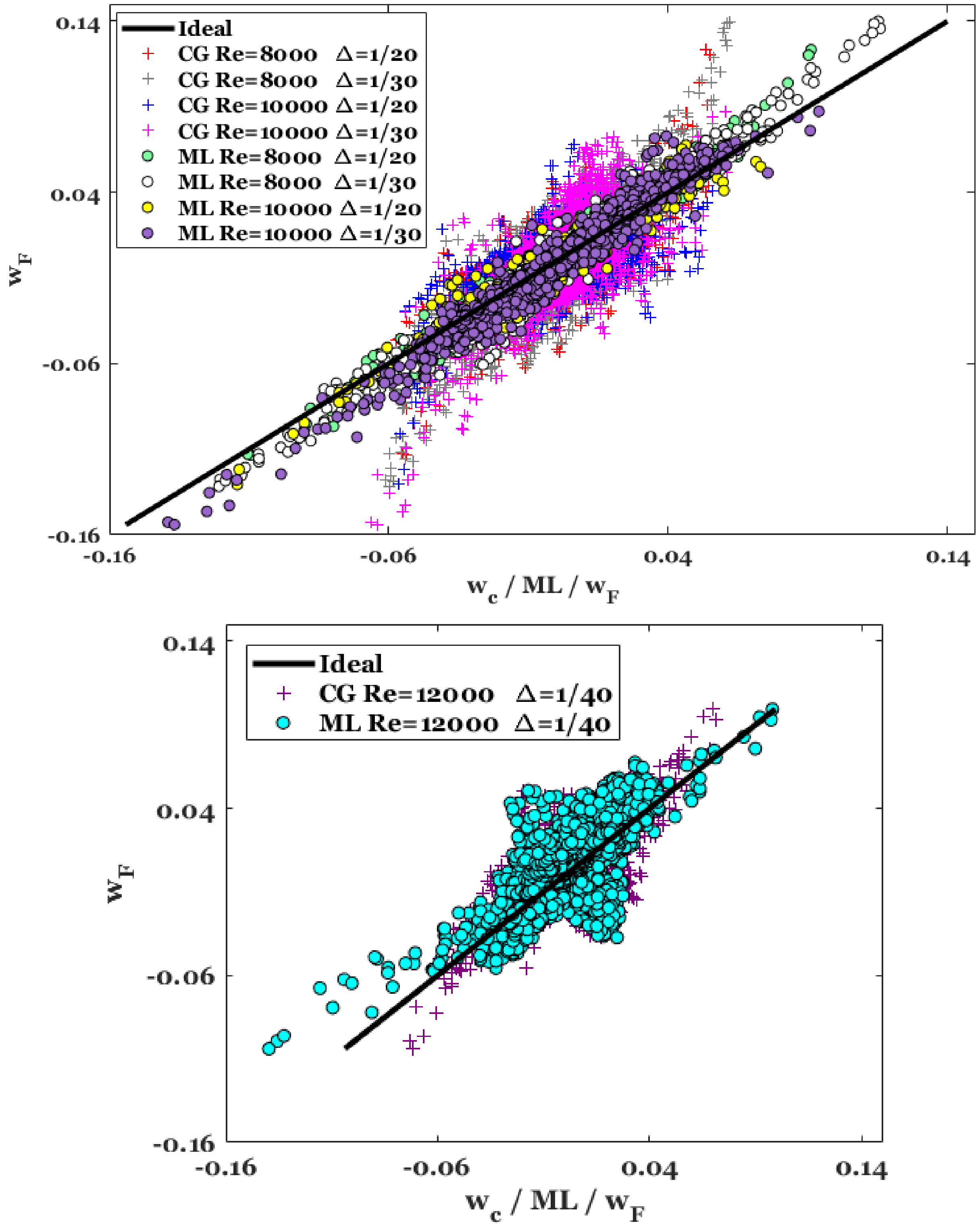


Figure 33. Scenario VIII (extrapolation to a higher Reynolds number and a finer grid) for *Uz* (by RFR). Training data (above) and testing data (below).

المنارة للاستشارات
www.manaraa.com

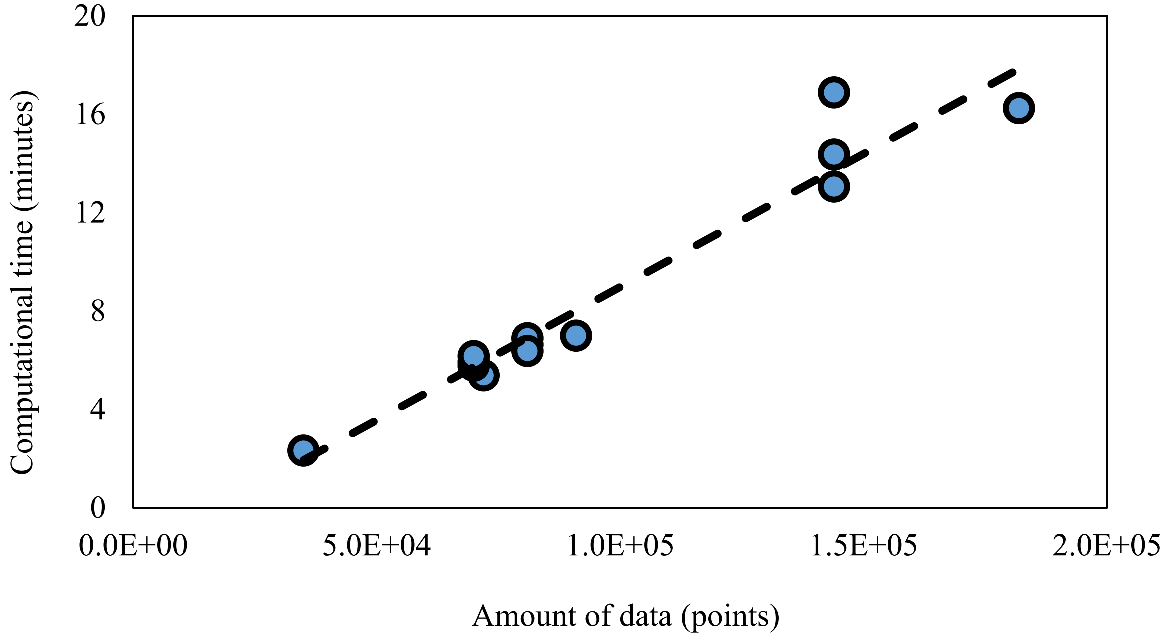


Figure 34. Big-data computational cost using random forest regression.

Four studies of data convergence are presented in Table 11 (assuming that U_x is the variable of interest). In the first study, all the training and testing flows have the same grid size ($\Delta = 1/30$ m) while the Reynolds number changes. Adding more training data with Reynolds number closer to the testing data leads to a lower E_{ML}. In the second study, adding more training data with Reynolds number far from the testing data, led to a lower mean E_{ML} and a little change in the maximum E_{ML}. The studies III and IV focus on adding data with different grid size while the Reynolds number is fixed. The studies III and IV led to results similar to the first and the second studies.

المنارة للاستشارات
www.manaraa.com

Table 11. Data convergence study.

#	Testing flows	Training flows	Testing data mean E_{ML}	Testing data maximum E_{ML}
I	$Re = 12000,$ $\Delta = 1/30m$	$Re = 6000, \Delta = 1/30m$	0.4	3.21
		$Re = 6000, \Delta = 1/30m$ $Re = 8000, \Delta = 1/30m$	0.36	2.78
		$Re = 6000, \Delta = 1/30m$ $Re = 8000, \Delta = 1/30m$ $Re = 10000, \Delta = 1/30m$	0.35	2.71
II	$Re = 12000,$ $\Delta = 1/30m$	$Re = 10000, \Delta = 1/30m$	0.4	2.67
		$Re = 10000, \Delta = 1/30m$ $Re = 8000, \Delta = 1/30m$	0.36	2.63
		$Re = 10000, \Delta = 1/30m$ $Re = 8000, \Delta = 1/30m$ $Re = 6000, \Delta = 1/30m$	0.35	2.71
III	$Re = 12000,$ $\Delta = 1/40m$	$Re = 12000, \Delta = 1/20m$	0.5	4.63
		$Re = 12000, \Delta = 1/20m$ $Re = 12000, \Delta = 1/30m$	0.38	4
IV	$Re = 12000,$ $\Delta = 1/40m$	$Re = 12000, \Delta = 1/30m$	0.39	3.89
		$Re = 12000, \Delta = 1/30m$ $Re = 12000, \Delta = 1/20m$	0.38	4

6.3.8. Scenario X: Different grid spacing in different directions.

The focus of this scenario is varying the grid size spacing in the three Cartesian directions. Training simulations are performed with 3 uniform grids while the testing simulation is performed

المنارة للاستشارات
www.manaraa.com

with different grid spacing in each direction. RFR predictions, for the testing case velocity spatial distribution, are reasonably better than the coarse-grid predictions as illustrated in Figure 35.

6.3.9. Scenario XI, XII and XIII (Extrapolation to Aspect Ratio =1.2).

RFR, that was trained based on the flow field in the cubic cavity, is tested with a rectangular cavity whose aspect ratio =1.2 (aspect ratio extrapolation). RFR predictions for the rectangular cavity is close to the ideal solid line (as depicted in Figure 36). Extrapolation is also performed in terms of aspect ratio and Reynolds number together (see Figure 37) or even in the direction of aspect ratio, Reynolds number, and grid size together (see Figure 38). In both cases, RFR predictions introduced significant improvement over the coarse-grid predictions. The ML error that occurs because of extrapolation to a higher aspect ratio is not different from the ML error when extrapolating in terms of Reynolds number and grid size (see the range of ML errors in Table 12).

6.3.10. Scenario XIV, XV, XVI, XVII and XVIII (Aspect Ratio Interpolation and Extrapolation up to Aspect Ratio = 4).

These scenarios are more challenging because the difference between the aspect ratios of the training and testing data is increased. Figure 39 illustrates training RFR based on the flow data of cavities with aspect ratios 1 and 4. Next, RFR model is tested with flow data of a cavity whose aspect ratio = 2 (interpolation mode). RFR predictions are significantly better than coarse-grid prediction (in Figure 39). However, it was found that RFR model performance can be improved by adding extra features (closest distance to the wall and the distance to the lid). These features improve the RFR predictions because the grid-induced error is typically higher near the boundaries (high flow variables' gradients). The improvement that results from adding extra flow features can be noticed in Figure 40 and Table 12 (ML maximum error decreases).

