# *A GENERAL DRAG MODEL FOR ASSEMBLIES OF NON-SPHERICAL PARTICLES CREATED WITH ARTIFICIAL NEURAL NETWORKS*

## SUBTITLE: PREDICTION OF THE FLOW DYNAMICS OF A PARTICLE TRANSLATING NEAR A PLANE WALL AT LOW REYNOLDS NUMBERS USING A MULTI-OUTPUT DEEP LEARNING MODEL

PI: Zhigang Feng, Ph.D.

Students: Joshua C. Conner (Presenter)

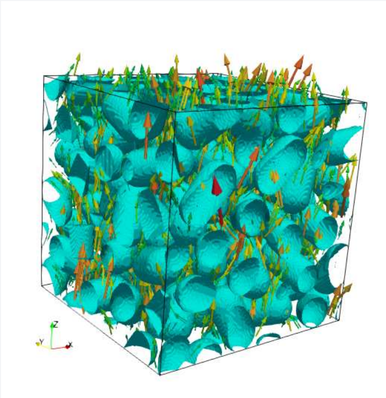
Andres Leon Islas

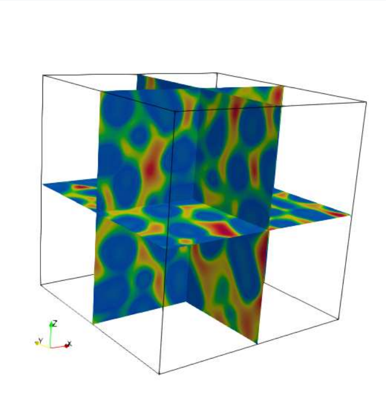
University of Texas at San Antonio

- **Acknowledgment:** “This material is based upon work supported by the Department of Energy Award Number DE-FE0031894.”

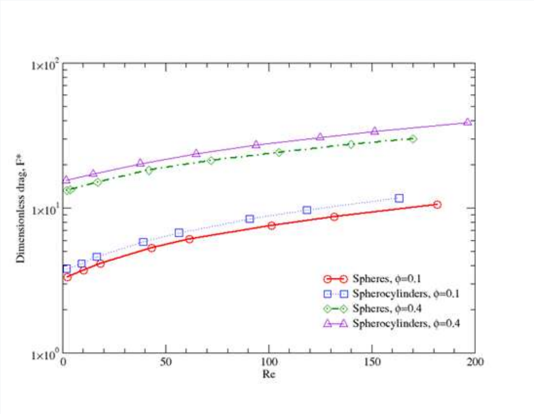
# Project Overview

- Development of Neural Network Models (This presentation)
- Simulations of Drags of Non-spherical Particles





*Velocity vectors and contours of 100 spherocylinders in a periodic cube at solid fraction 0.4.*




*Dimensionless mean drag of the assemblies*

# Introduction

- Wall-effect on the flow dynamics of sphere have been studied
  - Analytically
  - Experimentally
  - Numerically.
- Modeling this particulate flow system has applications in:
  - Microfluidic devices/systems
  - Fluidized beds
  - Sedimentation in streams/riverbeds

# Introduction

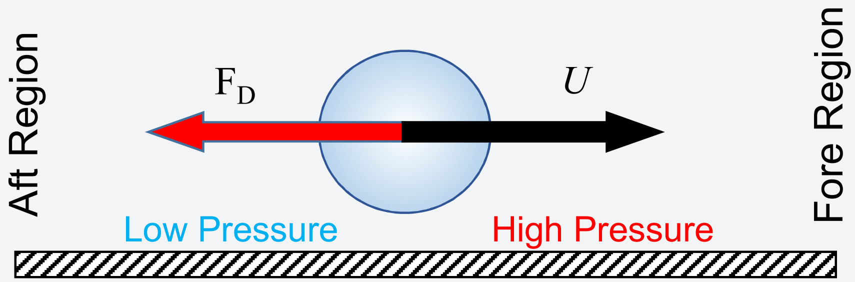
- Use of DL models to predict dynamics of particulate flow systems has increased in recent years

- Many DL models reported on employ strategy of developing multiple models to achieve desired level of accuracy

- Current work utilizes a multi-output DL regression model approach to model a simple particulate flow system
  - Well-trained model can decrease computation time
  - Continuous learning

# Wall Effect on Drag of a Sphere

- Wall boundary increases the drag of a sphere relative to the unbounded case
- The wall effect increases with decreasing Reynolds number and decreasing wall gap
- This effect arises from flow field asymmetry in the fore and aft regions of the sphere particle, resulting in an increased viscous effect
- Shear stress effects in the gap region also contribute to the net drag force when particle rotation is considered

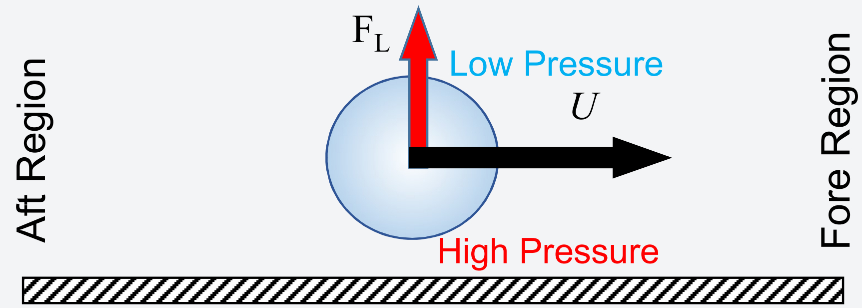


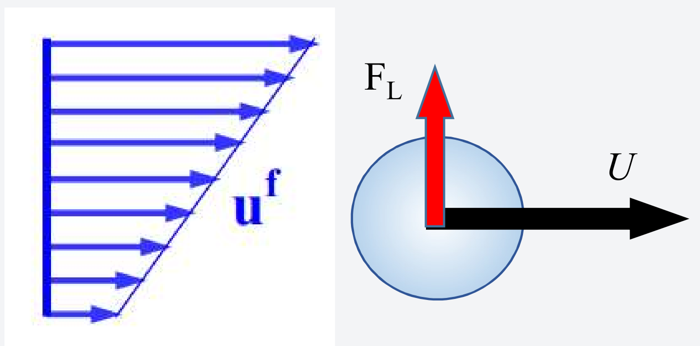

$$\frac{F_D}{F_{D,\infty}} > 1$$

Where $F_D$ = bounded drag force,
and $F_{D,\infty}$ = unbounded drag force

# Wall Effect on Lift of a Sphere

- A wall boundary has a similar, stronger influence on the lift force
  - Pressure differential exists in wall gap and unbounded region
- Other hydrodynamic contributions to the lift force to consider
  - Saffman lift force (linear shear flow)
  - Magnus lift force (rotating particle, uniform cross flow)
- Here, we study how the particle's rotational speed, direction, and proximity to wall affects net $C_L$
  - linear shear flow condition not imposed



