SAND2019-9698PE

Code Optimization via Machine Learning

Ki Tae Wolf

Org: 1555

PRESENTED BY

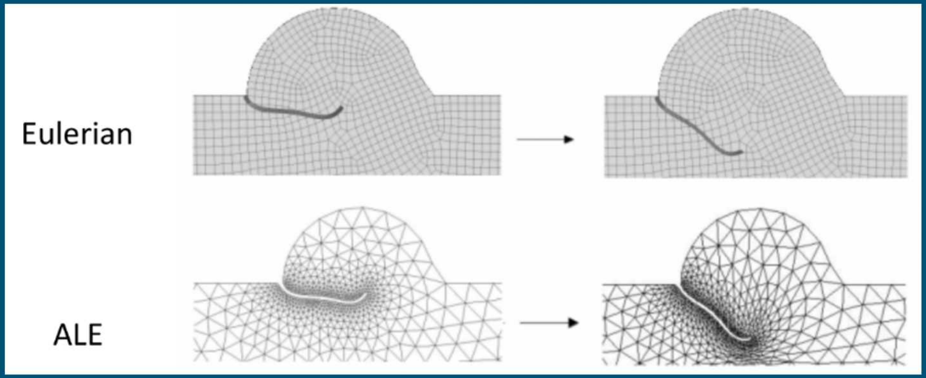
Ki Tae Wolf

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

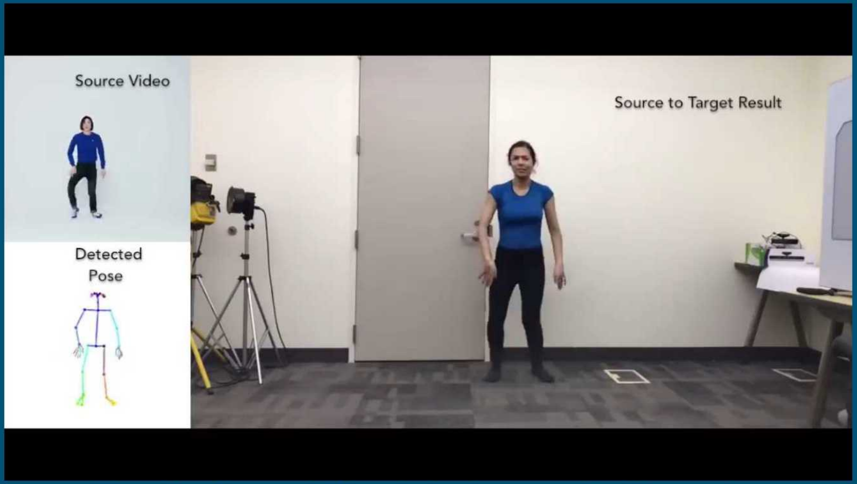
Objective/Approaches

- Objective of Project
  - Explore and optimize grid remapping using machine learning

- Approaches
  - Understanding and extraction of the remapping procedure and relevant data
  - Understanding and implementation of machine learning



