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

Botros Hanna (NMSU), Nam Dinh (NCSU), Igor Bolotnov (NCSU), Robert Youngblood (INL)

Big Data for Nuclear Power Plants Workshop
December 2018
Outline

❖ Introduction

❖ Machine Learning

❖ CG-CFD Error Prediction Method

❖ Numerical Results

❖ Summary

❖ Future Ideas
Motivation: Why CFD?
Nuclear Engineering Applications

❖ The Fukushima accident (2011) has drawn greater attention to the need to manage risk at nuclear plants.

❖ Nuclear reactor safety analysis requires analysis of a broad range of accident scenarios.

❖ The major safety defense barrier against nuclear fission products release is the containment.

❖ Modeling and simulation are essential to gain insights and identify sensitive parameters in any Containment Thermal Hydraulics (CTH) phenomena.

❖ CFD approach has an advantage over traditional physical modeling, because of its capability to provide detailed information about flow field.
Motivation: Why CG-CFD?

- The limitation of CFD is the high computational cost:
  - Highly turbulent flow
  - Multi-phase flows
  - Large domain and long transients.
  - The need for sensitivity analysis.

- Turbulence modeling hierarchy
  - Direct Numerical Simulation (DNS): No Modeling
  - Large Eddy Simulation (LES): Modeling of Small Scales
  - Reynolds Averaged Navier Stokes Equations (RANS): Modeling all scales

- High Computational expense of CFD is partly attributed to the need to grid-independent solution.
- In this work, CG-CFD simulations are performed while grid-induced error is predicted/reduced via machine learning.
Motivation: Why CG-CFD?

Example

❖ Simulating 10 sec of high-pressure steam blowdown from reactor cooling system and containment convective mixing.

❖ ESBWR containment actual geometry (active volume ≈ 7000 m³).

❖ Needs 1 week / 128 processors (RANS).

❖ One Million cells.

Even RANS simulations can be computationally expensive

Computational domain representing ESBWR containment design

Motivation: Why Machine Learning (ML)?

✓ **Available big data**: High-fidelity CFD simulation, including “first-principle” Direct Numerical Simulation (DNS) and advanced experiments produce an unprecedented amount of 4-D. This ‘big data’ are not usable and practically not used in the current research and analysis framework.

✓ **Lack of adaptability**: LES and RANS turbulence models perform well only for specific flow conditions while data-driven models could adapt via data assimilation.

✓ **Data-driven:**
  - Traditionally (i) data analysis → (ii) mechanistic model / correlation development → (iii) validation against data → (iv) model (compact form of data)
  - Using machine learning: (i) data analysis → (iv) relevant data is used in simulation. The more data become available, the more accurate the simulation will be. There is no data wasted in this framework
The Scope of This Work

- **The objective of this work:**
  - Investigating the feasibility of obtaining a correction for CG-CFD simulation results using machine learning algorithms.

- **Proposed CG-CFD Approach vs. CFD**

  - **In CFD:**
    - Grid-independent solution required.
    - New simulation is for each new flow problem (even if it is slightly different from old cases).

  - **In the Proposed CG-CFD Approach**
    - NS equations (no turbulence modeling) are solved with coarse grids (non-accurate results) and sufficiently grids to train a surrogate model to compute grid-induced error.
Machine Learning:
Random Forest Regression (RFR)

❖ RFR is a group of regression trees:

❖ Regression tree predicts responses to inputs from the root node down to a leaf node (the response $Y$).

![Diagram of a decision tree with nodes labeled 0, 1, 2, 3, 4, 5, 6 and conditions for splitting on $x_1$, $x_2$, and $x_3$. The leaves show values for $x_1$, $x_2$, and $x_3$.]
Machine Learning:
Random Forest Regression (RFR)

❖ RFR is a group of regression trees:

❖ For each input, all the possible binary splits are examined. The split is selected to minimize Mean Squared Error (MSE)

\[
MSE = \frac{1}{N} \sum_{k=1}^{N} (T_k - y_k)^2
\]

\(T_k\): Target
\(y_k\): output
Machine Learning: Random Forest Regression (RFR)

❖ Tree bagging: (RFR)

❖ Different sample of the training data is given to each regression tree (bootstrap sampling) → Each tree makes a different prediction.

