Field Inversion and Machine Learning in SU2

August 9th, 2019

Joint National Institute of Aerospace (NIA) & SU2 Foundation
User Workshop

Presenter: Jon Holland
Ph.D. Candidate, Aerospace Engineering
University of Maryland, College Park

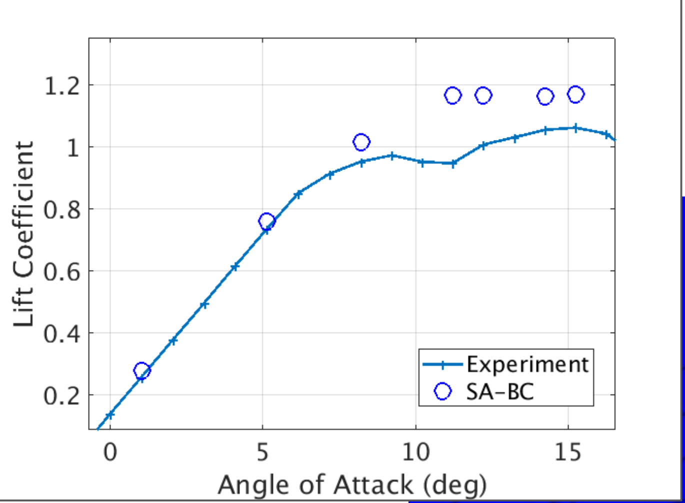
Dr. James Baeder
Professor, Aerospace Engineering
University of Maryland, College Park

Dr. Karthik Duraisamy
Associate Professor, Aerospace Engineering
University of Michigan, Ann Arbor

