A Framework for Turbulence Modeling using Big Data

Karthik Duraisamy
Department of Aerospace Engineering
Univ of Michigan

NASA Aeronautics Research Mission Directorate (ARMD)

LEARN Technical Seminar

Sep 6, 2017

Team

Karthik Duraisamy (Michigan): Turbulence Modeling, Inverse Modeling, Integration, Machine Learning
Juan Alonso (Stanford): Machine Learning, Integration
Paul Durbin (Iowa State): Turbulence modeling
Kaushik Das (Pivotal): Scalability of Machine Learning
Philippe Spalart (Boeing): Advisory

Students:
Brendan Tracey, Helen Zhang, Anand Pratap Singh, Eric Parish, Racheet Matai

Postdocs:
Shivaji Medida, Asitav Mishra

Pivotal:
Hulya Farinas, Grace Gee

Resolution requirements for aircraft wing ($Re_c = 2x10^7$)

DNS

LES

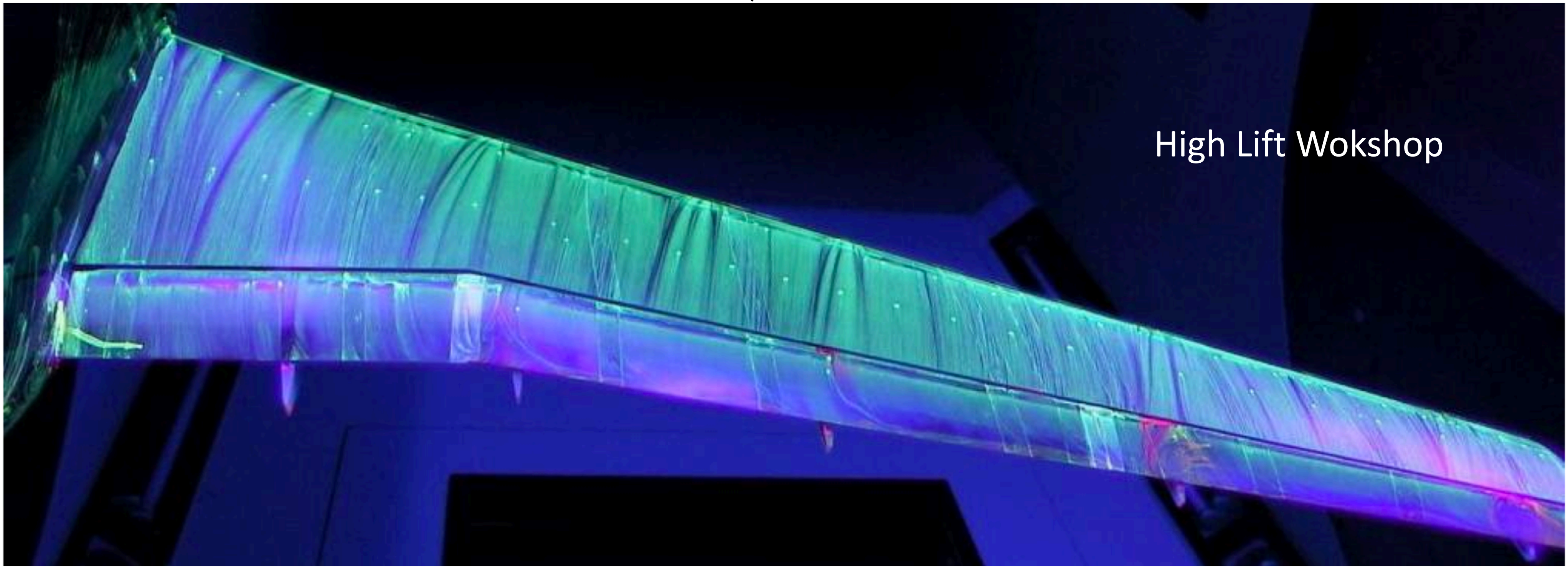
Hybrid RANS-LES / WMLES

RANS

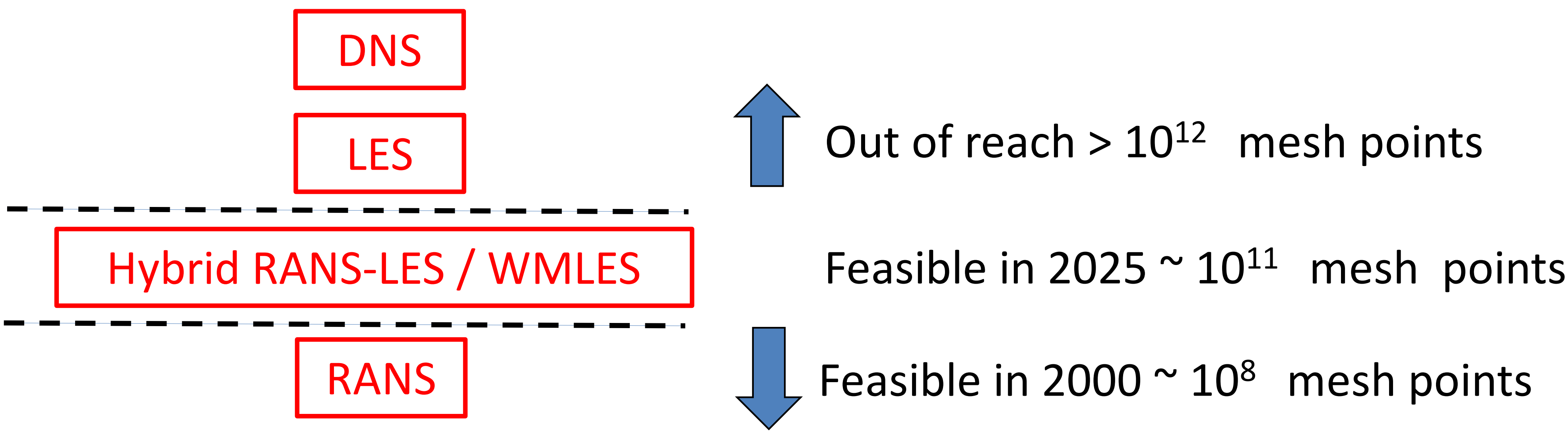
Out of reach > 10^{12} mesh points

Feasible in 2025 ~ 10^{11} mesh points

Feasible in 2000 ~ 10^{8} mesh points



Resolution requirements for aircraft wing (Re$_c$ = 2x10^7)

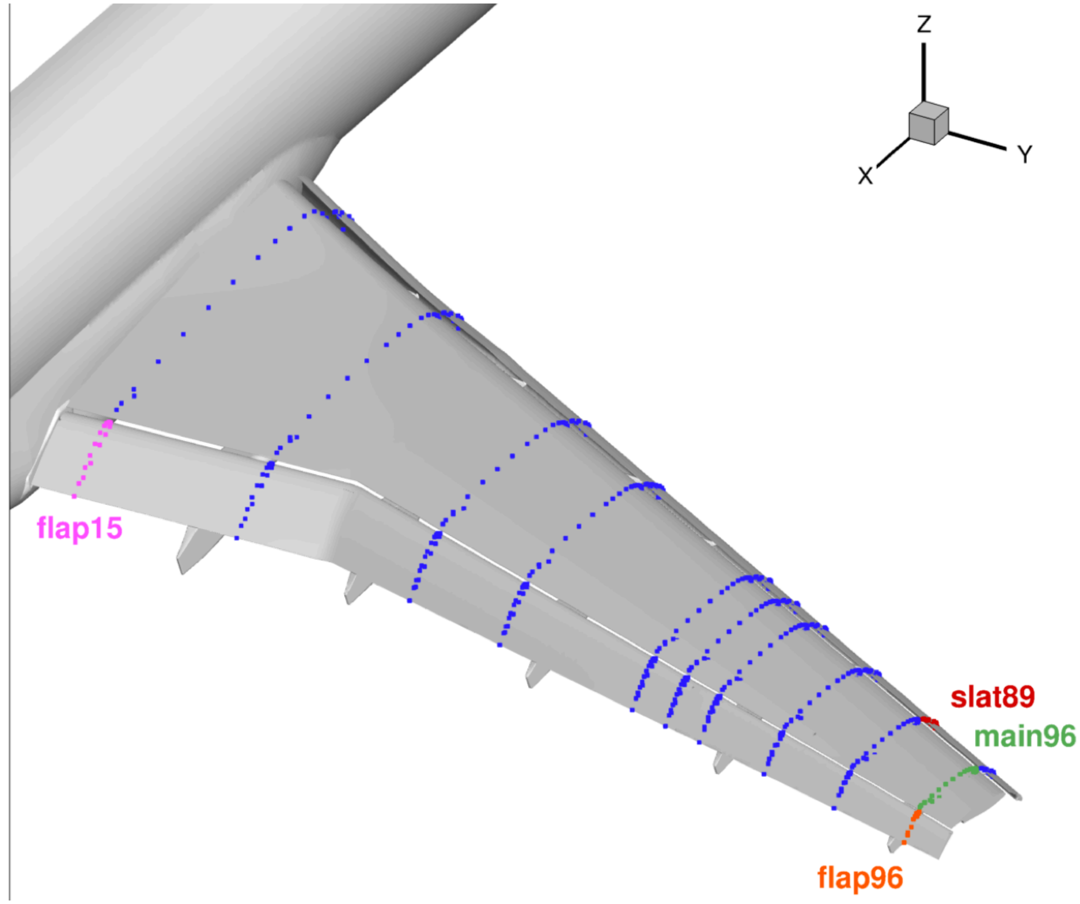


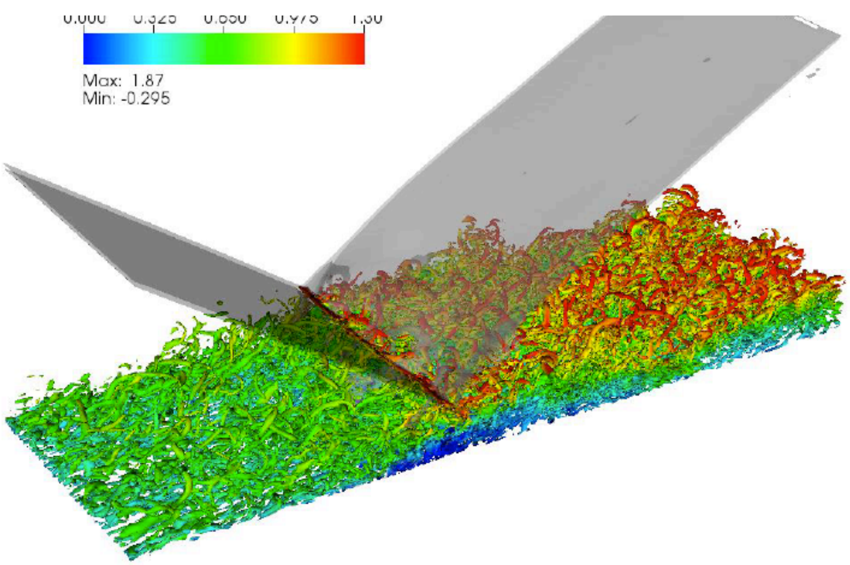
Near-wall modeling is here to stay for the next 20 years for analysis

Many other use cases: Full flight envelope, Parameter sweeps, Design, Trajectory prediction, mission planning. RANS will never go away.

Data deluge…

- DNS and LES have been produced in quantity
- Experimental PIV and MRV high-res data sets





- Data sets have not had a substantial impact on closure modeling

Commercial example : Face recognition

Data

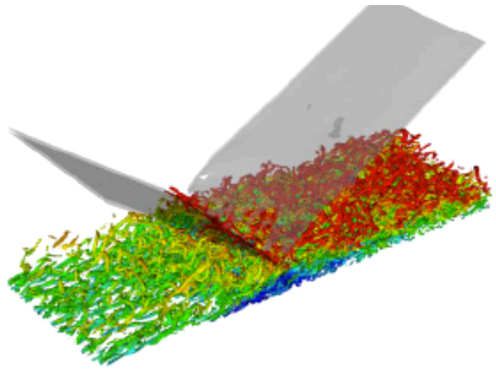
Mathematical model
+Machine Learning

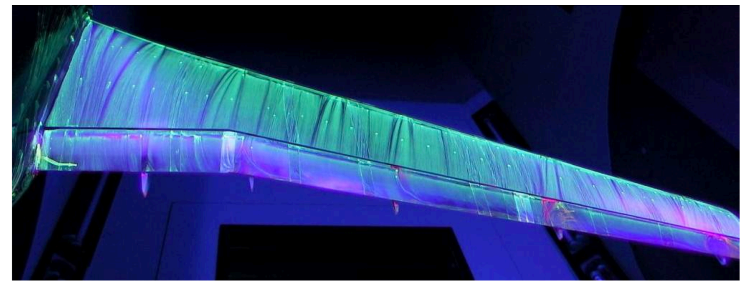
Predictive capability

- No physical law ;
- Data is directly useful for model;
- Large amounts of relevant data.

Can we replicate this type of success in turbulence modeling?

Data



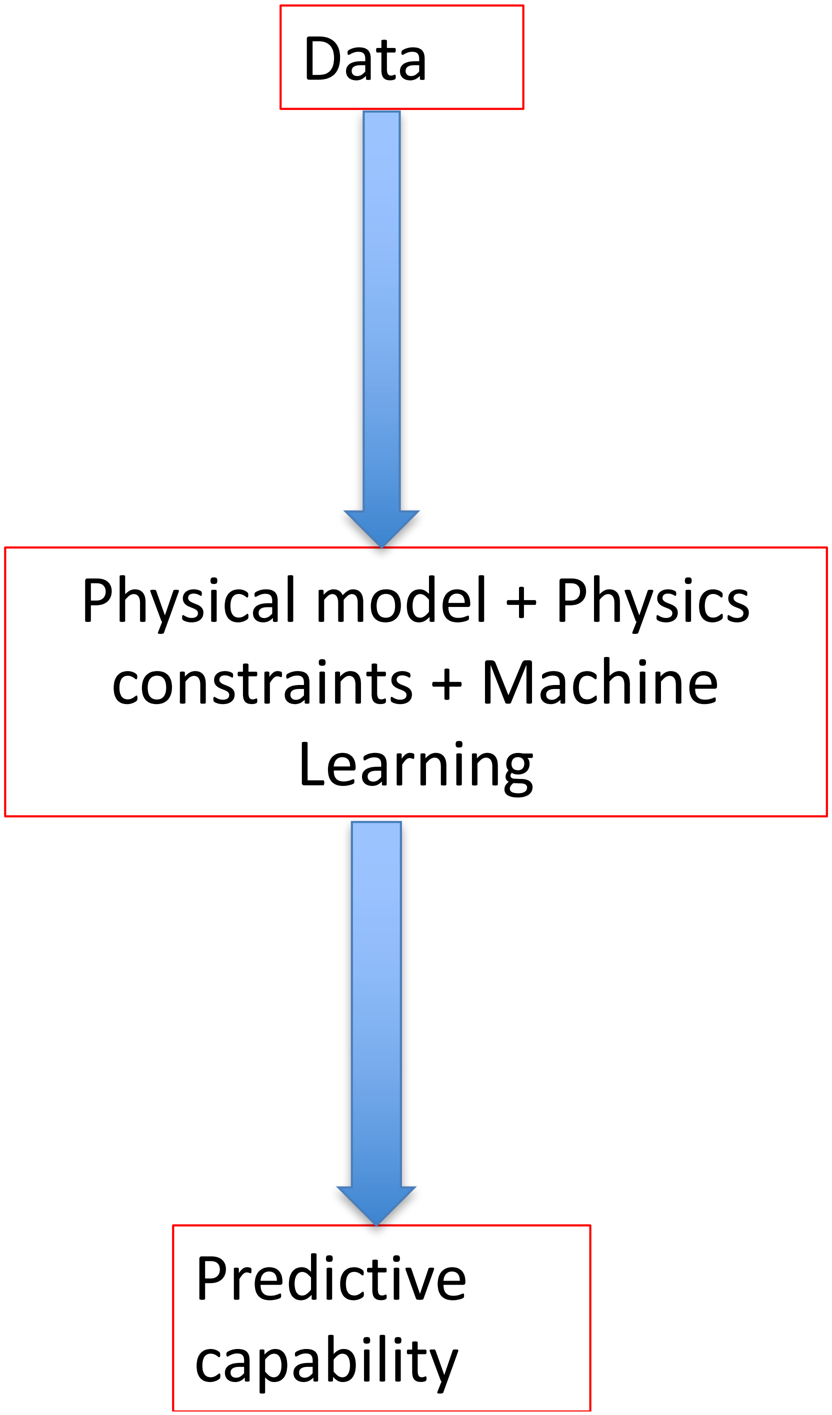


Physical model + Physics constraints + Machine Learning

Predictive capability

$$\frac{\partial \bar{u}}{\partial t} + \mathcal{R}(\bar{u}) = \mathbf{M}(\bar{u}, \bar{v})$$

Challenges



- Data contains real quantities; Model contains "modeled" quantities (loss of consistency is severe in turbulence models)
  ➔ k and in the model are not the k and eps in DNS
- Data will be only loosely connected to model (and not objective)
  ➔ How to improve a turbulence model if we only have pressure measurements (or images)?
- Data will be noisy and of variable quality,
- Inherent uncertainty

Outline

- Introduction
- Proof-of-concept
- How do we setup the data-driven turbulence modeling problem?
- What are the components?
- Demonstration
  - ➔ Predictions in Airfoil flows
- Dissemination / impact
- Vision / Perspectives

Turbulence models

$$\frac{\partial \overline{u_i' u_j'}}{\partial t} = C_{ij} + P_{ij} + V_{ij} + T_{ij} + \Pi_{ij} + D_{ij}$$

$$\frac{\partial \overline{u_i' u_j'}}{\partial t} = C_{ij} + P_{ij} + V_{ij} + \tilde{T}_{ij} + \delta_T + \tilde{\Pi}_{ij} + \delta_\Pi + \tilde{D}_{ij} + \delta_D$$

- One - seven transport eqns, and up to 30 adjustable constants.
- Modeling rests on large amounts of intuition and luck, in spite of starting with a “rigorous” approach
- Theories abound for parts of model, but not for output
- Model constants calibrated on very limited data
- Greater sophistication in RANS models, with mixed degree of success

  ➔ More constants to fit , still use canonical problems

Turbulence modeling discrepancies

$$\frac{\partial \overline{u_i' u_j'}}{\partial t} = C_{ij} + P_{ij} + V_{ij} + T_{ij} + \Pi_{ij} + D_{ij}$$

$$\frac{\partial \overline{u_i' u_j'}}{\partial t} = C_{ij} + P_{ij} + V_{ij} + \tilde{T}_{ij} + \delta_T + \tilde{\Pi}_{ij} + \delta_\Pi + \tilde{D}_{ij} + \delta_D$$

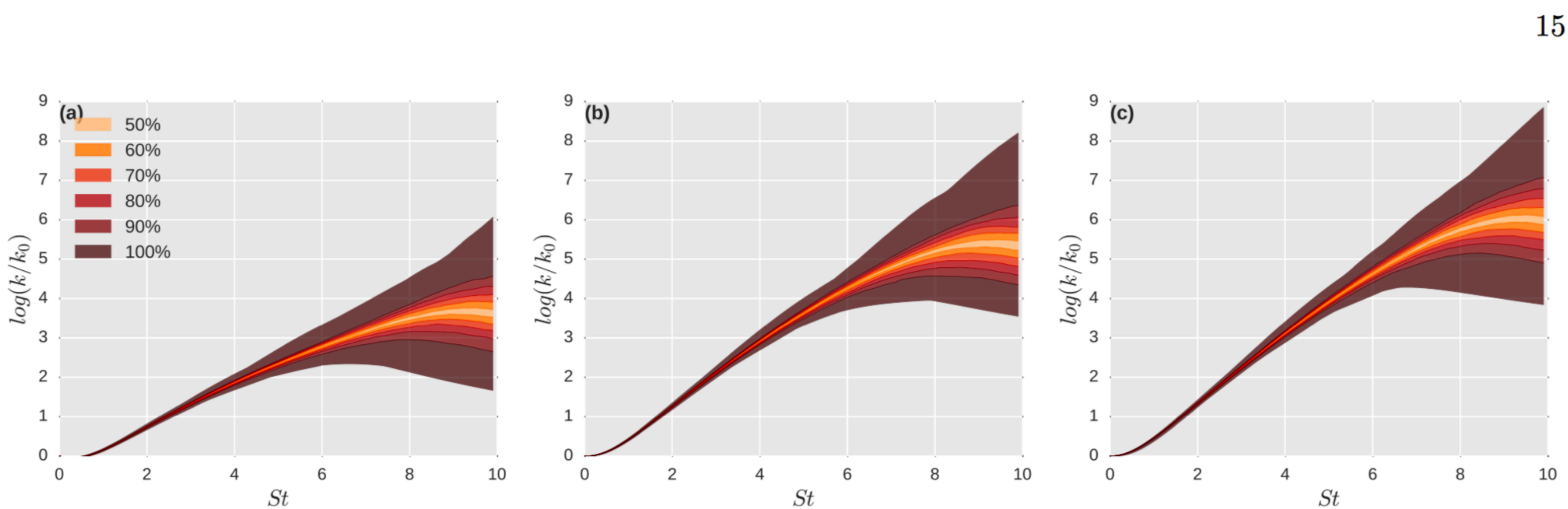
- Balance between the terms matters most (and not accuracy of individual terms)
  - ➔ Still respect invariance, symmetries, etc.
- Many "seemingly physical" quantities are just operational variables
  - ➔ Use of *apriori* analysis is of limited utility
- There is no beautiful turbulence model waiting to be discovered
  - ➔ Look for optimal model, conditional on data & constraints?

Turbulence models – inherent uncertainty

$$\frac{\partial u}{\partial t} + \mathcal{R}(u) = 0$$

$$\bar{u} = \mathcal{P}u$$

Same macrostate, different microstate – irreducible uncertainty



$\frac{Sk_0}{\epsilon_0} = 3$ $\frac{Sk_0}{\epsilon_0} = 15$ $\frac{Sk_0}{\epsilon_0} = 27$

Questions at the beginning of the program

- Is there merit in the general idea?
- How to setup a properly-posed data-driven-turbulence-modeling problem ?
- What are the most effective ways to use Machine Learning approaches?
- What data (and how much data) is needed to improve the predictive capabilities?
- What are the new modeling techniques and algorithms that must be developed to make these approaches a reality?
- What improvements can be shown in a number of flows of interest?
- Once a model has been learned, how is it best embedded in an existing RANS solver?