المنارة للاستشارات
www.manaraa.com

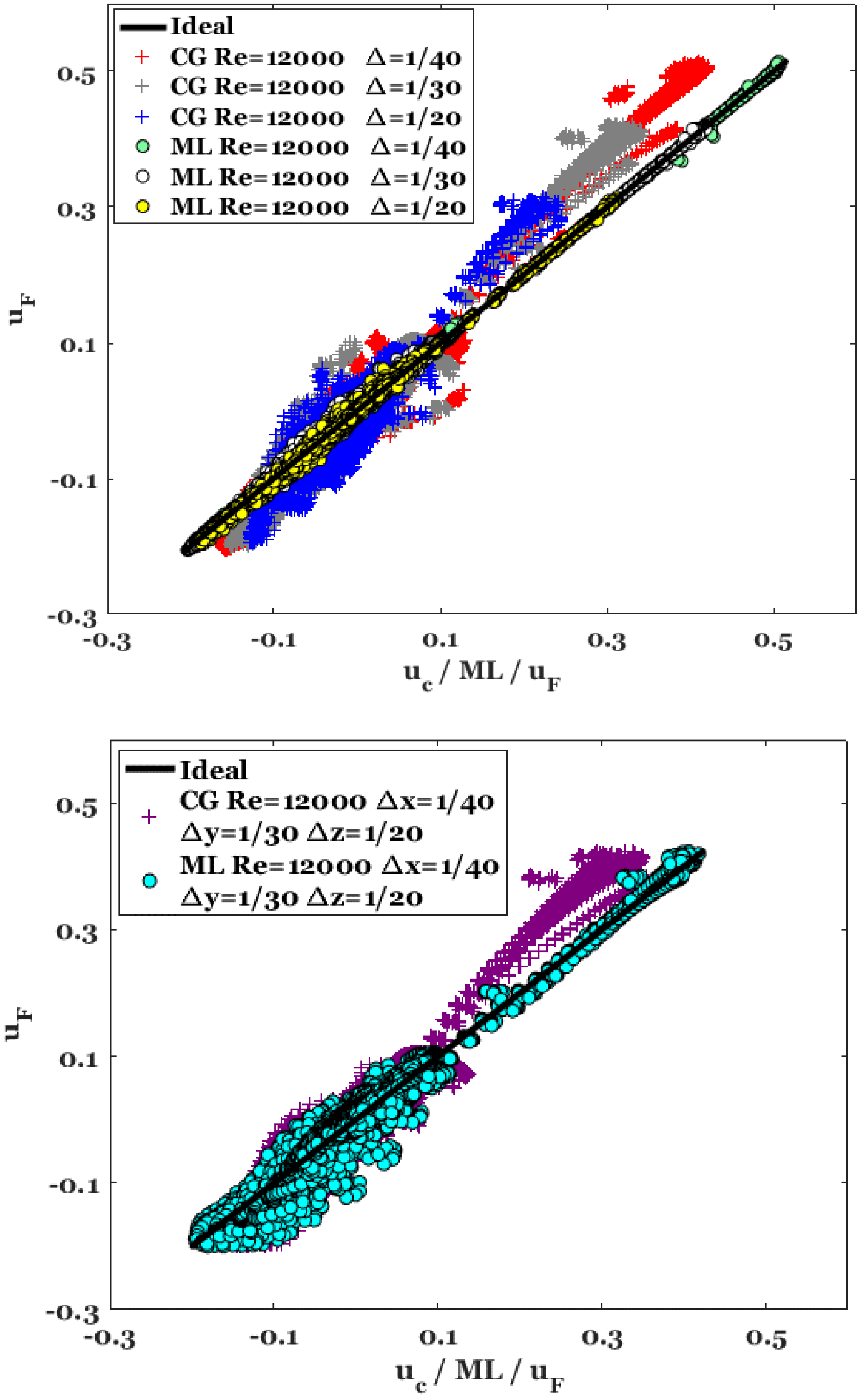


Figure 35. Scenario X (different grid spacing in different directions) for U_x (by RFR). Training data (above) and testing data (below).

المنارة للاستشارات
www.manaraa.com

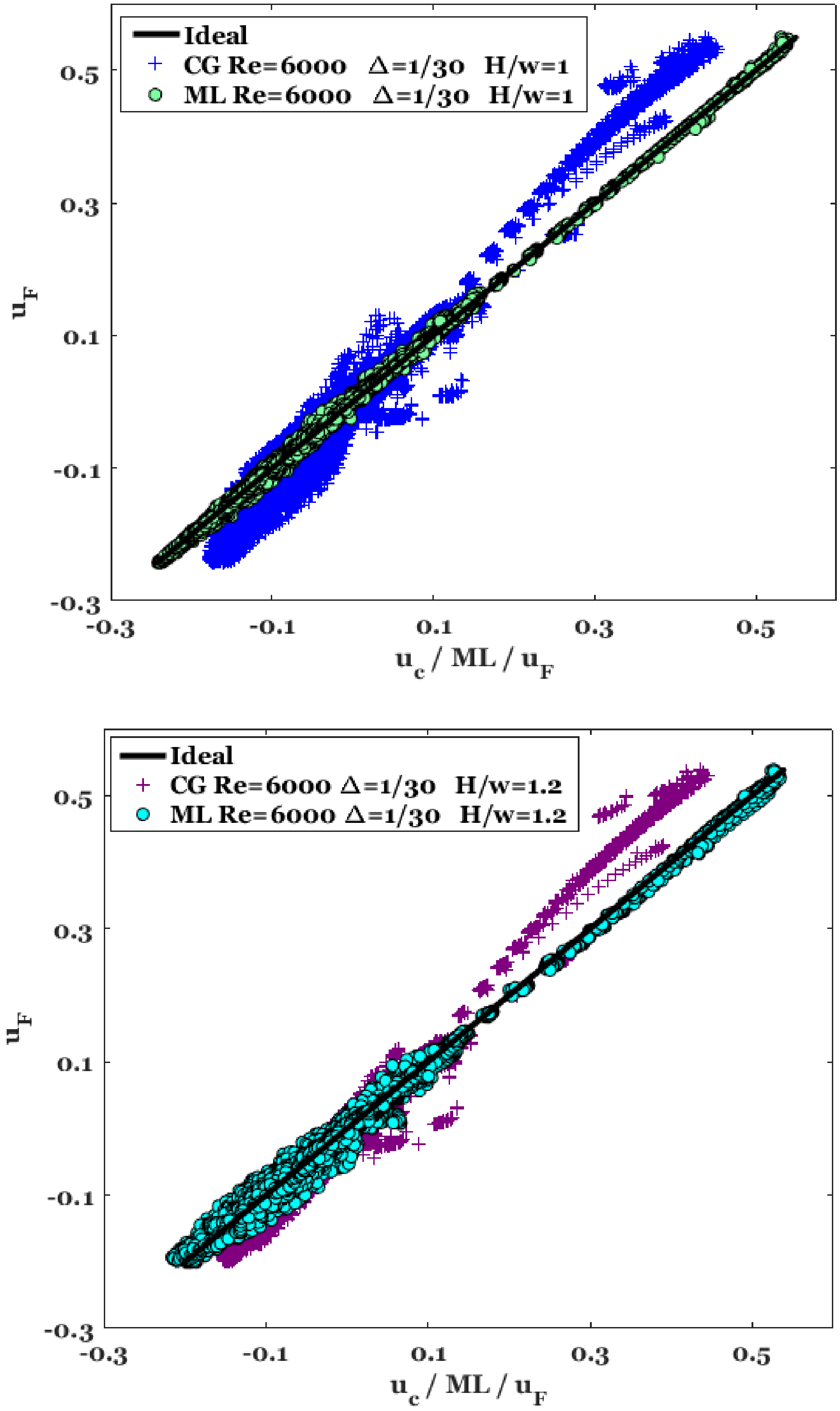


Figure 36. Scenario XI (aspect ratio extrapolation) for U_x (by RFR). Training data (above) and testing data (below).