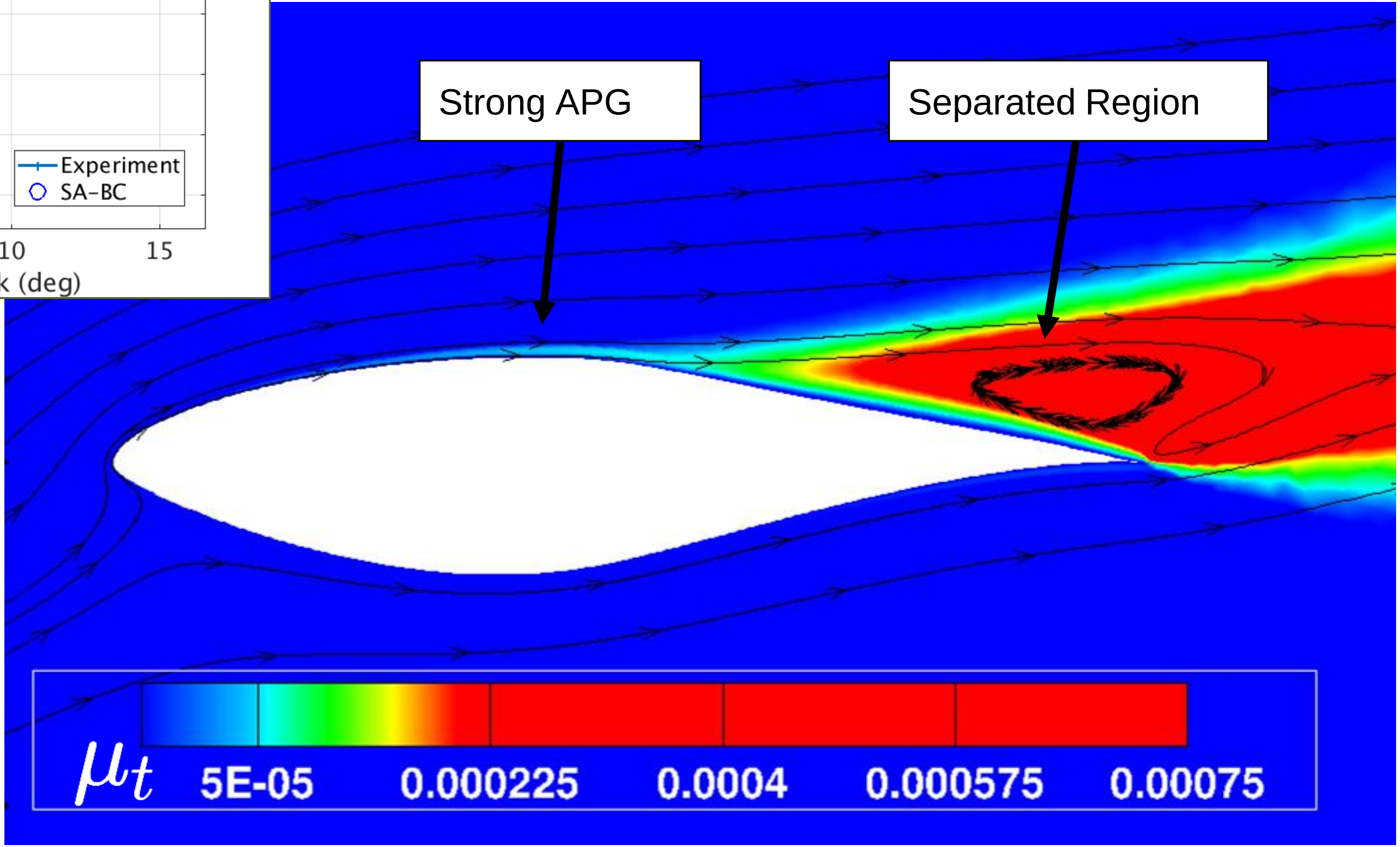
A. JAMES CLARK
SCHOOL OF ENGINEERING

DEPARTMENT of
AEROSPACE ENGINEERING

Motivation



Lift Coefficient
1.2
1
0.8
0.6
0.4
0.2
0
5
10
15
Angle of Attack (deg)
Experiment
SA-BC



Strong APG
Separated Region
μ_t
5E-05
0.000225
0.0004
0.000575
0.00075

Correction Field - RANS Applications

- Introduce Field Variable to Model

$$\frac{\partial \hat{\nu}}{\partial t} + u_j \frac{\partial \hat{\nu}}{\partial x_j} = \mathrm{P} - \mathrm{D} + \mathrm{Diffusion}$$