Outline

- Introduction
- Proof-of-concept
- How do we setup the data-driven turbulence modeling problem?
- What are the components?
- Demonstration
  ➔ Predictions in Airfoil flows
- Dissemination / impact
- Vision / Perspectives

Proof-of-concept

- Basic questions: Can machine learning work at all?:
  - Can a learning algorithm discover and replicate a known model?
  - Will the learned model destabilize a PDE solver?
- Isolate errors in learning from complexities of real-world data

Not just a matter of learning and prediction… Have to address convergence within framework

Proof-of-concept : Replicating Spalart Allmaras Model

$$\mu_t = \rho \hat{\nu} f_{v1}$$

$$\underbrace{\frac{\partial \hat{\nu}}{\partial t} + u_j \frac{\partial \hat{\nu}}{\partial x_j}}_{\text{Convection}} = \underbrace{c_{b1}(1 - f_{t2})\hat{S}\hat{\nu}}_{\text{Production}} - \underbrace{\left(c_{w1} f_w - \frac{c_{b1}}{\kappa^2} f_{t2}\right)\left(\frac{\hat{\nu}}{d}\right)^2}_{\text{Destruction}} + \frac{1}{\sigma}\left(\underbrace{\frac{\partial}{\partial x_j}\left((\nu + \hat{\nu})\frac{\partial \hat{\nu}}{\partial x_j}\right)}_{\text{Diffusion}} + \underbrace{c_{b2}\frac{\partial \hat{\nu}}{\partial x_i}\frac{\partial \hat{\nu}}{\partial x_i}}_{\text{Cross Production}}\right)$$

$$\chi = \hat{\nu}/\nu$$

$$f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3}$$

$$f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}}$$

$$\hat{S} = \Omega + \frac{\hat{\nu}}{\kappa^2 d^2} f_{v2}$$

$$r = min\left[\frac{\hat{\nu}}{\hat{S}\kappa^2 d^2}, 10\right]$$

$$g = r + c_{w2}(r^6 - r)$$

$$f_w = g\left[\frac{1 + c_{w3}^6}{g^6 + c_{w3}^6}\right]^{1/6}$$

$$f_{t2} = c_{t3} exp(-c_{t4}\chi^2)$$

$$W_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i}\right)$$

$$\Omega = \sqrt{2 W_{ij} W_{ij}}$$

Proof-of-concept : Replicating Spalart Allmaras Model

$$\underbrace{\frac{\partial \hat{\nu}}{\partial t} + u_j \frac{\partial \hat{\nu}}{\partial x_j}}_{\text{Convection}} = \text{[redacted]} + \frac{1}{\sigma}\left(\underbrace{\frac{\partial}{\partial x_j}\left((\nu + \hat{\nu}) \frac{\partial \hat{\nu}}{\partial x_j} \right)}_{\text{Diffusion}} + \text{[redacted]} \right)$$

Convection Production Destruction Diffusion Cross Production

Locally Non-Dimensional Input Features

$$\chi = \hat{\nu}/\nu$$

$$\bar{\Omega} = \frac{d^2}{\hat{\nu} + \nu}\Omega$$

$$\bar{N} = \frac{d^2}{(\hat{\nu} + \nu)^2} N$$

Locally Non-Dimensional Outputs

$$s_p = c_{b1}(1 - f_{t2})\hat{S}\hat{\nu}$$

$$s_d = \left(c_{w1} f_w - \frac{c_{b1}}{\kappa^2} f_{t2} \right) \left(\frac{\hat{\nu}}{d} \right)^2$$

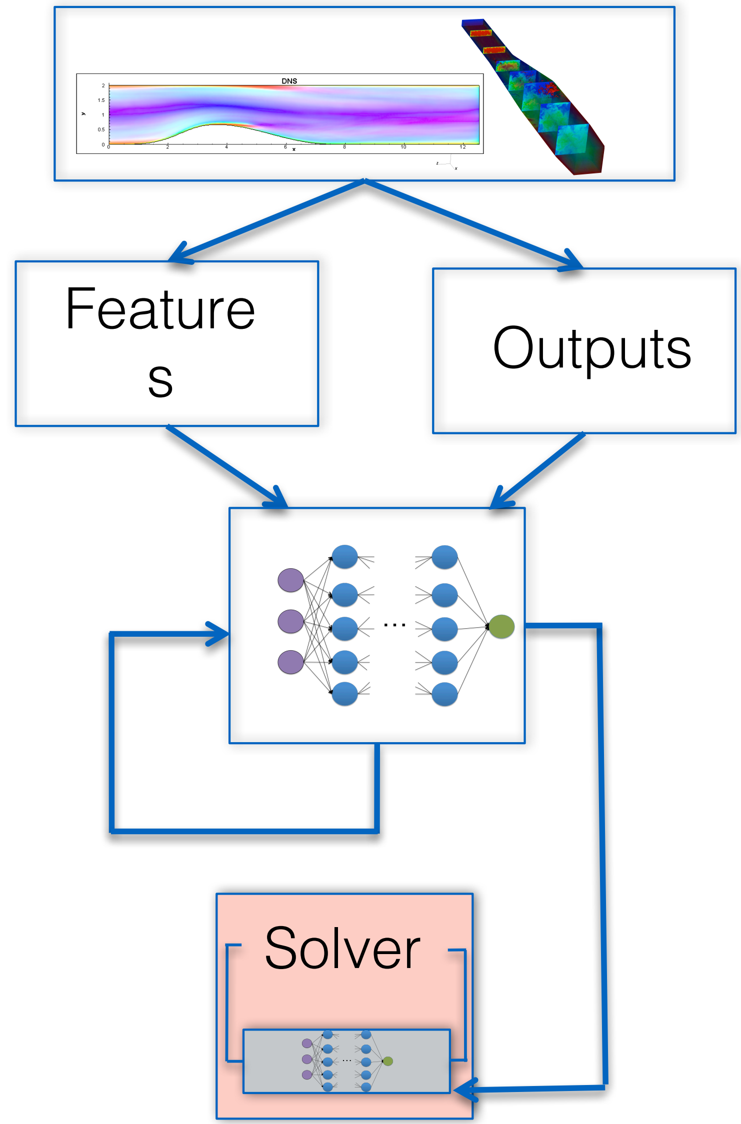
$$s_{cp} = \frac{c_{b2}}{\sigma} \frac{\partial \hat{\nu}}{\partial x_i} \frac{\partial \hat{\nu}}{\partial x_i}$$

$$s = s_p + s_d + s_{cp}$$

$$\bar{s}_i = \left(\frac{d}{\hat{\nu}} \right)^2 s_i$$

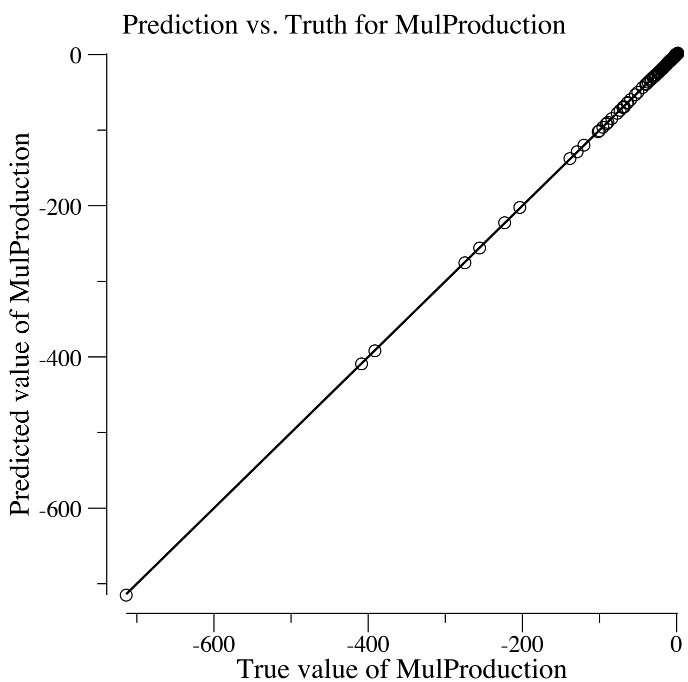
Procedure

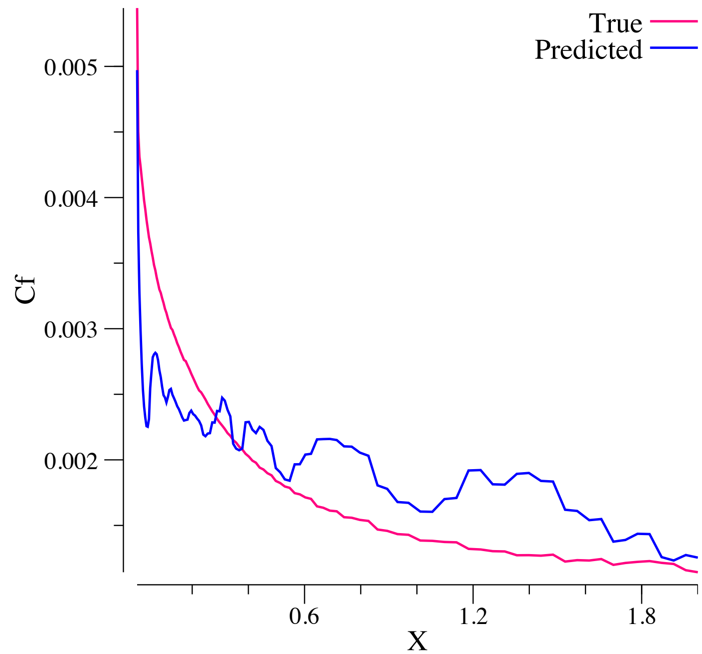
1) Select representative datasets
   - Flat plates, pressure-driven channels, airfoils
2) Choose and extract input and output features
   - Spalart-Allmaras quantities
3) Select learning algorithm
   - Neural network
4) Train learning algorithm
   - BFGS optimizer
5) Embed learned model within flow solver
   - SU2



We can learn and we can test, but …

- Favorable pressure gradient channel flow





- **Injection within a converging solver yields poor results**

The loss function

- Squared-Error

$$L = \sum_{i=1}^{k} (p_i - t_i)^2$$

  - Penalizes differences in the output value

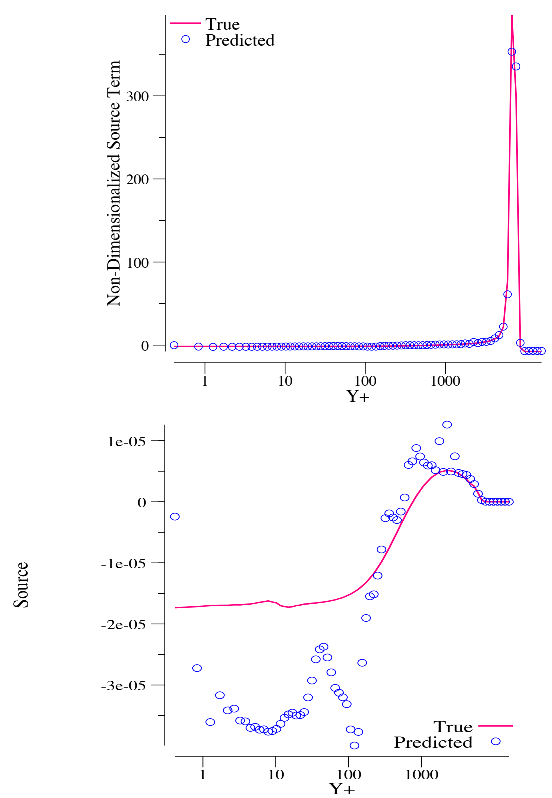
- Dimensionalized Squared-Error

$$L_2 = \sum_{i=1}^{k} \left(\left(\frac{d_i^2}{(\hat{\nu}_i + \nu_i)^2} \right) p_{\bar{s},i} - t_{s,i} \right)^2$$

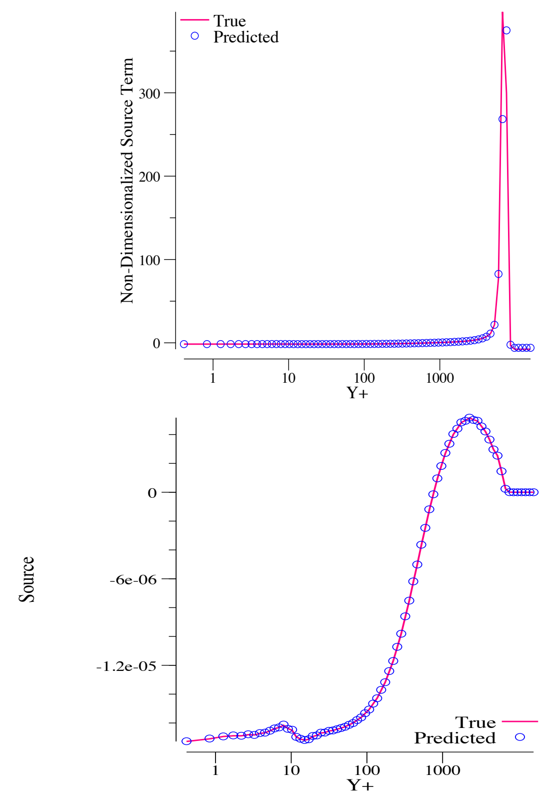
  - Penalizes differences in the dimensional output value

The loss function

Squared-distance Loss



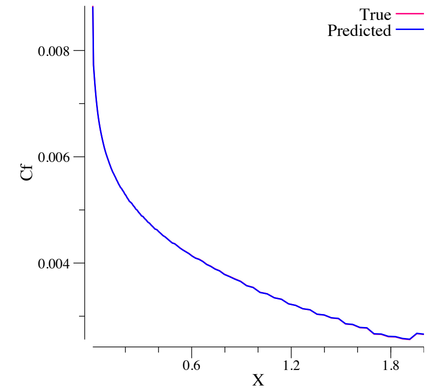
Dimensionalized Loss



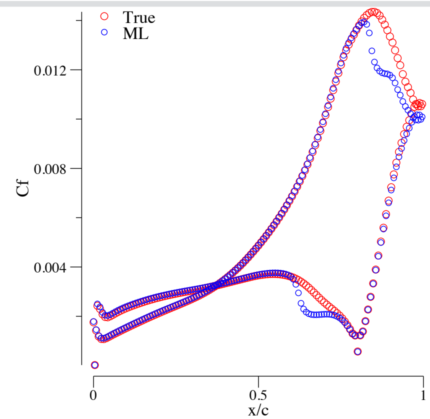
- **Must align loss function with CFD environment**

Test cases

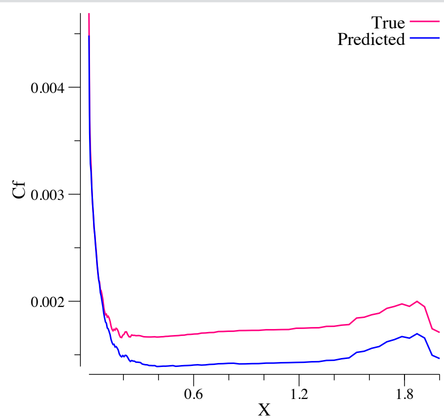
- G: No major difference



- F: Small region of difference



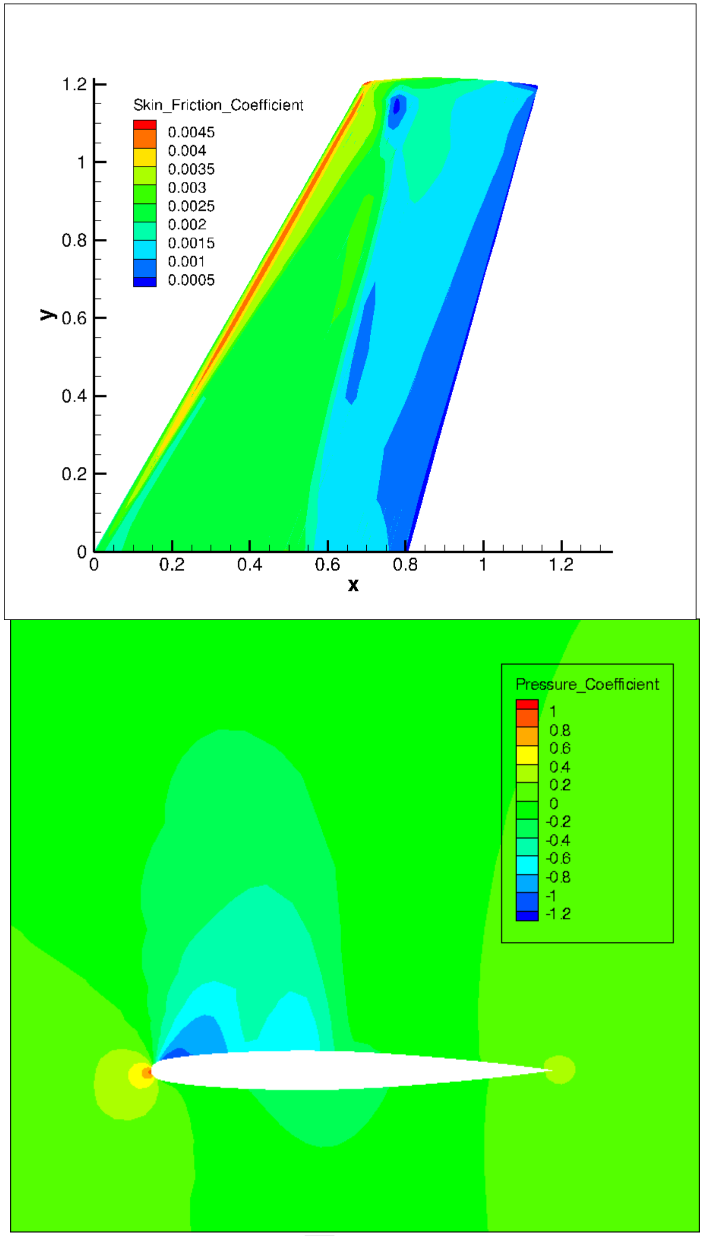
- P: Large discrepancy



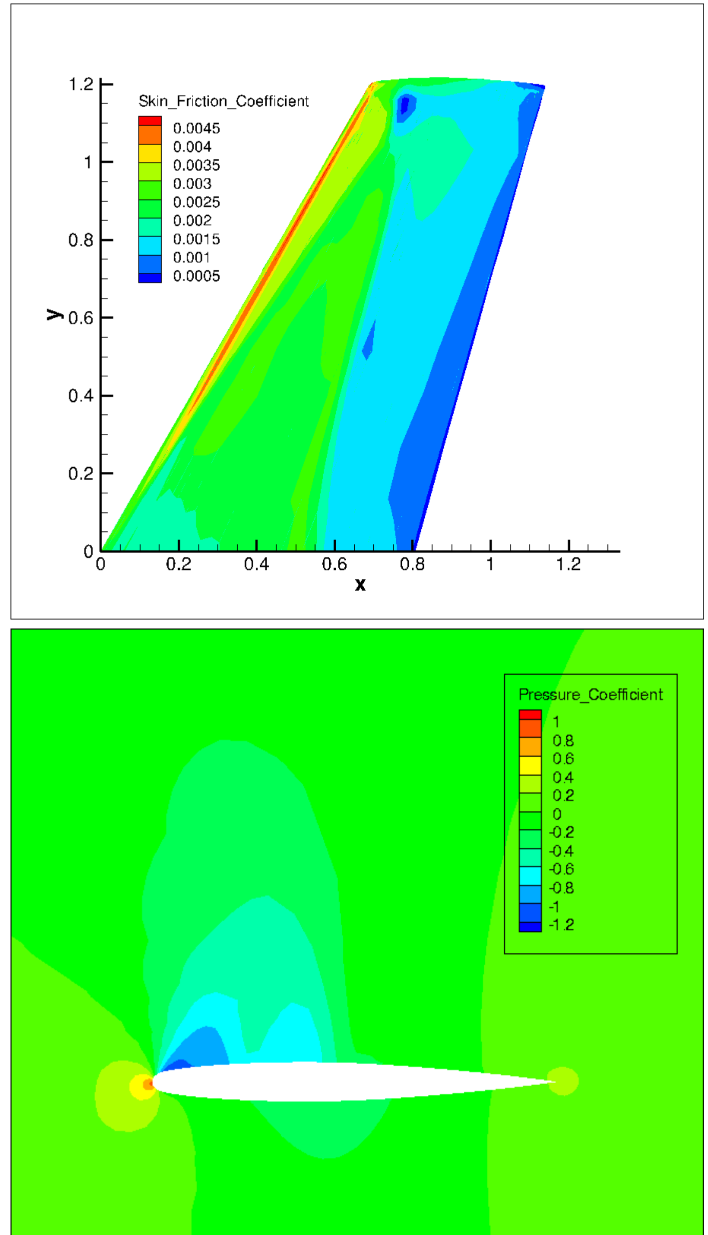
	Dest.	F_w	Mul. Dest.	Mul. Prod.	Prod.	Source
Flatplate 3e6	G	G	G	G	G	G
Flatplate 4e6	G	G	G	G	G	G
Flatplate 5e6	G	G	G	G	G	G
Flatplate 6e6	G	G	G	G	G	G
Flatplate 7e6	G	G	G	G	G	G
Channel $C_p = -0.3$	G	G	G	G	G	F
Channel $C_p = -0.1$	G	G	G	G	G	F
Channel $C_p = -0.03$	G	G	G	G	G	F
Channel $C_p = -0.01$	G	G	G	G	G	F
Channel $C_p = 0.01$	G	G	G	G	G	F
Channel $C_p = 0.03$	G	G	G	G	G	F
Channel $C_p = 0.1$	G	G	G	G	G	F
Channel $C_p = 0.3$	P	G	G	G	P	F
NACA 0	G	G	G	G	G	G
NACA 1	G	G	G	G	G	G
NACA 2	G	G	G	G	G	G
NACA 3	G	G	G	G	G	G
NACA 4	G	G	G	G	G	G
NACA 5	G	G	G	G	G	G
NACA 6	G	G	G	G	G	G
NACA 7	G	G	G	G	G	G
NACA 8	G	G	G	F	G	G
NACA 9	G	G	G	F	G	G
NACA 10	G	G	G	F	G	G
NACA 11	G	G	G	F	G	G
NACA 12	G	G	G	F	G	G

450+ cases

Test on 3D problem



True



ML

Takeaways

- Feature Scaling is important
- Testing within the CFD solver
- Alignment of loss function

If there is an underlying model, it is possible to discover it

Tracey, Brendan & Duraisamy, Karthik, & Alonso, Juan J. A Machine Learning Strategy to Assist Turbulence Model Development, Proc. AIAA SciTech, Kissimmee, FL 2015

Outline

- Introduction
- Proof-of-concept
- How do we setup the data-driven turbulence modeling problem?
- What are the components?
- Demonstration
  - ➔ Predictions in Airfoil flows
- Dissemination / impact
- Vision / Perspectives

Field Inversion & Machine learning (FIML)

Datasets $Y^1, Y^2 \ldots Y^n$

Field Inversion

$$\frac{DQ}{Dt} = R(Q) + \delta^j(x) \;:\; \min_{\delta^j(x)} ||Y^j - Y^j(Q)||$$

Information

Spatial discrepancy

Field Inversion & Machine learning (FIML)

Datasets $Y^1, Y^2 \ldots Y^n$

Field Inversion

$$\frac{DQ}{Dt} = R(Q) + \delta^j(x) \quad : \quad \min_{\delta^j(x)} ||Y^j - Y^j(Q)||$$

Information | Spatial discrepancy

$$\delta^1(x), \delta^2(x), \ldots \delta^n(x)$$

Machine Learning

Knowledge | Functional discrepancy

$$\hat{\delta}(\, f(Q) \,)$$

Field Inversion & Machine learning (FIML)

Datasets $Y^1, Y^2 \ldots Y^n$

Field Inversion

$$\frac{DQ}{Dt} = R(Q) + \delta^j(x) \;:\; \min_{\delta^j(x)} ||Y^j - Y^j(Q)||$$

Information | Spatial discrepancy

$$\delta^1(x), \delta^2(x), \ldots \delta^n(x)$$

Machine Learning

Knowledge | Functional discrepancy

$$\hat{\delta}(\,f(Q)\,)$$

Embedding

$$\frac{DQ}{Dt} = R(Q) + \hat{\delta}(f(Q))$$

Prediction : Injection into solver

Major insight from NASA LEARN project

Information Spatial discrepancy

$$\delta^1(x), \delta^2(x), \ldots \delta^n(x)$$

Machine Learning

Knowledge Functional discrepancy

$$\hat{\delta}(\, f(Q)\ )$$

How does it address the challenges?