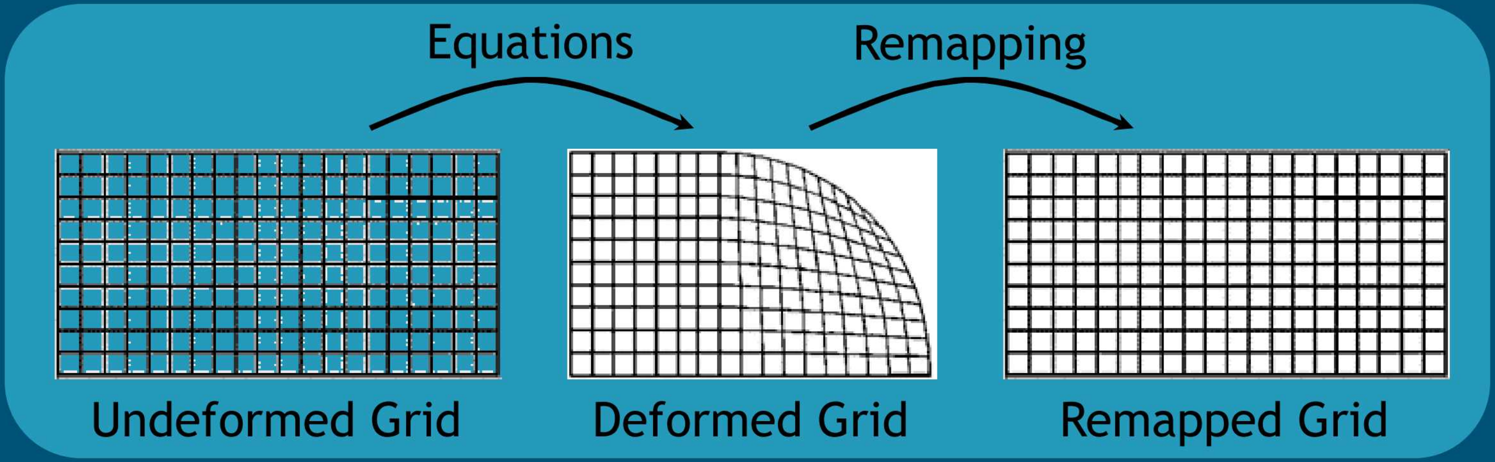
Bavo, Alessandra M., et al. "Fluid-structure interaction simulation of prosthetic aortic valves: comparison between immersed boundary and arbitrary Lagrangian-Eulerian techniques for the mesh representation." PloS one 11.4 (2016): e0154517.



Chan, Caroline, et al. "Everybody dance now." *arXiv preprint arXiv:1808.07371* (2018).

Remapping

- Remapping Algorithm
  - Lagrangian transformation into eulerian grid
  - Eulerian grid by default
  - Solving equations cause grid deformation, creating a temporary lagrangian grid
  - Remapping algorithm brings deformed grid back to eulerian grid

