Data-driven Predictive Modeling of Turbulent Flows : Current Status and Challenges

Karthik Duraisamy

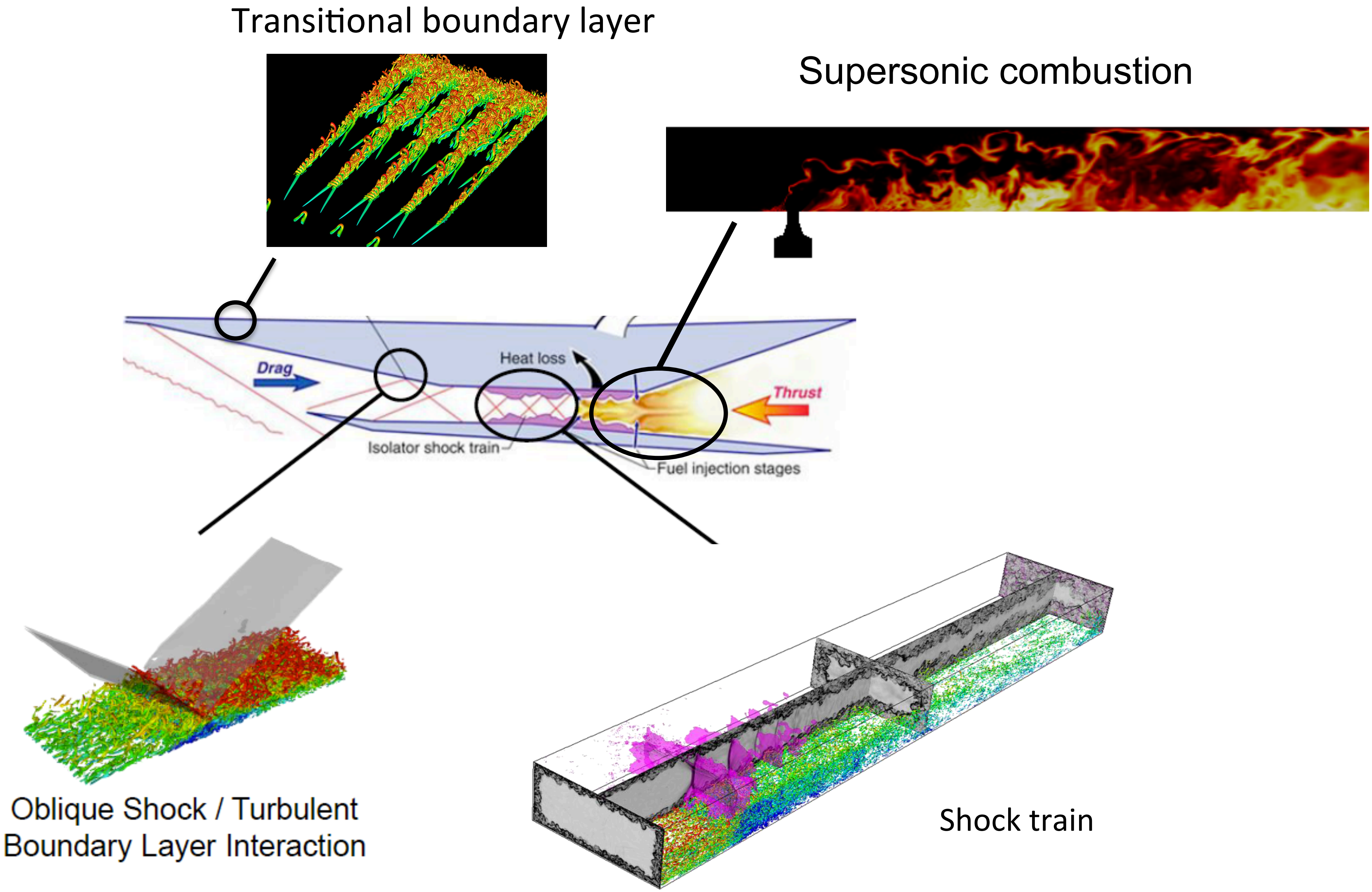
AEROSPACE ENGINEERING
UNIVERSITY of MICHIGAN

Thanks to:

A.P. Singh, B. Tracey, S. Medida, J. Alonso, P. Durbin

Aug 17 2016, Langley

Turbulence : a "tyranny of scales"



Turbulence Modeling: Coarse-graining the Navier-Stokes equations

NSE (DNS)

$$\rho \frac{\partial u_i}{\partial t} + \rho \frac{\partial u_i u_j}{\partial x_j} = \rho f_i + \frac{\partial}{\partial x_j}\left[-p + 2\mu S_{ij}\right]$$

Decompose

$$u_i = \bar{u}_i + u_i' \quad ; \quad p = \bar{p} + p'$$

Averaged (RANS)

$$\rho \frac{\partial \bar{u}_i}{\partial t} + \rho \frac{\partial \bar{u}_i \bar{u}_j}{\partial x_j} = \rho \bar{f}_i + \frac{\partial}{\partial x_j}\left[-\bar{p} + 2\mu \bar{S}_{ij} - \rho \overline{u_i' u_j'}\right]$$

The closure is

$$\tau_{ij}^{RANS} = -\rho \overline{u_i' u_j'}$$

Modeling the Reynolds stress tensor

Exact

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

Approx

- Models found lacking in accuracy in many complex flows
- It is the balance between the terms that matters (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
- Start with clean idea, but loss of rigor in final model
- Model constants calibrated on very limited data

Our approach

We propose large-scale data-driven to enable the construction of accurate models of turbulence

- ➔ Focus on non-parametric (functional) improvements
- ➔ Not replacing existing modeling knowledge, but just building on it

This technique enables

- ➔ The ability to "infer" what's missing in the closure
- ➔ The ability to convert that inference into modeling knowledge

A timeline of data in turbulence modeling

UQ perspective

Moser et al (2011-2013)	: Bayesian analysis and model averaging (coefficients)
Wang et al (2012)	: Infer correction to eddy viscosity coefficient
Tracey, Duraisamy, Alonso (2012)	: Machine Learning of Reynolds stress perturbations
Edeling (2014)	: Infer uncertainty in model coefficients
Ling et al (2015)	: Machine Learning to determine regions of model error
Xiao et al (2015)	: Infer reynolds stress perturbations

Modeling perspective

Duraisamy et al. (2014)	: Inversion and machine learning for Turbulence modeling
Tracey, Duraisamy, Alonso (2015)	: Machine learning + embedding
Xiao et al (2016)	: Inference + learning of Reynolds stress perturbations
Duraisamy (2016)	: Inference + machine learning + embedding

Perturbing eddy viscosity

UQ perspective

Moser et al (2011-2013)	: Bayesian analysis and model averaging (coefficients)
Wang et al (2012)	: Infer correction to eddy viscosity coefficient
Tracey, Duraisamy, Alonso (2012)	: Machine Learning of Reynolds stress perturbations
Edeling (2014)	: Infer uncertainty in model coefficients
Ling et al (2015)	: Machine Learning to determine regions of model error
Xiao et al (2015)	: Infer reynolds stress perturbations

Modeling perspective

Duraisamy et al. (2014)	: Inversion and machine learning for Turbulence modeling
Tracey, Duraisamy, Alonso (2015)	: Machine learning + embedding
Xiao et al (2016)	: Inference + learning of Reynolds stress perturbations
Duraisamy (2016)	: Inference + machine learning + embedding

$$R_p = 2(\mu_t + \delta)S_{ij} - \frac{2}{3}\rho k \delta_{ij}$$

Perturbing Reynolds stress tensor

UQ perspective

Moser et al (2011-2013)	: Bayesian analysis and model averaging (coefficients)
Wang et al (2012)	: Infer correction to eddy viscosity coefficient
Tracey, Duraisamy, Alonso (2012)	: Machine Learning of Reynolds stress perturbations
Edeling (2014)	: Infer uncertainty in model coefficients
Ling et al (2015)	: Machine Learning to determine regions of model error
Xiao et al (2015)	: Infer reynolds stress perturbations