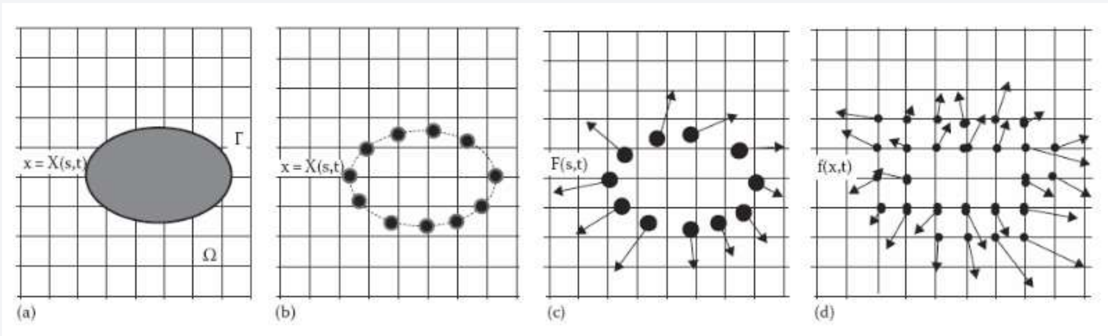



# Project Overview

- Generate a set of drag and lift data using a direct numerical simulation (DNS) method that is based on simple model of a sphere particle translating next to a plane wall
- Use the DNS data to develop and train a Deep Neural Network (DNN) to accurately predict the drag and lift coefficients simultaneously over a range of flow conditions
- Use the DNS data to develop correlations for drag and lift of the sphere particle

# IB-DNS Method

- Immersed Boundary method with direct forcing scheme used to simulate drag & lift data used for training.
  - (a) Description of the particle and fluid domains
  - (b) Discretization of the particle surface
  - (c) Transformation of particle-fluid boundary to Lagrangian point forces
  - (d) Transposition of boundary forces onto neighboring Eulerian grid points




**Adapted from E. E. Michaelides and Z.-G. Feng, "Computational Methods: The Immersed Boundary Method.," in Multiphase Flow Handbook, Boca Raton, CRC Press, 2016, pp. 126-135.**

# DNS Method - continued

- The no-slip boundary condition is maintained:

$$\frac{\partial \mathbf{X}(\mathbf{s},t)}{\partial t} = \mathbf{u}(\mathbf{X}(\mathbf{s},t),t) \qquad \mathbf{x} = \mathbf{X}(\mathbf{s},t)$$

- Governing momentum and continuity equations:

$$\boldsymbol{\rho}\left(\frac{\partial \mathbf{u}}{\partial \boldsymbol{t}} + \mathbf{u}\cdot\nabla\mathbf{u}\right) = \boldsymbol{\mu}\nabla^2\mathbf{u} - \nabla\boldsymbol{p} + \mathbf{f} \qquad \nabla\cdot\mathbf{u} = 0$$

- Direct forcing:

$$\mathbf{f} = \boldsymbol{\rho}\left(\frac{\partial \mathbf{u}}{\partial \boldsymbol{t}} + \mathbf{u}\cdot\nabla\mathbf{u}\right) - \boldsymbol{\mu}\nabla^2\mathbf{u} + \nabla\boldsymbol{p}$$

- Fluid body force density and velocity defined as:

$$\mathbf{f}(\mathbf{x},\boldsymbol{t}) = \int_{\Gamma} \mathbf{F}(\mathbf{s},\boldsymbol{t})\boldsymbol{\delta}\big(\mathbf{x} - \mathbf{X}(\mathbf{s},\boldsymbol{t})\big)\boldsymbol{ds}$$

$$\frac{\partial \mathbf{X}}{\partial \boldsymbol{t}} = \int_{\Omega} \mathbf{u}(\mathbf{x},\boldsymbol{t})\boldsymbol{\delta}\big(\mathbf{x} - \mathbf{X}(\mathbf{s},\boldsymbol{t})\big)\boldsymbol{d}\mathbf{x}$$

# DNS Method – continued

- Velocity at a boundary point $X_s$ :

$$\boldsymbol{U}_s = \boldsymbol{U}_c + \boldsymbol{\omega} \times (\boldsymbol{X}_s - \boldsymbol{X}_C)$$

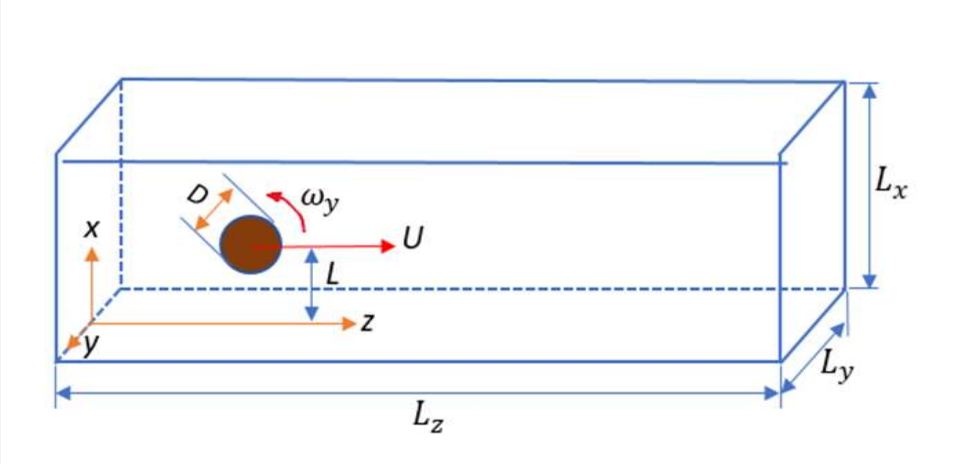
- Force spreading:

$$\delta(r) = \begin{cases} \frac{1}{4}\left[1 + \cos\left(\frac{\pi r}{2h}\right)\right], & |r| \leq 2h \\ 0, & |r| > 2h \end{cases}$$

$$D(\mathbf{x} - \mathbf{x}_{mn}) = \delta(x - x_{mn})\delta(y - y_{mn})\delta(z - z_{mn})$$

# DNS Model

- Simulation Box
  - Eulerian grid 120x80x320
  - Box size 9Dx6Dx24D
    - 1 < Re < 20
  - Box size 9Dx9Dx24D
    - Re = 0.5
- Sphere Surface
  - 3200 Lagrangian points




**Schematic diagram of a sphere translating and rotating in a simulation box filled with a viscous, quiescent fluid.**

# DNS Model

- Dimensionless parameters used to represent different particle movements:
  - $L/D$ [0.75, 2.5]
  - $Re = \frac{\rho U_z D}{\mu}$ [0.5, 20]
  - $Re_\omega = \frac{\rho \omega_y D^2}{\mu}$ [-5, 5]

- Drag and Lift Coefficients:
  - $C_D = \frac{2F_D}{\rho U^2 A} = f(\frac{L}{D}, Re, Re_\omega)$
  - $C_L = \frac{2F_L}{\rho U^2 A} = f(\frac{L}{D}, Re, Re_\omega)$