Data

↓

Physical model + Inference

↓

Machine learning

↓

Physical model + consistent augmentation

↓

Predictive capability

- Data contains real quantities; Model contains “modeled” quantities (loss of consistency is bad in turbulence models)

  ➔ Inference connects real quantities to modeled ones

- Data will be only loosely connected to model (and not objective)

  ➔ Inference connects secondary, non-objective data to model quantites

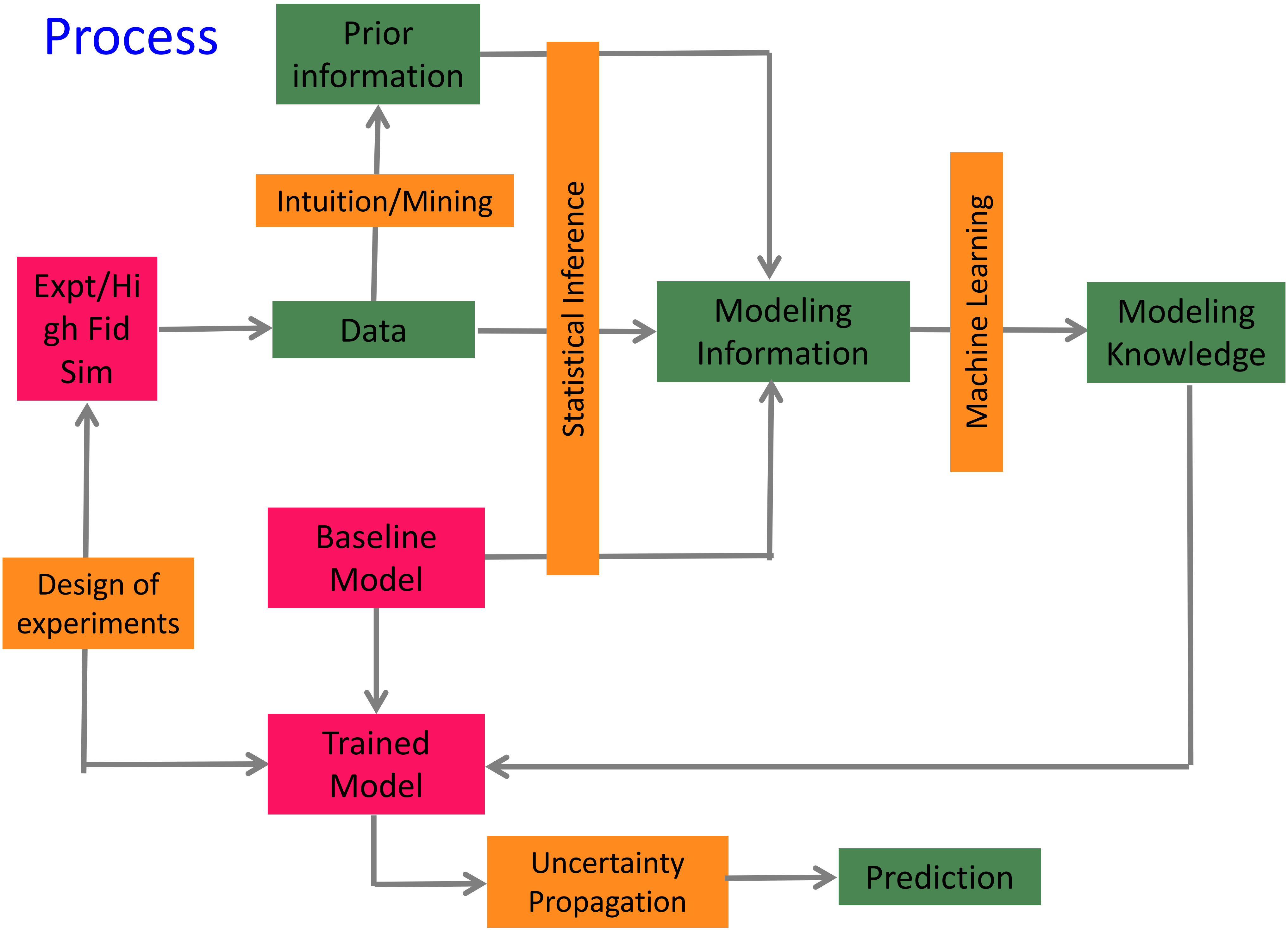
- Data will be noisy and of variable quality, inherent uncertainty

  ➔ Probabilistic casting of inference and learning

Outline

- Introduction
- Proof-of-concept
- How do we setup the data-driven turbulence modeling problem?
- What are the components?
- Demonstration
  - ➔ Predictions in Airfoil flows
- Dissemination / impact
- Vision / Perspectives

Process

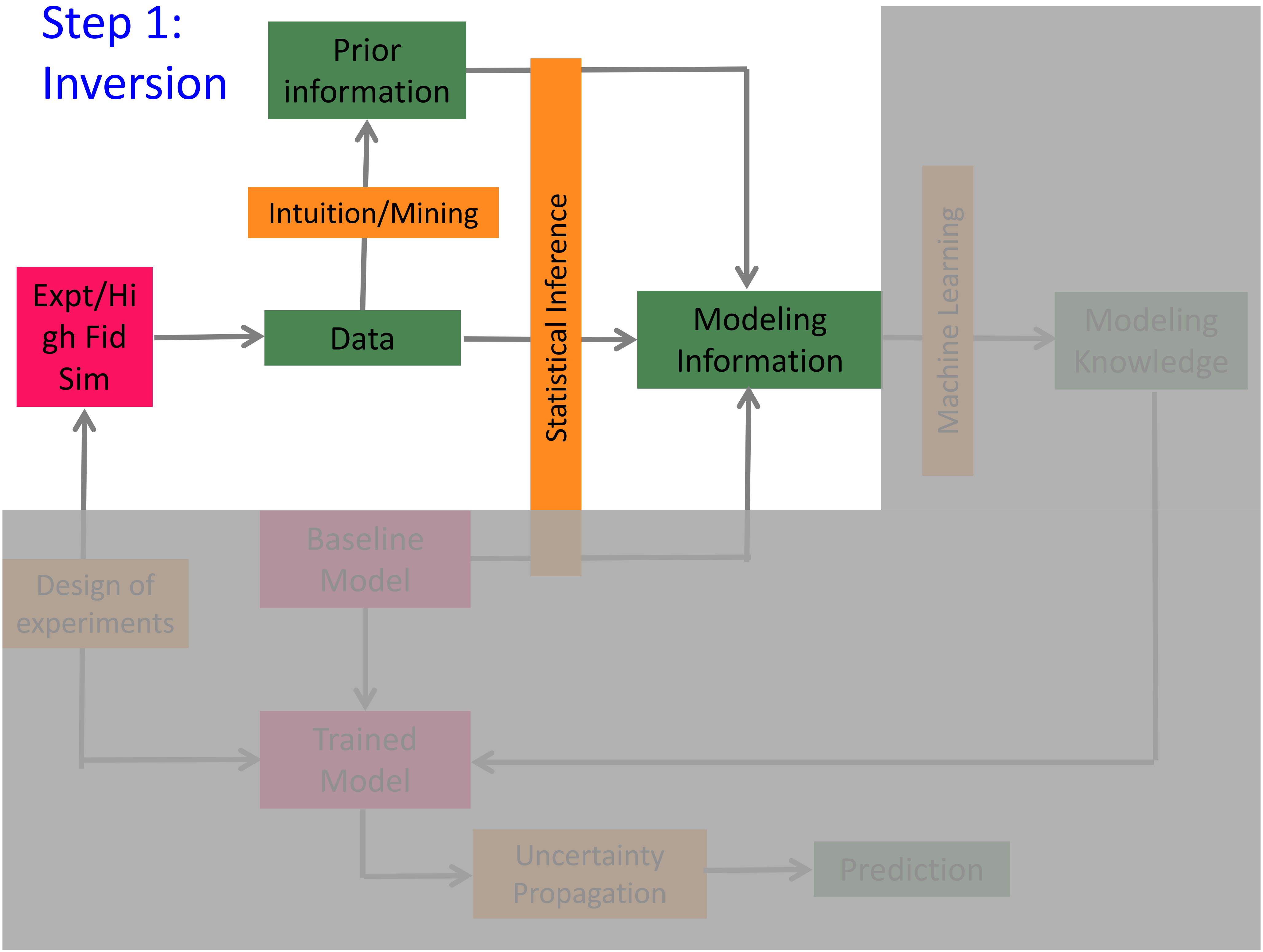

Prior information
Intuition/Mining
Expt/High Fid Sim
Data
Statistical Inference
Modeling Information
Machine Learning
Modeling Knowledge
Baseline Model
Design of experiments
Trained Model
Uncertainty Propagation
Prediction

1) Inference

3) Machine Learning

2) Design of Experiments

4) Prediction



Step 1:
Inversion
Prior information
Intuition/Mining
Expt/High Fid Sim
Data
Statistical Inference
Modeling Information
Machine Learning
Modeling Knowledge
Baseline Model
Design of experiments
Trained Model
Uncertainty Propagation
Prediction

Introducing discrepancies

$$\frac{D\omega}{Dt} = P_\omega - \beta(x) D_\omega + T_\omega$$

Singh & Duraisamy, PoF 2016

Parish & Duraisamy, Aviation 2014

$$\frac{DR_{ij}}{Dt} = C_{ij} + P_{ij} + T_{ij} + \Pi_{ij} + D_{ij} + \beta(x)_{ij} \epsilon_{ij}$$

$$\frac{DR_{ij}}{Dt} = \beta(x)_{ij} a_o \omega (R_{ij,eq} - R_{ij})$$

Singh & Duraisamy, Scitech 2016

$$\mathbf{R}_p = 2k \left[\frac{\mathbf{I}}{3} + \mathbf{V}(\Lambda + \vec{\beta}(x))\mathbf{V}^T \right]$$

Duraisamy, SIAM 2016

Bayesian FUNCTIONAL Inversion

$$\beta_{map} = arg\ \min \frac{1}{2}\left[\left(\mathbf{d} - h(\beta)\right)^T \mathbf{C_m}^{-1}\left(\mathbf{d} - h(\beta)\right) + \left(\beta - \beta_{prior}\right)^T \mathbf{C}_\beta^{-1}\left(\beta - \beta_{prior}\right)\right]$$

d – Data

β - Unknown function

h(β) – Model output

C_m - Observational covariance

C_β - Prior covariance

Parish, Eric & Duraisamy, Karthik, A paradigm for data-driven predictive modeling using field inversion and machine learning, Journal of Computational Physics, Volume 305, 15 January 2016, Pages 758–774 2016

Posterior

$$\mathbf{C}_{posterior} = \left[\frac{d^2\mathfrak{J}(\boldsymbol{\beta})}{d\boldsymbol{\beta}d\boldsymbol{\beta}}\right]^{-1}\Bigg|_{\boldsymbol{\beta}_{MAP}}$$

$$H_{ij} = \frac{\partial^2 \mathfrak{J}}{\partial \boldsymbol{\beta}_i \partial \boldsymbol{\beta}_j} + \psi_m \frac{\partial^2 R_m}{\partial \boldsymbol{\beta}_i \partial \boldsymbol{\beta}_j} + \mu_{i,m} \frac{\partial R_m}{\partial \boldsymbol{\beta}_j} + \nu_{i,m} \frac{\partial^2 \mathfrak{J}}{\partial u_n \partial \boldsymbol{\beta}_j} + \nu_{i,n} \psi_m \frac{\partial^2 R_m}{\partial u_n \partial \boldsymbol{\beta}_j}$$

where,

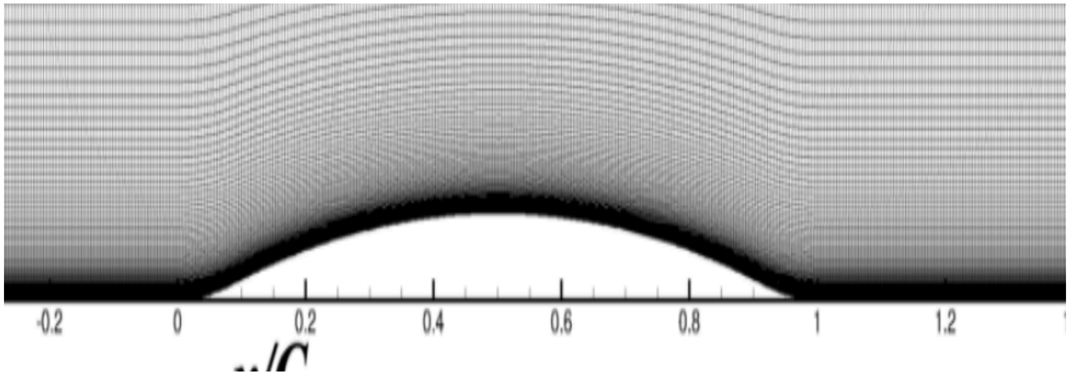
$$\nu_{i,n} \frac{\partial R_m}{\partial u_n} = -\frac{\partial R_m}{\partial \boldsymbol{\beta}_i}$$

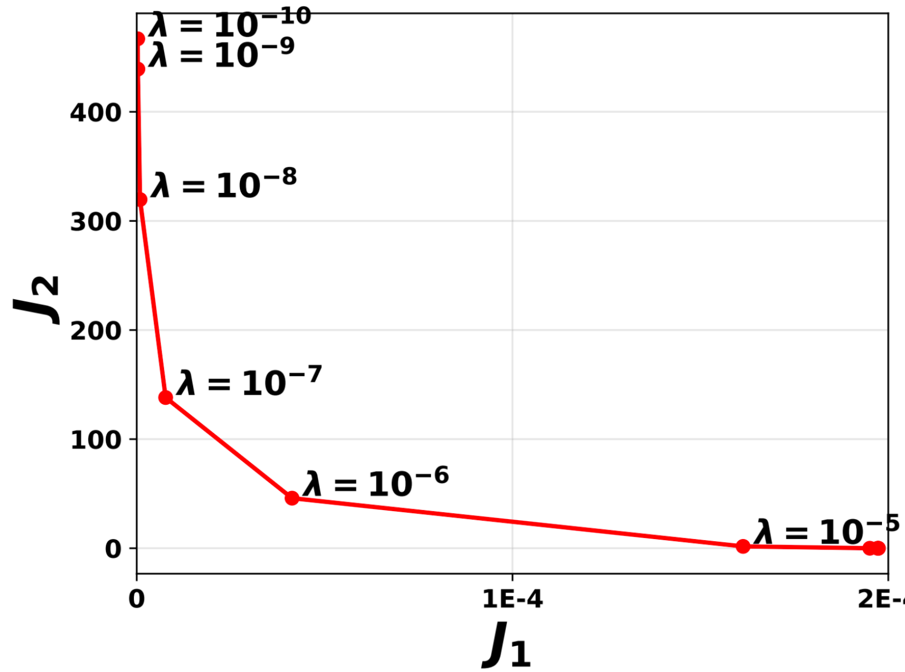
$$\mu_{i,m} \frac{\partial R_m}{\partial u_k} = -\frac{\partial^2 F}{\partial \boldsymbol{\beta}_i \partial u_k} - \psi_m \frac{\partial^2 R_m}{\partial \boldsymbol{\beta}_i \partial u_k} - \nu_{i,n} \frac{\partial^2 \mathfrak{J}}{\partial u_n \partial u_k} - \nu_{i,n} \psi_m \frac{\partial^2 R_m}{\partial u_n \partial u_k}$$

An approximate Hessian computation is additionally used for ill-posed problems

More complete PDFs with accelerated MCMC (with P. Constantine, Colorado Sc. Of Mines)

Example 1: Flow over a bump – Field inversion

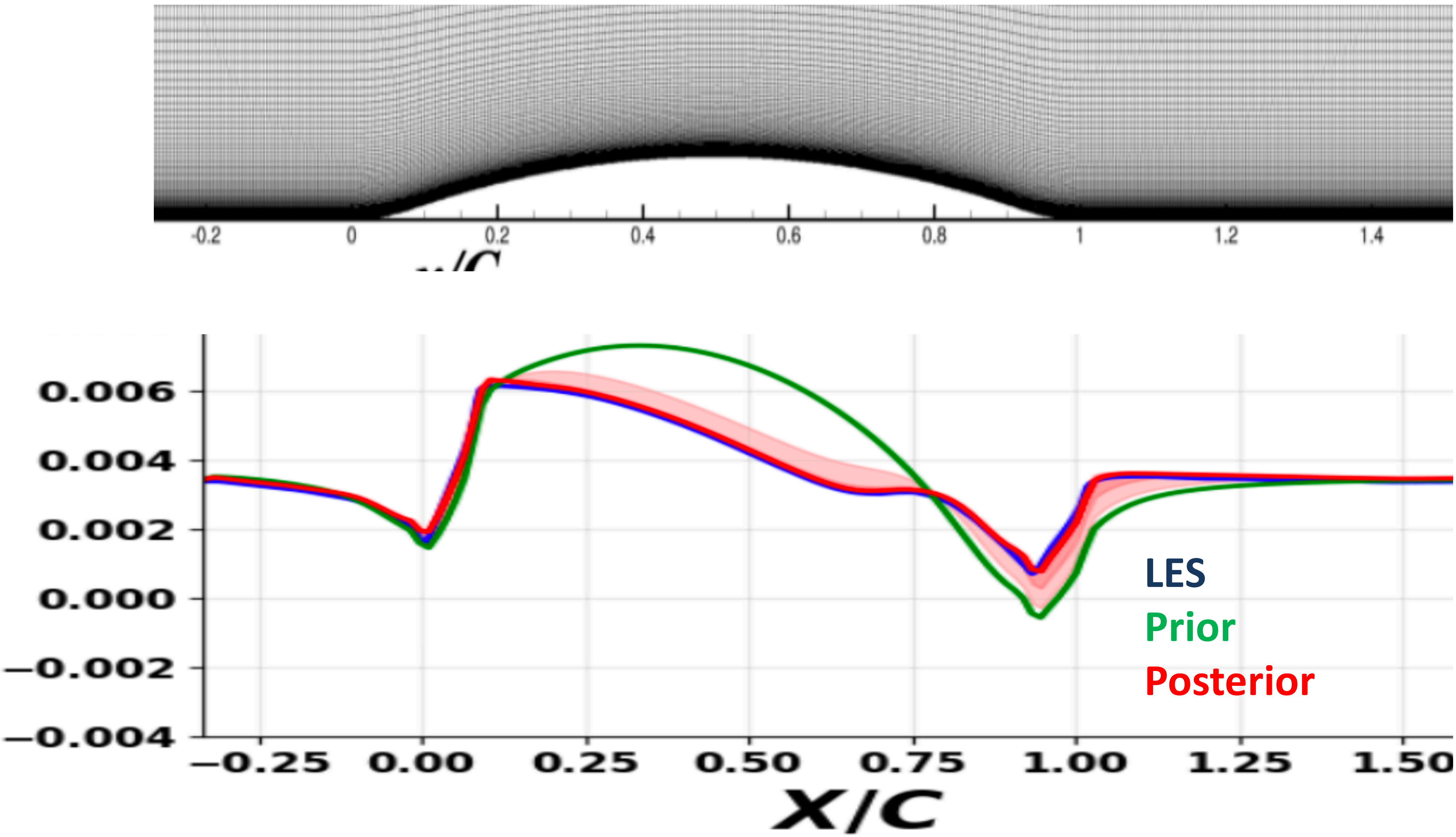




$$\frac{D\omega}{Dt} = \beta(\boldsymbol{x})P(k,\omega,\mathbf{U}) - D(k,\omega,\mathbf{U}) + T(k,\omega,\mathbf{U}).$$

$$\min_{\beta} \; J_1 + \lambda J_2 \equiv \min_{\beta} \sum_{j=1}^{N_d} [G_{j,d} - G_j(\beta)]^2 + \lambda \sum_{n=1}^{N_m} [\beta(x_n) - 1]^2$$