المنارة للاستشارات

www.manaraa.com

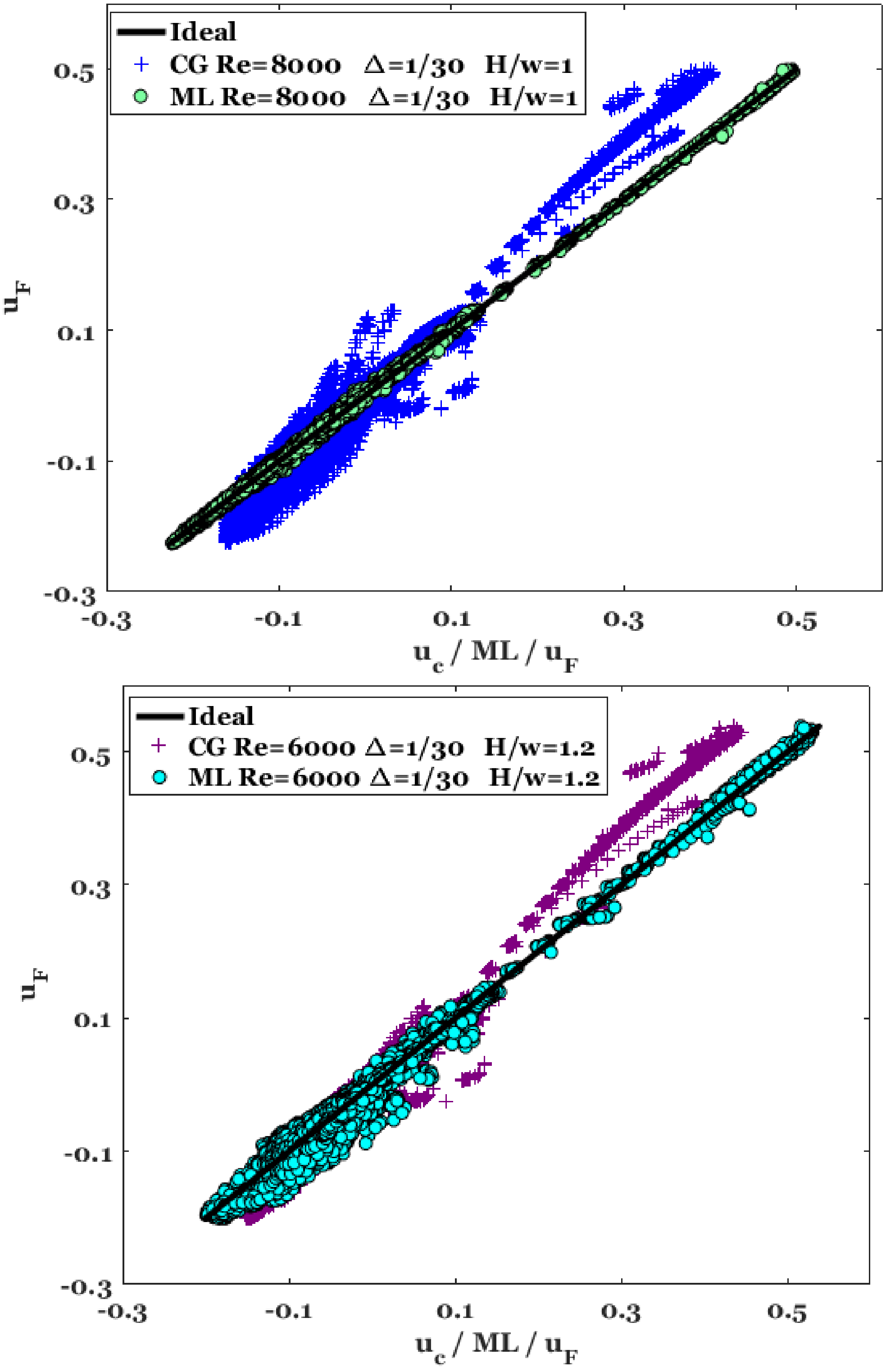


Figure 37. Scenario XII (aspect ratio and Reynolds number extrapolation) for U_x (by RFR). Training data (above) and testing data (below).

www.manaraa.com

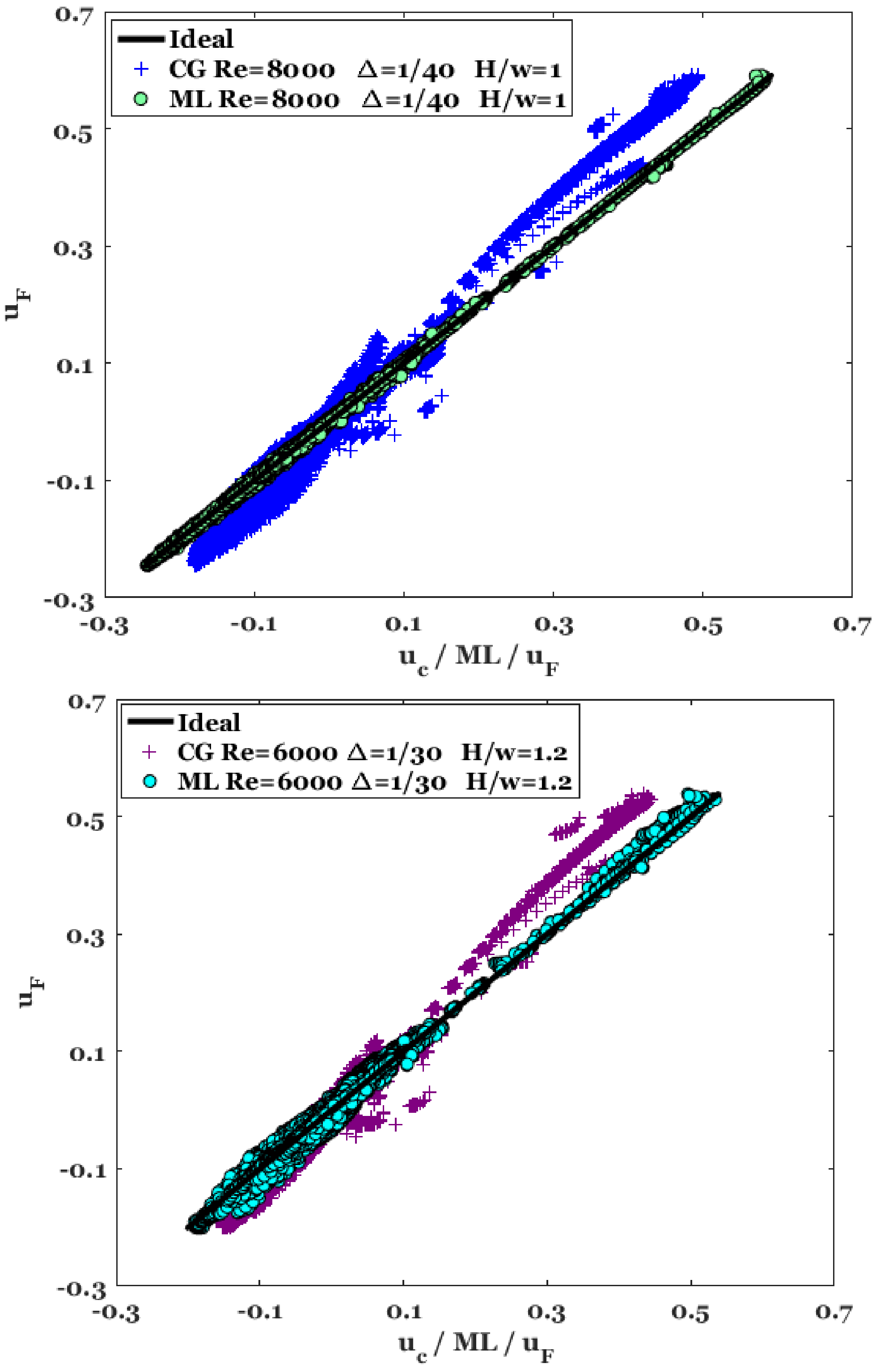


Figure 38. Scenario XIII (aspect ratio, Reynolds number, and grid size extrapolation) for U_x (by RFR). Training data (above) and testing data (below).

المنارة للاستشارات
www.manaraa.com

Table 12. A comparison between different scenarios (second set of scenarios in terms of Re, Δ and (H/W)) in terms of ML error.

Scenario	Training data		Testing data	
	Mean (E_{ML})	Max (E_{ML})	Mean (E_{ML})	Max (E_{ML})
XI: Extrapolation to $(H/W) = 1.2$	0.05	1	0.3	2.9
XII: Extrapolation to $(H/W) = 1.2$ and smaller Re.	0.05	0.8	0.3	3.9
XIII: Extrapolation to $(H/W) = 1.2$, smaller Re and coarser Δ.	0.04	0.9	0.4	2.4
XIV: (H/W) interpolation up to aspect ratio = 4	0.03	4.1	0.3	3.2
XIV: (H/W) interpolation up to aspect ratio = 4 (after adding new features).	0.02	1.3	0.3	2
XV: Extrapolation to a lower (H/W).	0.02	0.6	1	16.5
XVI: Extrapolation to a higher (H/W).	0.03	1	0.5	4

In both Figure 39 and Figure 40, some crossed points are observed to deviate away from the general trend of the whole data. These points are corresponding to the velocity field computed in the cells adjacent to the lid. The same set of features (after adding the new ones) are utilized when extrapolating to a lower and higher aspect ratio (Figure 41 and Figure 42 respectively). In both cases, the testing data have a small ML error, on the average, but the maximum ML error, when extrapolating to a lower aspect ratio, is high (RFR prediction, in some locations, is worse than the coarse-grid prediction as illustrated in Table 12).

المنارة للاستشارات
www.manaraa.com

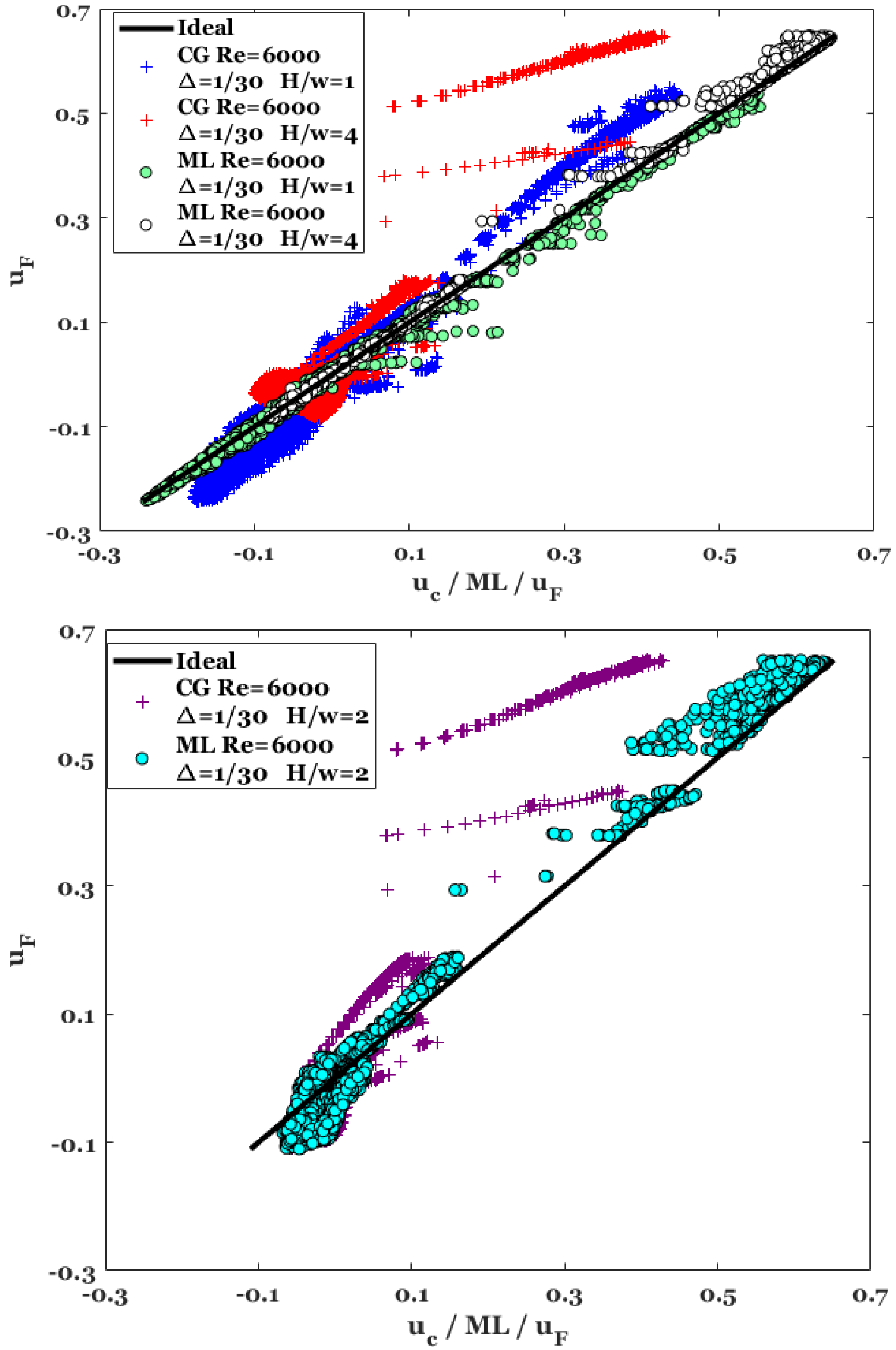


Figure 39. Scenario XIV (aspect ratio interpolation) for U_x (by RFR). Training data (above) and testing data (below).