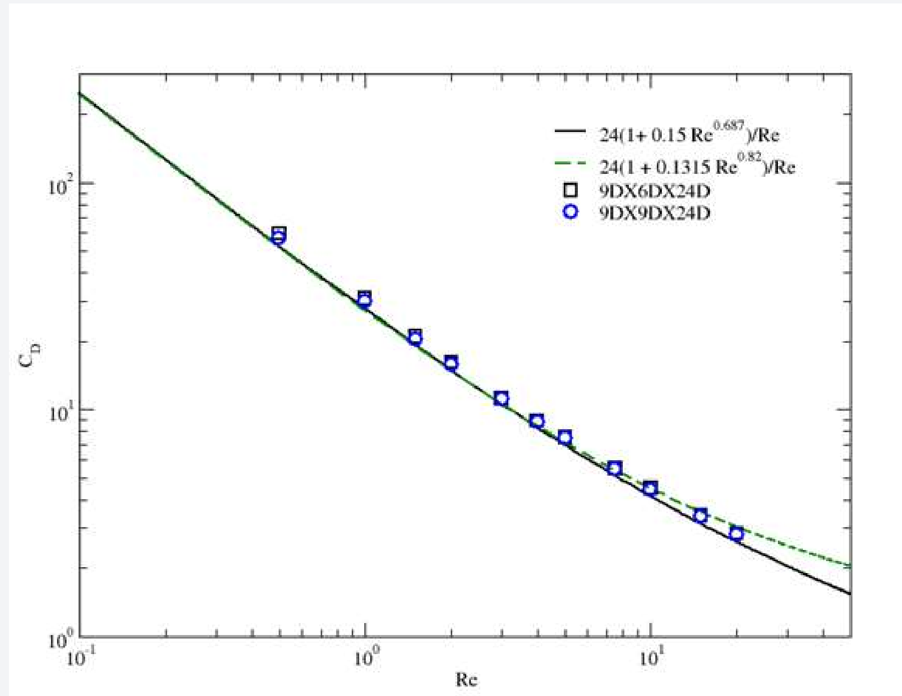
- 715 samples in total

# DNS Model Validation

- DNS drag compared to Beard, Schiller-Naumann correlations (no wall effect)
- DNS drag & lift compared with wall-effect simulation results found in literature

Table 1: Drag and lift coefficients of a sphere translating near a wall at $\frac{L}{D} = 1$ *and* $Re = 10$:

| | Zeng et al. | DNS (9Dx6Dx24D) | Difference in percentage |
|---|---|---|---|
| $C_D$ | 4.72 | 4.79 | 1.5% |
| $C_L$ | 0.351 | 0.362 | 3.1% |



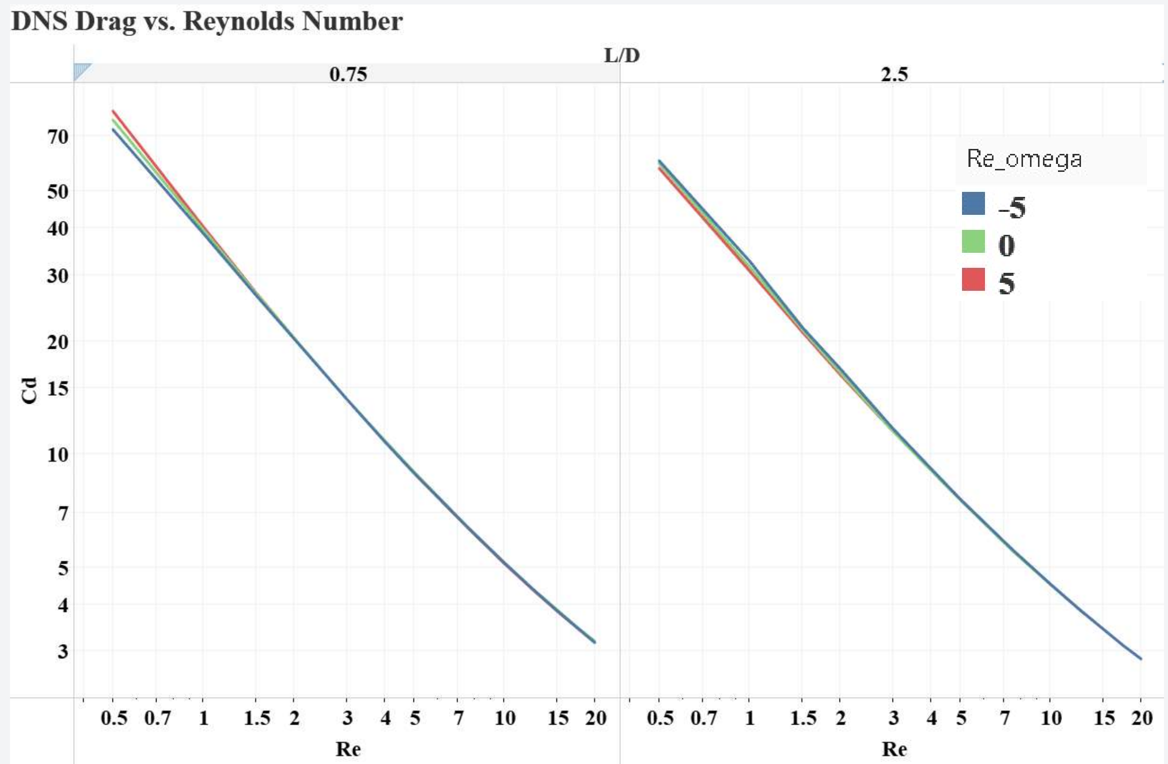

**Drag coefficient of a sphere at Reynolds numbers ranging from 0.5 to 20, $Re_{\omega} = 0$**

# DNS Drag Results

**Drag coefficients at Re = 0.5**

| $Re_\omega$ | L/D = 0.75 | L/D = 1.0 | L/D = 2.5 |
|---|---|---|---|
| -5.0 | 73.0 | 68.1 | 60.3 |
| 0 | 77.4 | 68.1 | 59.3 |
| 5.0 | 81.7 | 67.8 | 57.6 |

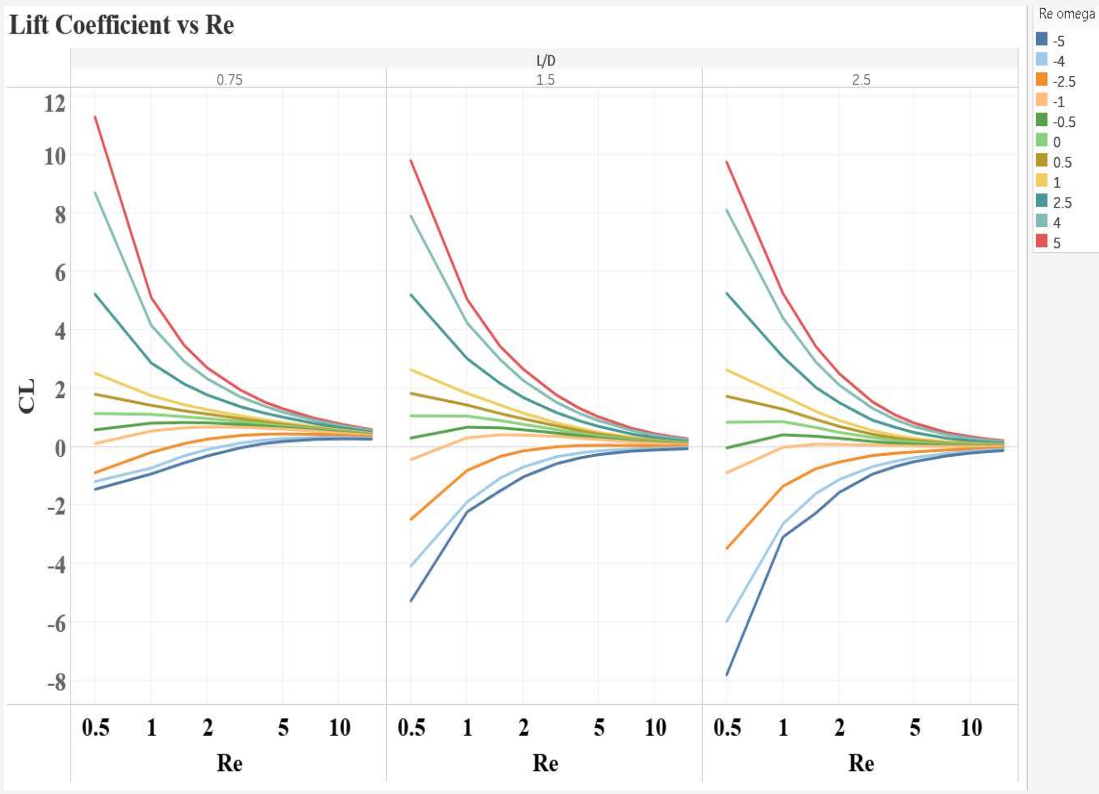
| $Re_\omega$ | L/D = 0.75 | L/D = 1.0 | L/D = 2.5 |
|---|---|---|---|
| -5.0 | -5.68% | 0.0 | 1.69% |
| 5.0 | 5.56% | -0.44% | -2.87% |