Modeling perspective

Duraisamy et al. (2014)	: Inversion and machine learning for Turbulence modeling
Tracey, Duraisamy, Alonso (2015)	: Machine learning + embedding
Xiao et al (2016)	: Inference + learning of Reynolds stress perturbations
Duraisamy (2016)	: Inference + machine learning + embedding

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

Structure proposed by Emory & Iaccarino (2012)

Introduce corrective terms in the model

UQ perspective

Moser et al (2011-2013) : Bayesian analysis and model averaging (coefficients)
Wang et al (2012) : Infer correction to eddy viscosity coefficient
Tracey, Duraisamy, Alonso (2012) : Machine Learning of Reynolds stress perturbations
Edeling (2014) : Infer uncertainty in model coefficients
Ling et al (2015) : Machine Learning to determine regions of model error
Xiao et al (2015) : Infer reynolds stress perturbations

Modeling perspective

Duraisamy et al. (2014) : Inversion and machine learning for Turbulence modeling
Tracey, Duraisamy, Alonso (2015) : Machine learning + embedding
Xiao et al (2016) : Inference + learning of Reynolds stress perturbations
Duraisamy (2016) : Inference + machine learning + embedding

$$\frac{DQ}{Dt} = \mathcal{R}(Q) + \delta(Q) \rightarrow \frac{DQ}{Dt} = \beta(Q)\mathcal{R}(Q) \rightarrow \beta \equiv 1 + \frac{\delta}{R}$$

Outline

- Proof-of-concept
  - ➔ If there is a known underlying model, can we discover it?
- The general framework
  - ➔ How do we setup the data-driven turbulence modeling problem
- Demonstration
  - ➔ Predictions in Airfoil flows
- Computer Science, Scaling, etc…

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

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

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

Convection — Production — Destruction — Diffusion — Cross Production

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

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

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

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

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

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

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

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

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

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

Proof-of-concept : Replicating Spalart Allmaras Model

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

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

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

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

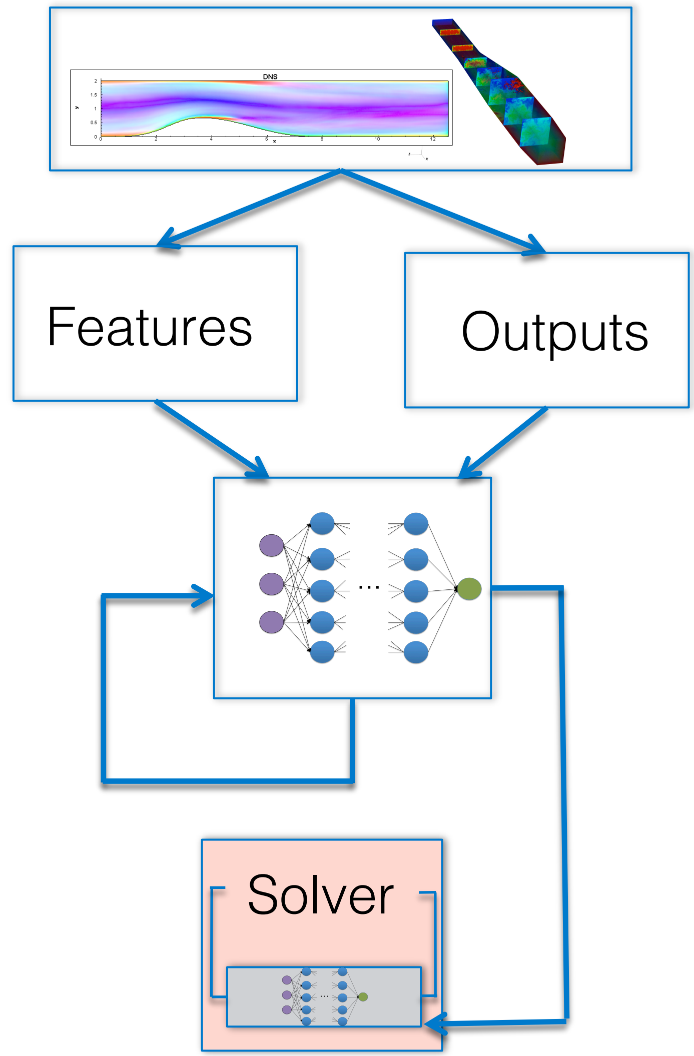
$$s_{cp} = \frac{c_{b2}}{\sigma}\frac{\partial \hat{\nu}}{\partial x_i}\frac{\partial \hat{\nu}}{\partial x_i}$$

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

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

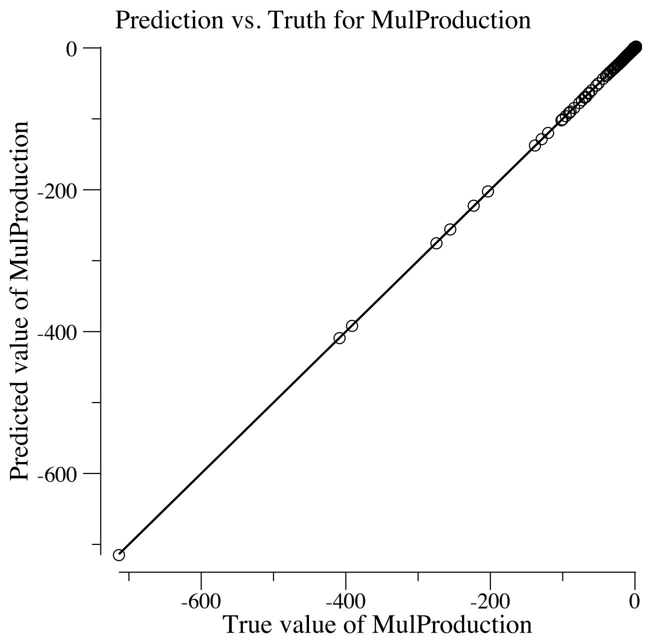
Procedure

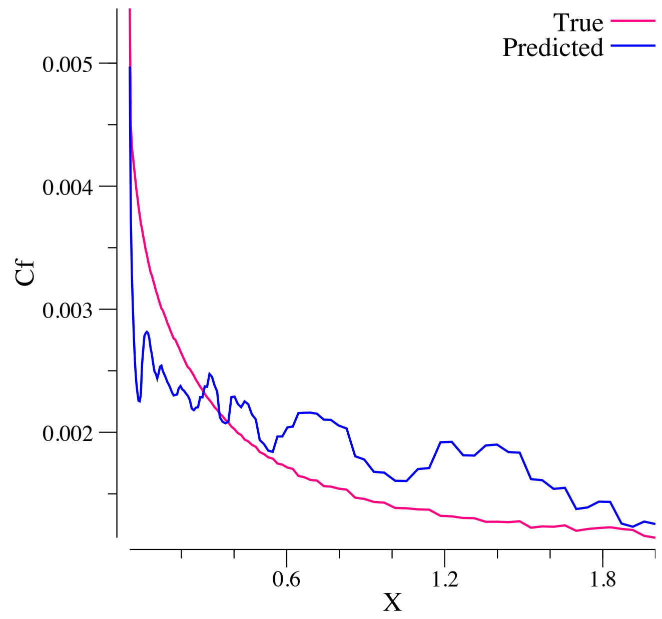
1) Select representative datasets
   - Flat plates, pressure-driven channels, airfoils
2) Choose and extract input and output features
   - Spalart-Allmaras quantities
3) Select learning algorithm
   - Neural network
4) Train learning algorithm
   - BFGS optimizer
5) Embed learned model within flow solver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