المنارة للاستشارات
www.manaraa.com

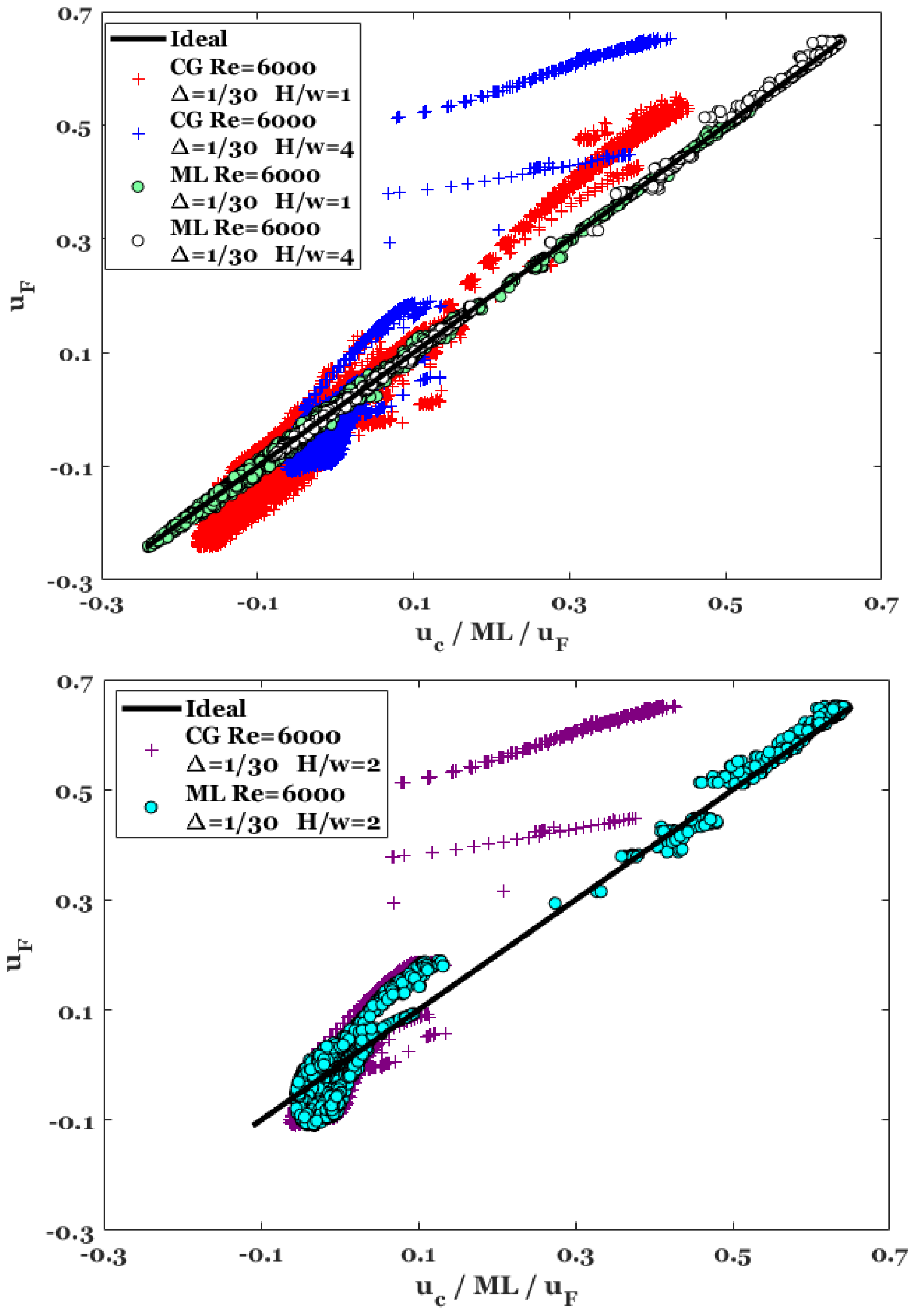


Figure 40. Scenario XIV (aspect ratio interpolation), after adding new features, for U_x (by RFR). Training data (above) and testing data (below).

المنارة للاستشارات

www.manaraa.com

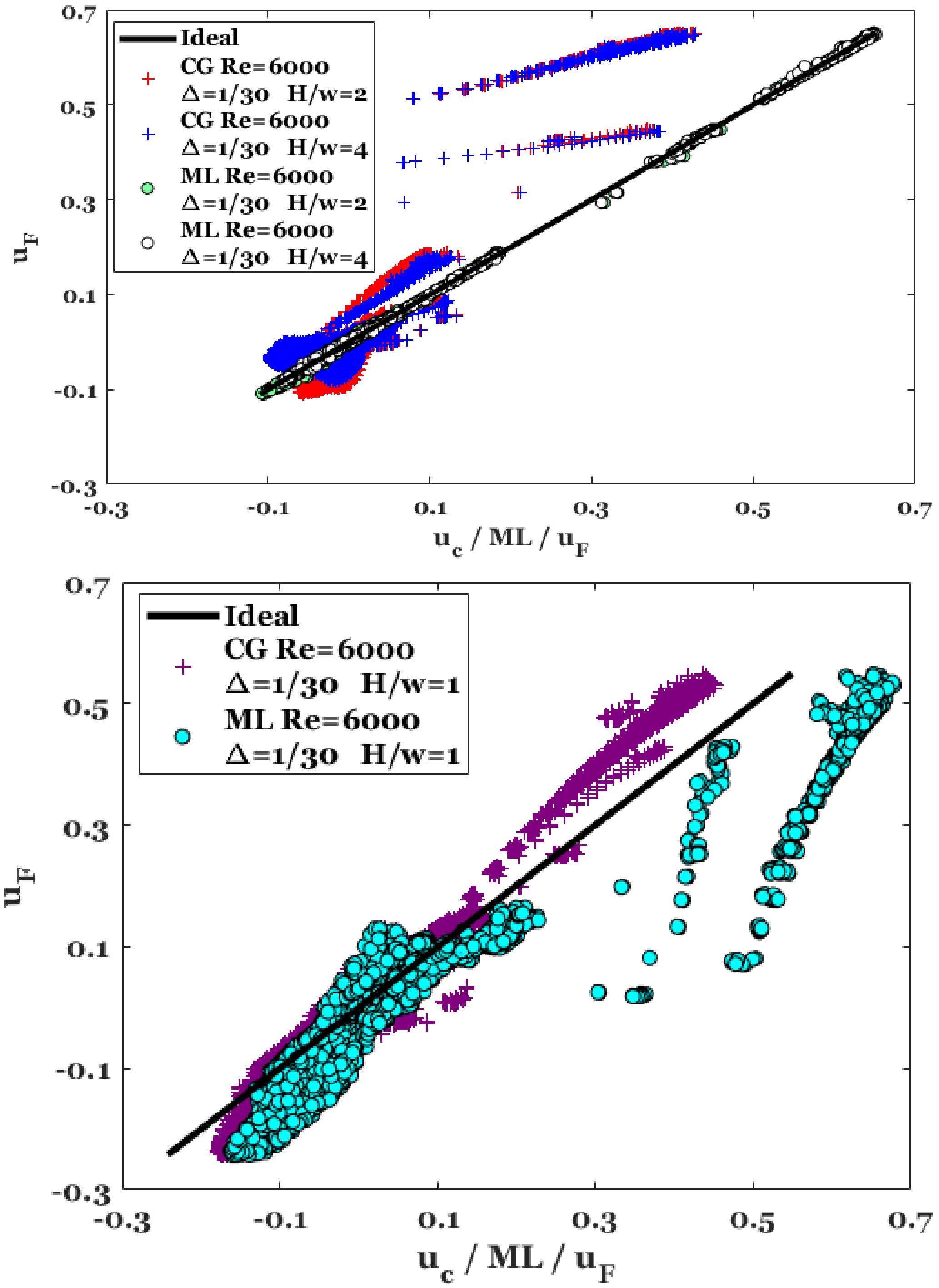


Figure 41. Scenario XV (aspect ratio extrapolation to a lower ratio), for U_x (by RFR). Training data (above) and testing data (below).