# DNS Lift Results

- Asymmetry in the lift coefficient distribution
- Opposing forces near wall
  - Magnus
  - Wall-Effect
- Dependent on rotational direction

| $Re_\omega$ | L/D = 0.75 | L/D = 1.5 | L/D = 2.5 |
|---|---|---|---|
| -5.0 | -1.47 | -5.3 | -7.83 |
| 0 | 1.13 | 1.05 | 0.83 |
| 5.0 | 11.3 | 9.80 | 9.75 |



# Related Work – Literature Survey

- Samantary et. al. (2017)
  - Used standard drag curve correlation to train ANN and predict $C_D$
  - 1 hidden layer, tanh activation, 745 samples, single input (Re)
- Nikolopoulos et. al. (2021)
  - Used simulation data to predict Heterogeneity index for fluidized Reactor
  - 3 models developed for different ranges of void fraction
  - 2 hidden layers in each model, 7 inputs, 1 output
- Viquerat & Hachem (2020)
  - Used CNN to predict drag of arbitrary 2-D shapes
  - Bezier curves → IB-DNS→CNN Learning (12,000 samples)

# Machine Learning Approach

- Python / Jupyter Notebook
- Tensorflow 2.3.0 / Keras API
  - Multi-output Regression
  - Densely connected
  - Multi-layer Perceptron (MLP)
- 3 Inputs/Features
  - L/D, Re, $Re_{\omega}$
- 2 Outputs/Labels
  - $C_D$, $C_L$

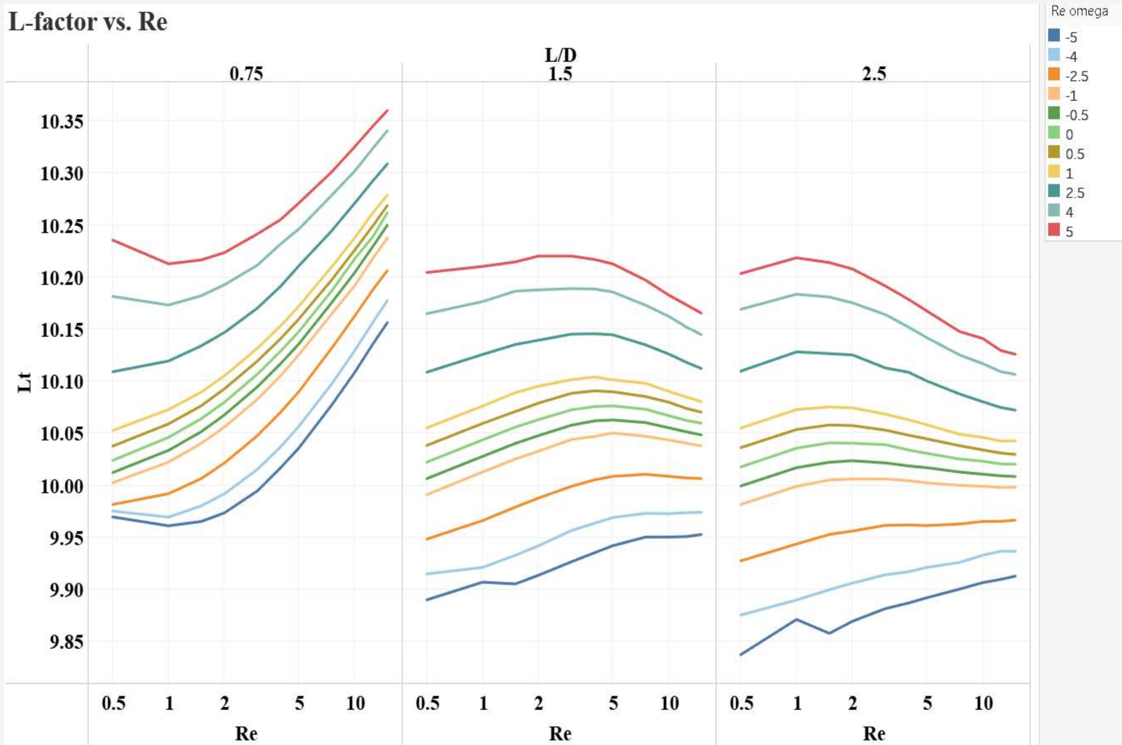
```
[1]: import matplotlib.pyplot as plt
     import numpy as np
     import pandas as pd
     import os

     from sklearn.model_selection import KFold
```

```
[2]: import tensorflow as tf

     from tensorflow import keras
     from tensorflow.keras import layers
     from tensorflow.keras import utils
     from tensorflow.keras.callbacks import LambdaCallback

     print(tf.__version__)

     2.3.0
```

```
[3]: import tensorflow_docs as tfdocs
     import tensorflow_docs.plots
     import tensorflow_docs.modeling
```

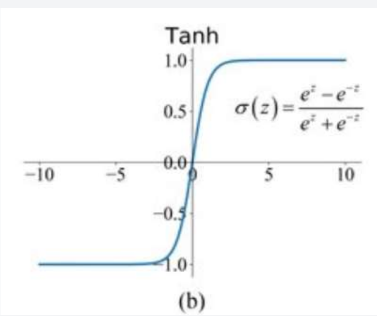
# Data Transformation & Normalization

- Drag Coefficient to "K-factor"
  - $K = \frac{C_D}{\frac{24}{Re}}$
- Lift Coefficient to "L-factor"
  - $L_T = 10 + \frac{C_L}{\frac{4}{Re}}$
- Data Normalization (-1 to 1)
  - $\boldsymbol{X}' = a + \frac{(\boldsymbol{X}-\boldsymbol{X}_{min})(b-a)}{\boldsymbol{X}_{max}-\boldsymbol{X}_{min}}$
- Data shuffled before each test