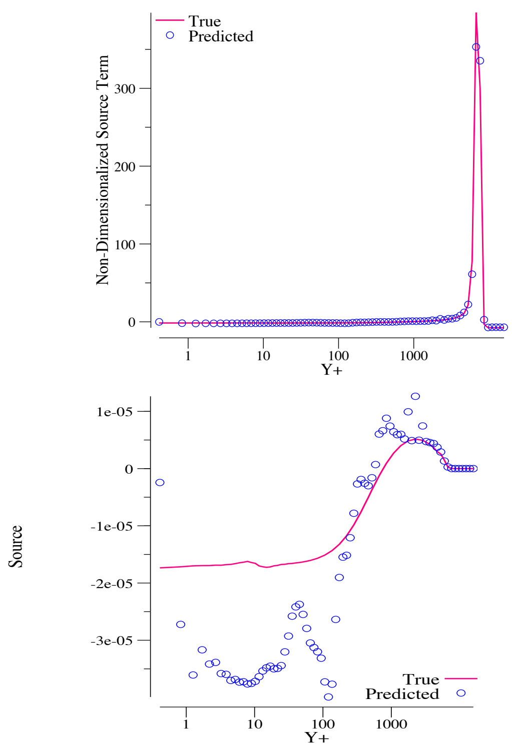
- Dimensionalized Squared-Error

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

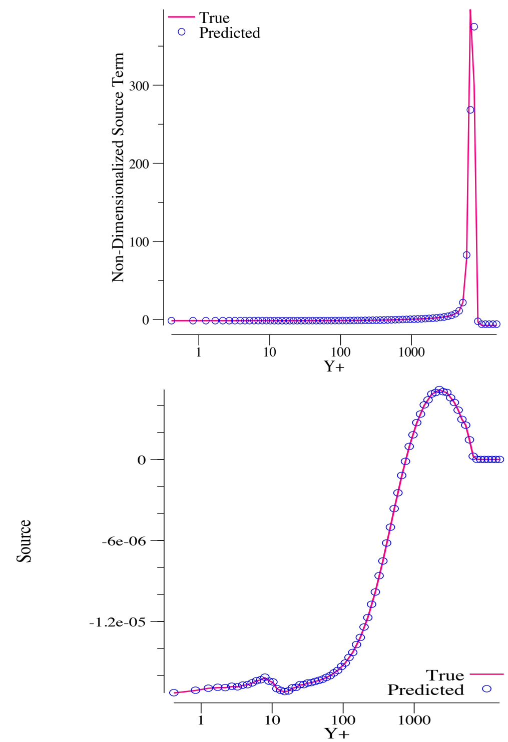
  - Penalizes differences in the dimensional output value

The loss function

Squared-distance Loss



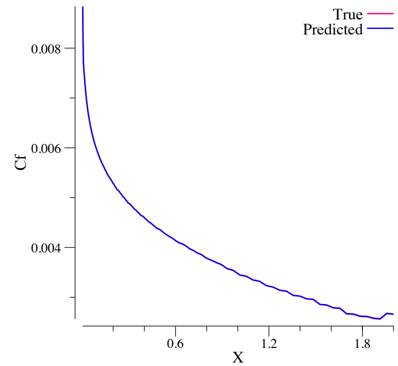
Dimensionalized Loss



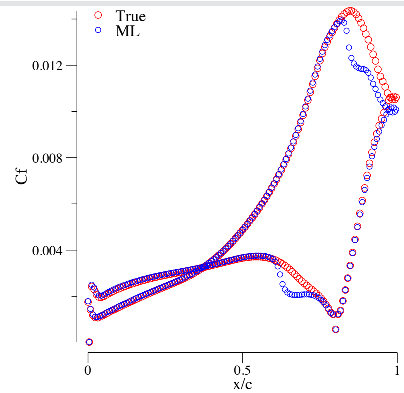
- Must align loss function with CFD environment