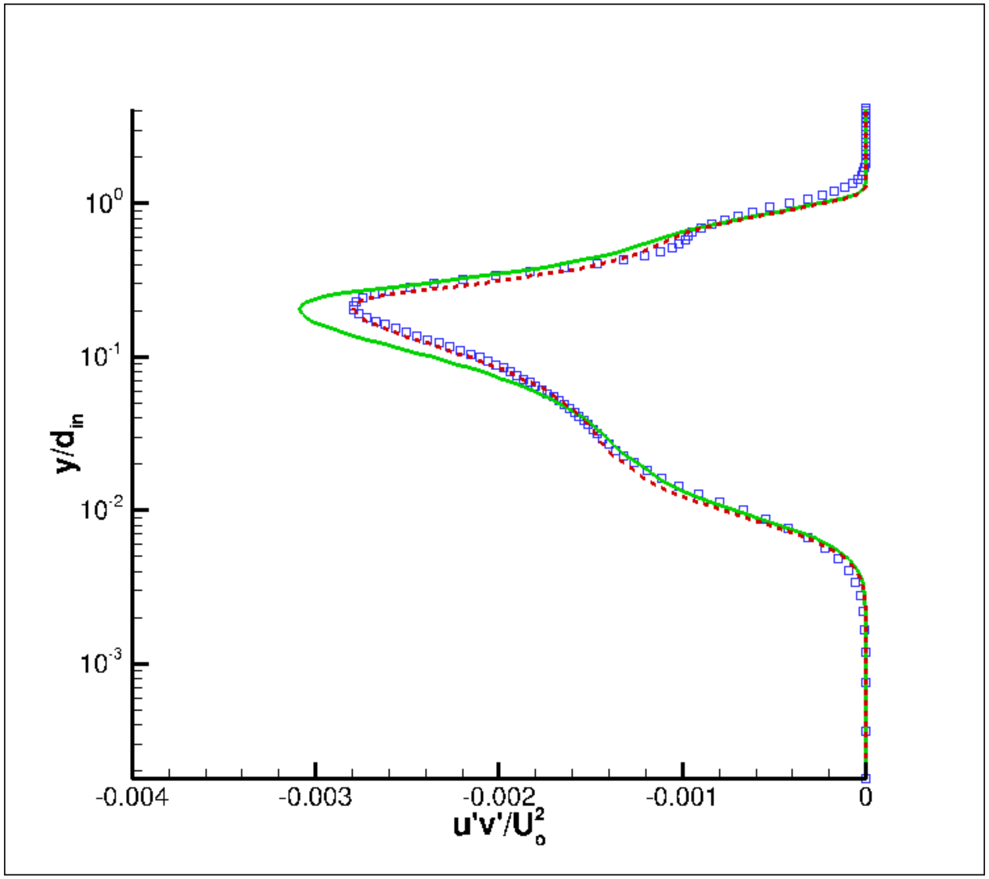
Inferred quantity - Cf

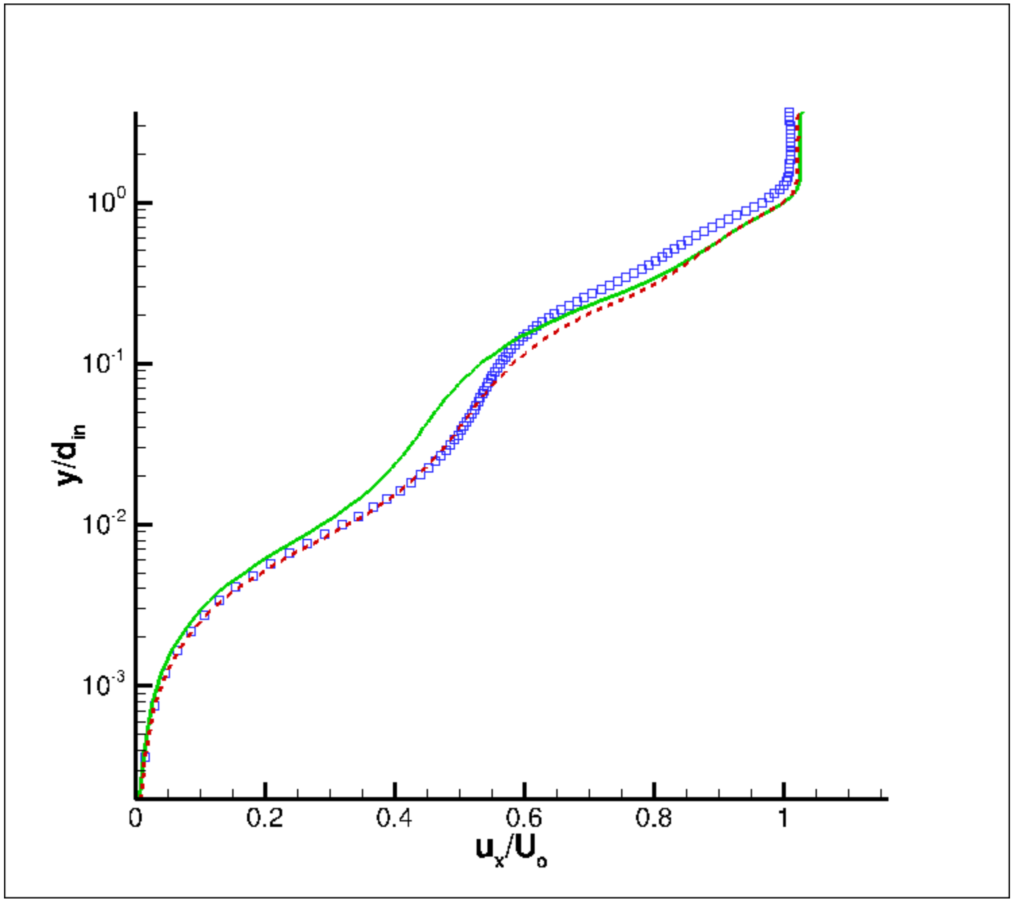


Secondary quantities

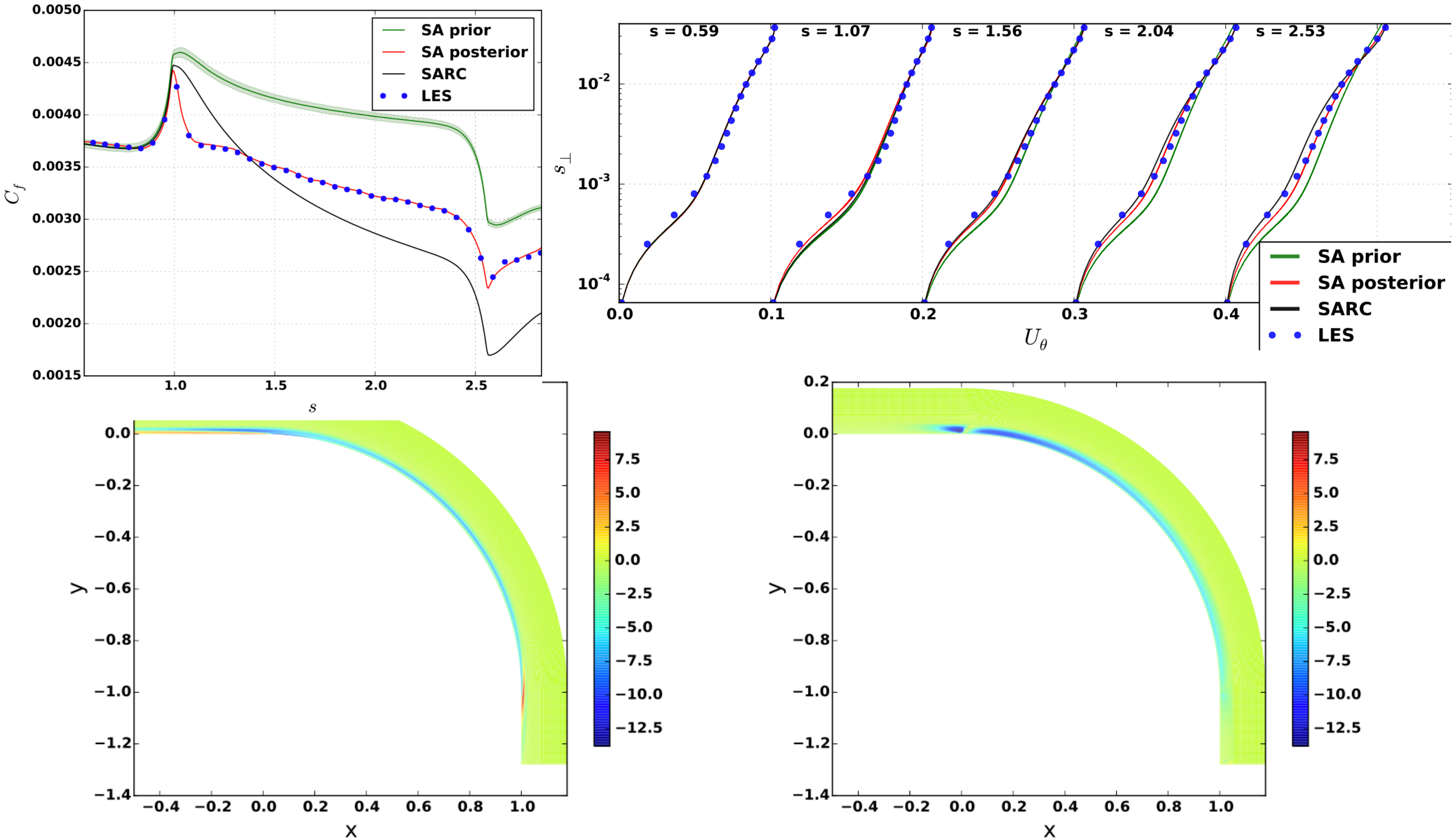
LES
Prior
Posterior

Data-driven augmentation of turbulence models for adverse pressure gradient flows AP Singh, R Matai, K Duraisamy, P Durbin, Proc. AIAA Aviation 2017



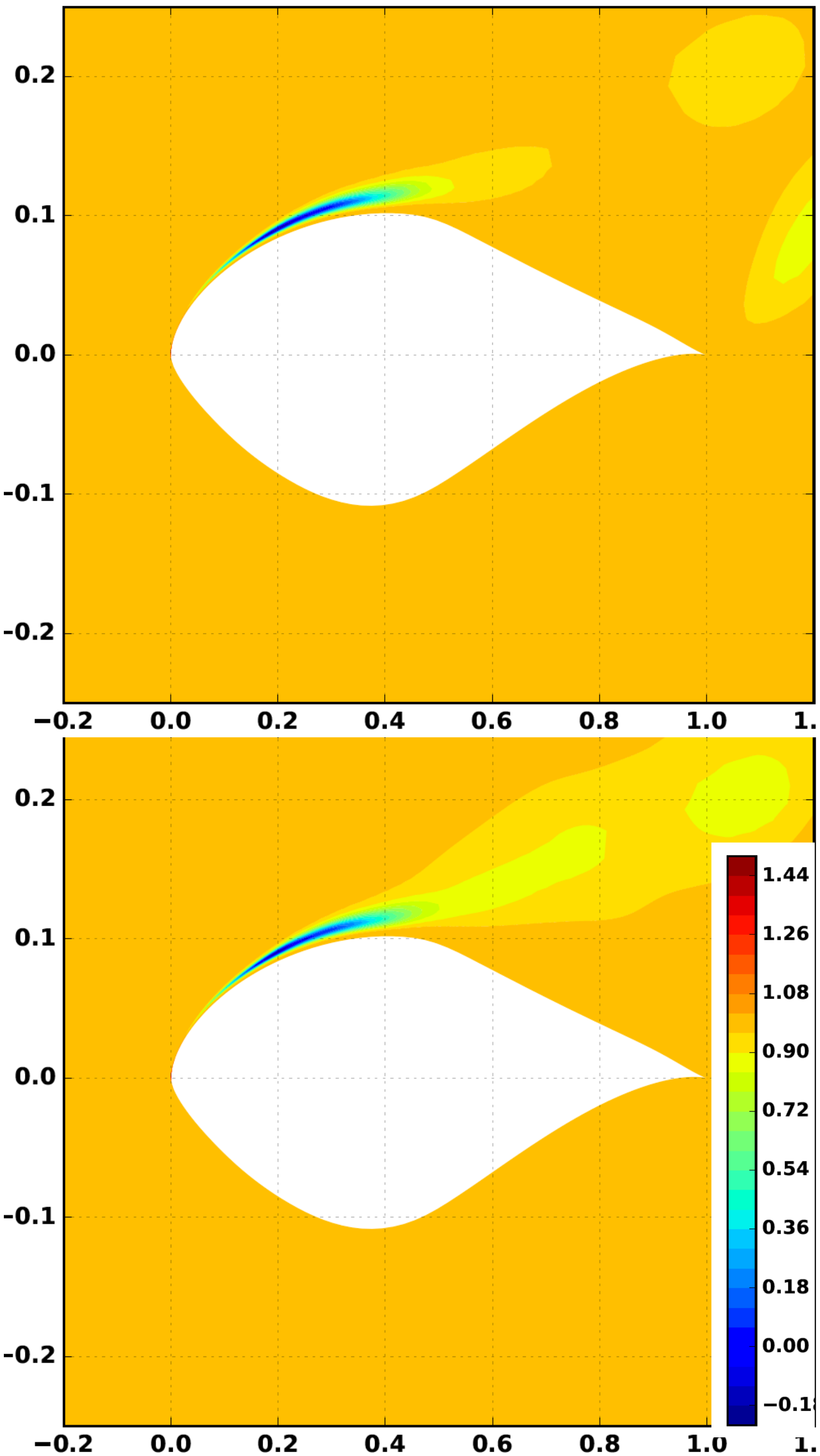


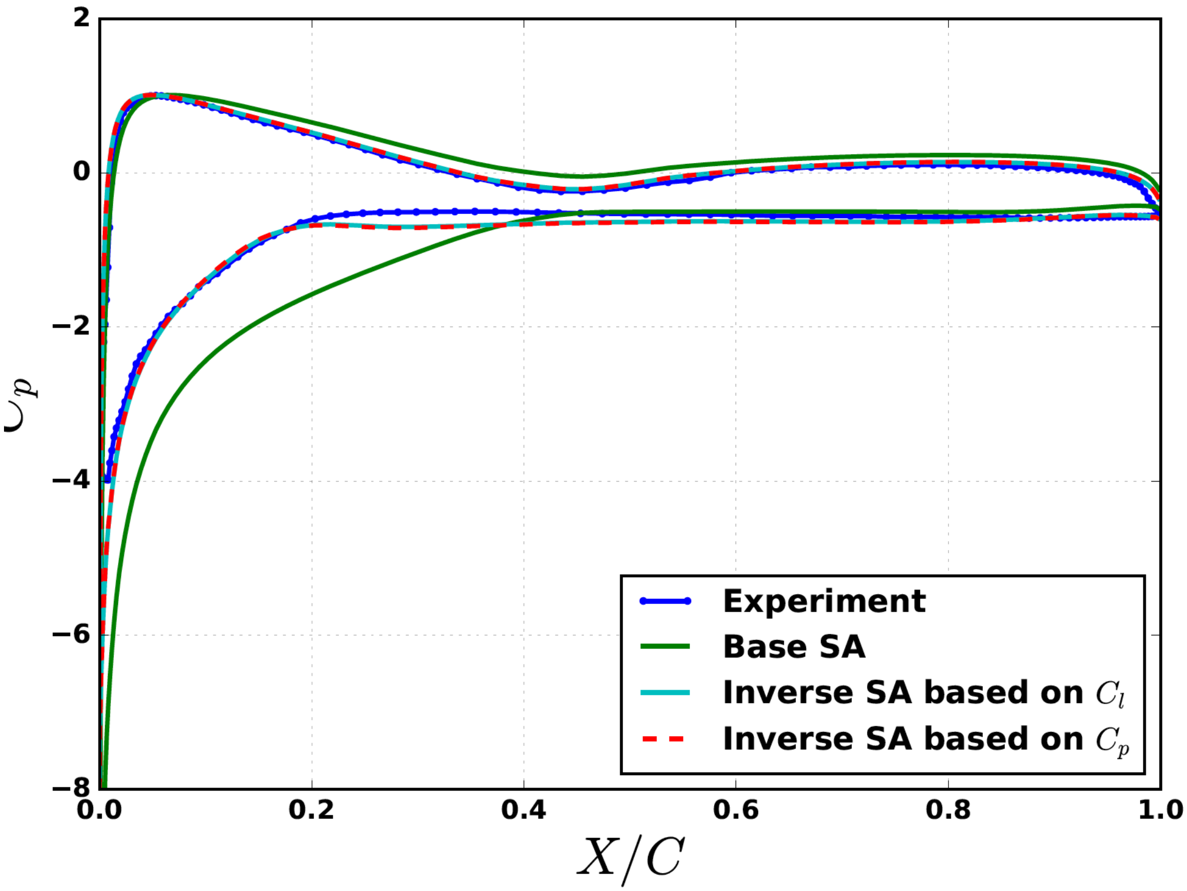
Example 2: Curved channel



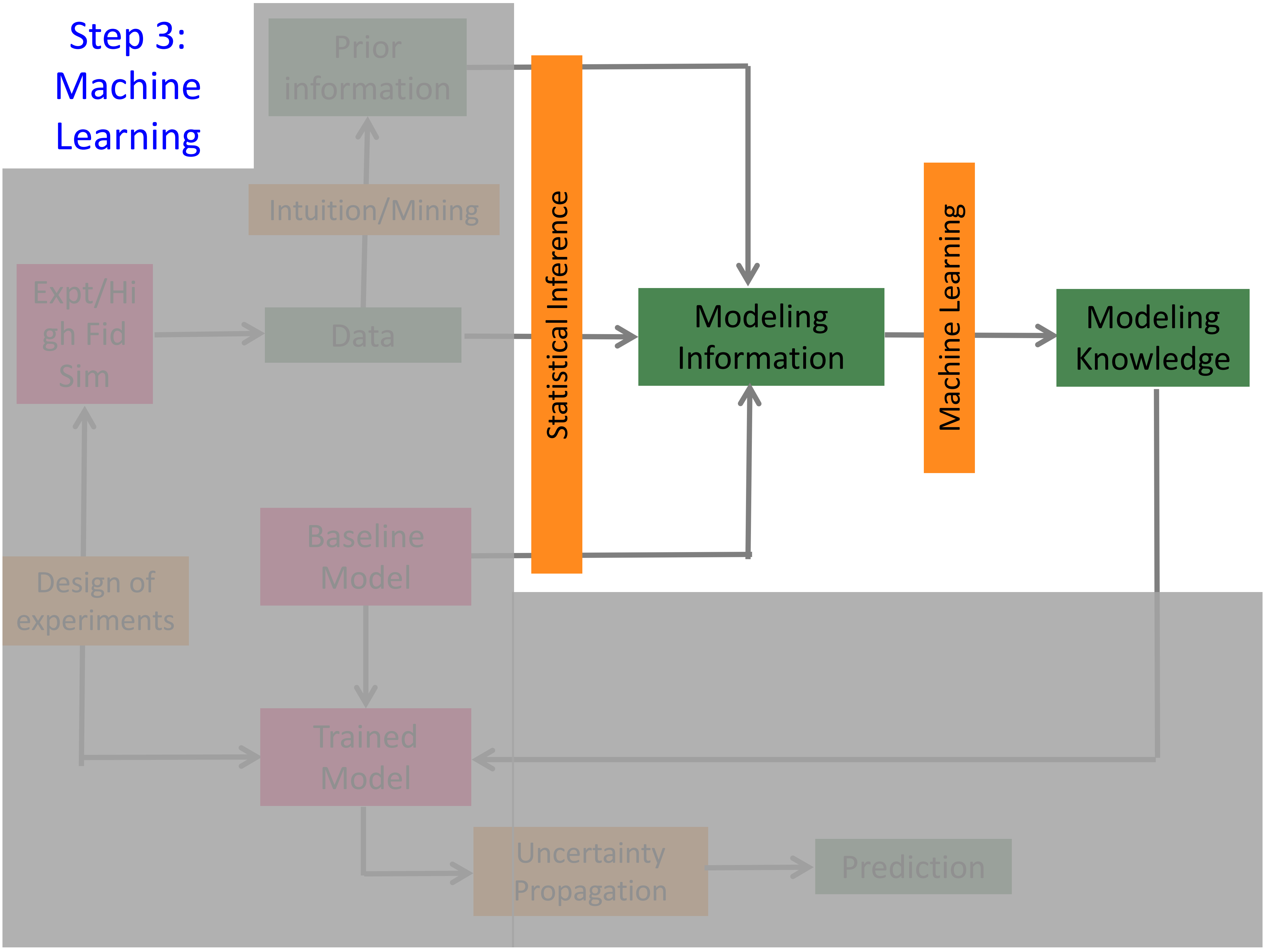
Singh, A.P. & Duraisamy, K. [Using Field Inversion to Quantify Functional Errors in Turbulence Closures,](#) Phys. Fluids 2016

Example 3: Airfoil
Inversion based on Pressures
vs
Inversion based on LIFT!





Ability to work on sparse amount of data is critical



Step 3:
Machine
Learning
Prior information
Intuition/Mining
Expt/High Fid Sim
Data
Statistical Inference
Modeling Information
Machine Learning
Modeling Knowledge
Baseline Model
Design of experiments
Trained Model
Uncertainty Propagation
Prediction

How to transform information to knowledge?

$$\beta^1(x,y)$$
$$\beta^2(x,y)$$
$$\beta^3(x,y)$$
$$\beta^4(x,y)$$
$$\ldots\ldots$$
$$\ldots\ldots$$
$$\beta^n(x,y)$$

$\Rightarrow$ **Machine Learning** $\Rightarrow$

$$\eta_1, \eta_2, \ldots$$
$$\beta(\eta_1, \eta_2, \ldots)$$

Selection of Features

Step 1: Look inside the baseline model

$$\chi = \hat{\nu}/\nu \qquad \bar{\Omega} = \frac{d^2}{\hat{\nu}+\nu}\Omega$$

$$\bar{s}_p = \frac{d^2}{(\hat{\nu}+\nu)^2} s_p = c_{b1}(1-f_{t2})\left(\frac{\chi}{\chi+1}\right)\left(\bar{\Omega} + \frac{1}{\kappa^2}\frac{\chi}{\chi+1} f_{t2}\right)$$

$$\bar{s}_d = \frac{d^2}{(\hat{\nu}+\nu)^2} s_d = \left(\frac{\chi}{\chi+1}\right)^2 c_{w1} f_w \;\; ,$$

Step 2: Look for relevant physics

$$S/\Omega, \Pi, s_p/s_d$$

Step 3: Feature-subset selection*

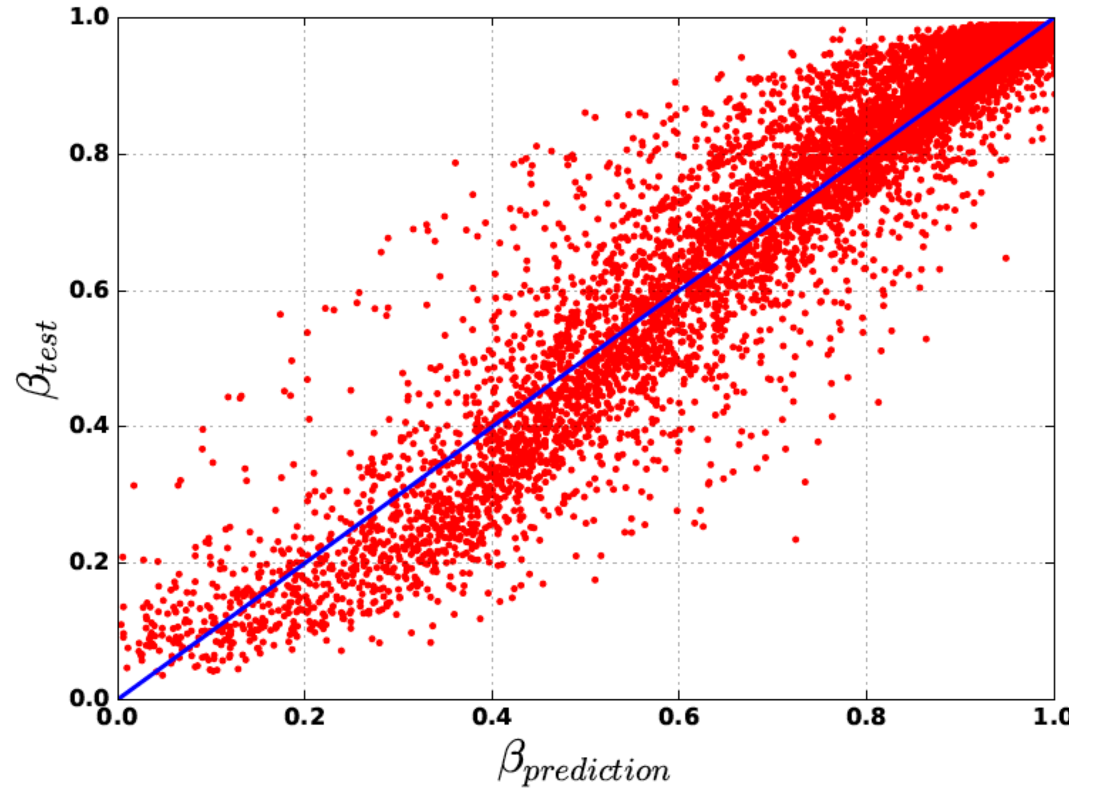
Hill-climbing algorithm

Kohavi, R. et al. "Wrappers for Feature Subset Selection," Artificial Intelligence, 1997

Features locally non-dimensional

Evaluation

Neural Networks
GP regression
*Multiscale GP regression**
Symbolic regression

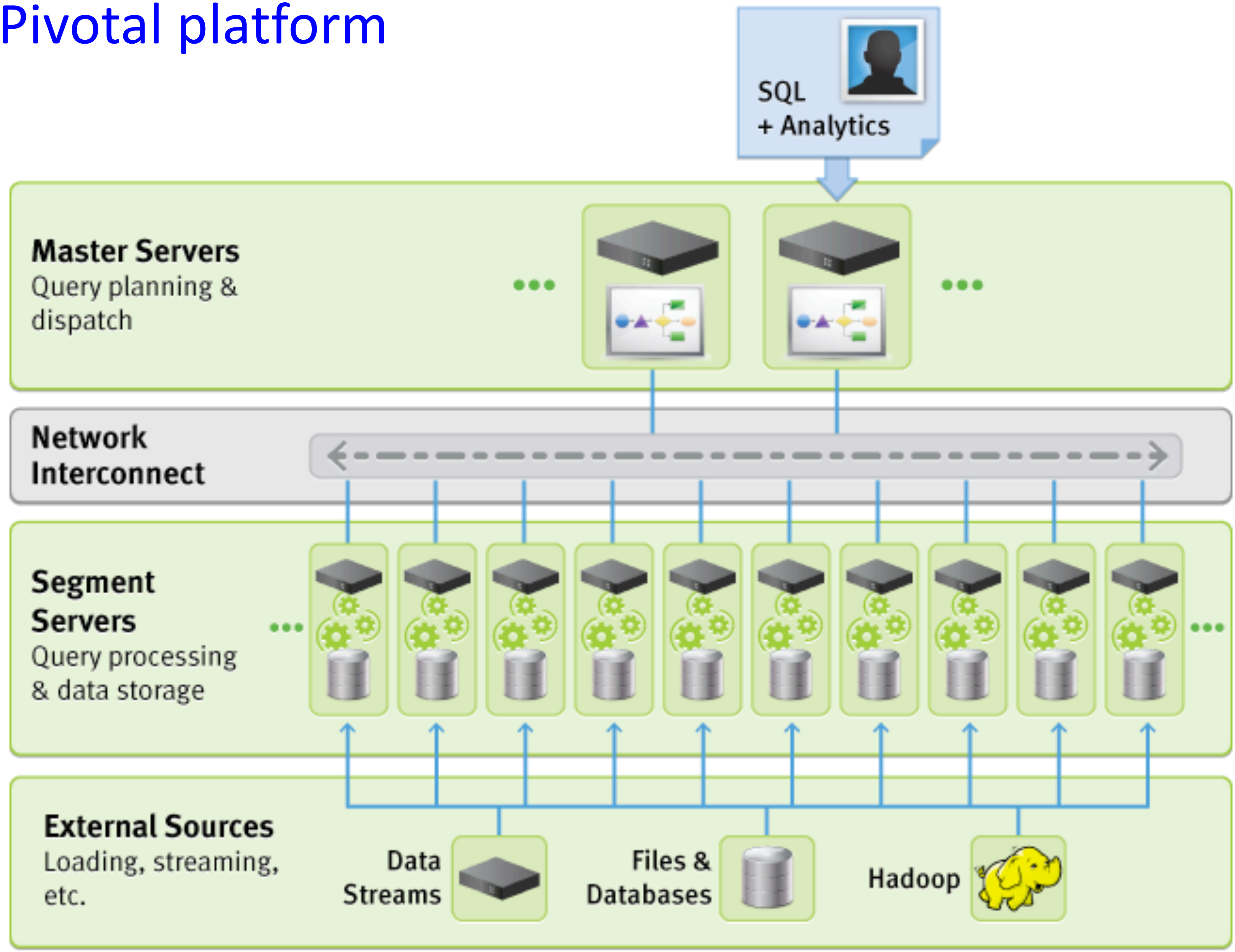


**Sparse Multiscale Gaussian Process Regression Using Hierarchical Clustering*, Z. Zhang, K. Duraisamy, N. Gumerov, Applied Numerical Mathematics 2017