❖ The prediction of the whole tree bagger is the average of all the trees’ predictions.
CG-CFD Error Prediction Method
(Problem statement)

Flow pattern characteristic number, $\psi$

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

Computational grid spacing, $\Delta$

$\varphi$: Flow variable (e.g. velocity), $\varphi = \varphi(x, y, z, t)$.

$\Psi$: Flow pattern characteristic number: Reynolds number, Rayleigh number, ...

$\Delta$: Grid spacing.

Database

Inform

Inform

Predict

Inform

Inform

Computeationally affordable coarse-grid simulations at various values of $\psi$ and $\Delta$:
$\varphi_{\Delta_1}(\psi_2)$,
$\varphi_{\Delta_2}(\psi_3)$,
$\varphi_{\Delta_1}(\psi_1)$,
......
CG-CFD Error Prediction Method (Hypothesis)

❖ The training flows (used to train a CG-CFD error prediction ML surrogate model) and the “testing flows” (used to test the model) have similar physics.

❖ In order to correct the grid-induced error over the whole domain (for the training flows), high-fidelity data over the whole domain are needed.

❖ The grid-induced error is a function of the coarse grid inaccurate flow features:

$$\varepsilon_\Delta = F(y(\varphi_\Delta))$$

❖ We are interested in all the data through the whole domain (flow variable value at each grid cell): thousand of data points are used for training machine learning model and thousands of data are expected (not just a linear velocity profile).
CG-CFD Error Prediction Method
(Proposed Framework)

1. Perform high fidelity simulation (with a fine grid).

2. Perform low fidelity simulation (solve the same equation on a coarse grid).

3. Map the results to a coarse grid.

4. Compute the grid induced error.

5. Compute the flow features based on coarse grid simulations.

6. Normalize the features and the grid induced error.

7. Construct a data driven model using machine learning algorithms, $\bar{e}_\Delta(\phi) = F(\bar{X}(\phi_\Delta))$
CG-CFD Error Prediction Method (Proposed Framework)

Construct a data driven model using machine learning algorithms, \( \hat{\varepsilon}_\Delta(\varphi) = F(\hat{X}(\varphi_\Delta)) \)

B- Testing flows

1. \( \hat{X}(\varphi^{te}_\Delta) \)
2. \( \hat{\varepsilon}_\Delta(\varphi^{te}_\Delta) \)

Normalize the features.

1. \( X(\varphi^{te}_\Delta) \)
2. \( \varepsilon_\Delta(\varphi^{te}_\Delta) \)

Reverse the normalization.

1. Compute the flow features based on coarse grid simulations.
2. Compute the correct variable of interest, \( \varphi^{te}_{f\rightarrow\Delta} = \varphi^{te}_\Delta + \varepsilon_\Delta(\varphi^{te}_\Delta) \)

Perform low fidelity simulation (solve the same equation on a coarse grid).
There is no universal method to select the optimal set of flow features that characterize the flow patterns.

Flow features are selected based on insights from physics or mathematics and numerical experiments.

Assuming a smooth flow variable, \( \varphi \rightarrow \) Taylor series expansion along the \( x \)-direction in a grid, \( \Delta \)

\[
\varphi = \varphi_0 + \Delta x \left( \frac{d\varphi}{dx} \right)_{\Delta x} + \frac{(\Delta x)^2}{2} \left( \frac{d^2\varphi}{dx^2} \right)_{\Delta x} + \cdots
\]

Taylor series terms \( \left( \Delta \left( \frac{d\varphi}{dx} \right), \Delta^2 \left( \frac{d^2\varphi}{dx^2} \right) \right) \) are flow features.

- Finer grid is needed near discontinuity or steep curve of the solution.
CG-CFD Error Prediction Method

Flow Features’ Selection

❖ The derivatives \( \frac{d\varphi}{dx} \) and \( \frac{d^2\varphi}{dx^2} \) are carrying the effect of the neighboring cells

❖ Flow is characterized by numbers like Reynolds number. → local \( Re \) is proposed (as flow feature) that accounts for the viscosity and the grid size

\[
Re_\Delta = |U|\Delta/\nu
\]

Thus, proposed flow features are:

\[
X(\varphi_\Delta) = \left( Re_\Delta, \Delta x_j \frac{du_i}{dx_j} \bigg|_{\Delta x_j}, (\Delta x_j)^2 \frac{d^2u_i}{dx_j^2} \bigg|_{\Delta x_j} \right)
\]

37 features = 9 first derivatives + 27 second derivatives + \( Re_\Delta \)
CG-CFD Error Prediction Method

Case studies

Training
Elementary flows

✓ Studied in this work

Prediction
Containment complex scenario

✓ Different $Re$
✓ Different grid size
✓ Larger geometry
❖ Different Boundary conditions
❖ Different geometry
❖ Combination of 2 or more elementary flows
❖ Unknown (The statistical model cannot predict).