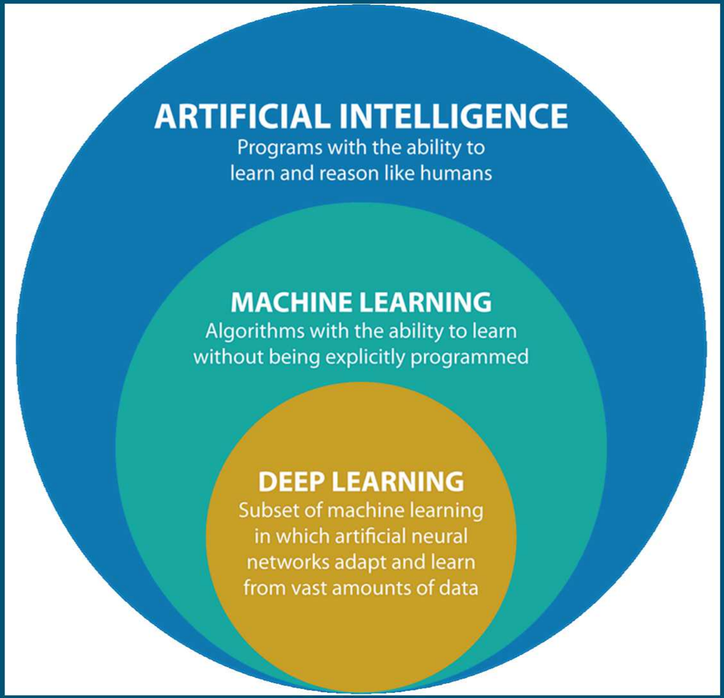


https://www.flow3d.com/resources/cfd-101/general-cfd/grid-systems/

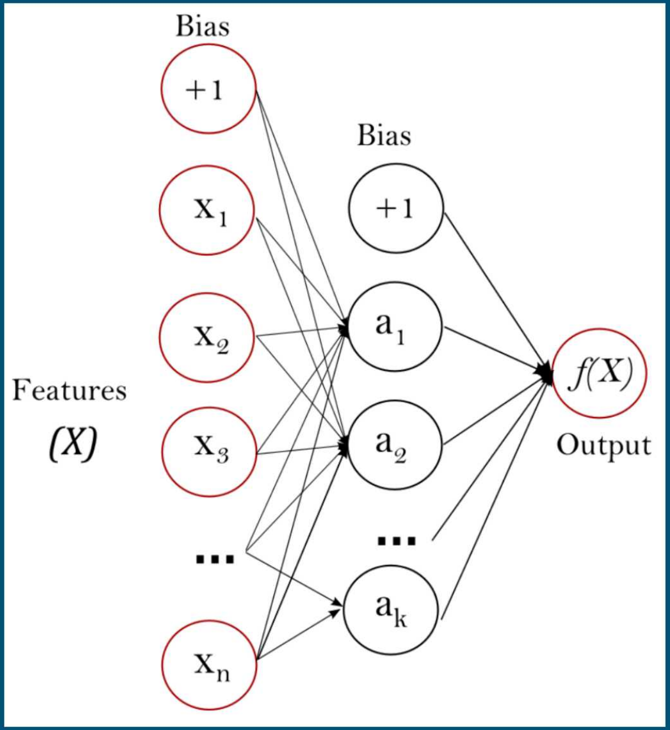
- Challenges
  - Remapping algorithm is costly
  - Remapping algorithm is hardwired

Machine Learning

- Machine Learning



https://www.argility.com/argility-ecosystem-solutions/iot/machine-learning-deep-learning/



https://scikit-learn.org/stable/modules/neural_networks_supervised.html

- Advantages
  - Can easily compartmentalize algorithms into modules
  - Easily transferrable to different architecture once a model is trained

Creation of Training Data

- Synthetic Data
  - Aimed to cover widest range as possible
  - Uniform distribution of dataset to limit bias (Latin hypercube?)

  - Focused on 2D dataset, time-dependent problem
  - Random velocity components ranging between -1 to 1 are initially assigned
  - Data extracted from remapping at first time step to minimize other sources of error that may dissipate the randomness of the initial profile
  - Displacement components due to remapping and velocity components before and after remapping are used for training
  - Differences in velocity components before and after remapping are used as predictive values

- Pre-processing
  - Filtered input parameters to ensure uniformity of predictive values
  - Normalization of input parameters to ensure parameters mostly range between 0 to 1 or -1 to 1
  - Dataset from multiple runs are combined to create dataset ranging from 1000s (lower accuracy) to 100,000s(higher accuracy)

Machine Learning Procedures

- Multilayer Perceptron (MLP)
  - Structured as: input layer, hidden layer, and output layer
  - Minimization of some loss function via manipulation of weight, learning rate, and many other parameters through each node and layer
  - Feed-forward method
  - Simple structure yet accurate

- Current Progress
  - Results from in-house machine learning code are not yet reliable
  - Help debugging the machine learning code
  - Machine learning test using simple multilayer perceptron code (available in python)

- Drawbacks
  - Large initial investment of computational resources for training
  - Sometimes difficult to verify the reliability of predictions based on trained model (bias, etc)