Machine Learning Requirements

- Highly multidimensional
- Since learning is in feature space, very highly multi-scale (coarse & rich)
- Multiscale learning is an active research area
- The training stage requires solution of a large ill-posed linear system of algebraic equations
- Regularization and speedups of solution can be achieved via employment of methods for efficient complexity reduction, including
  - Construction of compact bases via data structures
  - Nystrom methods (low-rank approximations)
  - Preconditioned iterative procedures
  - Specially designed Krylov subspace methods
- The test stage requires fast procedures for large matrix-vector products
- Computation of predictive variance can be also done efficiently using low-rank decompositions

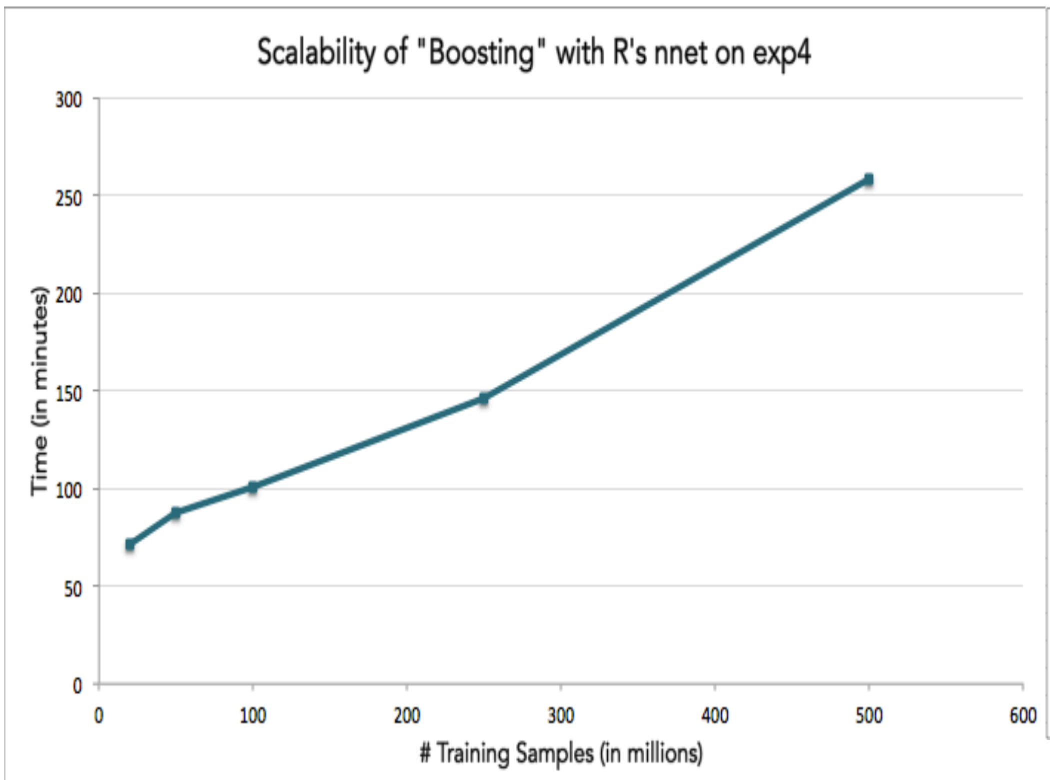
Pivotal platform



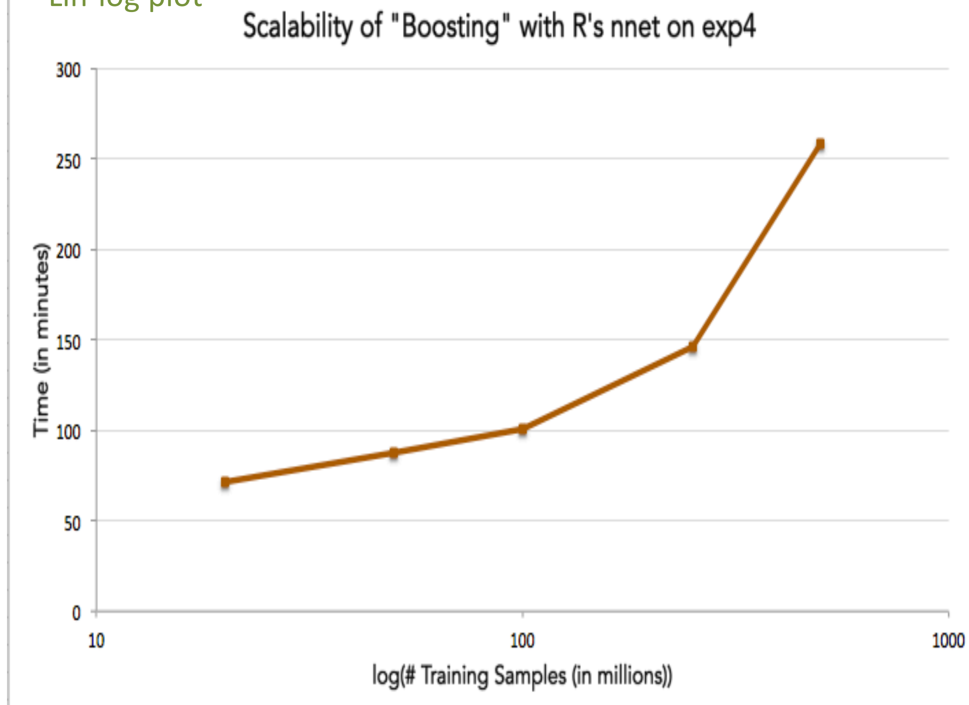
Scalability of Parallelized NN Implementation

Linearly scalable as data set scales up

- Boosting implementation proved to be linearly scalable up to at least 500 million rows of data
- Tests done on duplicated LES dataset
  - Model fits stayed relatively the same (no improvement due to no new data)



Lin-log plot



Multiscale GP : Model

output　input　noise　weights　basis functions

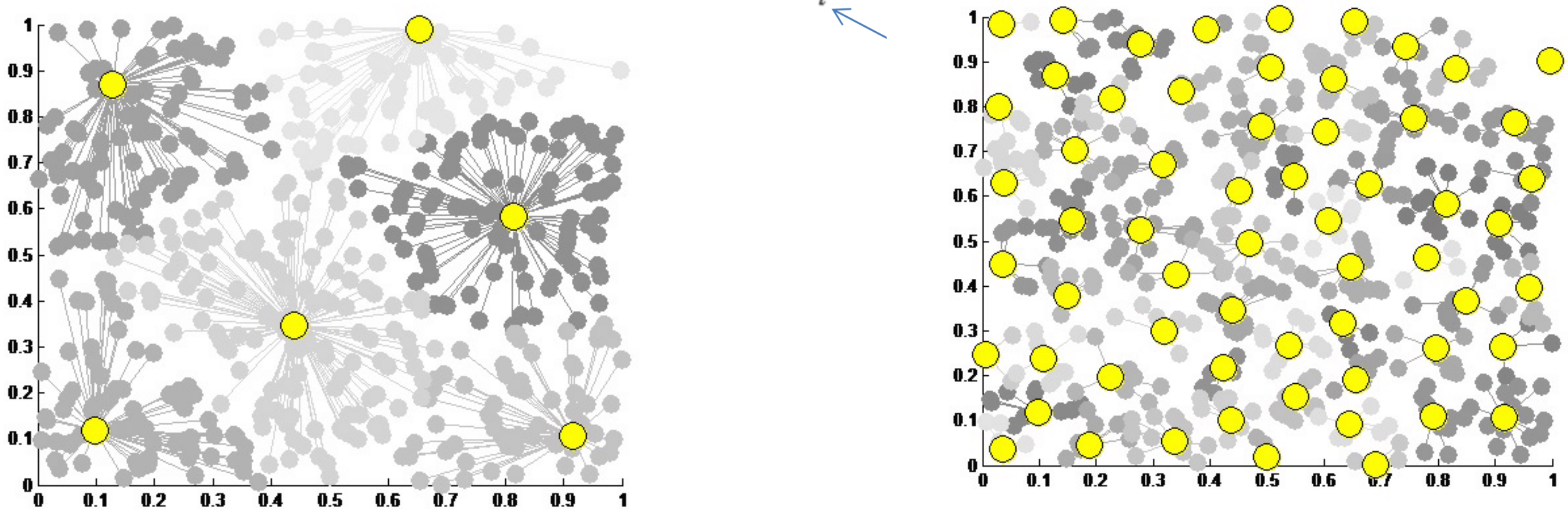
$$y = F(\mathbf{x}) + \epsilon, \quad F(\mathbf{x}) = \boldsymbol{\phi}(\mathbf{x})^T \mathbf{w} = \sum_{i=1}^{D} w_i \phi_i(\mathbf{x}), \quad \epsilon \sim \mathcal{N}(0, \sigma^2),$$

feature space　design matrix　noise variance

$$\boldsymbol{\phi}(\mathbf{x}) : \quad \mathbb{R}^d \to \mathbb{R}^D, \quad \Phi = \{\Phi_{ij}\}, \quad \Phi_{ij} = \phi_i(\mathbf{x}_j), \quad i = 1, \ldots, D, \quad j = 1, \ldots, n,$$

extended feature space　training point

$$\phi_i(\mathbf{x}) = \exp\left(-\frac{|\mathbf{x} - \mathbf{x}_i|^2}{h_i^2} \right), \quad i = 1, \ldots, D.$$

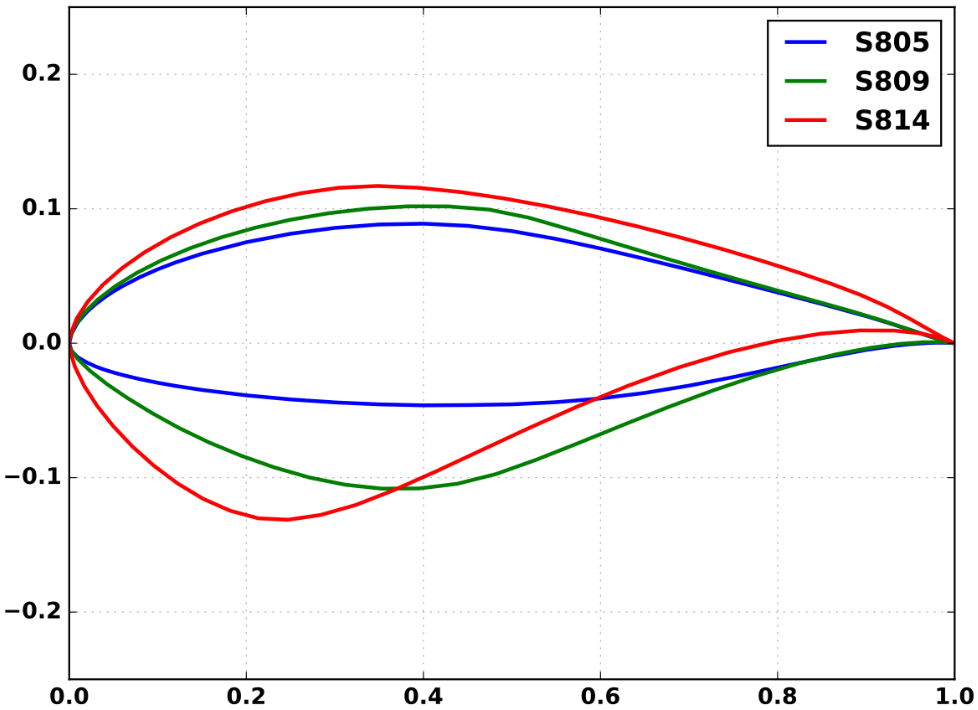


n = 500 uniform random distribution in [0,1] x [0,1]

Outline

- Introduction
- Proof-of-concept
- How do we setup the data-driven turbulence modeling problem?
- What are the components?
- Demonstration
  - ➔ Predictions in Airfoil flows
- Dissemination / impact
- Vision / Perspectives

Prediction in Airfoil flows

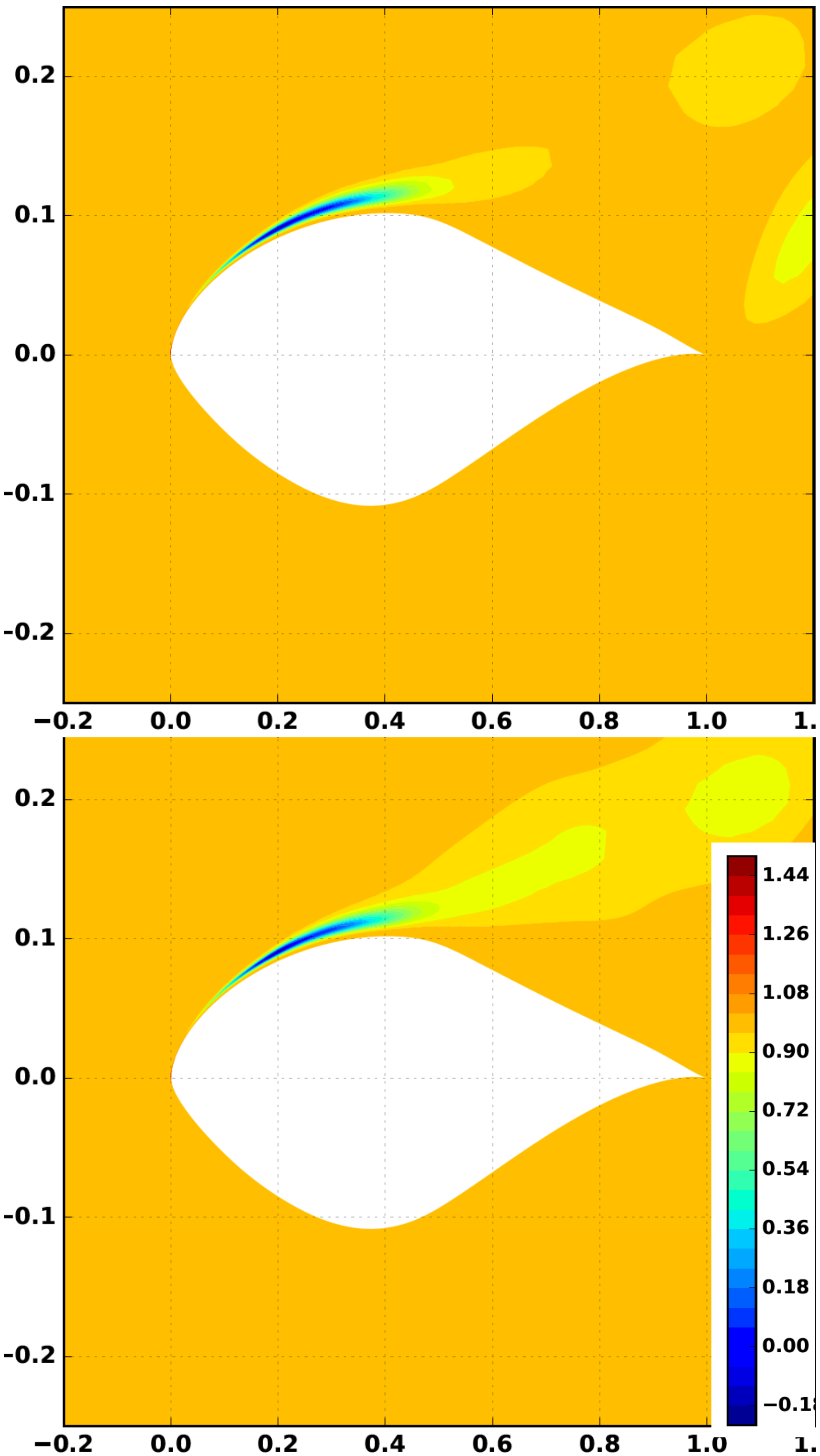


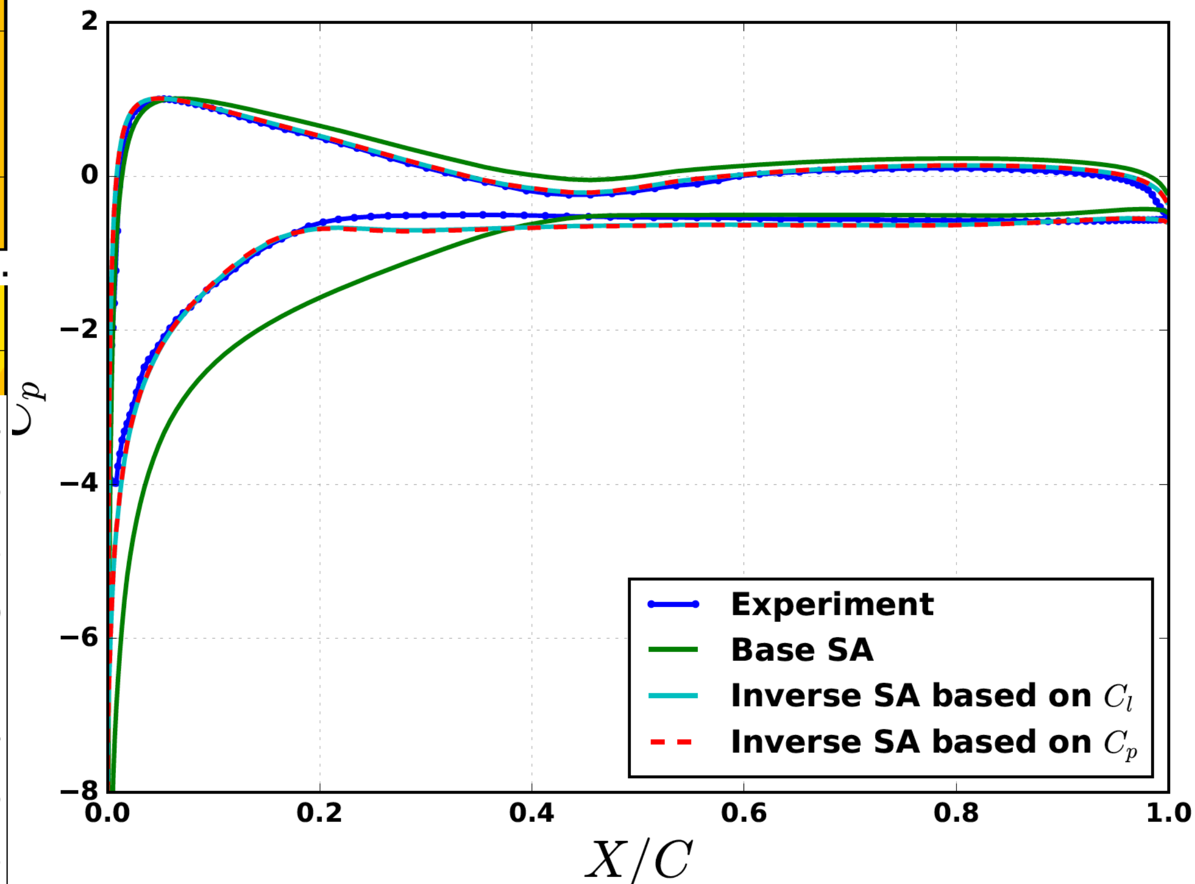
Singh, A., Medida, S. & Duraisamy, K., Data-augmented Predictive Modeling of Turbulent Separated Flows over Airfoils, AIAA Journal, 2017.

1	S805 at $Re = 1 \times 10^6$
2	S805 at $Re = 2 \times 10^6$
3	S809 at $Re = 1 \times 10^6$
4	S809 at $Re = 2 \times 10^6$
5	S805 at $Re = 1 \times 10^6, 2 \times 10^6$
6	S809 at $Re = 1 \times 10^6, 2 \times 10^6$
P	**S814** at $\mathbf{R}e = 1 \times 10^6, 2 \times 10^6$
7	S805, S809, S814 at $Re = 1 \times 10^6, 2 \times 10^6$

Training set →

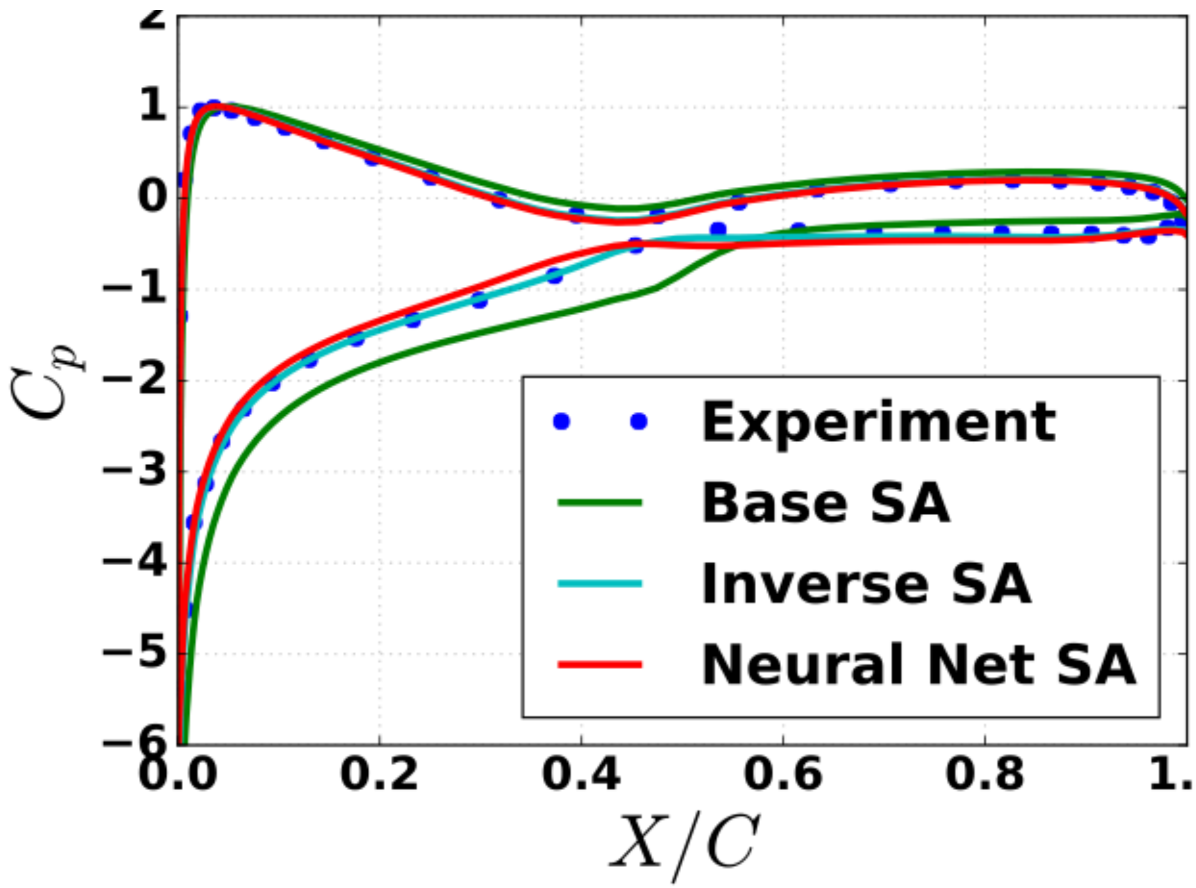
Inversion based on Pressures
vs
Inversion based on LIFT!