$$P = \gamma c_{b1}(1 - f_{t2})\hat{S}\hat{\nu} \quad \Rightarrow \quad P = \beta(x_j)\gamma c_{b1}(1 - f_{t2})\hat{S}\hat{\nu}$$

- Effectively changing entire model (not just production term)
- Correction Field Found by Inversion
- Goal: Find $\beta(\eta)$

FIML Classic

- Data:

$$k_d = \{C_L, C_D, C_f, T, \ldots\}$$

- Inversion:

$$\min_\beta (J_c)$$

$$J_c(\beta) = \|k_d - k_m(\beta)\|_2^2 + \lambda\|\beta - 1.0\|_2^2$$

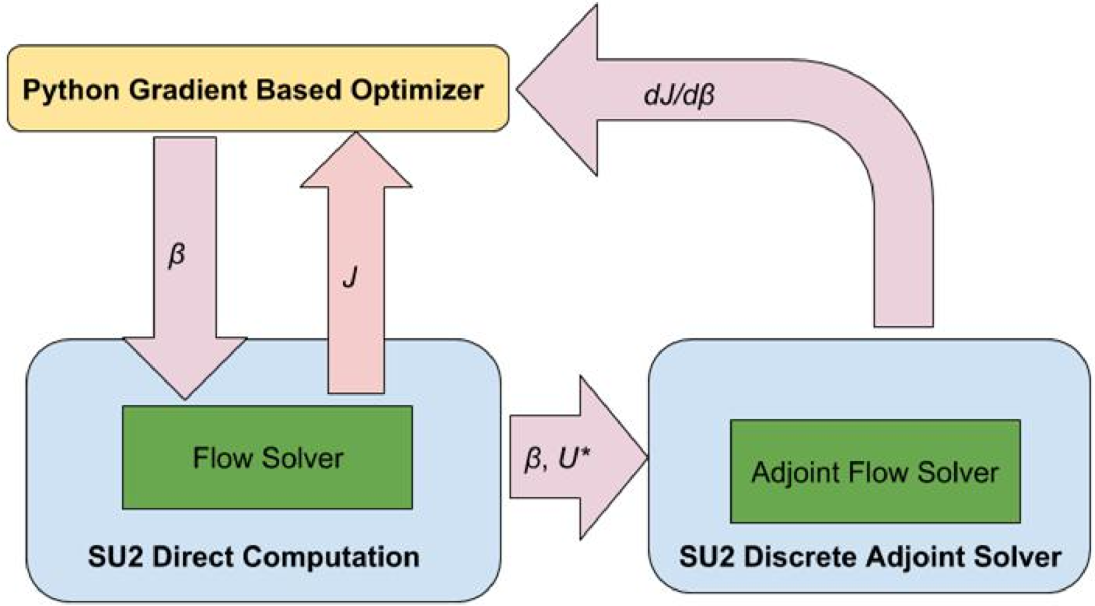
- Training: $\beta(x_j) \rightarrow \beta(\eta)$