Test cases

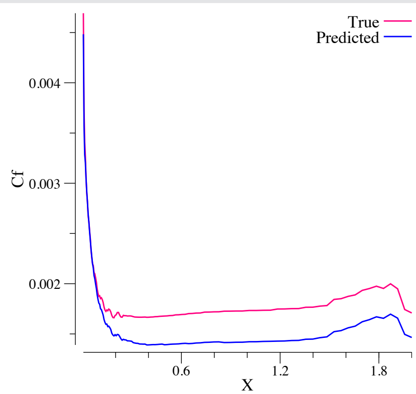
- G: No major difference



- F: Small region of difference



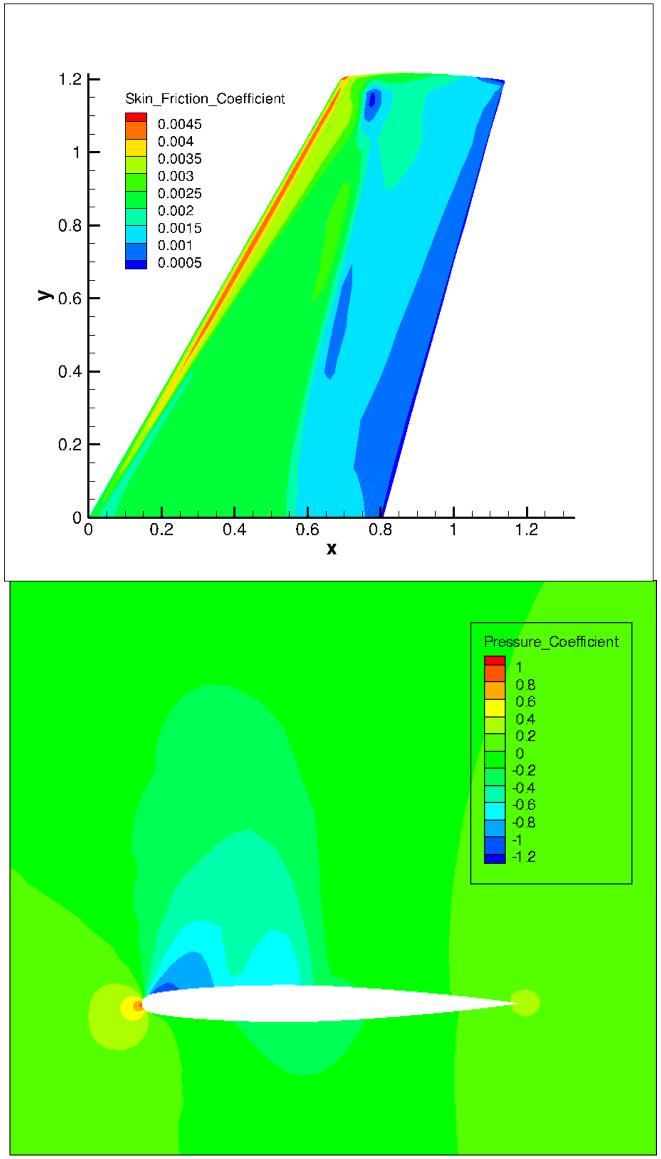
- P: Large discrepancy



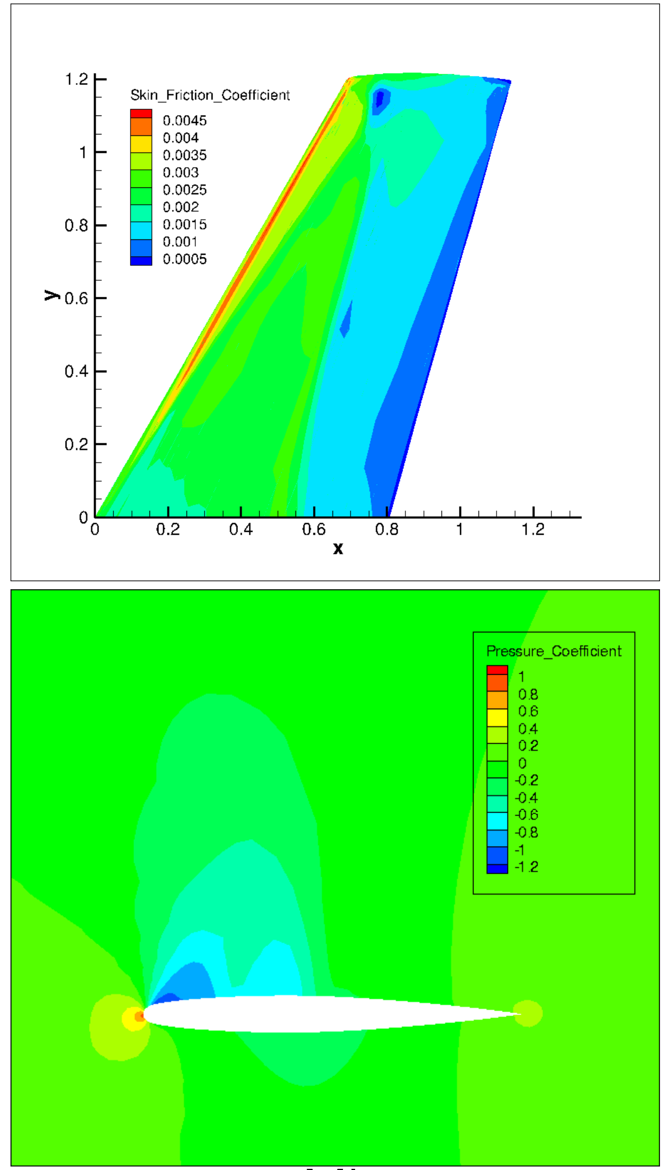
	Dest.	F_w	Mul. Dest.	Mul. Prod.	Prod.	Source
Flatplate 3e6	G	G	G	G	G	G
Flatplate 4e6	G	G	G	G	G	G
Flatplate 5e6	G	G	G	G	G	G
Flatplate 6e6	G	G	G	G	G	G
Flatplate 7e6	G	G	G	G	G	G
Channel $C_p = -0.3$	G	G	G	G	G	F
Channel $C_p = -0.1$	G	G	G	G	G	F
Channel $C_p = -0.03$	G	G	G	G	G	F
Channel $C_p = -0.01$	G	G	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 - 1

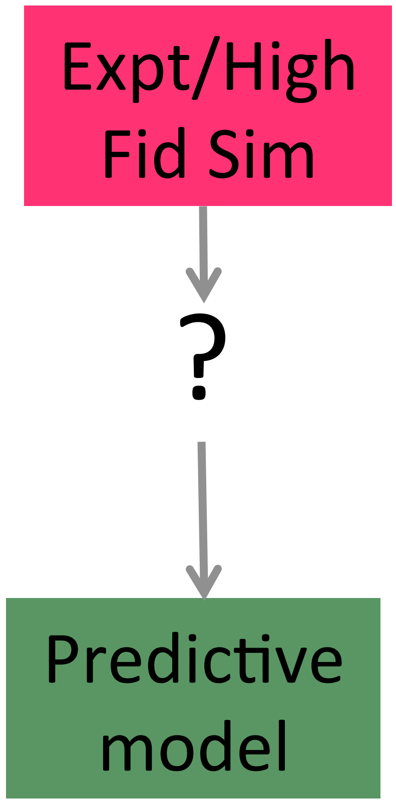
- Feature Scaling is important
- Testing within the CFD solver
- Alignment of loss function

IF THERE IS AN UNDERLYING MODEL, IT IS POSSIBLE TO DISCOVER IT

Outline

- Proof-of-concept
  ➔ If there is a known underlying model, can we discover it?
- The general framework
  ➔ How do we setup the data-driven turbulence modeling problem
- Demonstration
  ➔ Predictions in Airfoil flows
- Computer Science, Scaling, etc…

On to the real problem



- Data is not available in a form or context that is immediately useful,
- Right data isn't available
- Generalizing specific information into modeling knowledge is hard
- Uncertainties abound
- NO proof that there is an underlying model waiting to be discovered