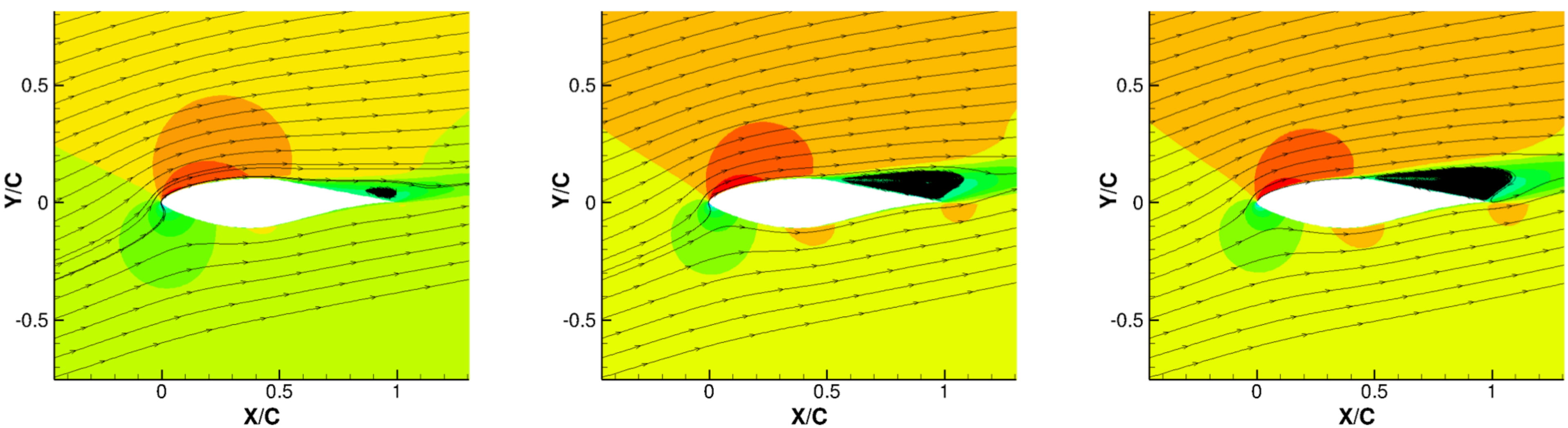


Ability to work on sparse amount of data is critical

True prediction !

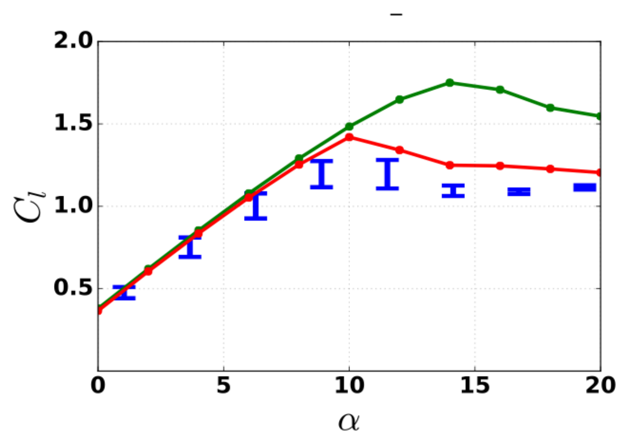


Singh, A., Medida, S. & Duraisamy, K., Data-augmented Predictive Modeling of Turbulent Separated Flows over Airfoils Submitted, AIAA Journal, 2016 (arXiv)

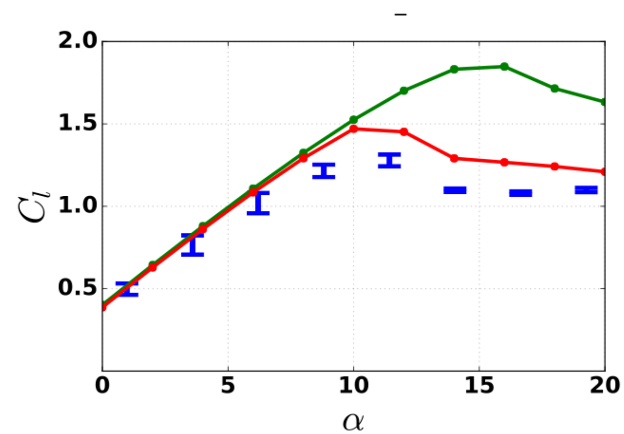


(a) Base SA (b) Inverse SA (c) NN-augmented SA (prediction)

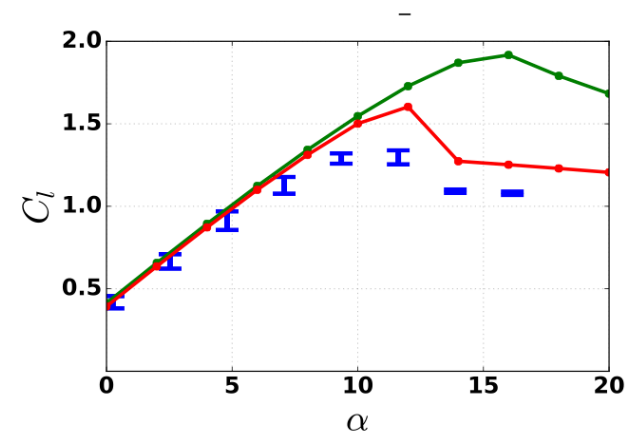
Prediction – S814



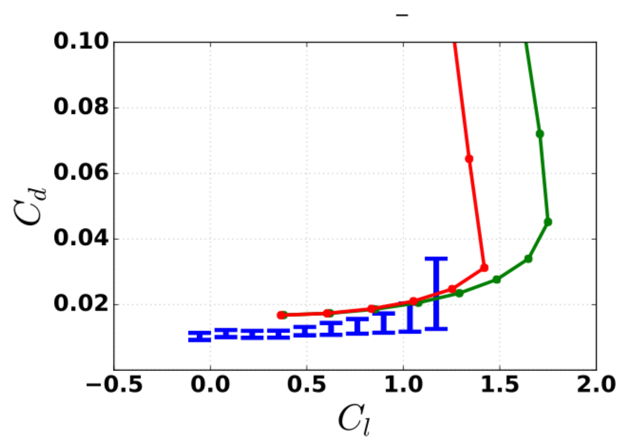
(a) $Re = 1 \times 10^6$



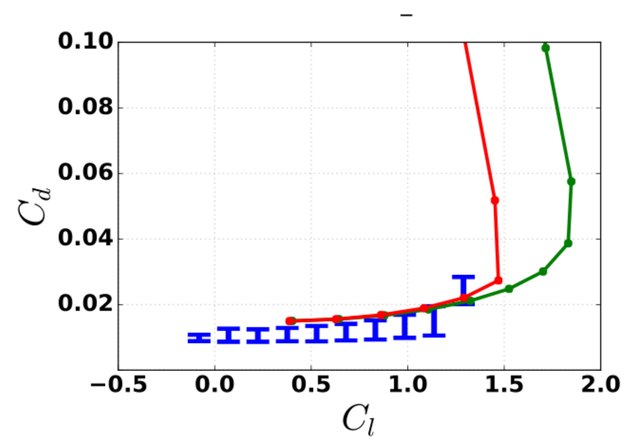
(b) $Re = 2 \times 10^6$



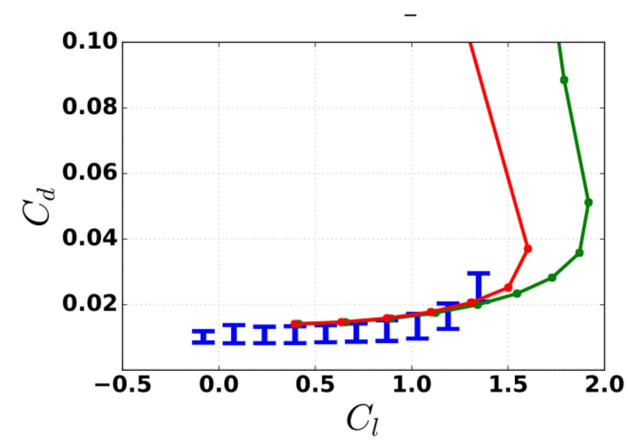
(c) $Re = 3 \times 10^6$



(d) $Re = 1 \times 10^6$



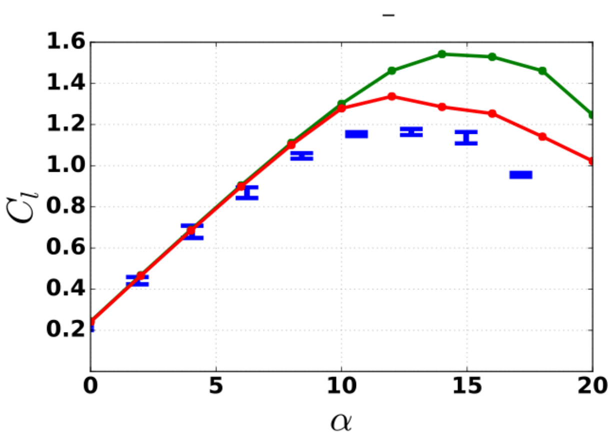
(e) $Re = 2 \times 10^6$



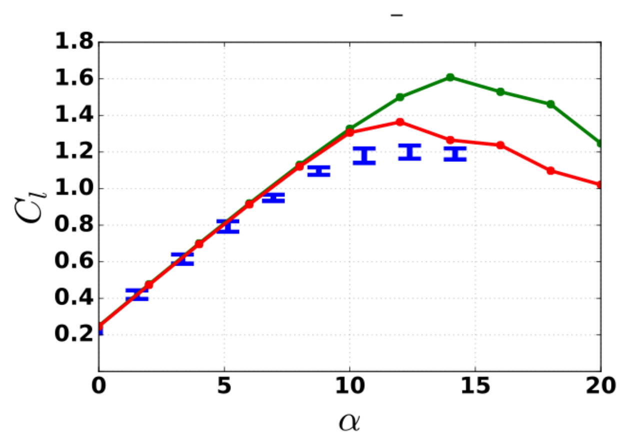
(f) $Re = 3 \times 10^6$

Collaboration with Altair, Inc.

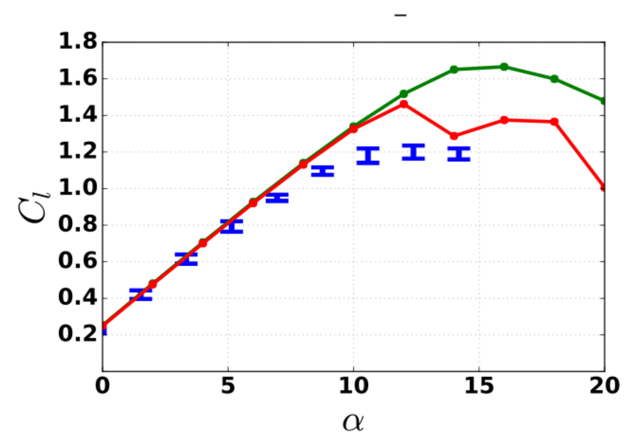
Prediction – S805



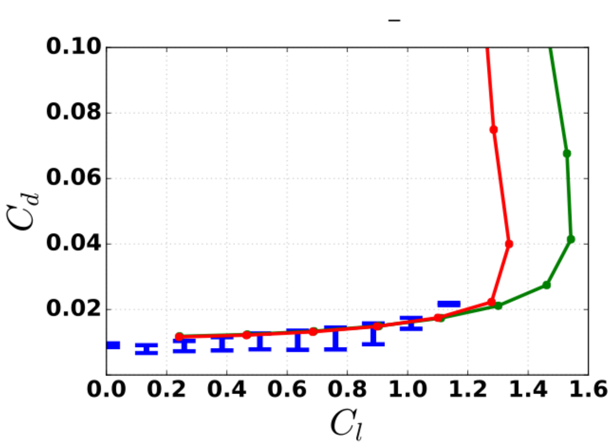
(a) $Re = 1 \times 10^6$



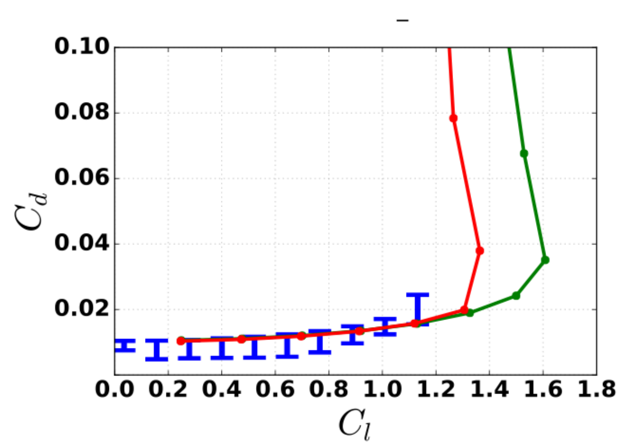
(b) $Re = 2 \times 10^6$



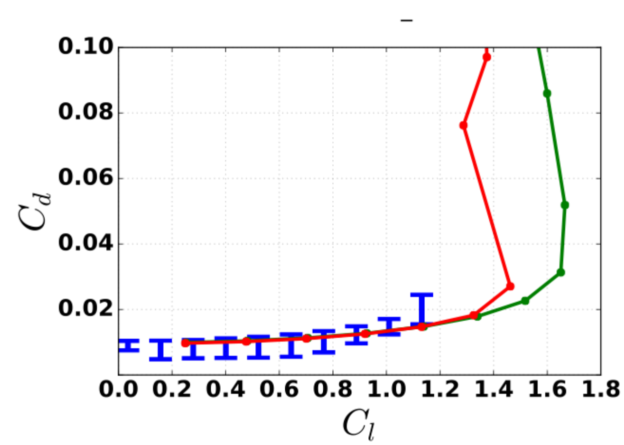
(c) $Re = 3 \times 10^6$



(d) $Re = 1 \times 10^6$



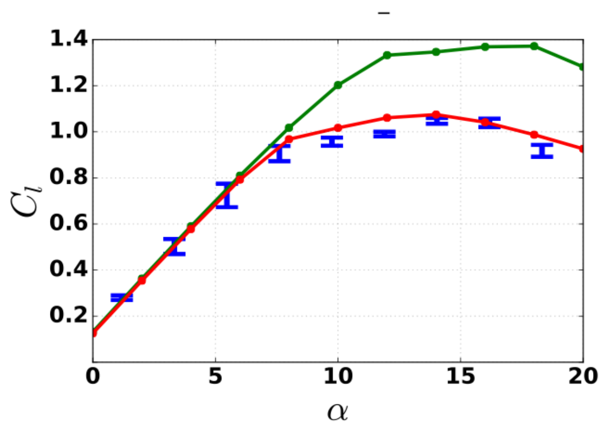
(e) $Re = 2 \times 10^6$



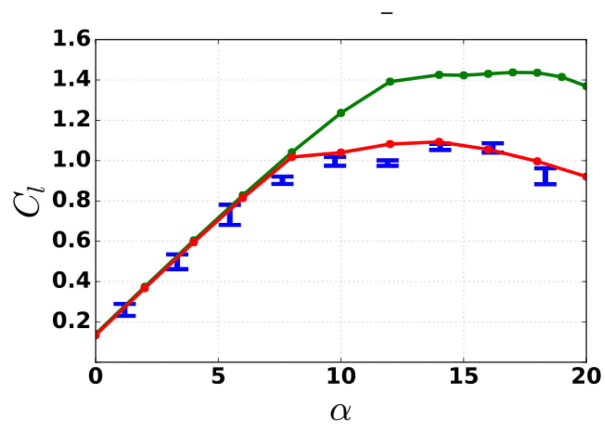
(f) $Re = 3 \times 10^6$

Collaboration with Altair, Inc.

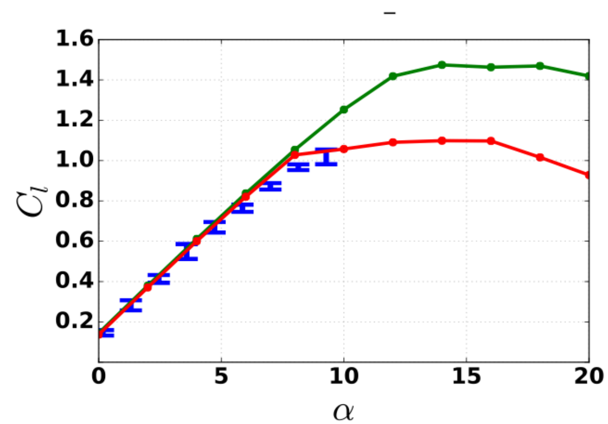
Prediction – S809



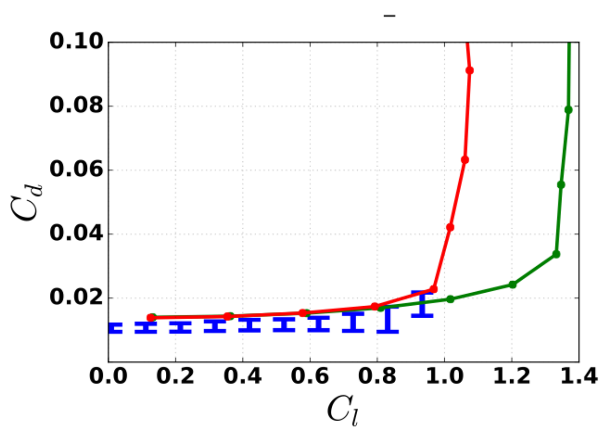
(a) $Re = 1 \times 10^6$



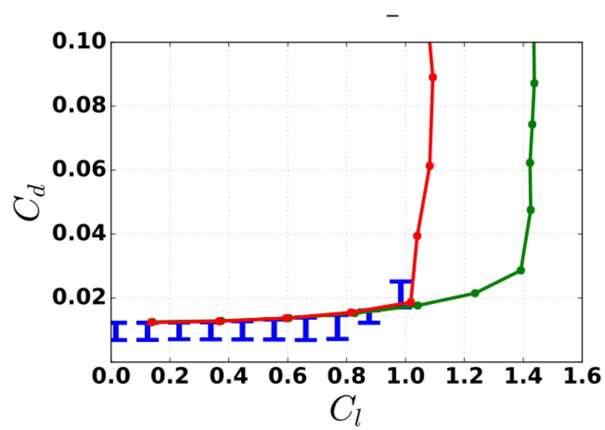
(b) $Re = 2 \times 10^6$



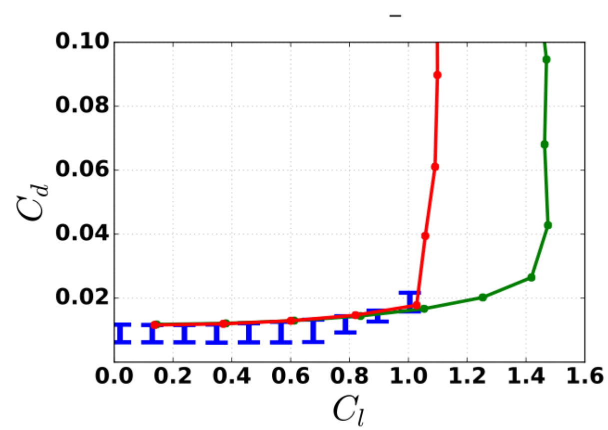
(c) $Re = 3 \times 10^6$



(d) $Re = 1 \times 10^6$



(e) $Re = 2 \times 10^6$

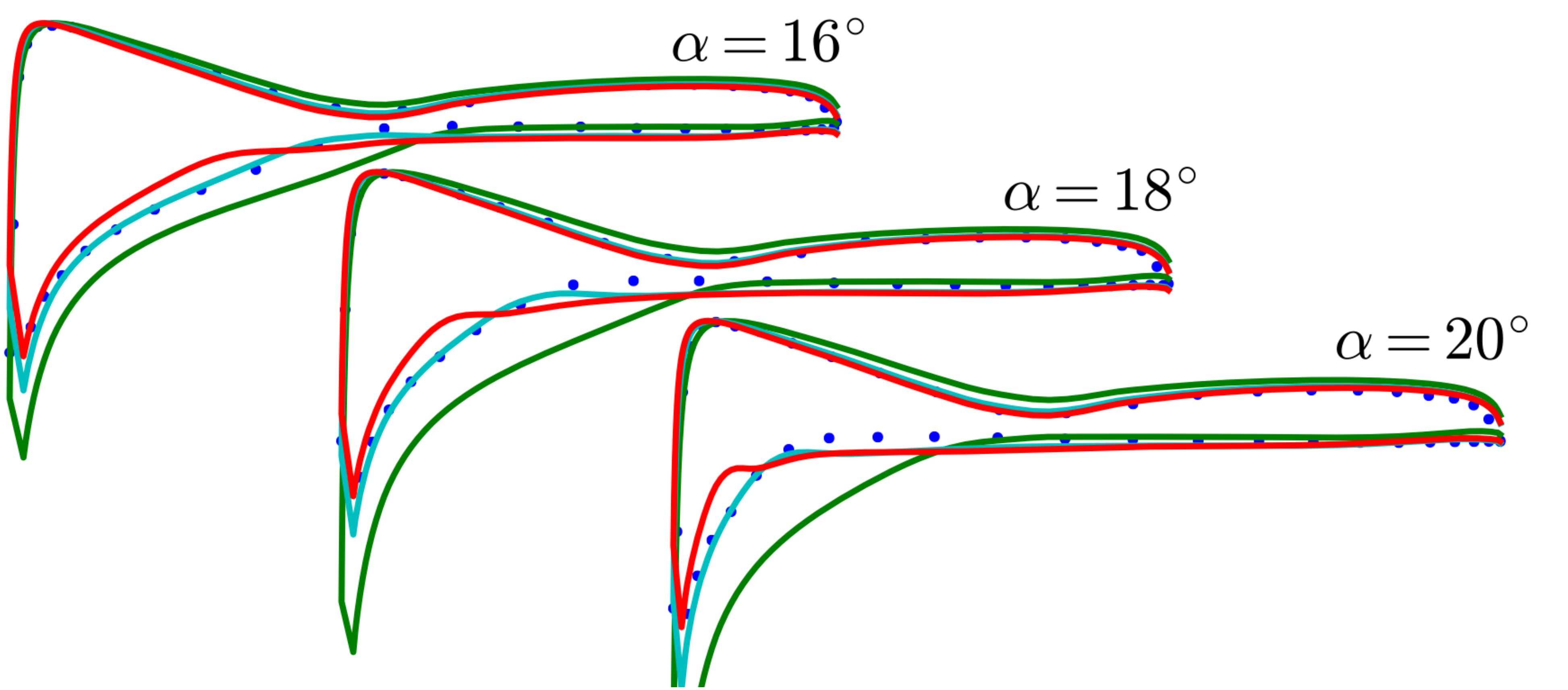


(f) $Re = 3 \times 10^6$

Collaboration with Altair, Inc.