www.manaraa.com

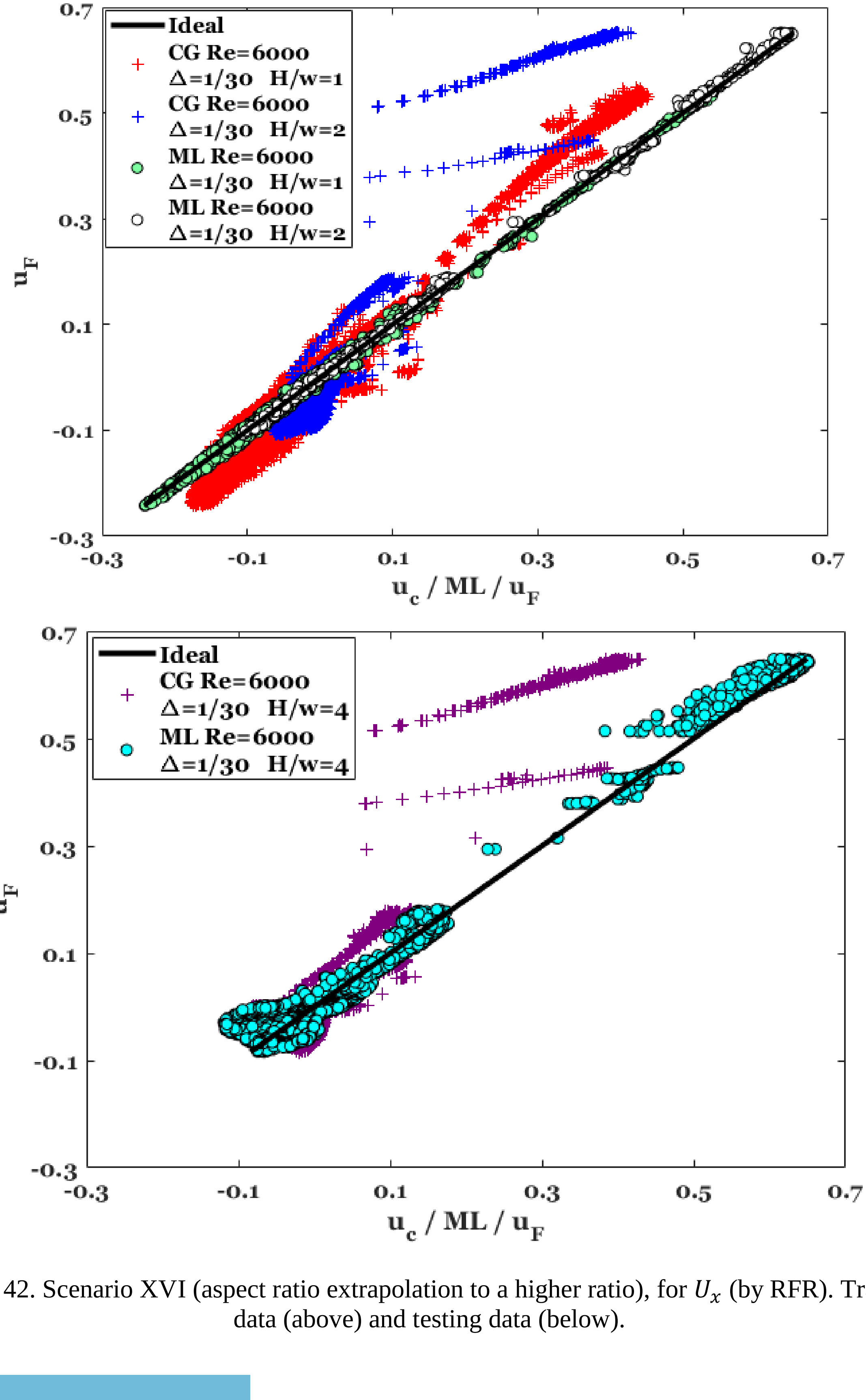


Figure 42. Scenario XVI (aspect ratio extrapolation to a higher ratio), for U_x (by RFR). Training data (above) and testing data (below).

المنارة للاستشارات
www.manaraa.com

To show the similarity or difference between the flow patterns in the training and testing data, flow patterns corresponding to different aspect ratios are presented in Figure 43. The flow patterns are significantly different (different number of vortices and different vortex location).

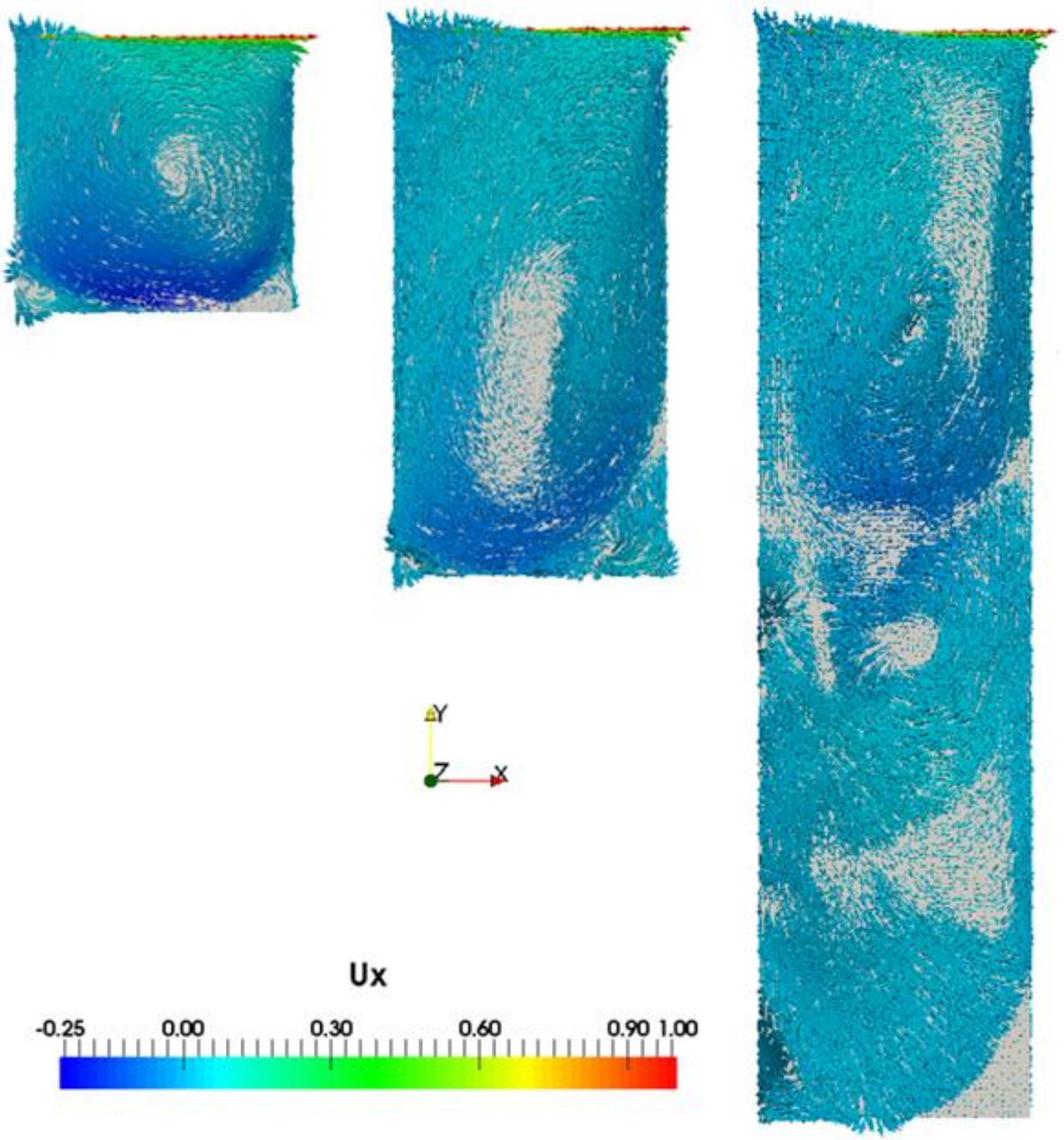


Figure 43. Velocity profile in *x*-direction inside the lid-driven cavity flow for different aspect ratios at $Re = 6000$.

المنارة للاستشارات
www.manaraa.com

In scenarios XVII and XVIII, aspect ratio extrapolation is performed at a higher Reynolds number ($Re = 12000$). In both scenarios, RFR is making good predictions over the coarse-grid simulations (see Figure 44, Figure 45 and Figure 46).

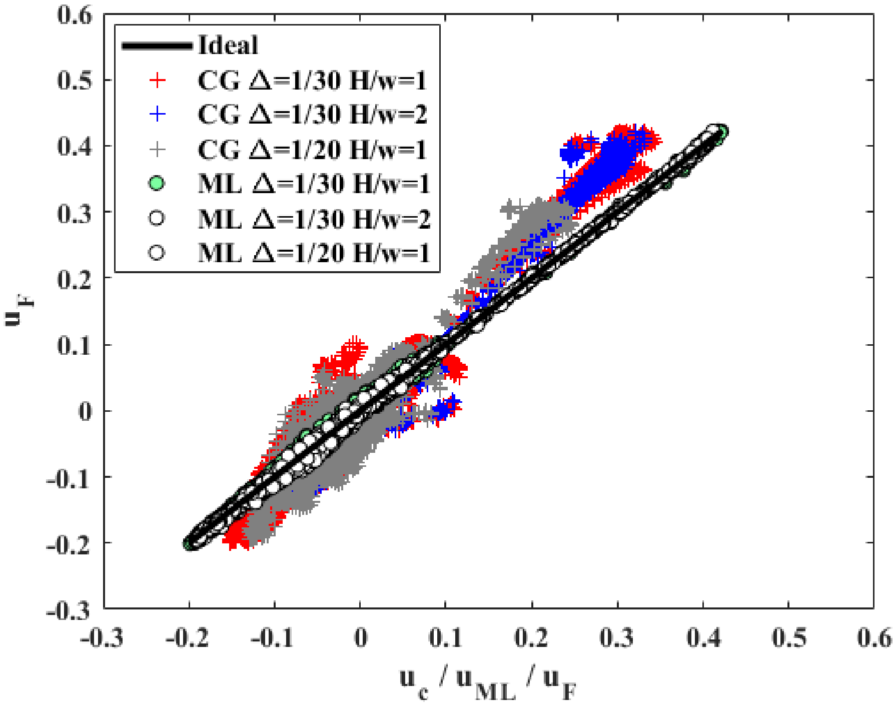


Figure 44. Training data from different aspect ratios and grid sizes for scenario XVII and XVIII for U_x (by RFR) at $Re = 12000$.