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

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

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

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

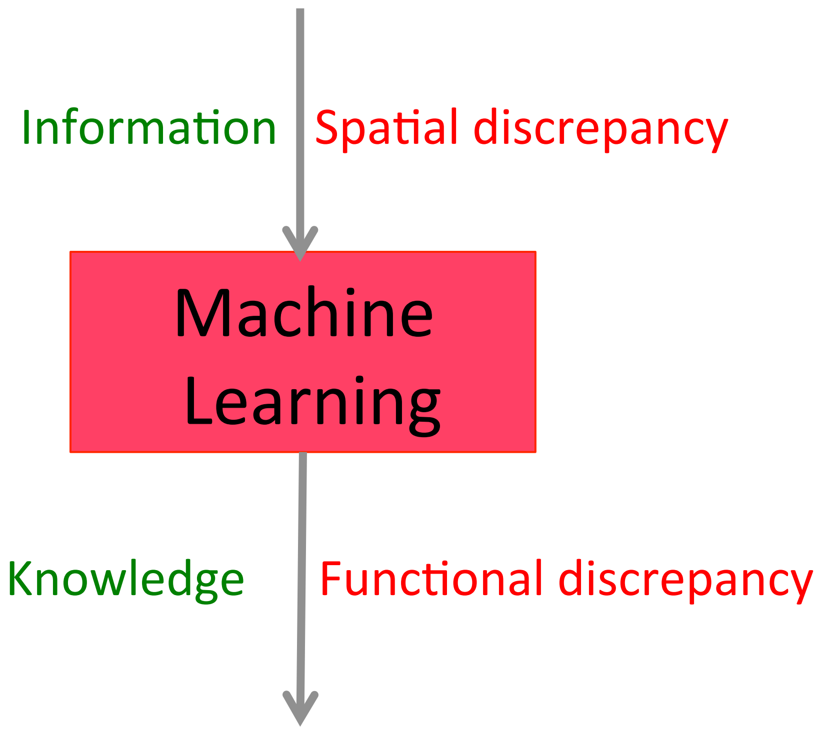
$$\hat{\delta}(\, f(Q) \,)$$

Embedding

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

Prediction : Injection into solver

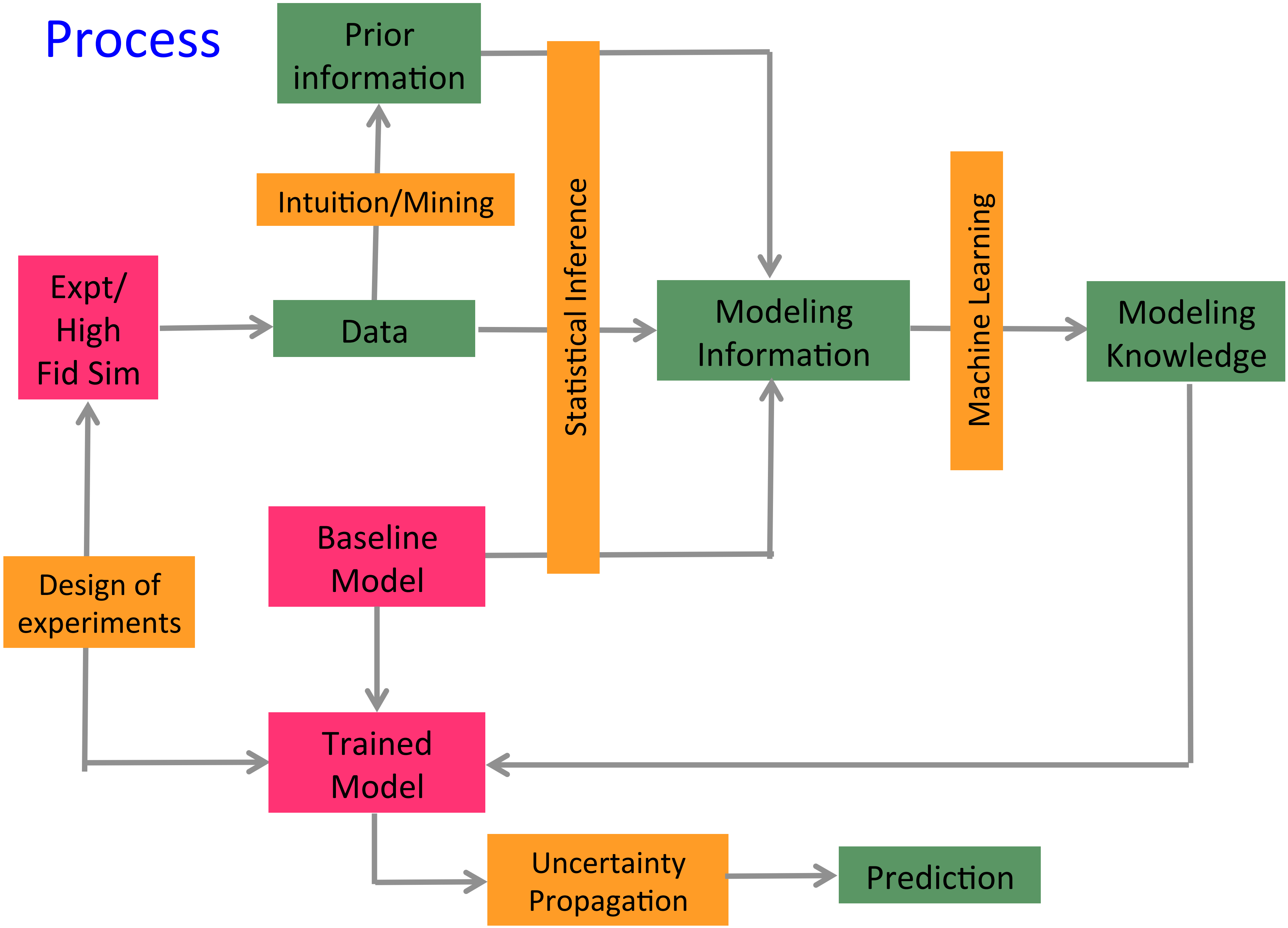
Major insight from NASA LEARN project



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

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

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

2) Design of Experiments

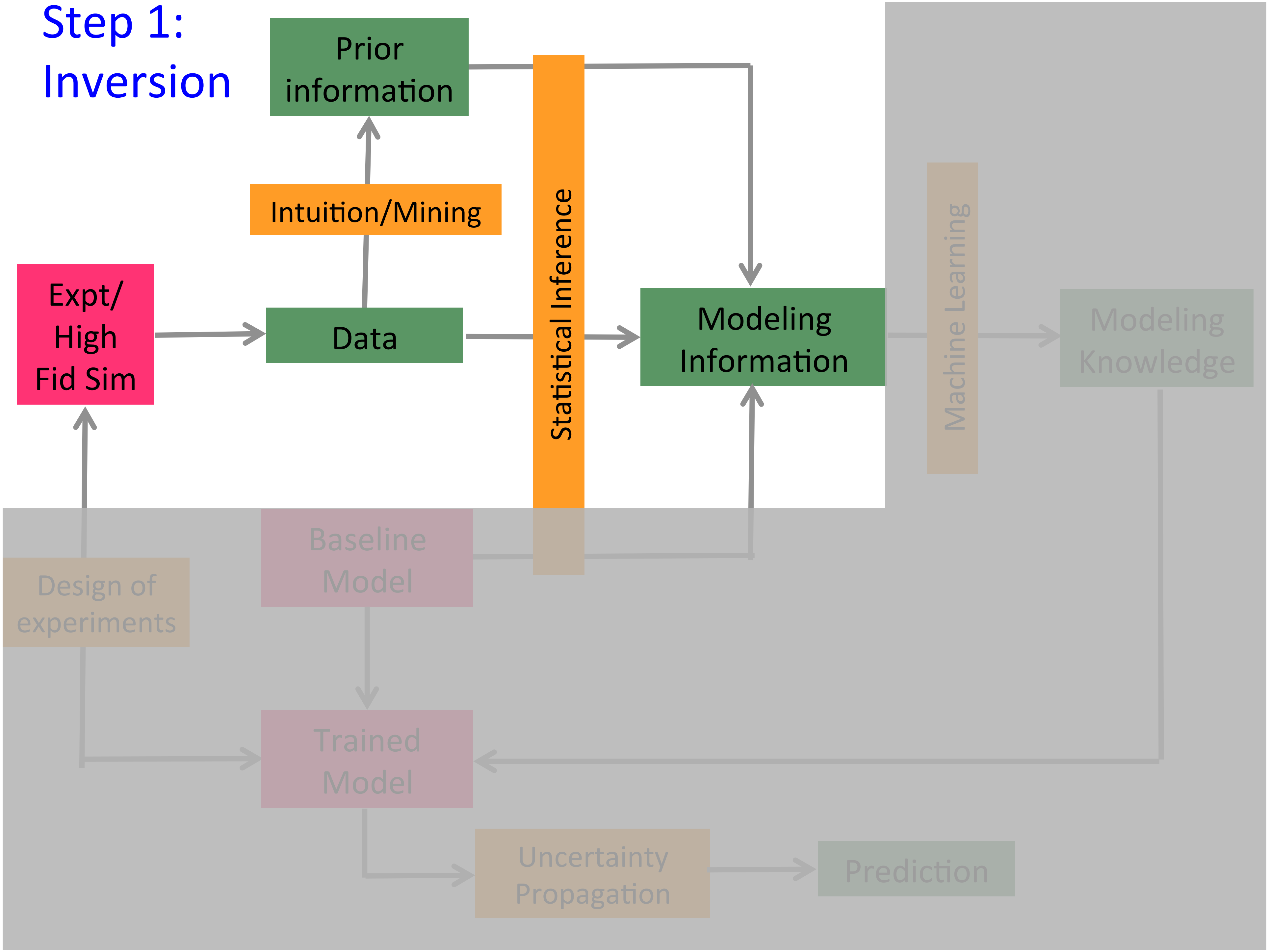
3) Machine Learning

4) Prediction

Outline

- Proof-of-concept
  - ➔ If there is a known underlying model, can we discover it?
- The general framework
  - ➔ How do we setup the data-driven turbulence modeling problem
- Demonstration
  - ➔ Predictions in Airfoil flows
- Computer Science, Scaling, etc…