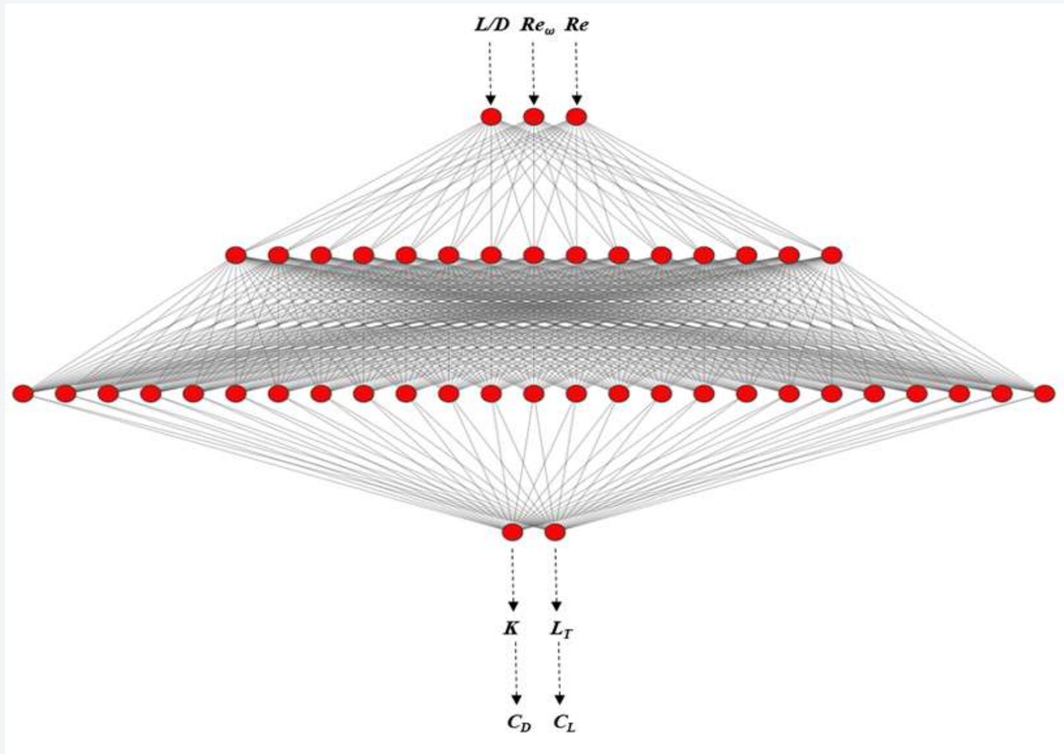
# Final Multi-Output FNN Architecture

| Layer | Nodes | Activation |
|---|---|---|
| Input | 3 | N/A |
| Hidden 1 | 15 | tanh |
| Hidden 2 | 25 | Softmax |
| Output | 2 | Identity |




**Softmax**

$$\varphi(\boldsymbol{\theta}_i) = f\left(\frac{e^{F_i}}{\sum_i e^{F_i}}\right)$$



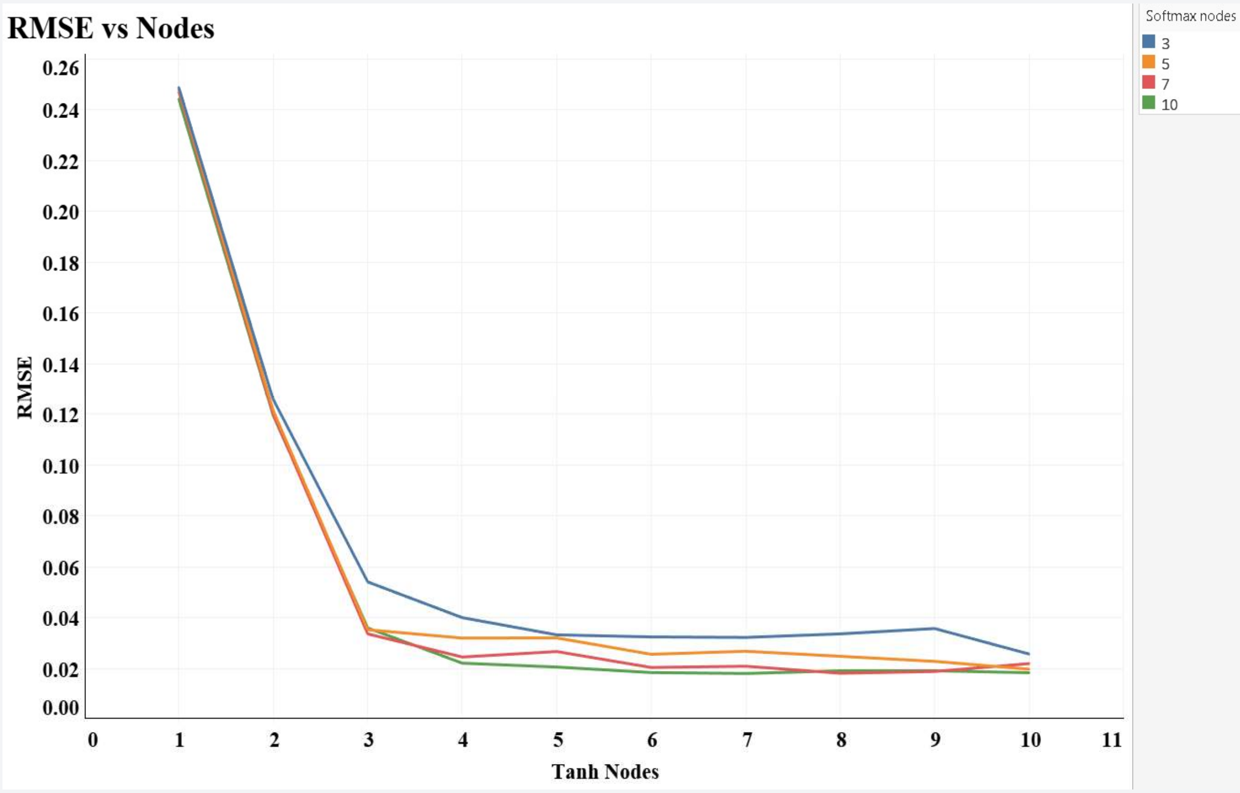
# Other Hyperparameters

- Kernel Initializer = HeUniform
  - Scales variance of initial weight tensor
- Learning Epochs = 4000
- Learning Rate = .0005
- Optimizer = RMSProp
  - $E[g^2]_t = 0.9E[g^2]_{t-1} + 0.1\left(\frac{\delta C}{\delta w}\right)^2$
  - $w_t = w_{t-1} - \frac{\eta}{\sqrt{E[g^2]_t}}\left(\frac{\delta C}{\delta w}\right)$
- Loss = Mean Squared Error (MSE)
  - $MSE = \frac{\sum(y_i - o_i)^2}{n}$
  - $MAE = \frac{\sum|y_i - o_i|}{n}$
  - $RMSE = \sqrt{\frac{\sum(y_i - o_i)^2}{n}}$
- Validation Split for Learning = 0.15

# Model Performance vs. Number of Nodes

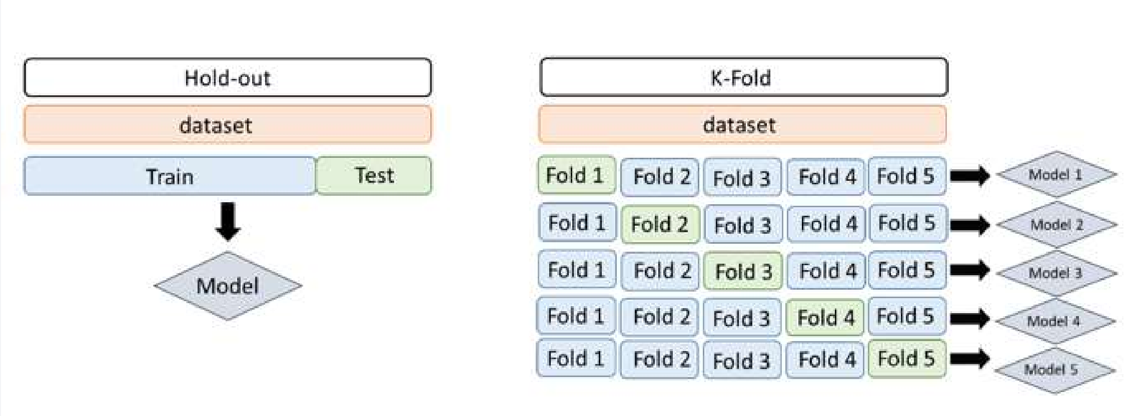
- 100 models trained adding a single node with each new model
- RMSE based on 715 samples
- Marginal improvement beyond 7 nodes in each hidden layer

| Parameter | Value |
|---|---|
| Learning Rate | 0.001 |
| Epochs | 1200 |



# K-Fold Cross Validation

- 8-Fold Cross Validation Test
  - ~20% test split for each fold
  - 0.0002 learning rate
  - 2500 epochs
  - 15% validation split