Why Use SU2 for FIML?

- Open Source
- Auto-differentiated Discrete Adjoint Solver
  - Enables Easy Experimentation
- SciPy Optimizers
  - Easy Interface to Variety of Optimizers
- Large User Base / Community for Support
  - Forum Q/A Was Immensely Helpful

FIML Process Analogous to SU2 Adjoint Shape Optimization, With Redefined Design Variables

SU2 FI-Classic Implementation

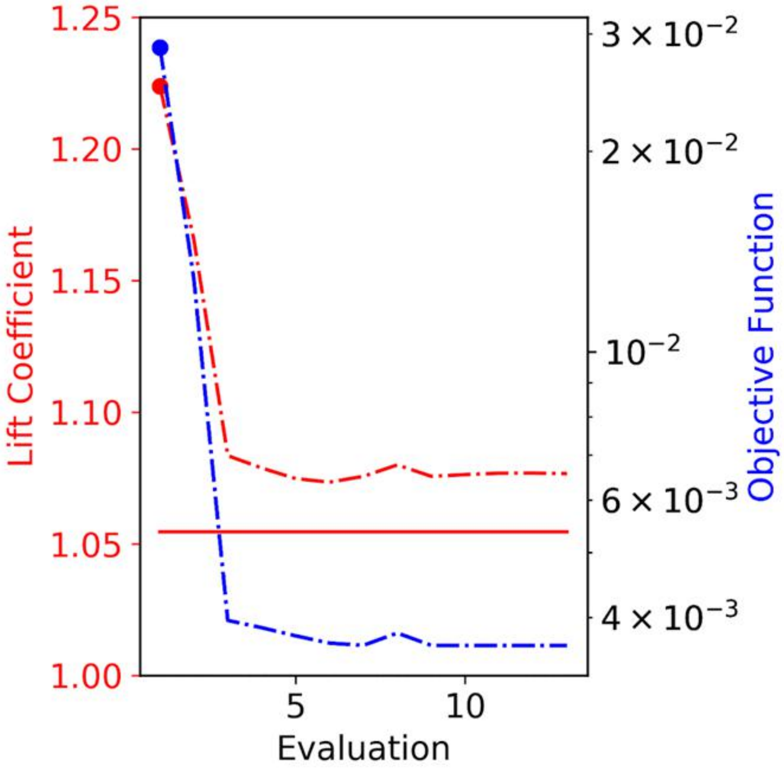


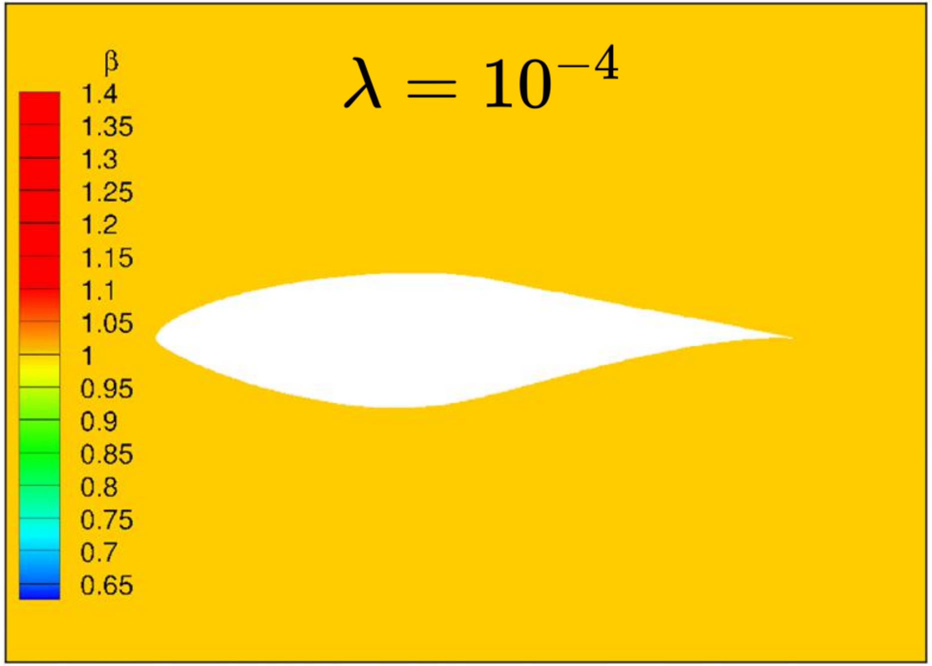
β	Turbulence Model Correction
U^*	Converged Flow Variables
J	Objective Function
$dJ/d\beta$	Gradient of Objective Function

Summary of Changes to SU2 v5.0.0 Code (FIML-Classic)

1. Redefine Design Variables to Modify Production Term of Turbulence Model
   a. Small Number of Design Variables Assumed, FIML Requires Design Variable At Every Node (~80,000), so Unsigned Shorts -> Unsigned Longs
   b. Python Scripting to Initialize Variables, Config Routines to Read and Store Variables in Turbulence Model (solver_direct_turbulent.cpp)
   c. Discrete Adjoint Solver Scripting to Set New DVs as Input, Retrieve and Store Gradient
   d. Output Routines to Store and Visualize Gradients (and Other Turbulence Model QOI)
   e. Store Variables and Gradients Separately From Config File, Python Routines to Read/Store/Copy Files as Necessary During Inversion
2. Define FIML Objective Functions
   a. solver_direct_mean.cpp, solver_direct_mean_inc.cpp
3. Modified Python Optimizers
   a. Added Interface to SciPy L_BFGS_B (Limited Memory BFGS), and Steepest Descent
   b. Added Routines to Validate, Test Gradient (Not Possible to Validate Gradients by FD)

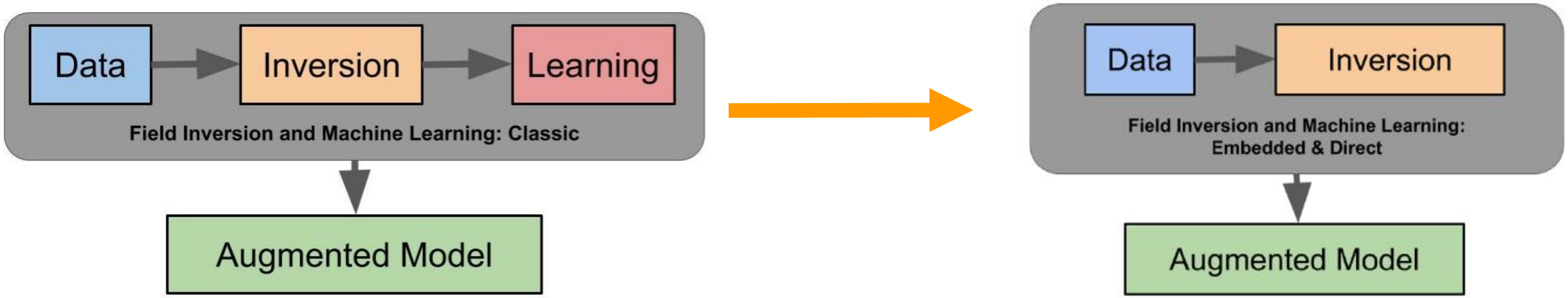
FI-Classic Inversion Results: S809 Airfoil