Step 1: Inversion



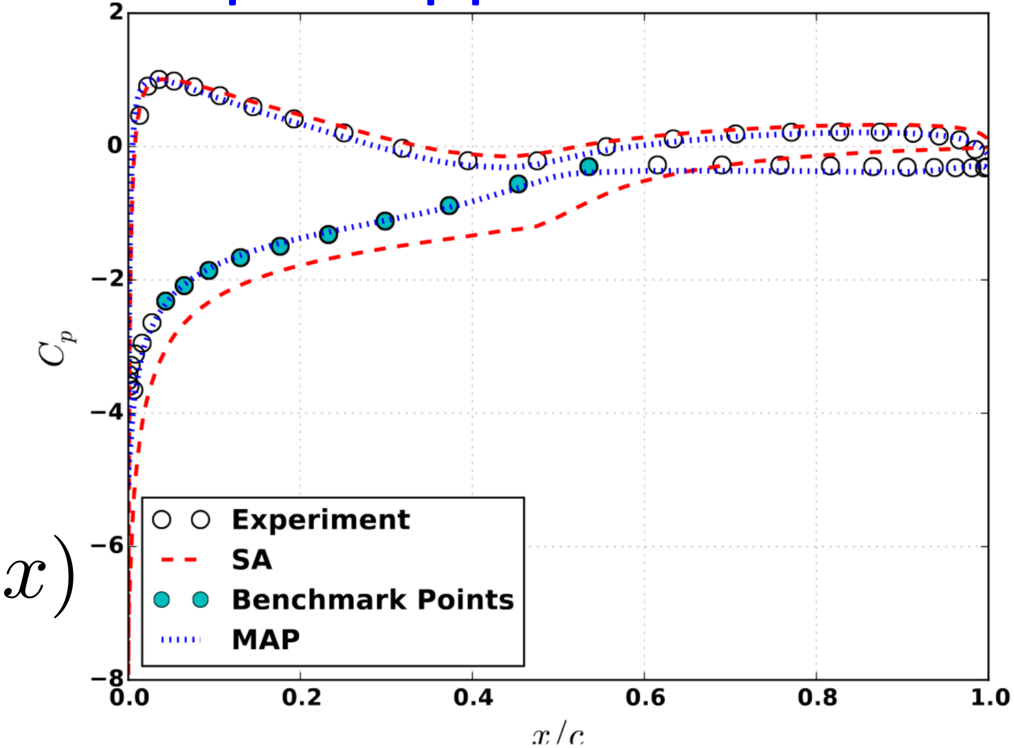
Example: Application to airfoil

Original Model

$$\frac{D\tilde{\nu}}{Dt} = P - D + T$$

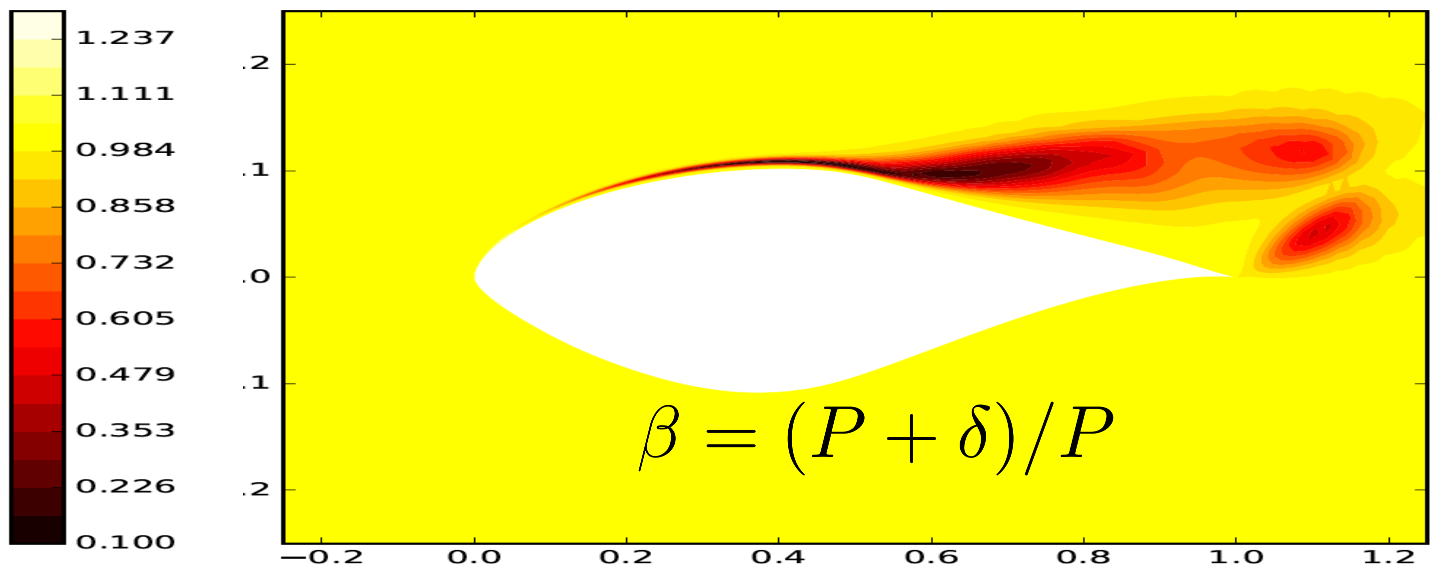
Introduce correction

$$\frac{D\tilde{\nu}}{Dt} = P - D + T + \delta(x)$$



Data used:

Wall pressure



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

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,

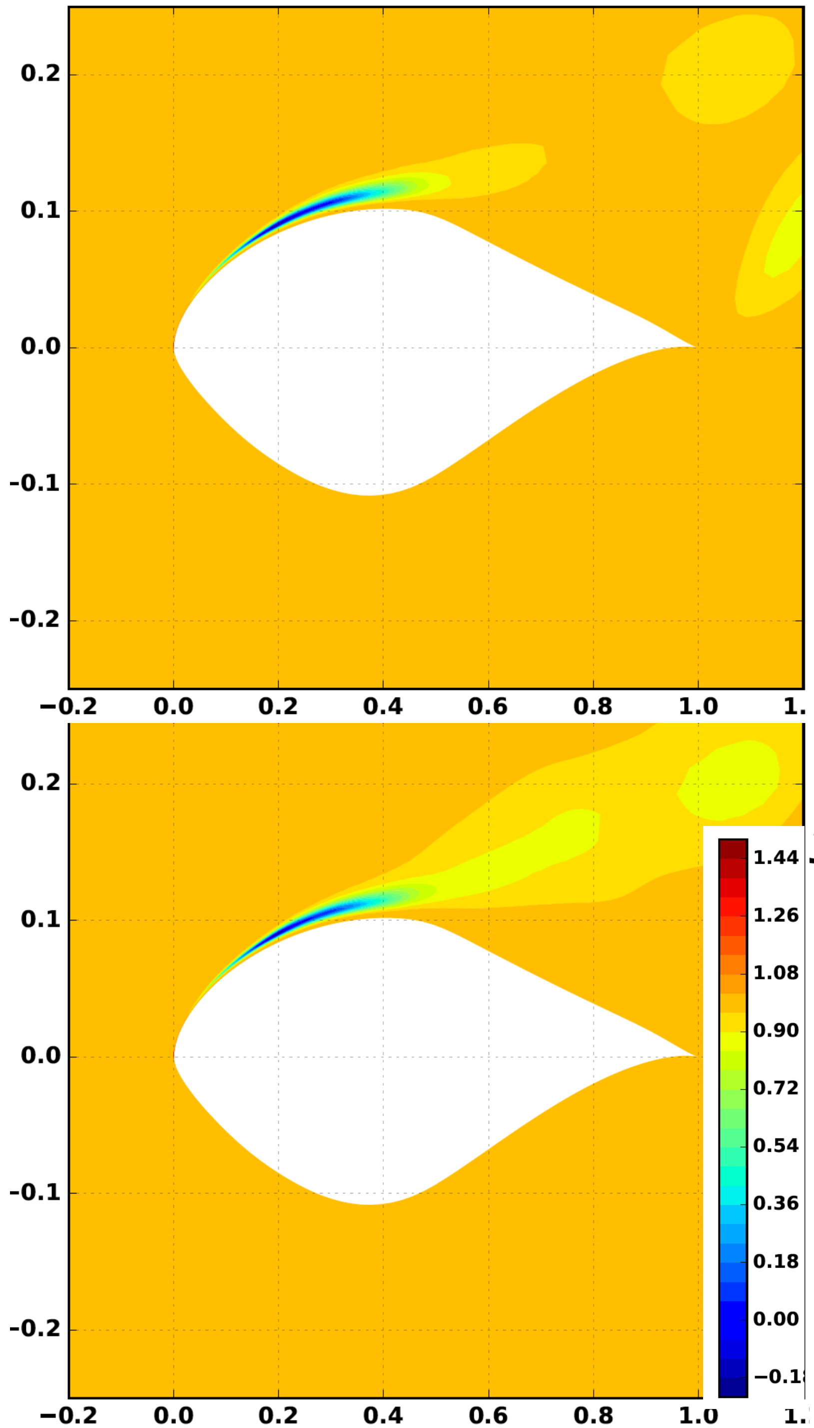
$$\nu_{i,n}\frac{\partial R_m}{\partial u_n} = -\frac{\partial R_m}{\partial\boldsymbol{\beta}_i}$$