المنارة للاستشارات
www.manaraa.com

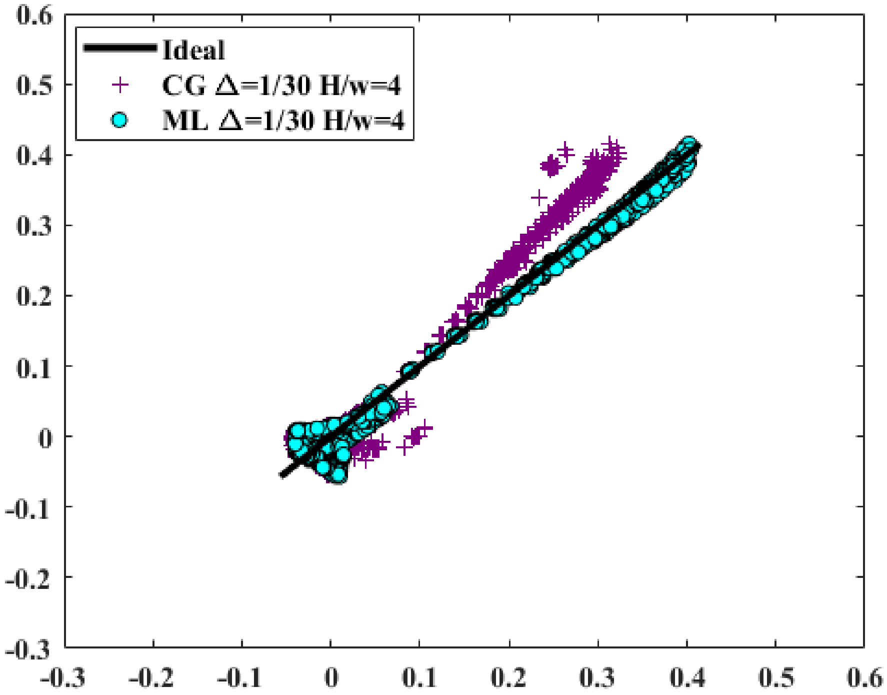


Figure 45. Testing data for scenario XVII (aspect ratio extrapolation) for U_x (by RFR) at $Re = 12000$.

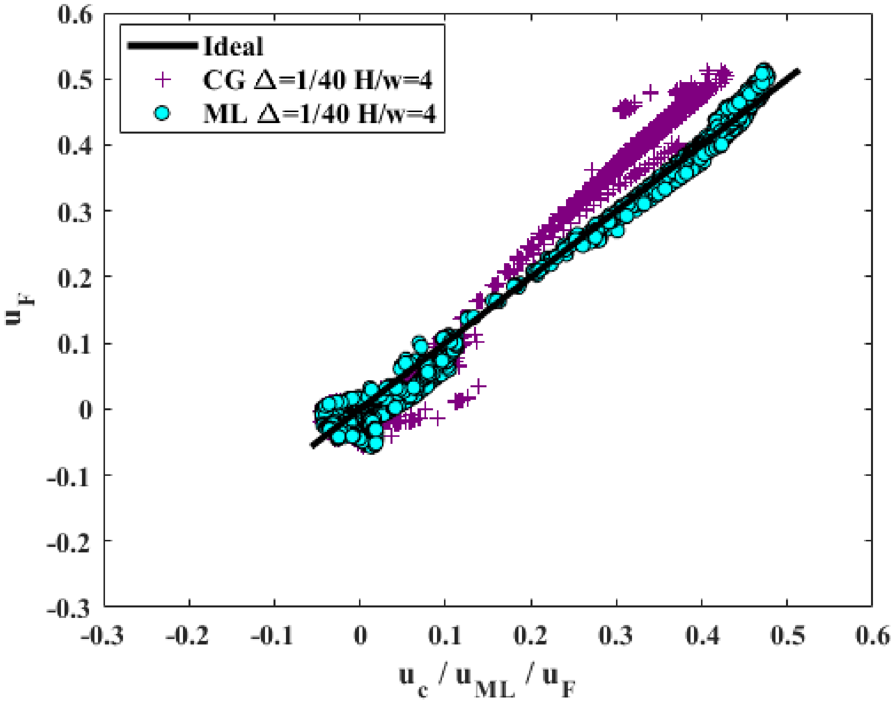


Figure 46. Testing data for scenario XVIII (aspect ratio and grid size extrapolation) for U_x (by RFR) at $Re = 12000$.

www.manaraa.com

Again, to show the similarity or difference between the flow patterns in the training and testing data, flow patterns corresponding to different aspect ratios are presented in Figure 47.

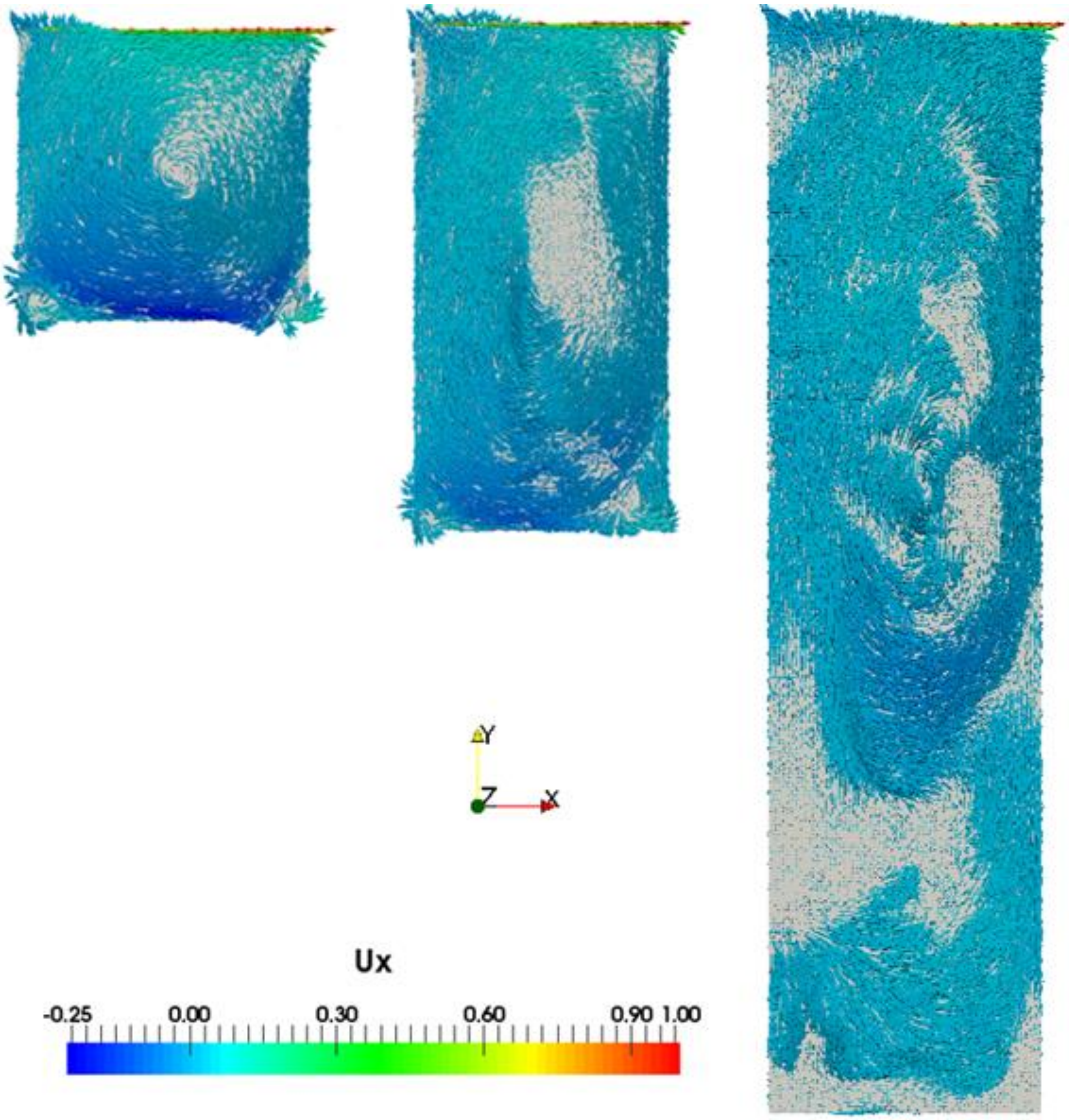


Figure 47. Velocity profile in x direction inside the lid-driven cavity flow for different aspect ratios at $Re\ = 12000$.

المنارة للاستشارات
www.manaraa.com

6.4. Chapter Summary

Initially, the fine-grid simulations were validated against experimental data to ensure that the resolution of the fine-grid simulation of the lid-driven cavity flow is sufficiently fine. The coarse grids that were utilized to produce CG-CFD results are very coarse (the number of cells per grid ranges between 8,000 to 64,000 cells) compared to 2 million cells per the validated fine-grid solution. Coarse grids are uniform while an additional refinement was inserted near the wall, when performing the fine-grid simulation. Hence, there is a significant difference between the fine-grid and the coarse-grid solution (grid-induced error).

The proposed statistical model was *trained* and *tested* in different scenarios. These scenarios include interpolation and extrapolation in terms of global Reynolds number or mesh size or aspect ratio. Typically, extrapolation scenarios are more challenging because the testing data do not lie in the range of the training data. Some scenarios were more challenging because of combining parameters together (e.g. Reynolds number and grid size extrapolation together).

The range of Reynolds numbers include flows in the turbulent and transitional regime. The range of aspect ratio starts from 1 (a cubic cavity) to 4 (a tall rectangular cavity). The proposed statistical model was reasonably capable of compensating for the grid-induced error for most studied scenarios.

The three-dimensional spatial distribution of the velocity grid-induced error was predicted. The velocity components (parallel and perpendicular to the cavity lid) were corrected given the computed grid-induced error while the third velocity component (spanwise velocity) was not corrected as it was too small (around zero) so it was very sensitive to small machine learning errors.

المنارة للاستشارات
www.manaraa.com

Data convergence study was performed (this means that adding more training data either improves the predictive capability of the model or at least does not affect it).

Machine Learning algorithms (Artificial Neural Network (ANN) and Random Forest Regression (RFR)) were investigated as regression algorithms. RFR performs better than ANN in terms of accuracy and computational expense. For the investigated cases, RFR performed even better than a three-layer ANN. The average RFR error for the testing data is around 3 times smaller than the standard deviation of the local grid induced error while the maximum RFR error is 3 to 4 times the standard deviation of the grid induced error.