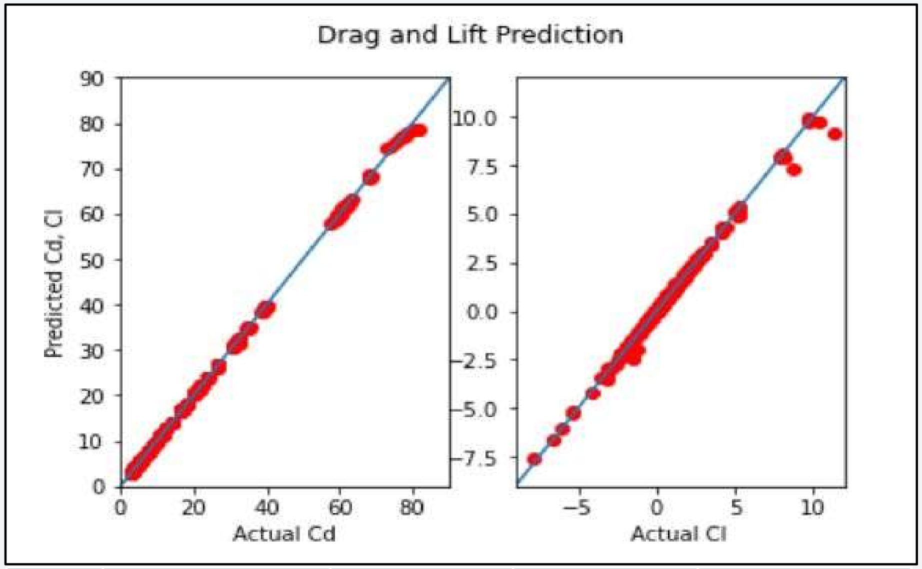

***Comparison of Hold-out CV vs K-fold CV method used for regression model validation.***

| | Avg. MAE | Std. Dev. MAE | Avg. RMSE | Std. Dev. RMSE |
|---|---|---|---|---|
| Test K | 0.0062 | 0.0011 | 0.0091 | 0.0019 |
| Test $L_T$ | 0.0174 | 0.0033 | 0.0292 | 0.0105 |

| | Avg. MAE | Std. Dev. MAE | Avg. RMSE | Std. Dev. RMSE |
|---|---|---|---|---|
| Train | 0.0088 | 0.0006 | 0.0144 | 0.0009 |
| Validation | 0.0090 | 0.0006 | 0.0127 | 0.0020 |
| Test | 0.0095 | 0.0017 | 0.0153 | 0.0046 |

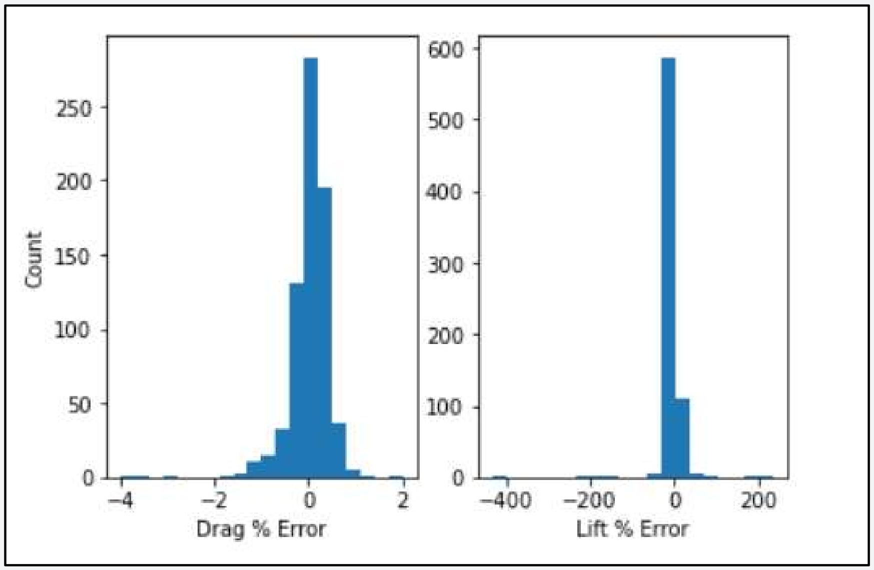
# Final Model Selection & Performance

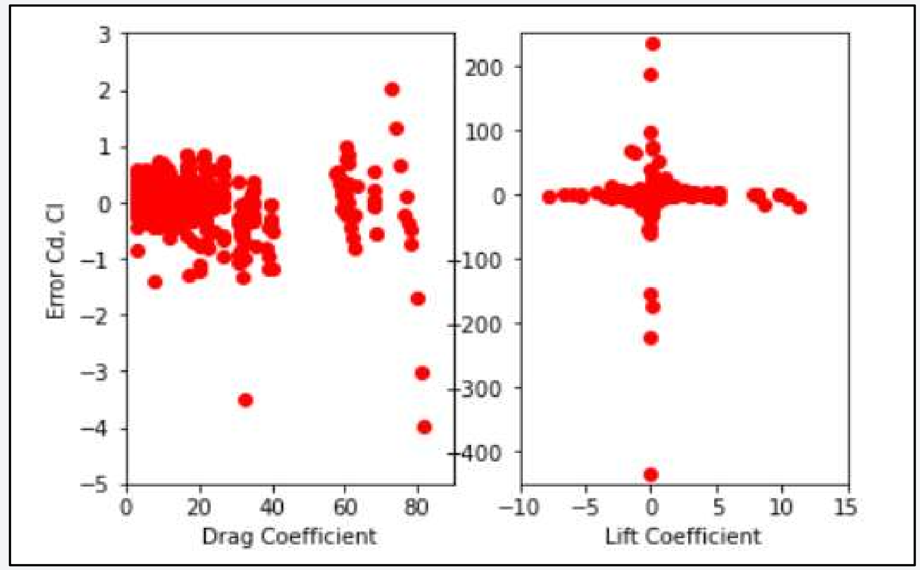
- K-fold models did not predict the full set of $C_D$ and $C_L$ data very well
- 715 samples used for final learning with 15% validation split
- Overall error similar to k-fold performance (K, $L_T$)




| | Train MAE | Validation MAE | Train RMSE | Validation RMSE |
|---|---|---|---|---|
| Start | 0.38777 | 0.36744 | 0.46303 | 0.43443 |
| End | 0.00616 | 0.00824 | 0.01149 | 0.01849 |