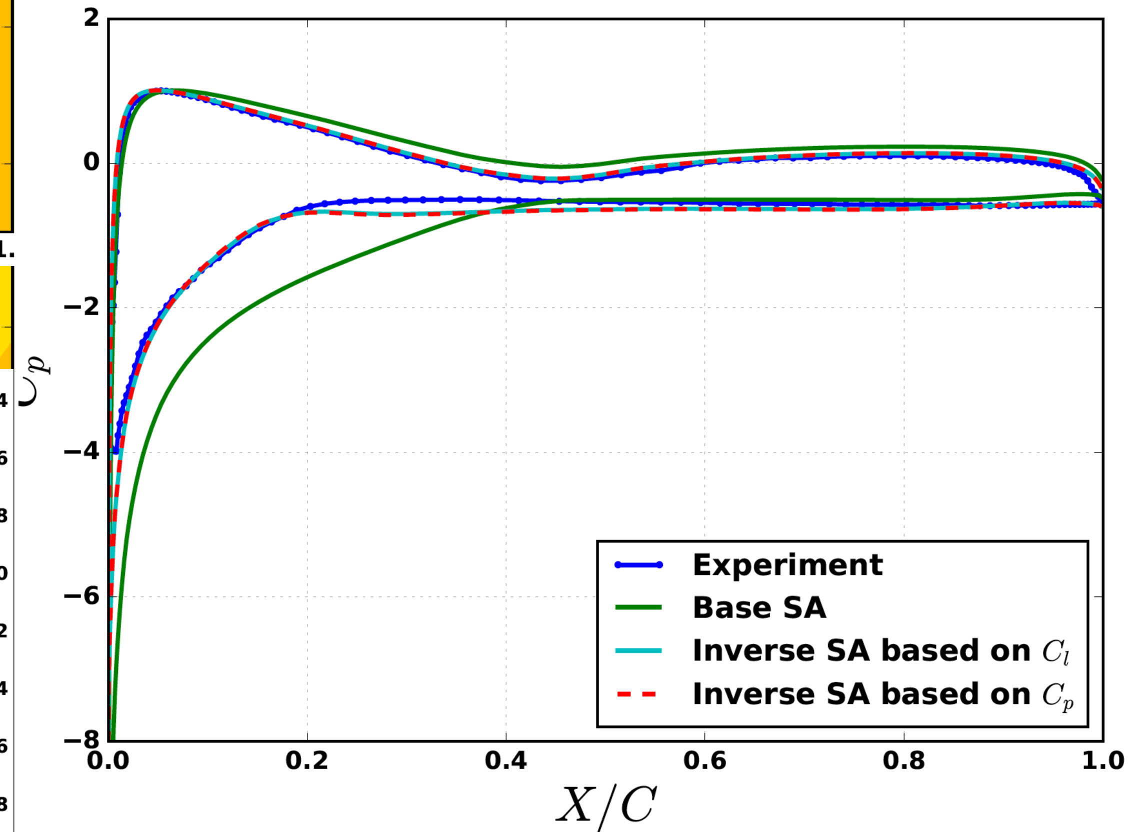
$$\mu_{i,m}\frac{\partial R_m}{\partial u_k} = -\frac{\partial^2 F}{\partial\boldsymbol{\beta}_i\partial u_k} - \psi_m\frac{\partial^2 R_m}{\partial\boldsymbol{\beta}_i\partial u_k} - \nu_{i,n}\frac{\partial^2\mathfrak{J}}{\partial u_n\partial u_k} - \nu_{i,n}\psi_m\frac{\partial^2 R_m}{\partial u_n\partial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

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

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

Inversion based on Pressures
vs
Inversion based on LIFT!