FIML Direct in SU2

- Integrate Neural Network In Turbulence Model!
- Train Weights Directly!
- Primary Difference from FIML-Classic in SU2 Code is All in Turbulence Model
  - Gather / Scale Turbulent Features, Forward Propagate to Obtain Correction Field
  - Backpropagate to Give Regularization Term Component Gradient (NOT Used to Update Weights)
  - Weights Held Constant, Are Now the Design Variables for Discrete Adjoint Solver



FIML Direct: Weights as Design Variables

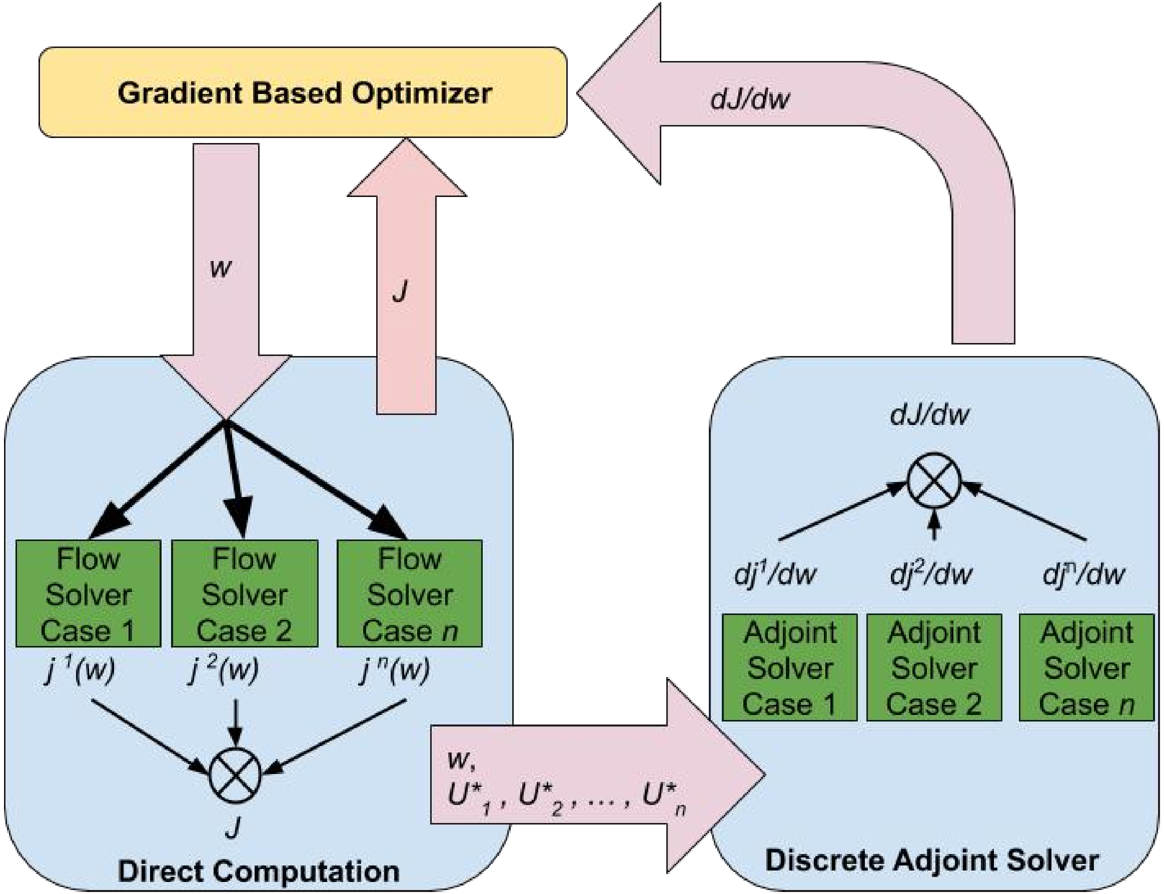
- Train Neural Network Directly By Treating Weights as the Design Vars:

$$J_d(w) = ||k_d - k_m(w)||_2^2 + \lambda||\beta(w) - 1.0||_2^2$$

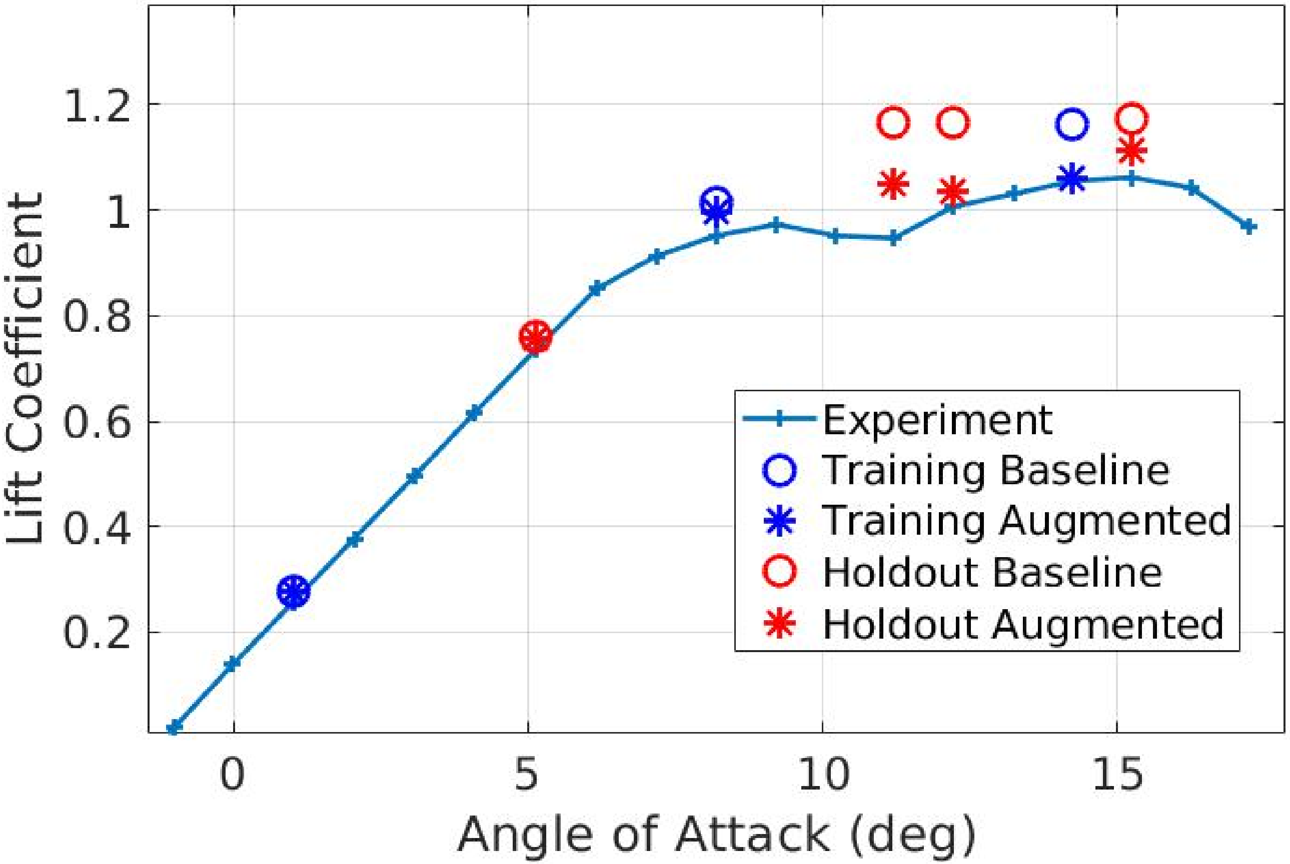
- Correction is Output of Neural Network $(\beta(w))$
  - Updated Every Flow Iteration With Current Features
- Weights Held Constant For Each Evaluation
  - Initialized to Small Random Values
- Inversion is Simply:

$$\min_w(J_d)$$

SU2 FIML Direct Implementation for Multiple Cases

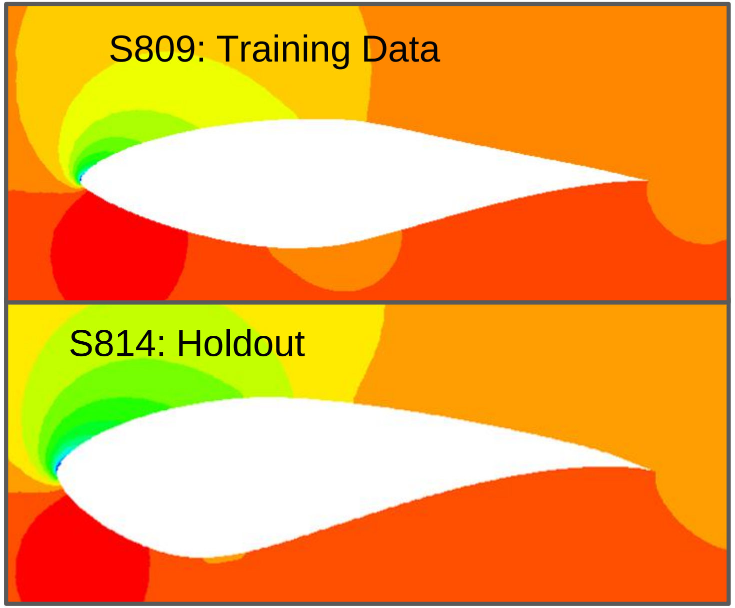


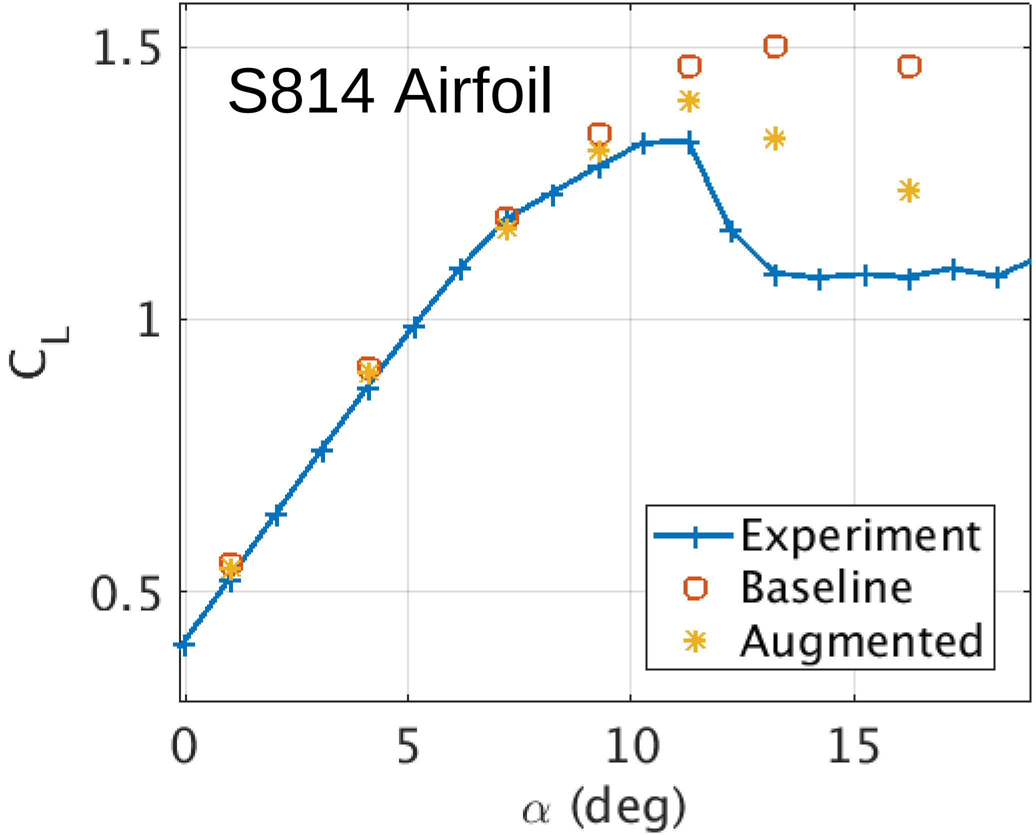
FIML-Direct S809 Trained at 3 AoAs



Testing on S814 Airfoil

- Network Trained on 7 S809 AoAs
- S814 *NOT* in Training Set
- Model Augmentation Improves Predictions on S809 *AND* S814 Airfoil





Questions?

https://github.com/jholland1

https://www.researchgate.net/profile/Jonathan_Holland5