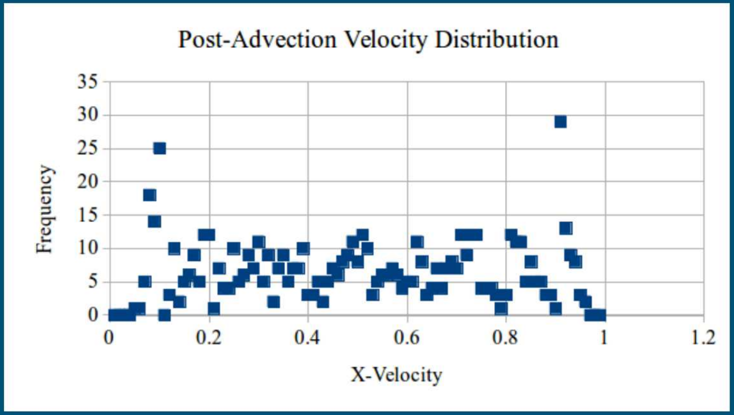
Machine Learning Procedures

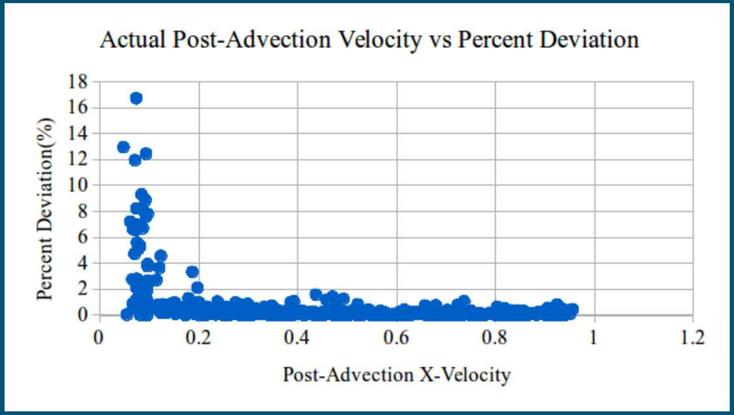
- ML code test
  - Pytorch
  - Simple structure with 3 hidden layer
  - Smooth L1 loss function (alternating loss function between absolute value difference and mean squared difference)

- Three ML code tests
  - Uniform post-remap velocity distribution with training and validation using the same dataset (highest accuracy expected)
  - Uniform post-remap velocity distribution with training and validation using different datasets
  - Un-uniform post-remap velocity distribution with training and validation using different datasets (lowest accuracy expected)

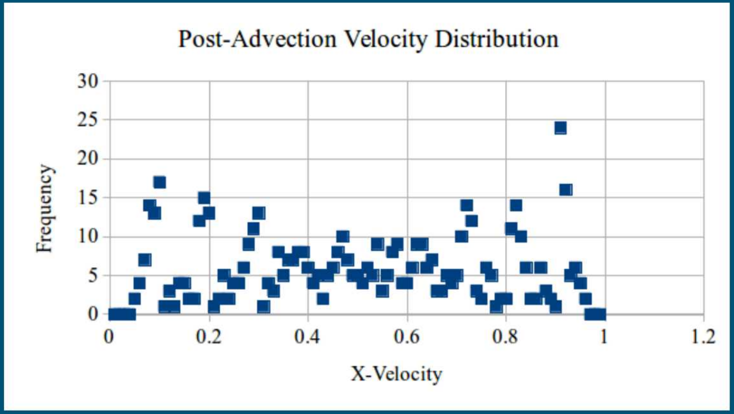
Machine Learning Procedures

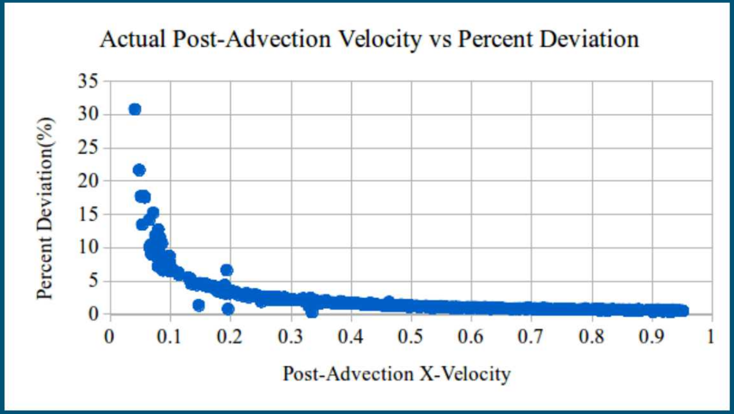
- X-Velocity Component Test
  - 1st Case: Average Deviation 0.54%