# Final Model Prediction Error Distribution



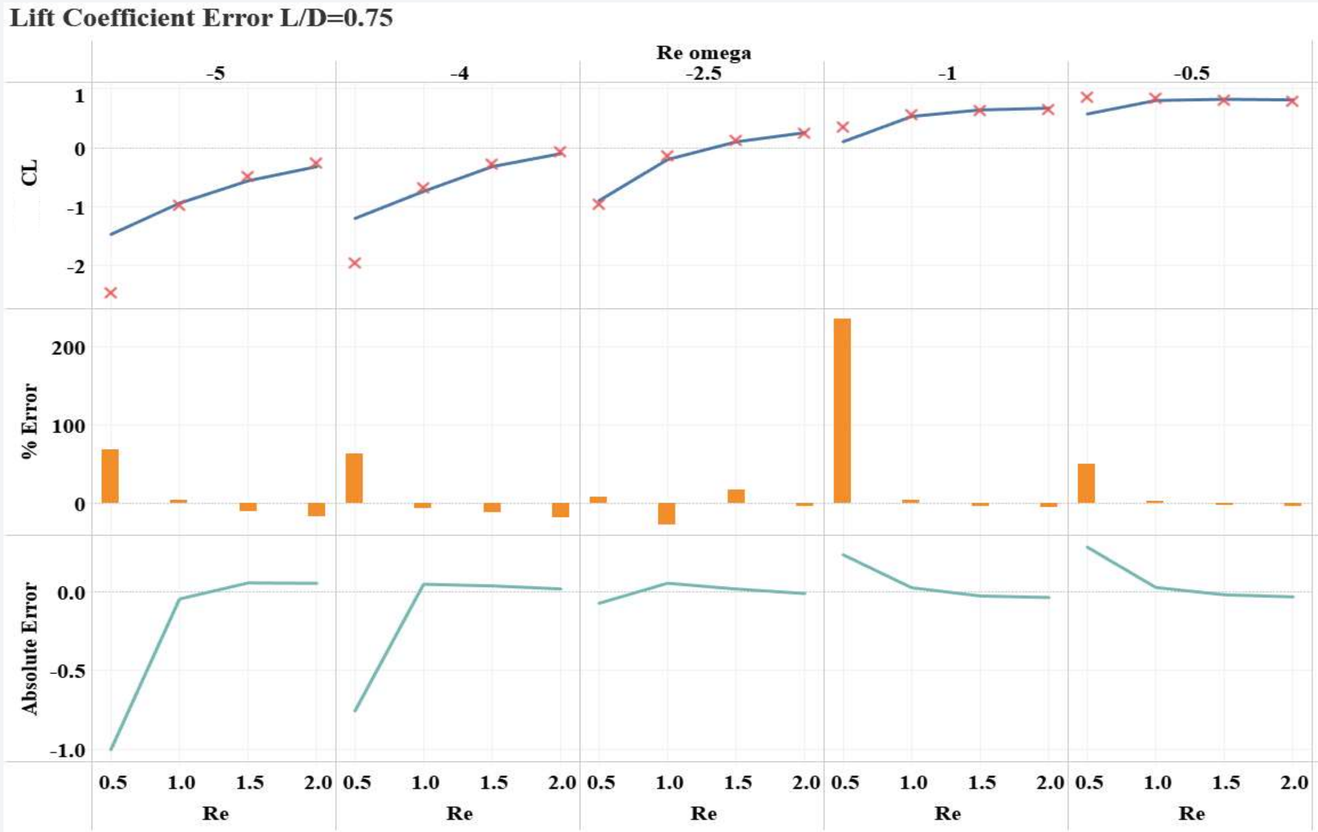

- 100% drag predictions within ± 5% from DNS
- 91% lift predictions within ± 10% from DNS
- 96% lift predictions within ± 20%

- Largest error in regimes were $C_L < 0.01$
- ~50% of high error predictions $-2.5 < Re_\omega < -1.0$
  - sign change in $C_L$

# $C_L$ Prediction Error – Near Wall

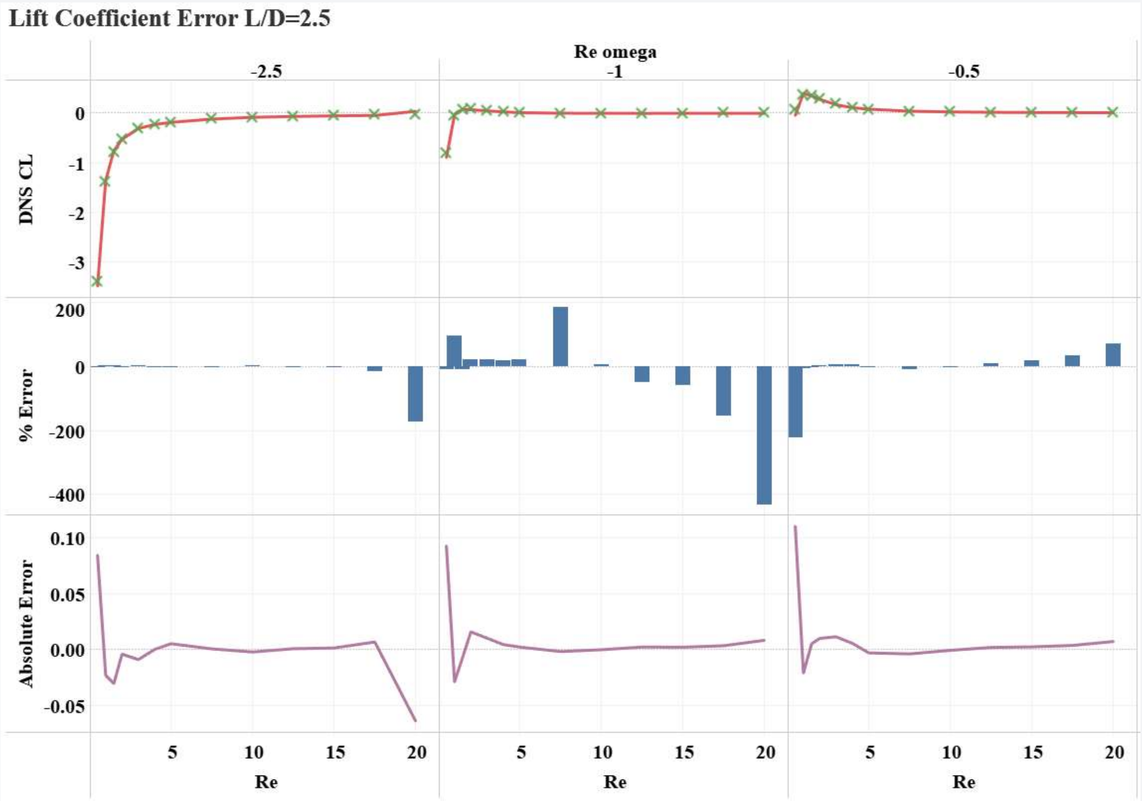
- Largest prediction errors in creeping flow (Re=0.5), near wall, clockwise rotation

- NN Model tends to over-predict $C_L$ in the creeping flow, near wall condition



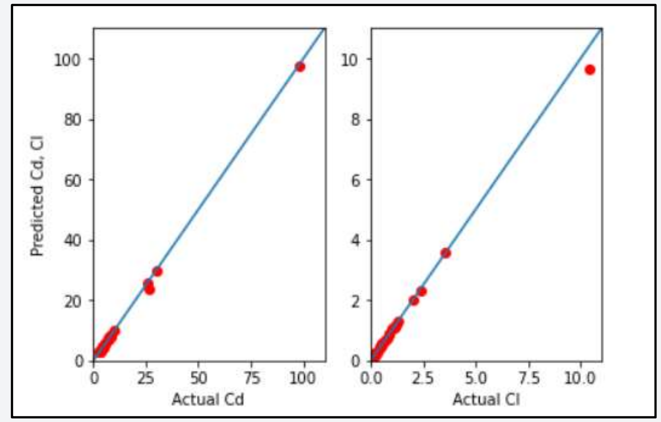
# $C_L$ Prediction Error – Far from Wall