المنارة للاستشارات
www.manaraa.com

7. CONCLUSIONS

The thermal-hydraulics' phenomena modeling approaches used in nuclear industry include system codes with emerging trend of utilizing CFD-scale analysis. CFD codes have the advantage of capturing the three-dimensional flow behavior but they are computationally expensive. To mitigate the computational expense, a Coarse-grid CFD (CG-CFD) approach is proposed in this work. A general framework for simulating containment thermal hydraulics with CG-CFD is presented. CG-CFD framework involves relying on coarse-grid simulations which introduce discretization error (grid-induced error). Next, the grid-induced error is captured with the data-driven surrogate model developed by Machine Learning. This data-driven model predicts the grid-induced error as a function of the inaccurate flow features computed by coarse-grid simulations. Regression among the grid-induced error and the coarse-grid flow features were performed with machine learning algorithms (one layer, multi-layer neural network, and random forest regression).

This approach was applied successfully with a three-dimensional turbulent flow inside a lid-driven cavity. The surrogate model which predicts the grid-induced error was trained with different sets of training flows and applied on testing flows (new data) which have different Reynolds number, aspect ratio, and grid size. The proposed method succeeded in predicting the three-dimensional spatial distribution of the velocity. This method optimizes using the available high-fidelity data and has a good predictive capability.

The contributions of this work can be summarized as follows:

7.1. Contributions

1. A General framework to utilize coarse grids for simulating containment thermal hydraulics

Typically, the available experimental data or high-fidelity simulation may help to understand some phenomena (separate effects) but it would be challenging to simulate the

www.manaraa.com

combined effects in a nuclear accident scenario on the containment scale. Hence, A conceptual framework is proposed to benefit from fine-grid simulation results, to correct CG-CFD results for each potential CTH single phenomenon, while the multiple phenomena scenario is simulated with coarse grids. Part of this framework is already actualized in this work (correcting the coarse-grid simulation results).

2. A new method (CG-CFD) is developed to predict the grid-induced error using machine learning

Getting a grid-independent solution is not possible in many practical scenarios. Therefore, a new method is developed to correct the local grid-induced error distribution of the flow of interest. The method has the following features: 1- Local flow features are computed for each cell in the computational domain. 2- Flow variable value in each cell in the domain is assumed to be of interest. 3. Maximum benefit of the available data is achieved given more than 30 local flow features computed for each cell in the computational domain for a variety of cases studies.

The proposed method predicts the discretization error (grid-induced error) given the results computed by the grids which are away from the asymptotic convergence range. Typical discretization error estimators (like Richardson extrapolation) demands to have numerical results that lie in the asymptotic range. Reaching the asymptotic range is computationally expensive for three-dimensional Navier–Stokes equations.

3. Exploration of the proposed method capability to extrapolate beyond the training limits.

Using the proposed method, a surrogate model is trained using a smaller geometry (e.g. a cubic cavity) to predict the flow variable distribution for a bigger geometry (higher aspect ratio). It can also predict the flow variable distribution for a different Reynolds number (different viscosity corresponding to a different fluid).

المنارة للاستشارات
www.manaraa.com

4. Evaluation of the capability of machine learning algorithms to predict the grid-induced error.

Shallow (one-layer) neural network, deep (multi-layer) network and random forest regression algorithms were evaluated as regression algorithms. For the investigated cases, random forest performed better in terms of accuracy and computational expense.

7.2. Future Work

The CG-CFD approach presented in this work with grid-induced error, predicted by the data-driven surrogate model, is still far from practical application in the field of CTH. To cover the gap between the presented approach and CTH applications, future work is needed in three directions: (1) Completeness of the CG-CFD CTH framework (depicted in Figure 1), (2) Studying cases which are more relevant to CTH phenomena, (3) Assessing the confidence in the surrogate model prediction for different cases.

1. Completeness of CG-CFD CTH framework

CG-CFD use in CTH as depicted in Figure 1 includes 8 steps. The third step (that was highlighted in yellow) was studied in this work. The other steps are still challenging as following:

- Coarse-grid simulation of the containment scenario (the fourth step in Figure 1): The coarse grid can be a relative term because a grid that is coarse for one of the CTH phenomena may not be as coarse for another phenomenon. Additionally, because of the interplay between different phenomena, there is no guarantee if the grid-induced error model based on a single phenomenon will behave in the same way if the phenomenon is a part of a complicated scenario.
- Pattern recognition (in Figure 1) is needed for every single phenomenon given the coarse-grid flow variable profile. There is still an open question: to what extent is it possible to

المنارة للاستشارات
www.manaraa.com

coarsen the grid and get results which still identify the physics/phenomena although the results are inaccurate?

2. Studying cases which are more relevant to CTH phenomena

CTH phenomena typically involve the following physics: 1- turbulent flow, 2- multidimensional behavior, 3- non- isothermal (temperature dependent) flows, 4- multi-phase flow, 5- multiphysics, 6- transient flow. The flow in a lid-driven cavity is turbulent and three-dimensional flow but it still lacks the other properties.

To deal with temperature dependent flows, additional features (e.g. temperature gradients, Rayleigh number, etc.) will be required.

For the transient flows, time derivatives' terms may be involved as flow features. Alternatively, we may hypothesize that the local time scale is much shorter than the characteristic time scale of the transient. Hence, the current CG-CFD method that was tested on a quasi-steady state flow may be applied to transient problems.

3. Assessing the confidence in the surrogate model prediction

In this work, 18 scenarios were studied which represent different possibilities about which flows are available for training a surrogate model and which flow is targeted by the surrogate model. It may be expected that the capability of the surrogate model is related to the closeness between the training flows and the testing flows. There is no metric for this closeness. This metric should serve as a priori assessment for the prediction confidence.

Another challenge is the error boundedness: surrogate models may give good predictions, but the predictions are never perfect. Estimating the error bounds when using a specific statistical model is essential to identify the safety margin related to the variable of interest.

www.manaraa.com

8. REFERENCES

[1] Sehgal, B.R., (2011) "Nuclear Safety in Light Water Reactors", 1 ed. Academic Press, Cambridge, MA.

[2] Bestion, D., (2017). in: F. D'Auria (Ed.), "Thermal-Hydraulics of Water Cooled Nuclear Reactors", 1 ed., Woodhead Publishing, Sawston, United Kingdom, pp. 639.

[3] Fletcher, C., R. Schultz, (1995) "RELAP5/MOD3 Code Manual Volume V: User's Guidelines", 1 ed. Idaho National Engineering Laboratory, Idaho Falls, ID.

[4] Gauntt, R., R. Cole, S. Hodge, S. Rodriguez, R. Sanders, R. Smith, D. Stuart, R. Summers, M. Young, (2000) "MELCOR Computer Code Manuals", 1 ed. Sandia National Laboratories, Albuquerque, NM.

[5] George, T.L., (2001) "GOTHIC Containment Analysis Package Installation and Operations Manual ", 1 ed. Numerical Applications, Inc, Richland, WA.

[6] Bury, T., (2013). "Coupling of CFD and lumped parameter codes for thermal-hydraulic simulations of reactor containment" *Computer Assisted Methods in Engineering and Science*, 20(3), 195.

[7] Austregesilo Filho, H., (1997). *"A survey of new trends in nuclear thermal-hydraulics" Meeting on reactor physics and thermal hydraulics*, 28, 433, Brazil.

[8] A. Fluent, "ANSYS FLUENT 12.0 User's guide" 6, (2009).

[9] CD-adapco, (2009) "Star-CCM+ User Guide", 4.02 ed.

[10] OpenFOAM, R., (2011). "The open source CFD toolbox" *The OpenFOAM Foundation homepage: http://openfoam.com*.

[11] Hassan, Y., (2017). in: F. D'Auria (Ed.), "Thermal-Hydraulics of Water Cooled Nuclear Reactors" , 1 ed., Woodhead Publishing, Sawston, United Kingdom, pp. 729.

[12] Pope, S.B., (2001) "Turbulent Flows", 1 ed. Cambridge University Press, Cambridge, United Kingdom.

[13] Hanna, B., (2014). "Evaluation of CFD Capability for Simulation of Energetic Flow in Light Water Reactor Containment.".

[14] Youngblood, R., V. Mousseau, D. Kelly and T. Dinh, (2010). *"Risk-Informed Safety Margin Characterization (RISMC): Integrated Treatment of Aleatory and Epistemic Uncertainty in Safety Analysis" The 8th International Topical Meeting on Nuclear Thermal-Hydraulics, Operation and Safety (NUTHOS-8)*, Shanghai, China.

المنارة للاستشارات
www.manaraa.com

[15] Viellieber, M. and A. Class, (2015). "Coarse-Grid-CFD for the Thermal Hydraulic Investigation of Rod-Bundles" *PAMM · Proc. Appl. Math. Mech.*, 15(1), 497-498.

[16] Smagorinsky, J., (1963). "General circulation experiments with the primitive equations: I. The basic experiment" *Mon. Weather Rev.*, 91(3), 99-164.

[17] Grinstein, F.F., L.G. Margolin, W.J. Rider, (2007) "Implicit Large Eddy Simulation: Computing Turbulent Fluid Dynamics", 1 ed. Cambridge university press, Cambridge, United Kingdom.

[18] Zhou, Y. and B. Thornber, (2016). "A Comparison of Three Approaches to Compute the Effective Reynolds Number of the Implicit Large-Eddy Simulations" *Journal of Fluids Engineering*, 138(7), 070905.

[19] Launder, B.E. and D.B. Spalding, (1974). "The numerical computation of turbulent flows" *Comput. Methods Appl. Mech. Eng.*, 3(2), 269-289.

[20] Spalart, P.R. and S.R. Allmaras, (1992). "A one-equation turbulence model for aerodynamic flows" *AIAA Paper*, 92, 0439.

[21] Lucy, L.B., (1977). "A numerical approach to the testing of the fission hypothesis" *Astronomical Journal*, 82(1), 1013-1024.

[22] King, R.N., P.E. Hamlington and W.J. Dahm, (2015). *"Autonomic Subgrid-Scale Closure for Large Eddy Simulations" 53rd AIAA Aerospace Sciences Meeting*, 1285, Kissimmee, Florida.

[23] King, R.N., P.E. Hamlington and W.J. Dahm, (2016). "Autonomic closure for turbulence simulations" *Phys. Rev.*, 93(3), 031301.

[24] Ma, M., J. Lu and G. Tryggvason, (2015). "Using statistical learning to close two-fluid multiphase flow equations for a simple bubbly system" *Physics of Fluids*, 27(9), 092101.

[25] H. Demuth and M. Beale, "Neural network toolbox for use with MATLAB" (1998).