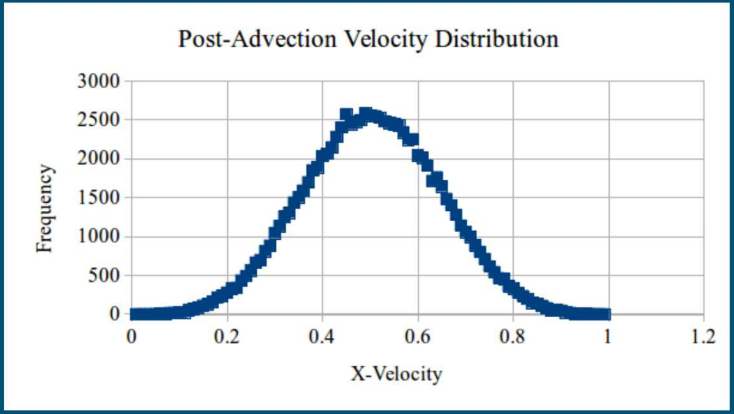
  - 2nd Case: Average Deviation 2.40%

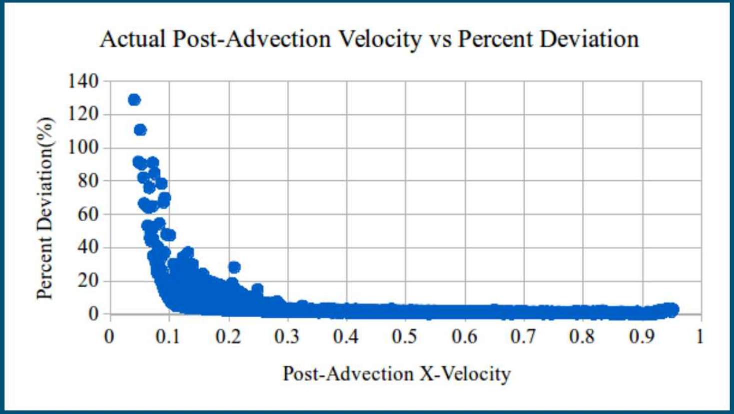




Machine Learning Procedures

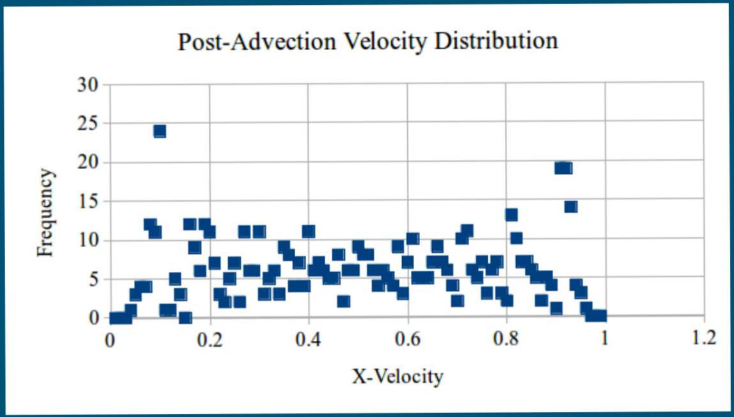
- X-Velocity Component Test
  - 3rd Case: Average Deviation 1.43%