✓ Turbulent
✓ Multi-dimensional
✓ Available validation data
❖ Thermal
❖ Multi-phase
Case study: 3D Turbulent flow inside a lid-driven cavity

- It is a turbulent three dimensional flow with available experimental data for validation.
- CFD software: OpenFOAM

Cubic cavity (H=1m).
Lid velocity is parallel to x axis.
\[ U_{lid} = 1 \text{ m/s} \]
Dashed axis lines → Validation
CG-CFD  Error  Prediction  Method

Case study: 3D Turbulent flow inside a lid-driven cavity

- Fine-grid simulations: 120 × 120 × 120 cells grid + boundary refinement. The length of the cells touching the wall is 0.0014 meters → 2 × 10^6 cells, with the guidance of Damián, Nigro 2010.

CG-CFD Error Prediction Method

Case Study: Validation

Grid size requirements increase with Reynolds number → validating OpenFOAM flow with $Re = 12000$ (max $Re$ in this work).
CG-CFD Error Prediction Method

Case Study: Scenarios

How similar/different are the training data and the testing data?

1. Different global Reynolds number (different viscosity)
2. Different grid size
3. Different grid spacing in different directions.
4. Larger geometry
5. Different $Re$ and grid size combined
6. Larger geometry and grid size combined

✓ Different flow variables of interest: $U_x, U_y, U_z$

✓ Interpolation
✓ Extrapolation (more challenging)
Numerical Results

Some Scenarios

Reynolds number extrapolation by RFR for $U_x$

Training data (left) and testing data (right)
Numerical Results
Some Scenarios

Reynolds number and Grid size extrapolation by RFR for $U_x$

Training data (left) and testing data (right)
Numerical Results

Some Scenarios

Reynolds number and Grid size extrapolation by RFR for $U_y$

Training data (left) and testing data (right)
Numerical Results
Some Scenarios

Reynolds number and Grid size extrapolation by RFR for $U_y$
Numerical Results

Some Scenarios

Aspect ratio extrapolation

Different flow patterns for different aspect ratios.

Given data from smaller geometries to predict velocity for a bigger geometry.
Numerical Results

Some Scenarios

Aspect ratio extrapolation by RFR for $U_x$

Training data (left) and testing data (right)

- Features were added (distance to the closest wall and the lid).
- Training data are fewer than the testing data.
Summary

❖ High-resolution results from simulations/experiments produce an enormous amount of data. These “big data” are not optimally usable because, for each new scenario, a sufficiently fine grid CFD simulation needs to be performed. This approach is computationally overwhelming for many applications.

❖ In the present work, CG-CFD simulations are performed and the CG-CFD induced error is learned by a surrogate model constructed by ML.

❖ The surrogate model is trained given the available fine grid and coarse grid data. Both fine and coarse grid data were performed with the same set of conservation equations (no turbulence modeling).

❖ The coarse grid-induced local error distribution was predicted (velocity distribution corrected), given features computed from the coarse grid simulations. The function that relates the error to the features is constructed using ML technique, Random Forest Regression.
Summary

❖ The proposed method was applied successfully with a three-dimensional turbulent flow inside a lid driven cavity.

❖ The proposed approach was found to be capable of correcting the coarse grid results for new cases (having different Reynolds number, computed using different grid sizes or having different aspect ratios).

❖ The fine-grid simulations need around 1000 (CPU. Hours) compared to 8 (CPU. Hours) for the coarse-grid simulation and training the surrogate model (combined). This emphasizes the computational gain when using the proposed CG-CFD method.

❖ To our knowledge, the proposed CG-CFD method is the first approach to reduce the grid-induced error using machine learning algorithms.

❖ The method still needs further assessment in scenarios when the testing and the training fluid flows are less similar.