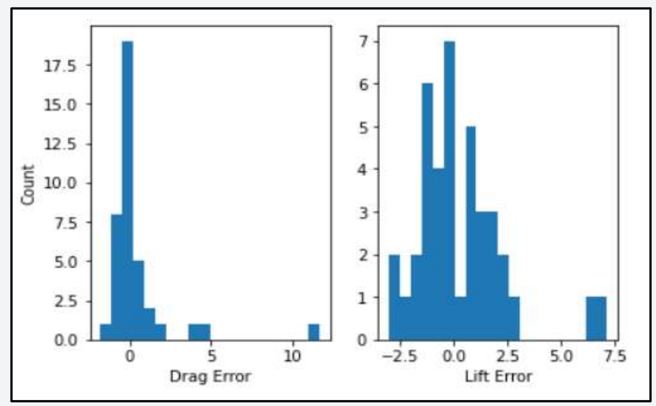
- Far from wall, clockwise rotation, small $Re_{\omega}$, very small $C_L$
- Small absolute error amplified when expressed as %



# Final Hold Out Validation Test

- New dataset for final hold out test
- 39 Randomly generated DNS inputs/combinations for ($Re_\omega > 0$)
- Predictions evaluated for deviation from DNS drag and lift coefficients
- 97.4% $C_D$ predictions within ± 5% Error
- 100% $C_L$ predictions within ± 7.5% Error



# Wall Effect Impact Factor

- $K_f = \frac{C_D}{C_{D,\infty}}$
- Faxen Analytical Solutions
  - $K_{fF3O\parallel} = 1 + \frac{9}{16}\kappa + \frac{81}{256}\kappa^2 + \frac{217}{4096}\kappa^3$
  - $K_{fF5O\parallel} = \left[1 - \frac{9}{16}\kappa + \frac{1}{8}\kappa^3 - \frac{45}{256}\kappa^4 - \frac{1}{16}\kappa^5\right]^{-1}$
- $K_{f,FNN}$ used Schiller –Naumann correlation for $C_{D,\infty}$

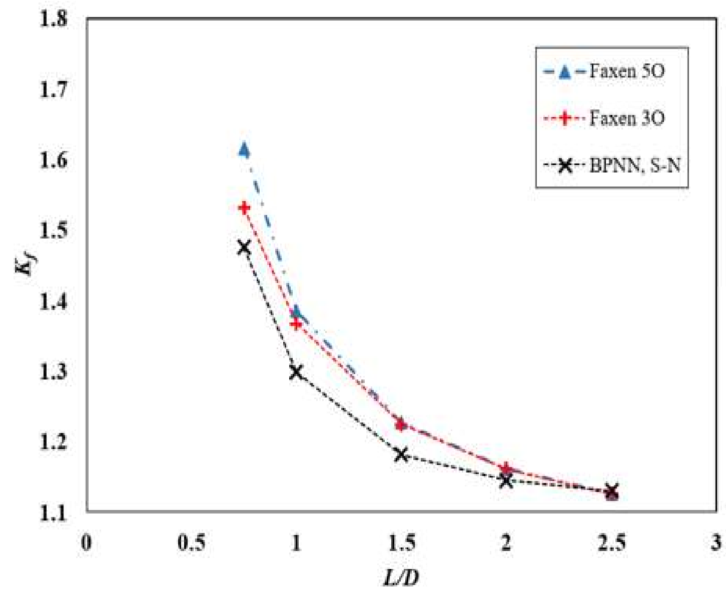
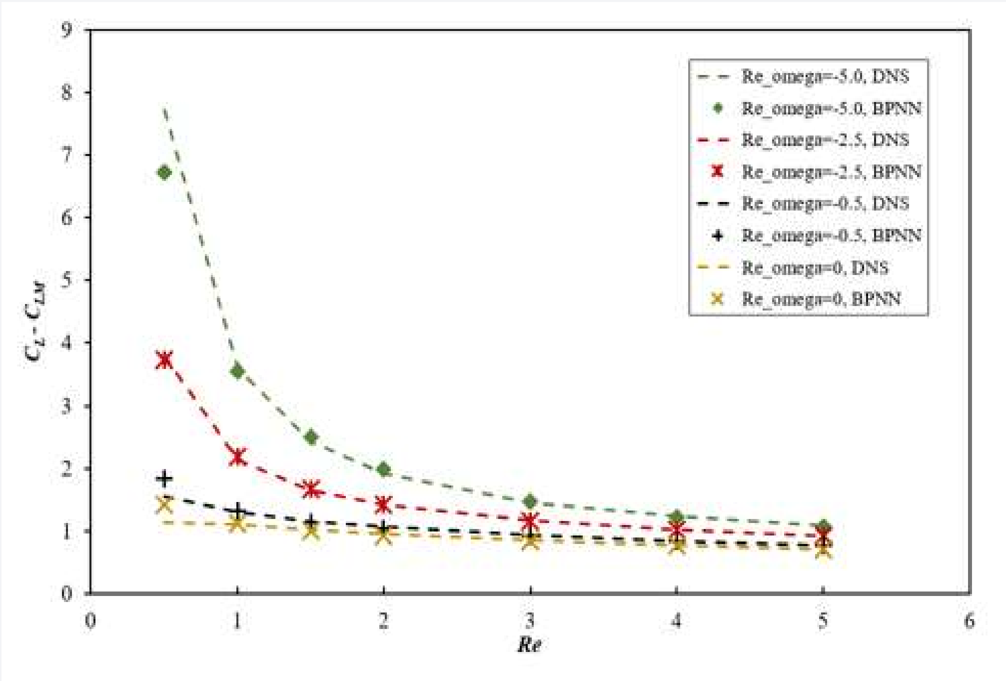



**Figure 19. Comparison of Faxen 3rd and 5th order analytical solutions for wall effect drag correction factor $K_f$ vs. L/D with $K_{FNN,S-N}$ (Re=0.5, $Re_\omega$=0).**

# Wall effect and Magnus Lift Coefficient

- Oesterle & Bui Dinh Correlation for used for Magnus lift coefficient
  - $C_{LM} = 0.45 + \left(\frac{Re_{rot}}{Re_s} - .45\right) e^{-.05684 Re_{rot}^{0.4} Re_s^{0.3}}$
  - Symmetry with respect to $Re_\omega$
- Greatest deviation from $C_{LM}$ near wall, clockwise rotation with $Re_\omega$
  - Opposing forces




**Difference between DNS, NN model $C_L$ and the Magnus lift coefficient equation 12 $C_{LM}$ versus Re at different angular speeds ($Re_\omega$=-5.0, -2.5, -0.5, 0, L/D=0.75)**

# Conclusions

- A Multi-Output FNN Regression Model was developed to predict drag and lift coefficients simultaneously at low Reynolds numbers
- DNS data showed particle rotation has little effect on drag, but heavily influences net lift force.
- FNN lift prediction errors are higher when sign of $Re_{\omega}$ is negative (clockwise rotation) and when $C_L$ is small.
- Wall-effect drag and lift correlations were developed for a fixed $Re_{\omega}$ and showed good agreement with Analytical and other numerical simulation results