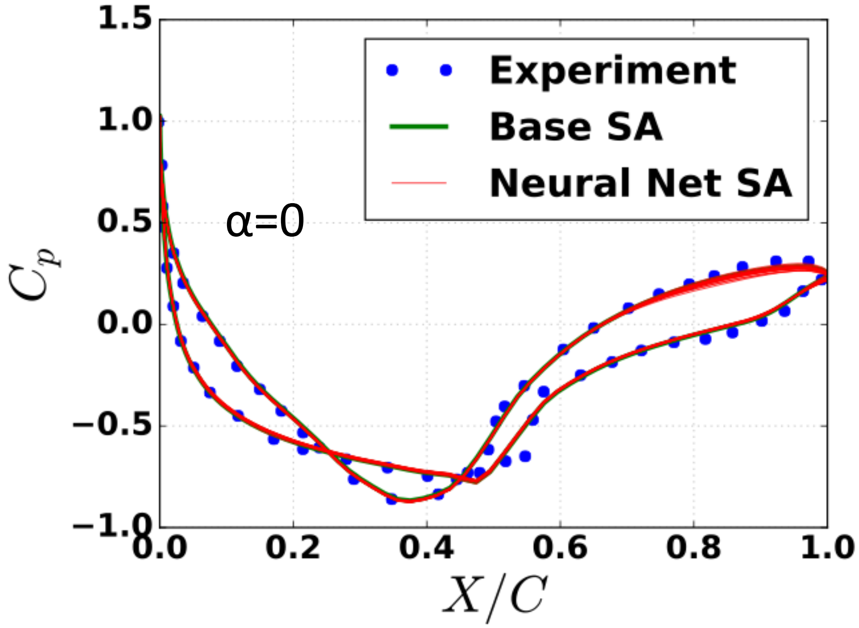
True prediction !

S 809, Re=2 Million

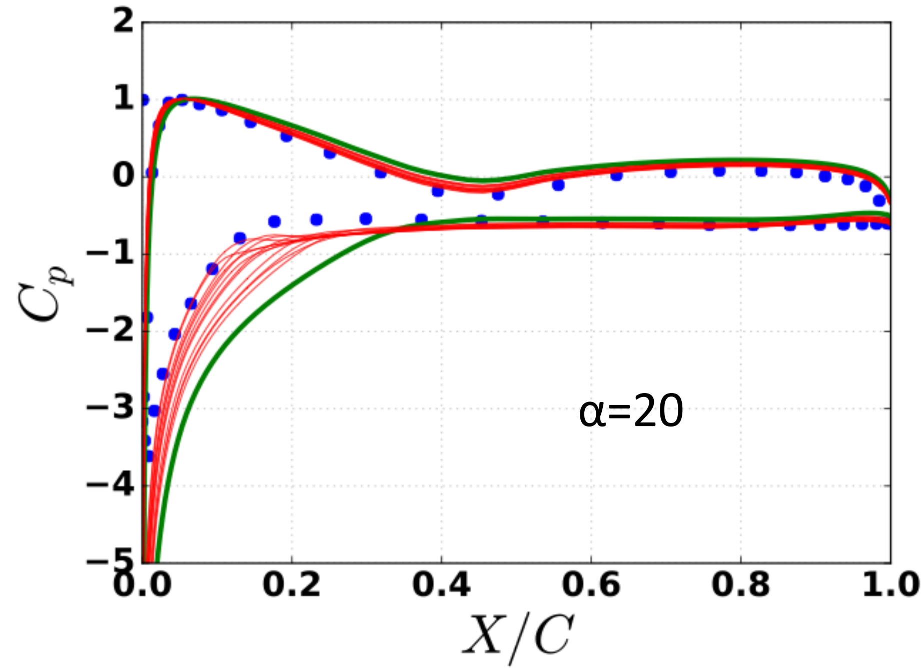


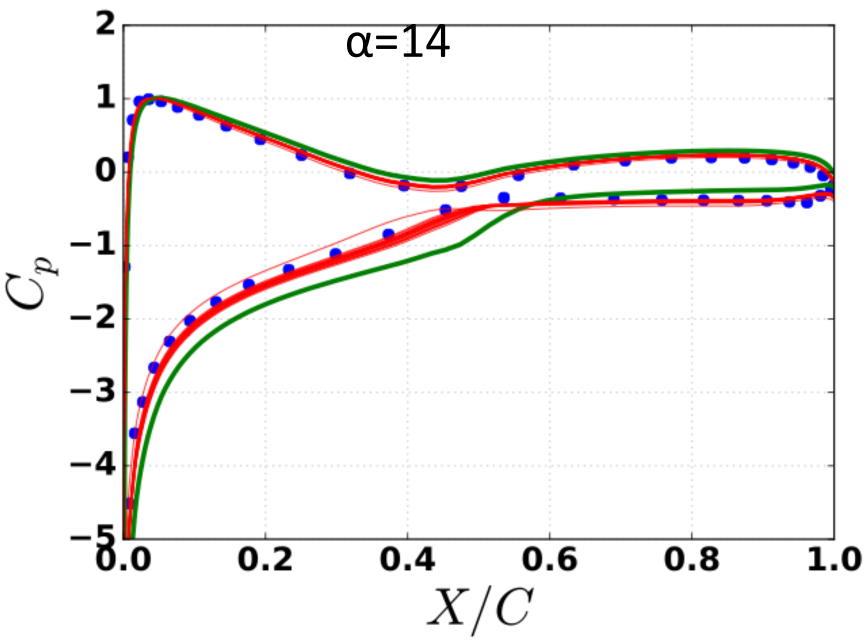
Inference used only CL data, NN-augmented model provides considerable predictive improvements of Cp

Variability



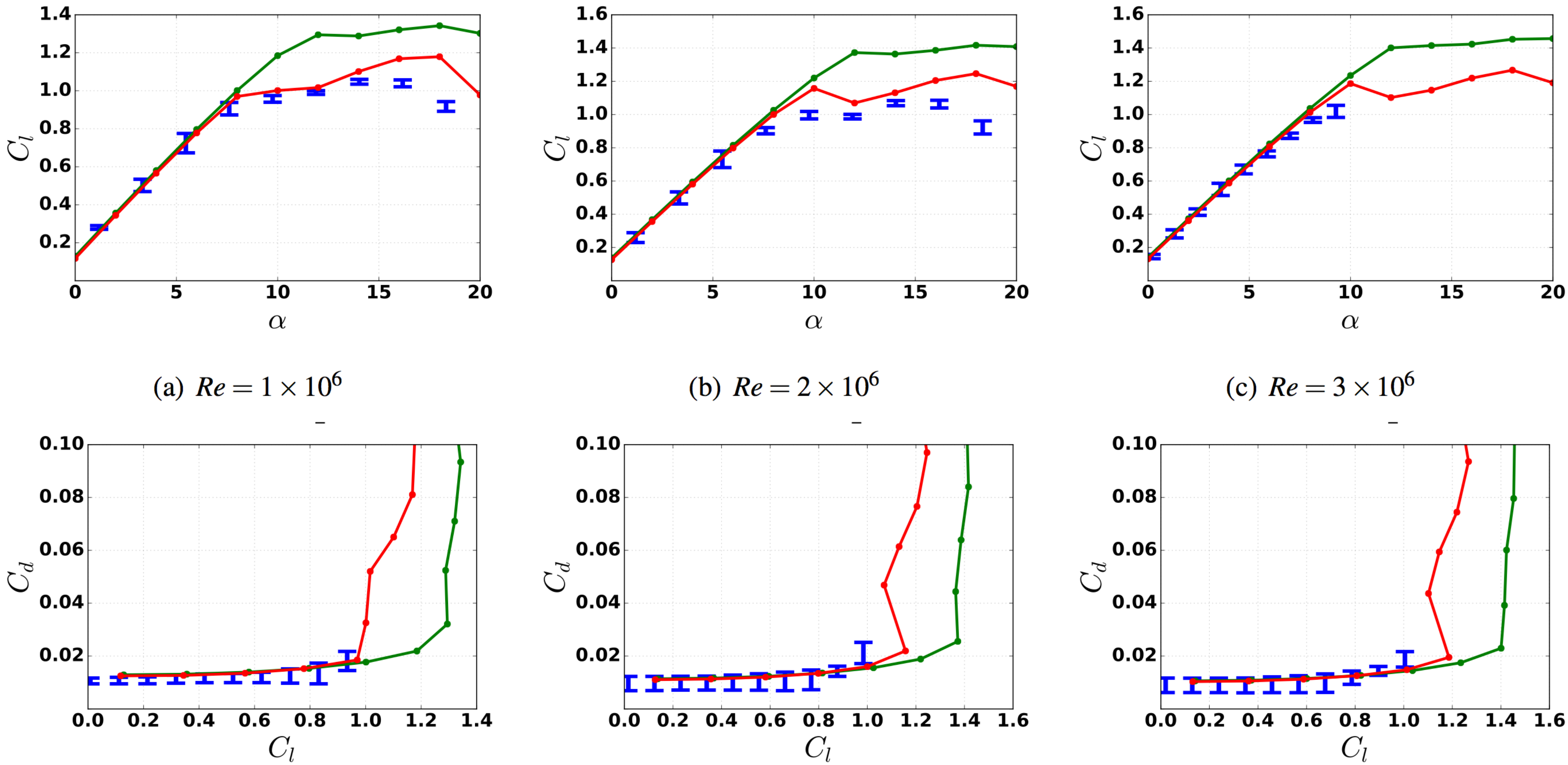
S 809, Re=2 Million





Training from different sets

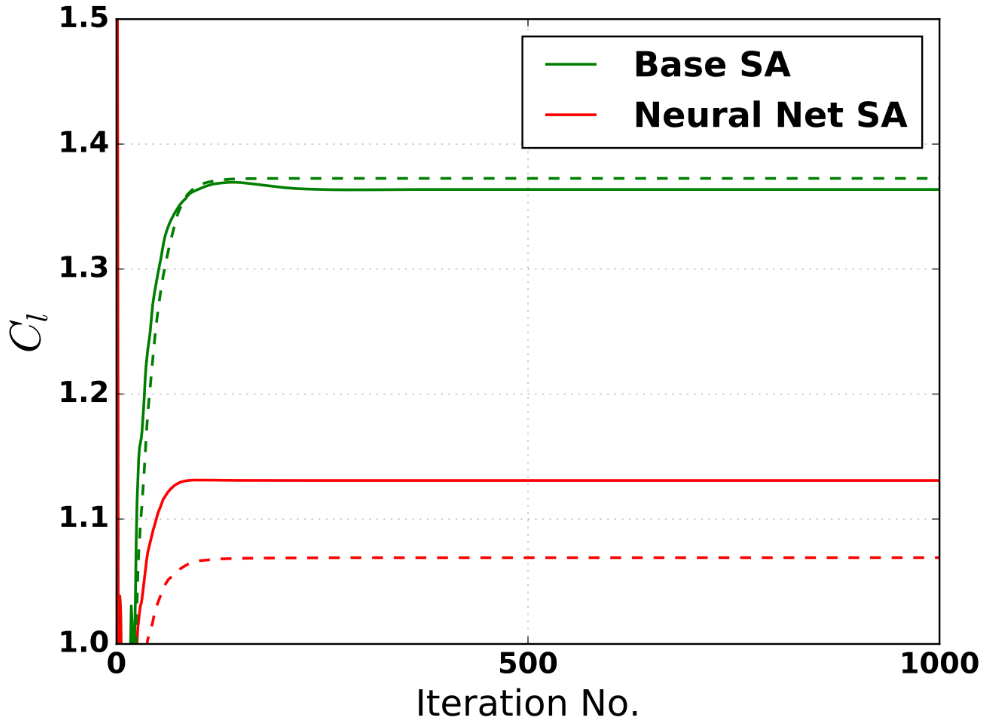
Portability :
Implementation in AcuSolve



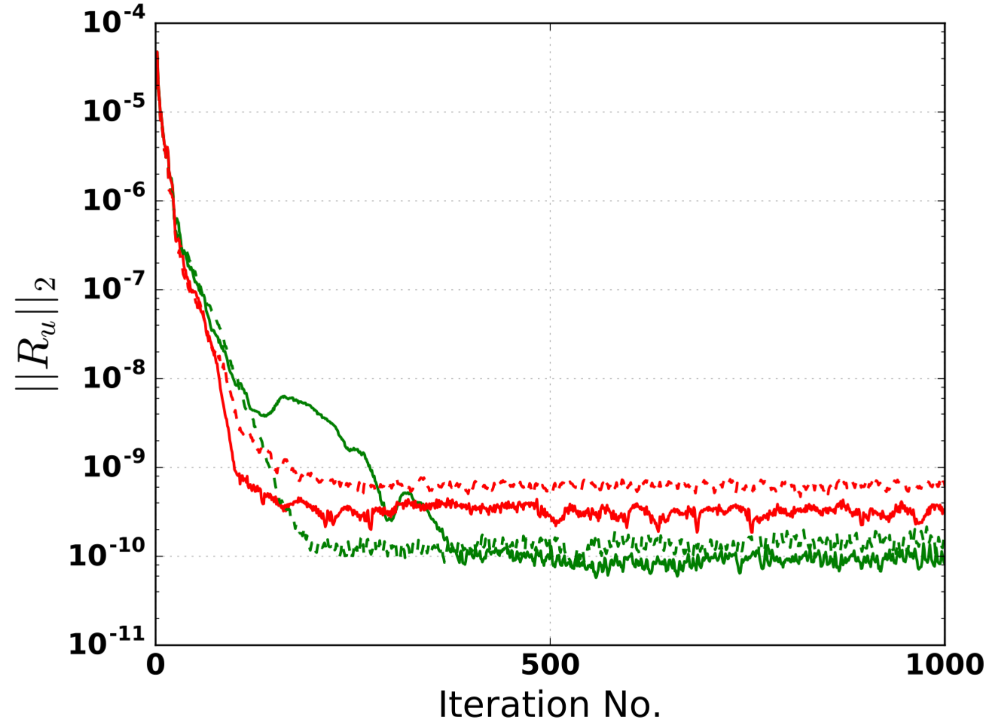
(a) $Re = 1 \times 10^6$ (b) $Re = 2 \times 10^6$ (c) $Re = 3 \times 10^6$

S809 Airfoil : Predictive results in Commercial CFD solver

Robustness: Implementation in AcuSolve



(a) Lift coefficient



(b) L2 norm of solver residual

S809 Airfoil : Predictive results in Commercial CFD solver

Outline

- Introduction
- Proof-of-concept
- How do we setup the data-driven turbulence modeling problem?
- What are the components?
- Demonstration
  - ➔ Predictions in Airfoil flows
- Dissemination / impact
- Vision / Perspectives

Dissemination : UM/NASA Symposium

Attendees: 88

NASA + Other National labs : 12 + 8 = 20
U of M + Other academia : 22 + 31 = 53
Industry = 15

Status, Emerging Ideas and Future Directions of Turbulence Modeling Research in Aeronautics

K. Duraisamy*, P.R. Spalart†, C.L. Rumsey‡

Dissemination

- AFRL starting a 6.3 project in data-driven turbulence modeling!
- In talks with DLR Braunschweig
- Project website

- Participation in NASA/Stanford Summer Turbulence Research Program (2014/2016)
- Presentation at NASA Langley (2013/2014/2016)
- Visit by Gary Coleman to Michigan (Dec 2014)
- Discussion with Big data analytics group at NASA Langley
- Discussions with several NASA researchers (Ames, Langley)

Some Key papers

- Singh, A.P. & Medida, S. & Duraisamy, K. Machine Learning-augmented Predictive Modeling of Turbulent Separated Flows over Airfoils, AIAA Journal, Vol. 55, No. 7 (2017), pp. 2215-2227. 2017
- Duraisamy, K. & Singh, A.P. & Pan, S. Augmentation of Turbulence Models Using Field Inversion and Machine Learning, Proc. AIAA SciTech, Grapevine, TX 2017
- Singh, A.P. & Duraisamy, K. Using Field Inversion to Quantify Functional Errors in Turbulence Closures, Phys. Fluids 28, 045110 2016
- Parish, Eric & Duraisamy, Karthik, A paradigm for data-driven predictive modeling using field inversion and machine learning, Journal of Computational Physics, Volume 305, 15 January 2016, Pages 758–774 2016
- Tracey, Brendan & Duraisamy, Karthik, & Alonso, Juan J. A Machine Learning Strategy to Assist Turbulence Model Development, Proc. AIAA SciTech, Kissimmee, FL 2015
- Duraisamy, Karthik; Zhang, Ze Jia & Singh, A.P., New Approaches in Turbulence and Transition Modeling Using Data-driven Techniques, Proc. AIAA SciTech, Kissimmee, FL 2015

Also..

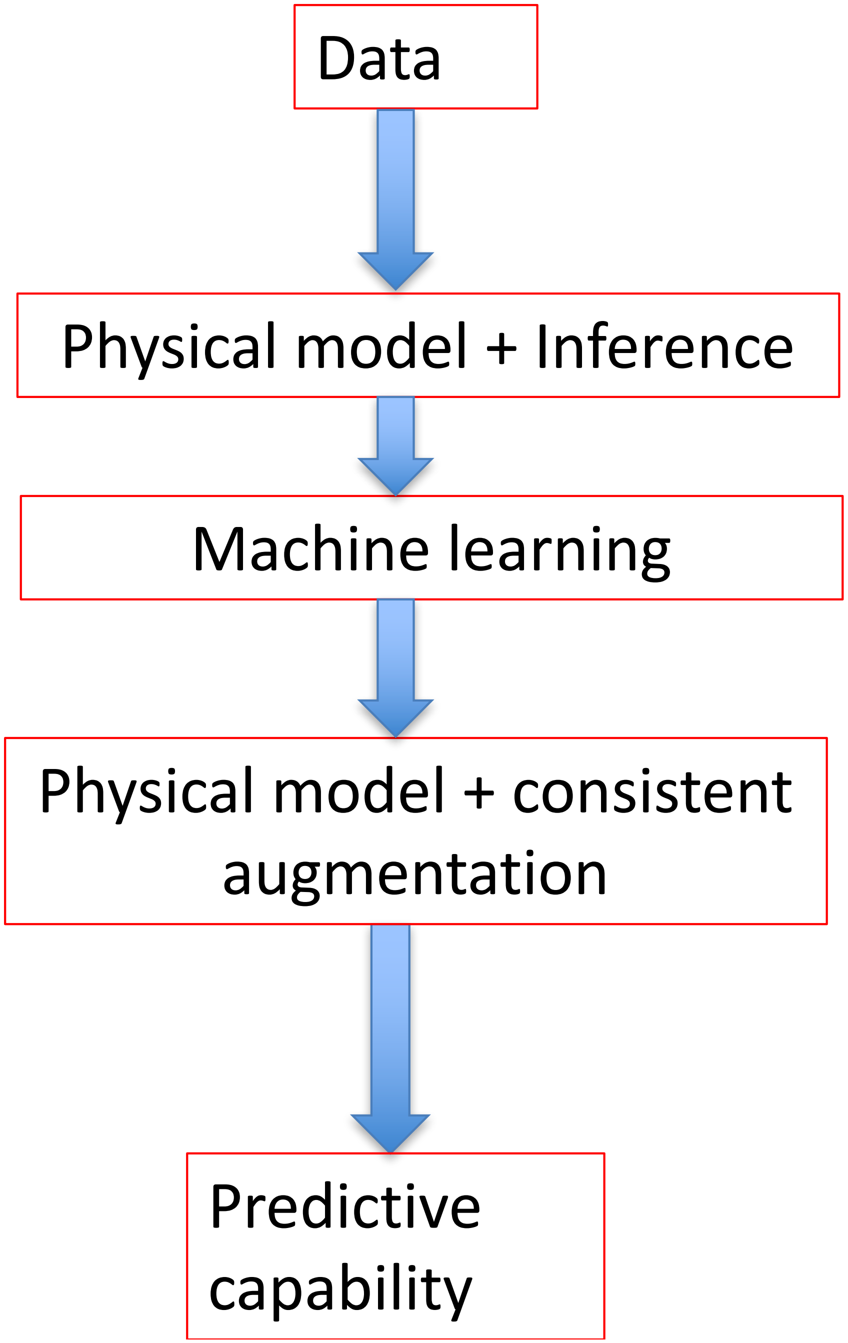
- Data-driven augmentation of turbulence models for adverse pressure gradient flows AP Singh, R Matai, K Duraisamy, P Durbin, Proc. AIAA Aviation 2017
- Singh, A.P. & Pan, S. & Duraisamy, K. Characterizing and Improving Predictive Accuracy in Shock-Turbulent Boundary Layer Interactions Using Data-driven Models, Proc. AIAA SciTech, Grapevine, TX 2017
- Zhang, Z. & Duraisamy, K. & Gumerov, N. Efficient Multiscale Gaussian Process Regression using Hierarchical Clustering, Submitted, Machine Learning Journal, 2016
- Duraisamy, Karthik & Singh, A.P., Informing Turbulence Closures With Computational and Experimental Data, Proc. AIAA SciTech, San Diego, CA 2016
- Zhang, Ze Jia & Duraisamy, Karthik, Machine Learning Methods for Data-Driven Turbulence Modeling, Proc. AIAA Aviation, Dallas, TX 2015
- Parish, Eric & Duraisamy, Karthik, Quantification of Turbulence Modeling Uncertainties Using Full Field Inversion, Proc. AIAA Aviation, Dallas, TX 2015
- Duraisamy, Karthik & Durbin , P.A., Transition modeling using data driven approaches, Center of Turbulence Research, Proceedings of the Summer Program 2014

Growing community for data-driven turbulence modeling – thanks to NASA LEARN !

2013: Tracey, Duraisamy, Alonso (ML for non-parametric UQ)
---------------- LEARN BEGINS Jan 2014 ---------------------
2014: Duraisamy et. al (Inversion + ML for model improvement)
2015: Ling & Templeton, Weatheritt & Sandberg (apriori ML)
2016: Xiao et al. (ML for model improvement)
2017: Fabbiane, Mishra, Iaccarino, Edeling (physics, data-based)
Also, Dwight, Cinella, Arunjatesan et al.,

Companies: Altair, Inc. ; UTRC;
Labs: AFRL, DLR

Field Inversion + Machine learning to Augment Physics-based, Consistent Models



- Data contains real quantities; Model contains “modeled” quantities (loss of consistency is bad in turbulence models)

  ➔ Inference connects real quantities to modeled ones

- Data will be only loosely connected to model (and not objective)

  ➔ Inference connects secondary, non-objective data to model quantites

- Data will be noisy and of variable quality, inherent uncertainty

  ➔ Probabilistic casting of inference and learning

Perspectives 1/2

- Framework: Data -> Information -> Knowledge -> Prediction
- Machine learning
  - ➔ Can function as indicator
  - ➔ Is an optional step
  - ➔ Can be fed by theory and asymptotics
- If there is an underlying “exact” model, we can discover it
- There is no (and will ever be a) universally accurate model waiting to be discovered
  - ➔ Optimal model, conditional on data and assumptions possible
  - ➔ Avoid tendency to overfit
  - ➔ Small number of sensible features (Galilean invariant)
  - ➔ Absolutely the most sensible thing to do in an industrial setting

(Lots of data for a class of problems, Lots of expertise/knowhow)

Perspectives 2/2

- Modeling has ALWAYS been data-driven & we have always been using machine learning (and inversion too)

- Data-driven approach is not a substitute to turbulence modeling

- Data-driven approach is not a new way of modeling. It is a new tool.