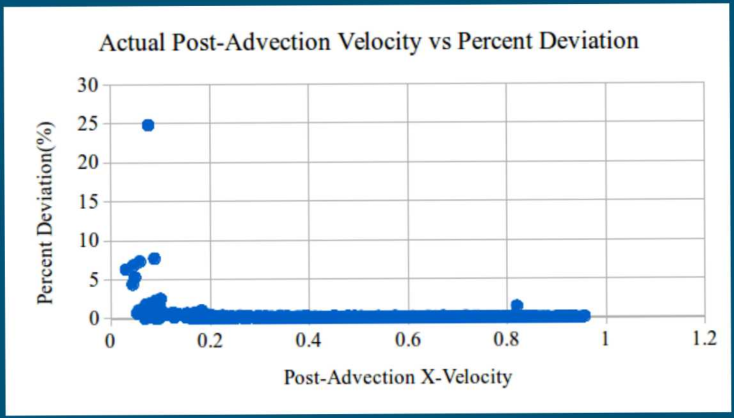




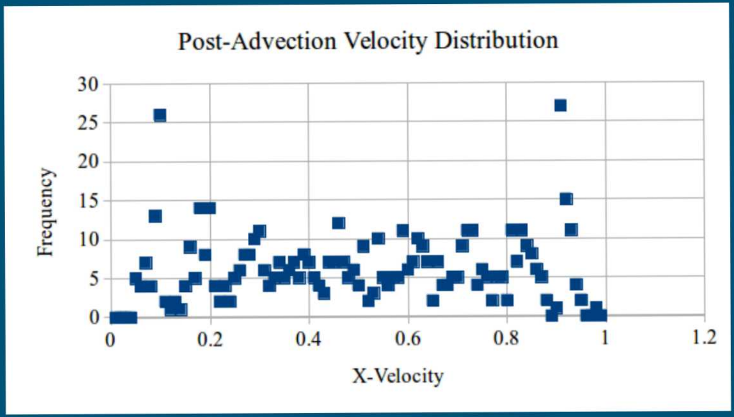
Machine Learning Procedures

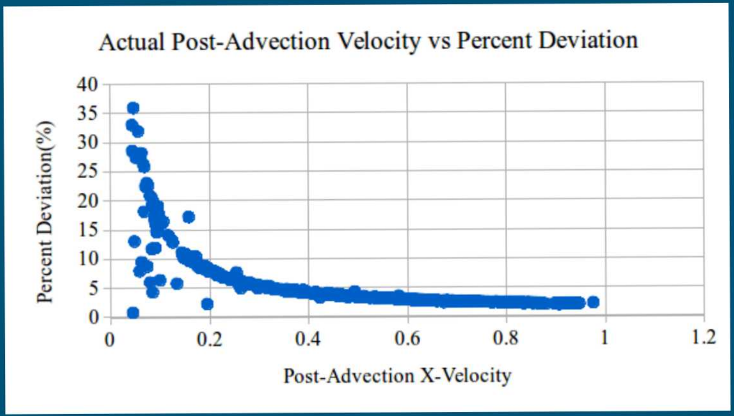
- Y-Velocity Component Test
  - 1st Case: Average Deviation 0.27%





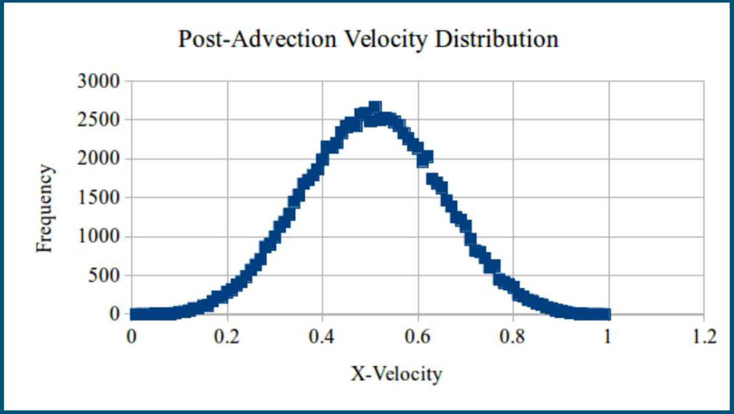
  - 2nd Case: Average Deviation 5.50%

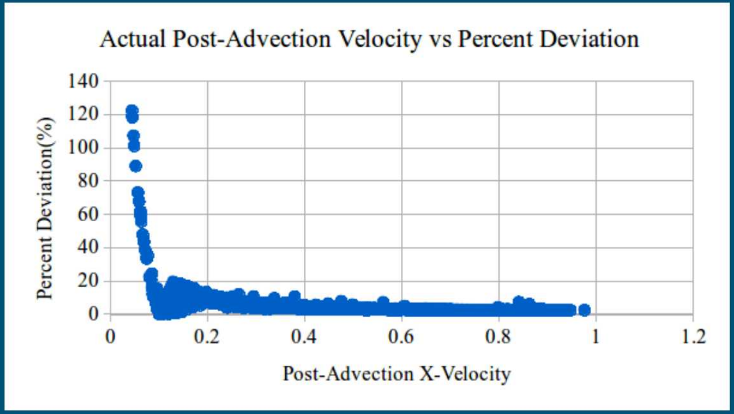




Machine Learning Procedures

- Y-Velocity Component Test
  - 3rd Case: Average Deviation 3.73%





  - Un-uniform velocity distribution had the worst accuracy in terms of maximum velocity deviation (possibly caused by bias?)
  - All cases had average velocity deviation around 1~5%
  - Importance of proper accuracy metric selection (R^2~0.99 for all)

Applications

- Complex Behavior
  - Biological applications
  - Materials behavior
- Image Processing
  - Input with complex noise and distortion
  - Restoration of images
- Many Others