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

[27] Wang, J., J. Wu and H. Xiao, (2017). "Physics-informed machine learning approach for reconstructing Reynolds stress modeling discrepancies based on DNS data" *Phys. Rev. Fluids*, 2(3), 034603.

[28] J. Wang, J. Wu, J. Ling, G. Iaccarino and H. Xiao, "A Comprehensive Physics-Informed Machine Learning Framework for Predictive Turbulence Modeling" "Preprint submitted to Elsevier," (2017).

المنارة للاستشارات
www.manaraa.com

[29] Barone, M.F., J.A. Fike, K.S. Chowdhary, W.L. Davis IV, J. Ling and S. Martin, (2017). *"Machine Learning Models of Errors in Large Eddy Simulation Predictions of Surface Pressure Fluctuations" 47th AIAA Fluid Dynamics Conference*, 3979, Denver, Colorado.

[30] LeCun, Y., Y. Bengio and G. Hinton, (2015). "Deep learning" *Nature*, 521(7553), 436-444.

[31] Rasmussen, C.E., C.K. Williams, (2006) "Gaussian Processes for Machine Learning", 1 ed. MIT press, Cambridge, MA.

[32] Breiman, L., (2001). "Random forests" *Mach. Learning*, 45(1), 5-32.

[33] J. Wu, J. Wang, H. Xiao and J. Ling, "Physics-informed machine learning for predictive turbulence modeling: A priori assessment of prediction confidence" "submitted to Flow Turbulence Combust.," (2016).

[34] De Maesschalck, R., D. Jouan-Rimbaud and D.L. Massart, (2000). "The mahalanobis distance" *Chemometrics Intellig. Lab. Syst.*, 50(1), 1-18.

[35] Silverman, B.W., (1986) "Density Estimation for Statistics and Data Analysis", 1 ed. Chapman and Hall, United Kingdom.

[36] Ling, J., A. Kurzawski and J. Templeton, (2016). "Reynolds averaged turbulence modelling using deep neural networks with embedded invariance" *J. Fluid Mech.*, 807, 155-166.

[37] Jeong, S., B. Solenthaler, M. Pollefeys and M. Gross, (2015). "Data-driven fluid simulations using regression forests" *ACM Transactions on Graphics (TOG)*, 34(6), 199.

[38] Abdo, M., (2016). "Multi-Level Reduced Order Modeling Equipped with Probabilistic Error Bounds".

[39] Chapman, D.R., (1979). "Computational aerodynamics development and outlook" *AIAA J.*, 17(12), 1293-1313.

[40] Lilly, D.K., (1992). "A proposed modification of the Germano subgrid-scale closure method" *Physics of Fluids A: Fluid Dynamics*, 4(3), 633-635.

[41] Van Driest, E.R., (2012). "On turbulent flow near a wall" *Journal of Communication and Computer*, 9, 1104.

[42] Moin, P. and J. Kim, (1982). "Numerical investigation of turbulent channel flow" *J. Fluid Mech.*, 118, 341-377.

[43] Davidson, L., (2011) "Fluid Mechanics, Turbulent Flow and Turbulence Modeling" Chalmers University of Technology, Goteborg, Sweden.

المنارة للاستشارات

www.manaraa.com

[44] Baldwin, B. and Lomax, (1978). *"Thin-layer approximation and algebraic model for separated turbulentflows" 16th Aerospace Sciences Meeting,* 257, Huntsville,AL.

[45] Wilcox, D.C., (2002) "Turbulence Modeling for CFD" DCW Industries.

[46] Menter, F. and T. Esch, (2001). *"Elements of industrial heat transfer predictions" 16th Brazilian Congress of Mechanical Engineering (COBEM),* 109, Uberlandia, Brazil.

[47] Gaitonde, U., D. Laurence and A. Revell, (2008). "Quality Criteria for Large Eddy Simulation" University of Manchester.

[48] Celik, I., Z. Cehreli and I. Yavuz, (2005). "Index of resolution quality for large eddy simulations" *J. Fluids Eng*, 127(5), 949-958.

[49] Georgiadis, N.J., D.P. Rizzetta and C. Fureby, (2010). "Large-eddy simulation: current capabilities, recommended practices, and future research" *AIAA Journal,* 48(8), 1772-1784.

[50] Eça, L., M. Hoekstra, A. Hay and D. Pelletier, (2007). "On the construction of manufactured solutions for one and two-equation eddy-viscosity models" *Int. J. Numer. Methods Fluids*, 54(2), 119-154.

[51] Blazek, J., (2015). in: J. Blazek (Ed.), "Computational Fluid Dynamics: Principles and Applications" , 3 ed., Butterworth-Heinemann, Oxford, United Kingdom, pp. 29-72.

[52] Donea, J., A. Huerta, (2005) "Finite Element Methods for Flow Problems", 1 ed. John Wiley & Sons, Hoboken, NJ.

[53] Greenshields, C.J., (2015) "OpenFOAM the Open Source CFD Toolbox Programmer’s Guide", 3.0.1 ed. CFD Direct Ltd, United Kingdom.

[54] Issa, R.I., (1986). "Solution of the implicitly discretised fluid flow equations by operator-splitting" *Journal of Computational Physics*, 62(1), 40-65.

[55] Roy, C., (2010). *"Review of discretization error estimators in scientific computing" 48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, 126, Orlando, Florid.

[56] Richardson, L.F., (1911). "IX. The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam" *Philosophical transactions of the Royal Society A*, 210(459-470), 307-357.

[57] Solom, M. and K.V. Kirkland, (2016). "Experimental investigation of BWR Suppression Pool stratification during RCIC system operation" *Nucl. Eng. Des.*, 310(1), 564-569.

www.manaraa.com

[58] Allelein, H., K. Fischer, J. Vendel, J. Malet, E. Studer, S. Schwarz, M. Houkema, H. Paillere and A. Bentaib, (2007). "International standard problem ISP-47 on containment thermal hydraulics" *NEA News*, 25(2), 28.

[59] Grötzbach, G. and M. Wörner, (1999). "Direct numerical and large eddy simulations in nuclear applications" *Int J Heat Fluid Flow*, 20(3), 222-240.

[60] Tanskanen, V., (2012). "CFD modelling of direct contact condensation in suppression pools by applying condensation models of separated flow" *Acta Universitatis Lappeenrantaensis*, 14(472), 1-184.

[61] Alpaydin, E., (2014) "Introduction to Machine Learning", 3 ed. MIT press, Cambridge, MA.

[62] Lloyd, S., (1982). "Least squares quantization in PCM" *IEEE Trans. Inf. Theory*, 28(2), 129-137.

[63] Pearson, K., (1901). "LIII. On lines and planes of closest fit to systems of points in space" *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11), 559-572.

[64] Lang, Y., A. Malacina, L.T. Biegler, S. Munteanu, J.I. Madsen and S.E. Zitney, (2009). "Reduced order model based on principal component analysis for process simulation and optimization" *Energy Fuels*, 23(3), 1695-1706.

[65] Suykens, J.A., T. Van Gestel, J. De Brabanter, (2002) "Least Squares Support Vector Machines", 1 ed. World Scientific, Hackensack, NJ.

[66] Wang, H. and L. Zhang, (2009). "Identification of two-phase flow regimes based on support vector machine and electrical capacitance tomography" *Measurement Science and Technology*, 20(11), 114007.

[67] Hagan, M.T. and M.B. Menhaj, (1994). "Training feedforward networks with the Marquardt algorithm" *IEEE Trans. Neural Networks*, 5(6), 989-993.

[68] Marquardt, D.W., (1963). "An algorithm for least-squares estimation of nonlinear parameters" *Journal of the society for Industrial and Applied Mathematics*, 11(2), 431-441.

[69] Breiman, L., J. Friedman, C.J. Stone, R.A. Olshen, (1984) "Classification and Regression Trees", 1 ed. CRC press, Boca Raton, FL.

[70] Hastie, T., R. Tibshirani, J. Friedman, (2009). in: T. Hastie, R. Tibshirani, J. Friedman (Eds.), "The Elements of Statistical Learning" , 2 ed., Springer-Verlag, New York, pp. 1.

[71] Efron, B., R. Tibshirani, (1993) "An Introduction to the Bootstrap Boca Raton", 1 ed. CRC press, Boca Raton, FL.

المنارة للاستشارات

www.manaraa.com

[72] Baker, T.J., (1997). "Mesh adaptation strategies for problems in fluid dynamics" *Finite Elements Anal. Des.*, 25(3-4), 243-273.

[73] Venditti, D.A. and D.L. Darmofal, (2000). "Adjoint error estimation and grid adaptation for functional outputs: Application to quasi-one-dimensional flow" *Journal of Computational Physics*, 164(1), 204-227.

[74] Nadal, E., A. Leygue, F. Chinesta, M. Beringhier, J. Ródenas and F. Fuenmayor, (2015). "A separated representation of an error indicator for the mesh refinement process under the proper generalized decomposition framework" *Comput. Mech.*, 55(2), 251-266.

[75] Ghia, U., K.N. Ghia and C. Shin, (1982). "High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method" *Journal of Computational Physics*, 48(3), 387-411.

[76] Erturk, E., (2009). "Discussions on driven cavity flow" *Int. J. Numer. Methods Fluids*, 60(3), 275-294.

[77] Botella, O. and R. Peyret, (1998). "Benchmark spectral results on the lid-driven cavity flow" *Comput. Fluids*, 27(4), 421-433.

[78] Horton, B., Y. Song, J. Feaster and J. Bayandor, (2017). "Benchmarking of Computational Fluid Methodologies in Resolving Shear-Driven Flow Fields" *Journal of Fluids Engineering*, 139(11), 111402.

[79] Iwatsu, R., K. Ishii, T. Kawamura, K. Kuwahara and J.M. Hyun, (1989). "Numerical simulation of three-dimensional flow structure in a driven cavity" *Fluid Dyn. Res.*, 5(3), 173-189.

[80] Koseff, J. and R. Street, (1984). "The lid-driven cavity flow: a synthesis of qualitative and quantitative observations" *Journal of Fluids Engineering*, 106(12), 390-398.

[81] Bouffanais, R., M.O. Deville and E. Leriche, (2007). "Large-eddy simulation of the flow in a lid-driven cubical cavity" *Phys. Fluids*, 19(5), 055108.

المنارة للاستشارات

www.manaraa.com