- “Kitty Hawk” Stage. Community effort required

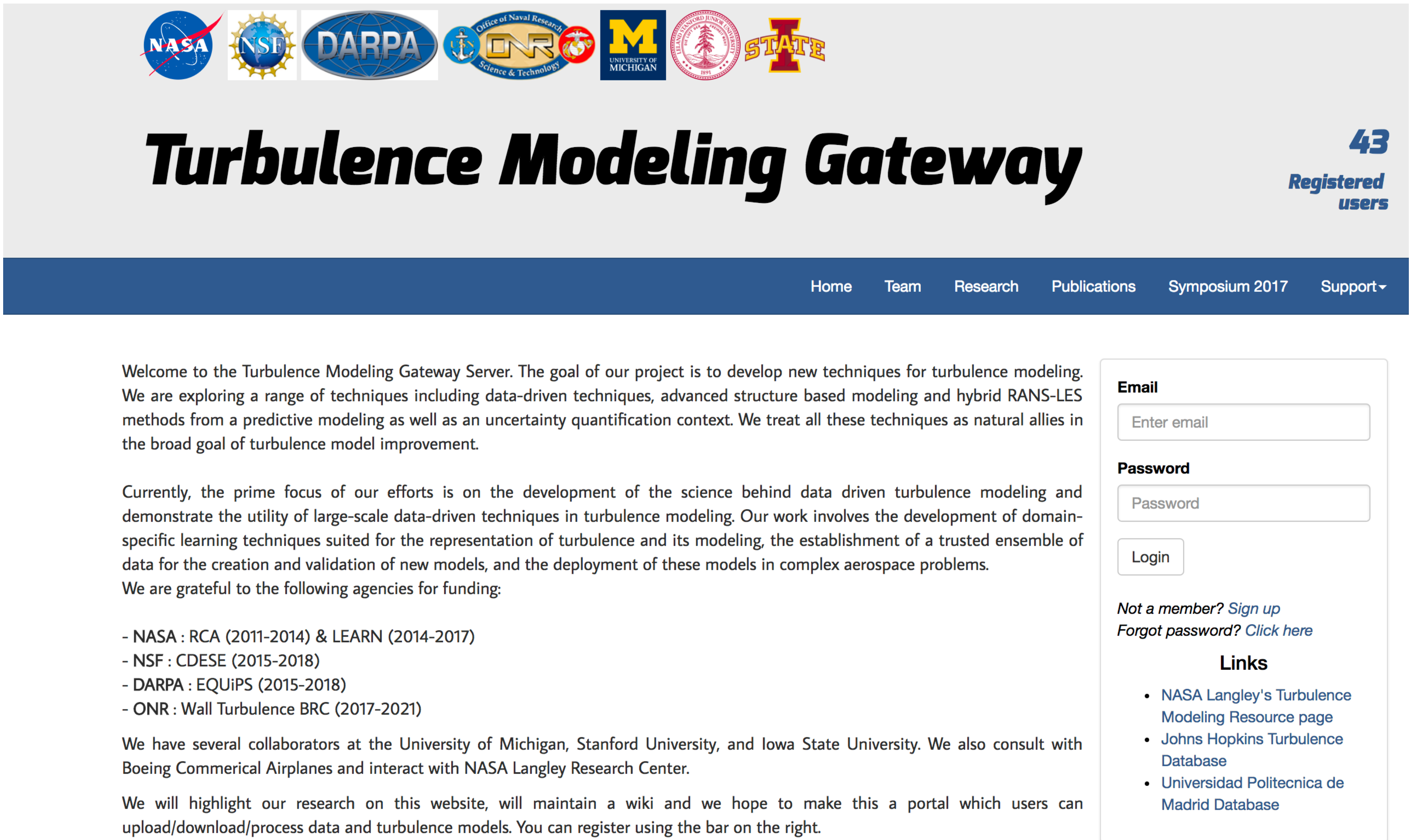
  - Uses (other than prediction):

    ➔ Model credibility: Can validate/invalidate model structures
    ➔ Uncertainty quantification: Can obtain modeling error bounds
    ➔ Robust design
    ➔ Feature selection
    ➔ Input for modeler (forget machine learning)

Vision

A continuously augmented curated database / website of inferred corrections that are input to the machine learning process

Users upload/download/process data, generate maps.



Welcome to the Turbulence Modeling Gateway Server. The goal of our project is to develop new techniques for turbulence modeling. We are exploring a range of techniques including data-driven techniques, advanced structure based modeling and hybrid RANS-LES methods from a predictive modeling as well as an uncertainty quantification context. We treat all these techniques as natural allies in the broad goal of turbulence model improvement.

Currently, the prime focus of our efforts is on the development of the science behind data driven turbulence modeling and demonstrate the utility of large-scale data-driven techniques in turbulence modeling. Our work involves the development of domain-specific learning techniques suited for the representation of turbulence and its modeling, the establishment of a trusted ensemble of data for the creation and validation of new models, and the deployment of these models in complex aerospace problems.
We are grateful to the following agencies for funding:

- NASA : RCA (2011-2014) & LEARN (2014-2017)
- NSF : CDESE (2015-2018)
- DARPA : EQUiPS (2015-2018)
- ONR : Wall Turbulence BRC (2017-2021)

We have several collaborators at the University of Michigan, Stanford University, and Iowa State University. We also consult with Boeing Commerical Airplanes and interact with NASA Langley Research Center.

We will highlight our research on this website, will maintain a wiki and we hope to make this a portal which users can upload/download/process data and turbulence models. You can register using the bar on the right.

Email

Enter email

Password

Password

Login

Not a member? *Sign up*
Forgot password? *Click here*

Links

- NASA Langley's Turbulence Modeling Resource page
- Johns Hopkins Turbulence Database
- Universidad Politecnica de Madrid Database

Acknowledgements

NARI (Koushik Datta, Michael Dudley)

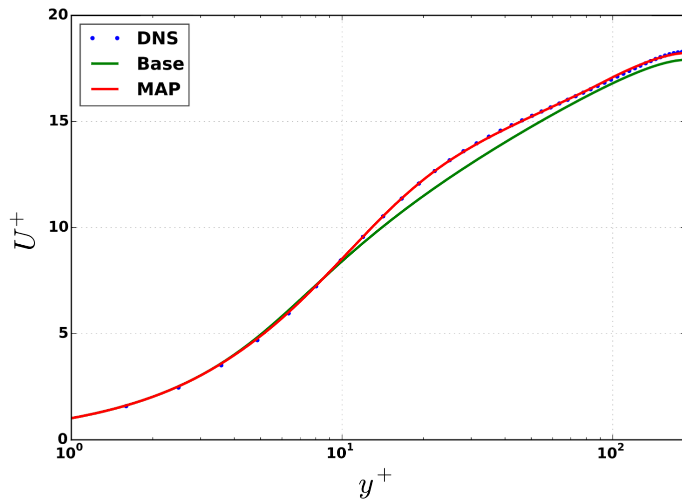
Monitors (Gary Coleman, Mujeeb Malik)

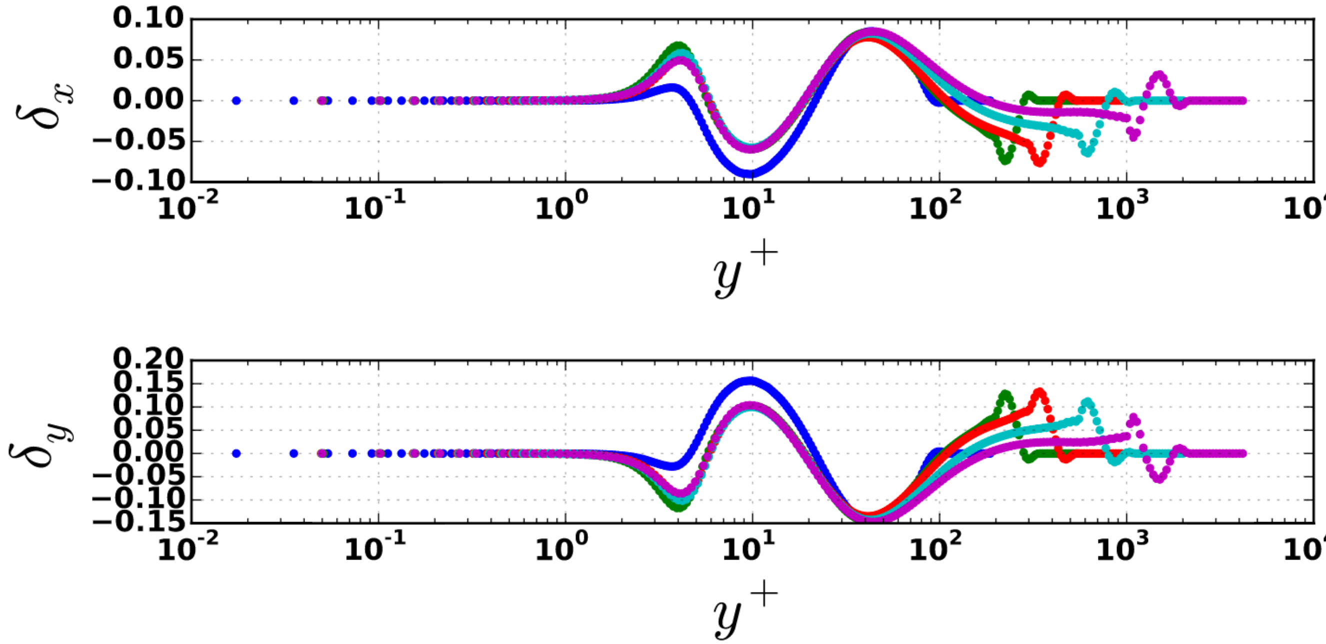
Backup slides

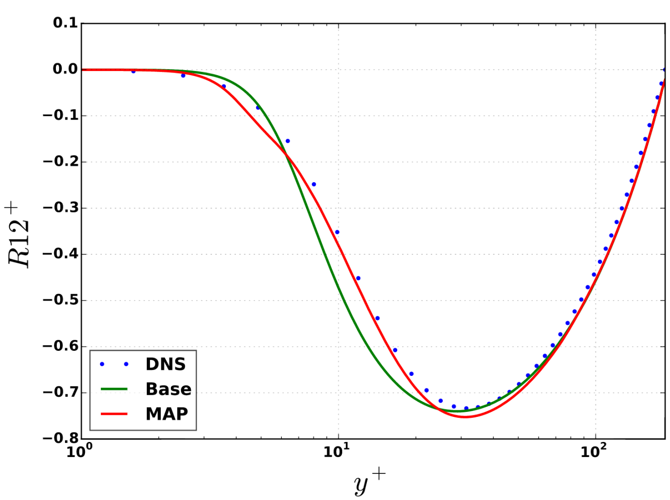
Introducing discrepancies in stress perturbations

$$\mathbf{R}_p = 2k\left[\frac{\mathbf{I}}{3} + \mathbf{V}(\Lambda + \vec{\beta}(x))\mathbf{V}^T\right]$$

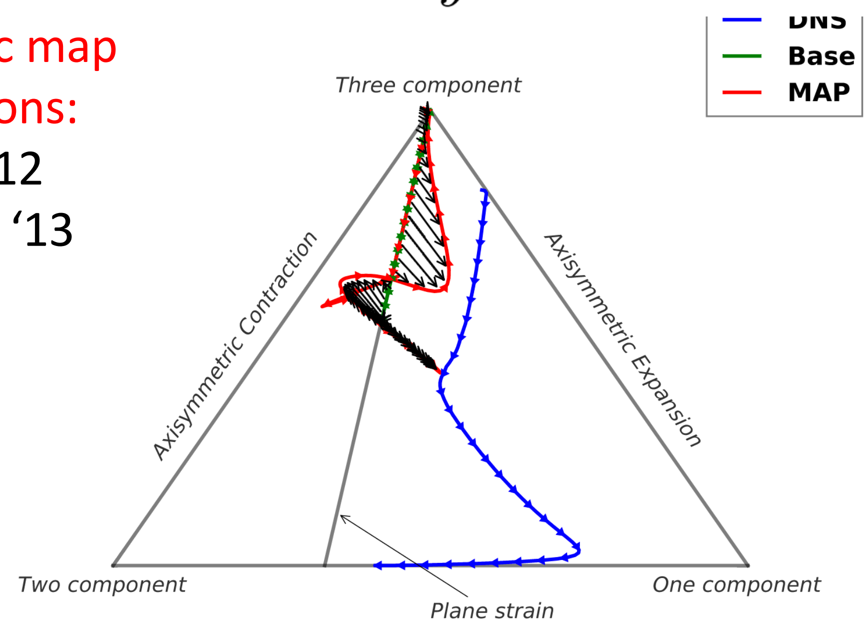
k-ω model



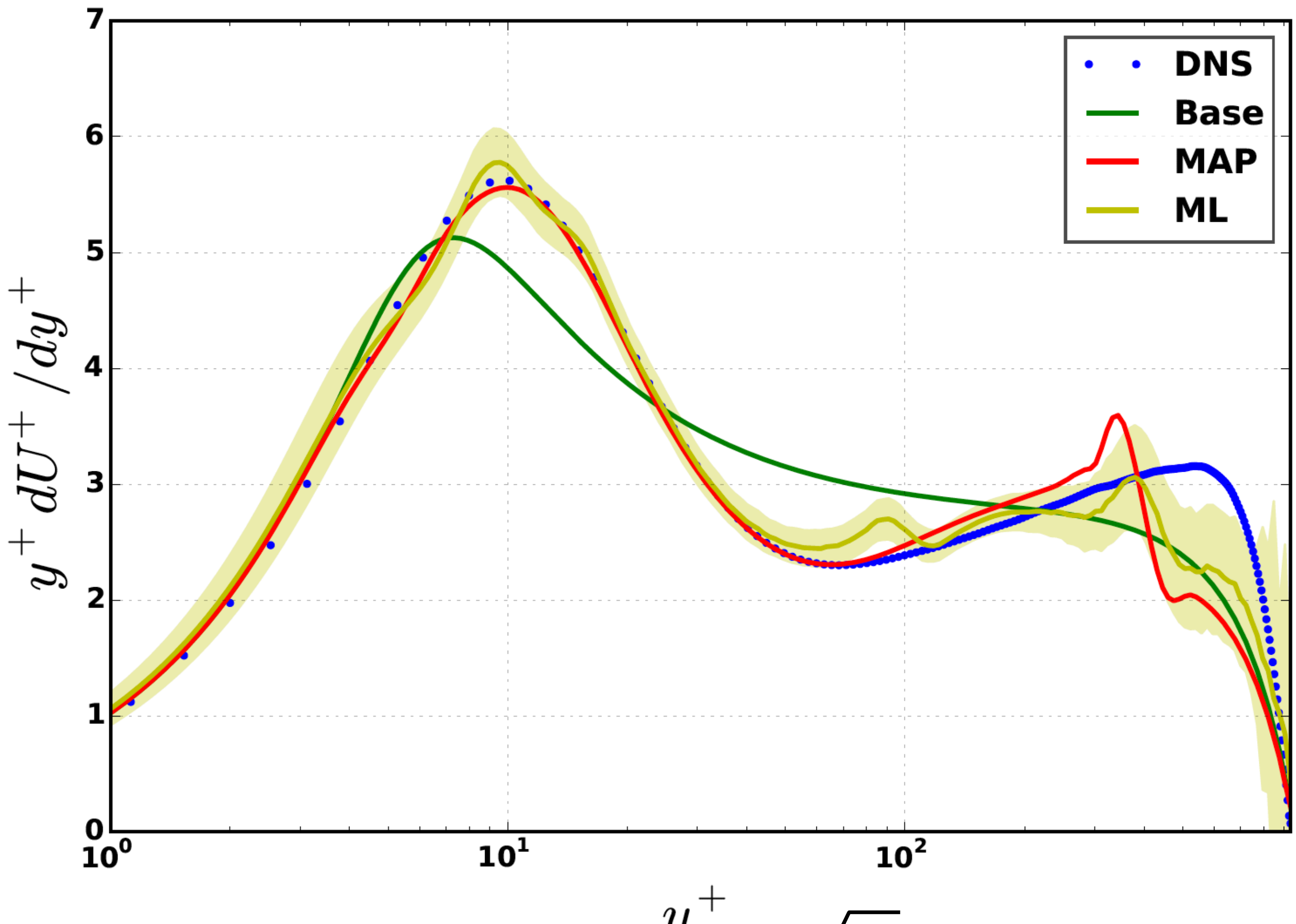




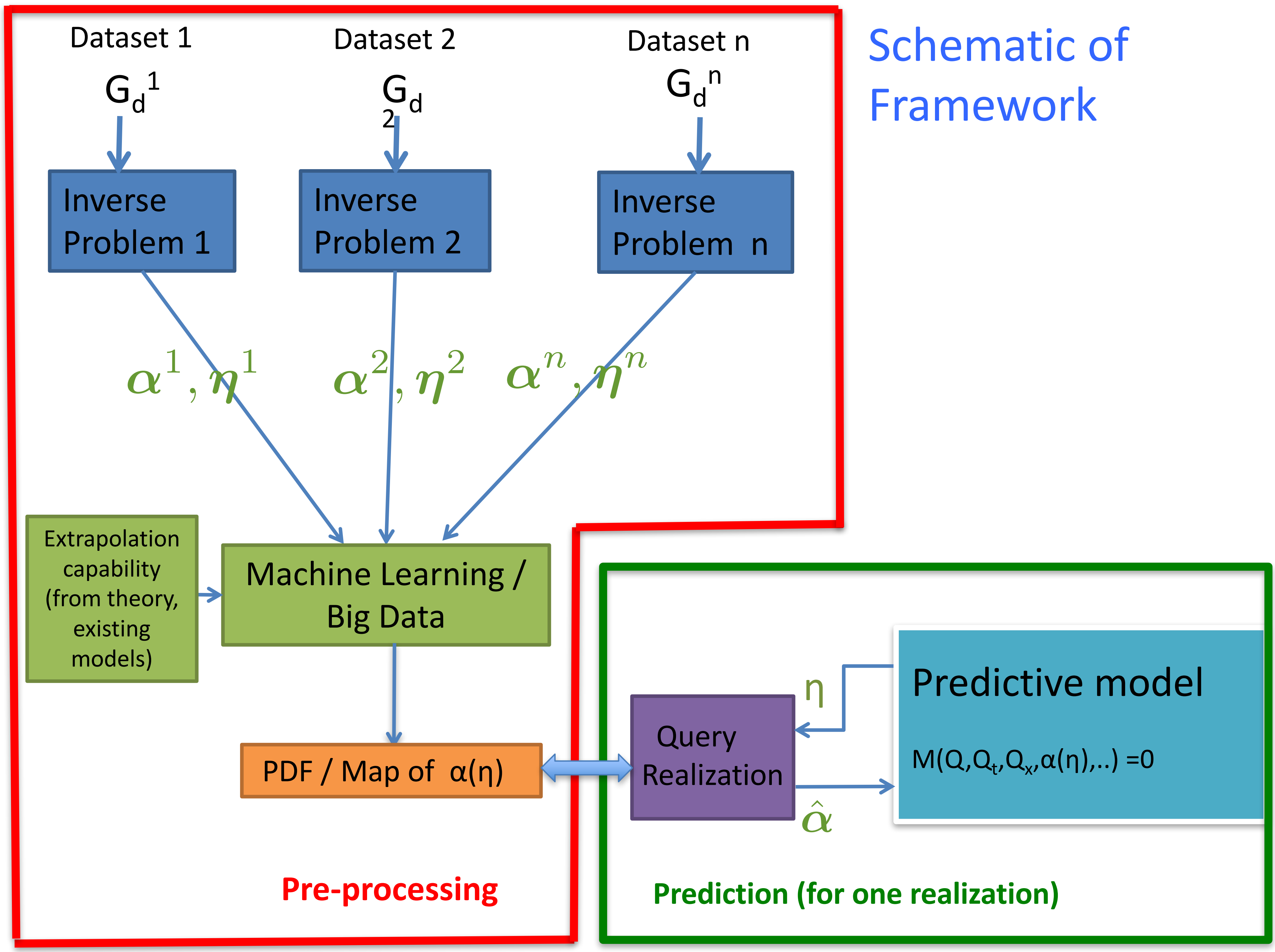
Barycentric map perturbations:
Iaccarino '12
Duraisamy '13
Xiao '15



Prediction with Machine-Learning Injection (Re_τ = 950)

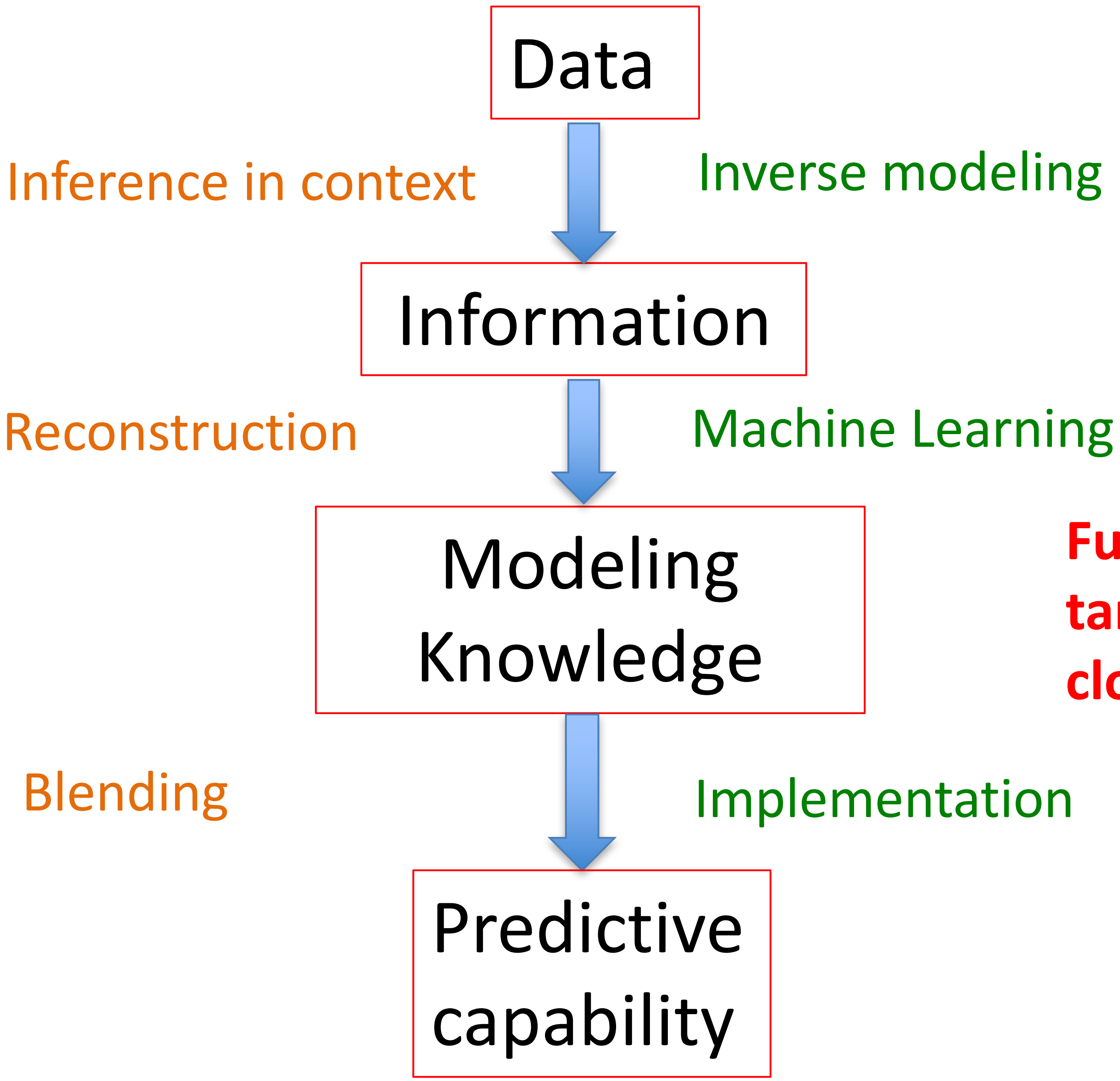


$$\eta = \{Sk/\epsilon, P/\epsilon, y\sqrt{k}/\nu\}$$



Schematic of Framework
Dataset 1
G_d^1
Dataset 2
G_d^2
Dataset n
G_d^n
Inverse Problem 1
Inverse Problem 2
Inverse Problem n
α^1, η^1
α^2, η^2
α^n, η^n
Extrapolation capability (from theory, existing models)
Machine Learning / Big Data
PDF / Map of α(η)
Pre-processing
Query Realization
η
$\hat{\alpha}$
Predictive model
$M(Q,Q_t,Q_x,\alpha(\eta),..) = 0$
Prediction (for one realization)

Summary



Functional form is targeted and not just closure parameters

Disclaimer on RANS models

- Single point closures based on local `well-behaved' quantities
  - ➔ Miss out on spectral and structural information
  - ➔ Do not process disparity of turbulence scales
  - ➔ Cannot distinguish inactive motions and low frequency unsteadiness

- But, room for improvement is vast