Ability to work on sparse amount of data is critical

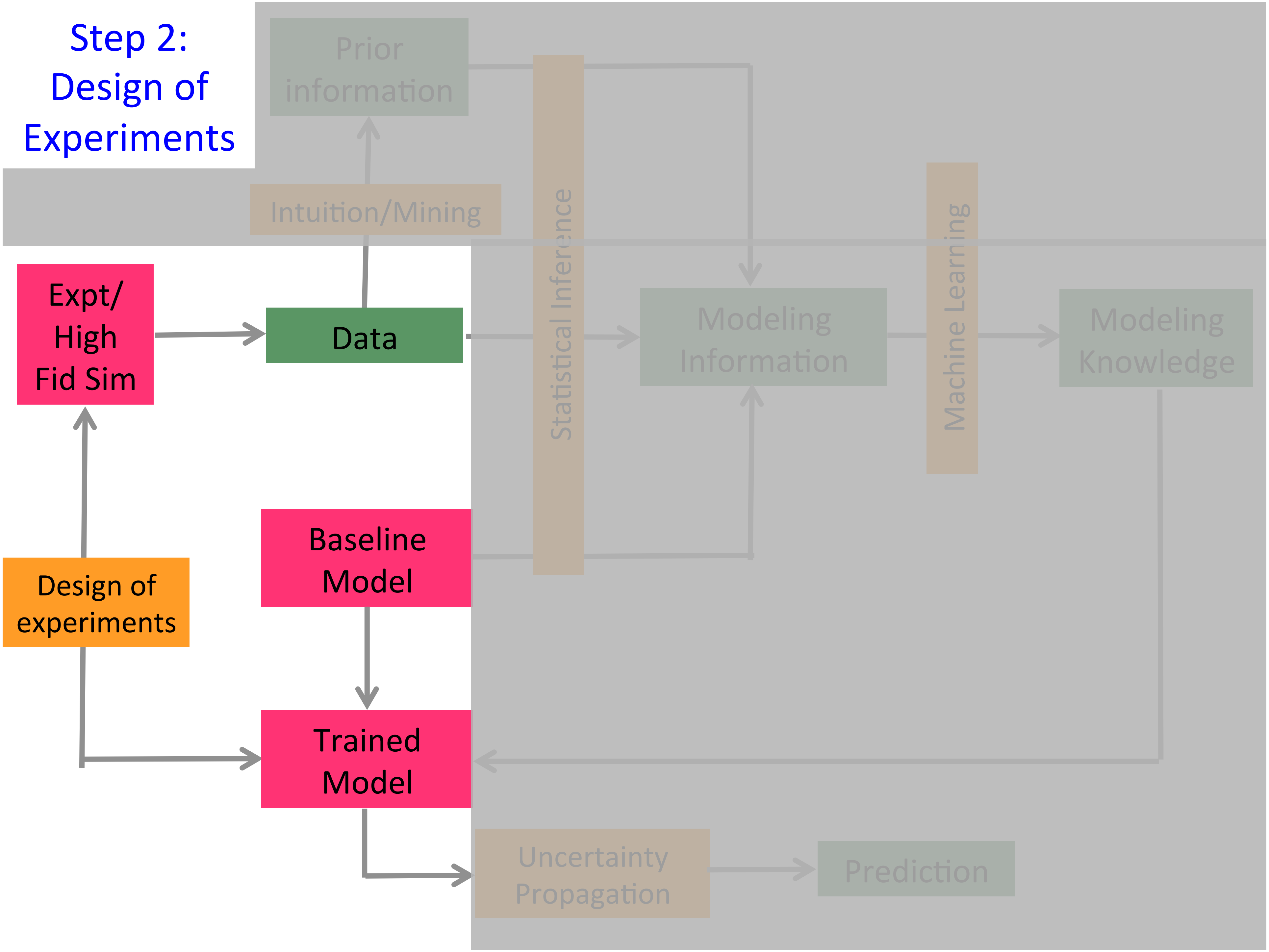
Other ways of introducing discrepancies

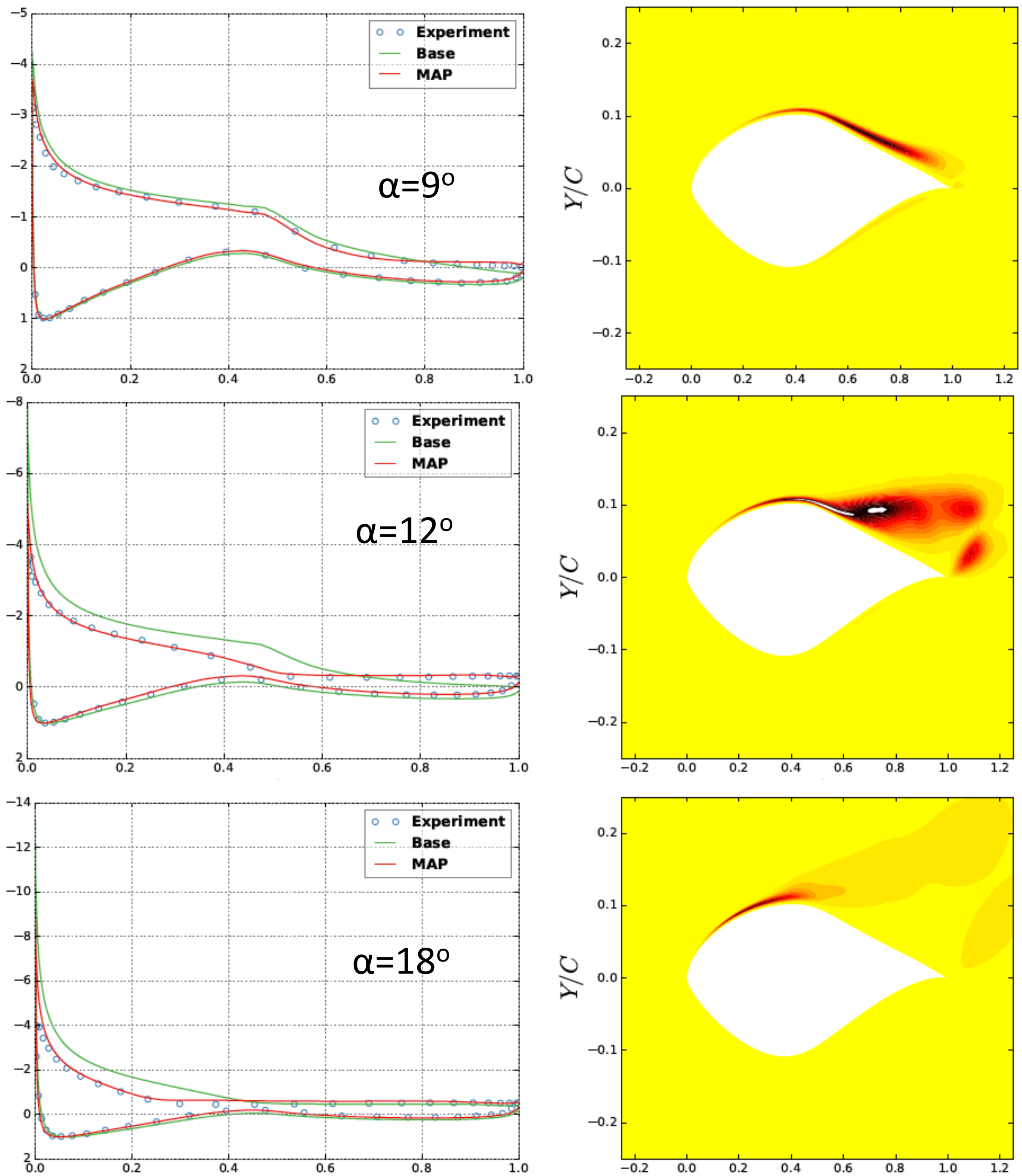
$$\frac{DR_{ij}}{Dt} = C_{ij} + P_{ij} + T_{ij} + \Pi_{ij} + D_{ij} + \beta(x)_{ij}\epsilon_{ij}$$

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

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

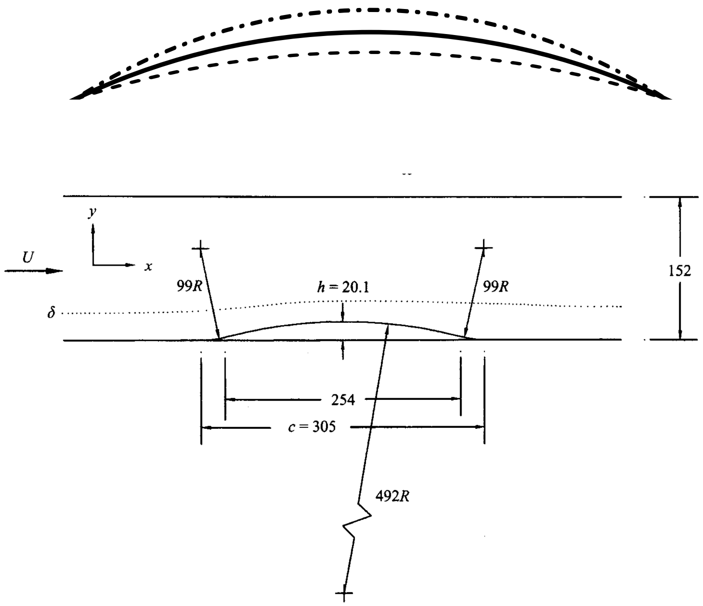
Step 2: Design of Experiments





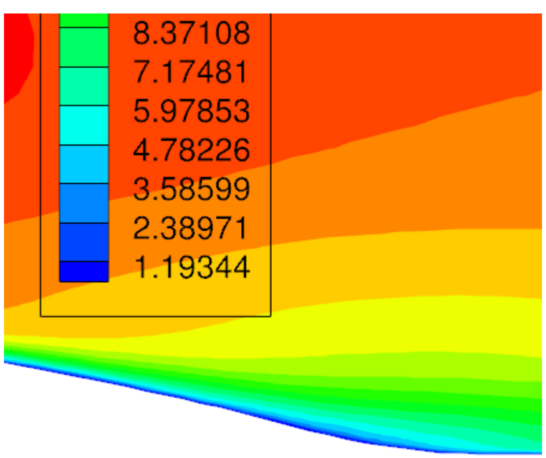
Application to Airfoil flows

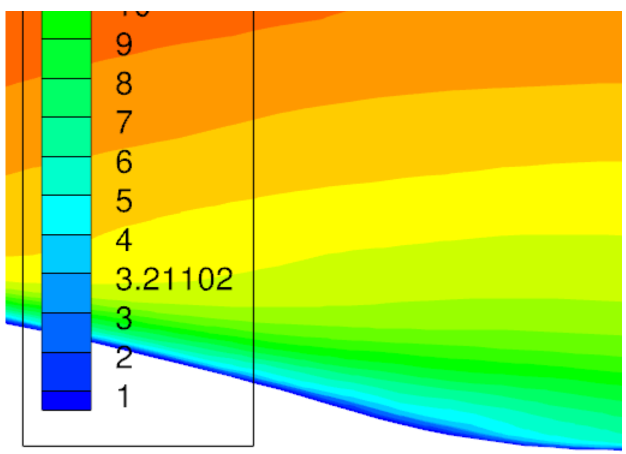
Application to flows over bumps

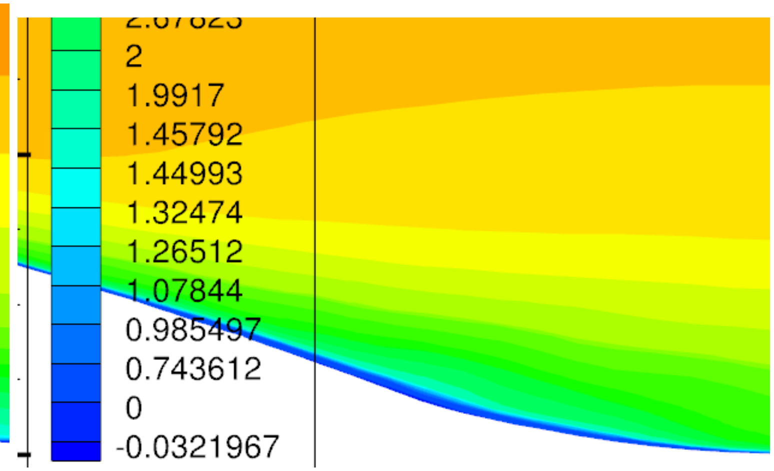


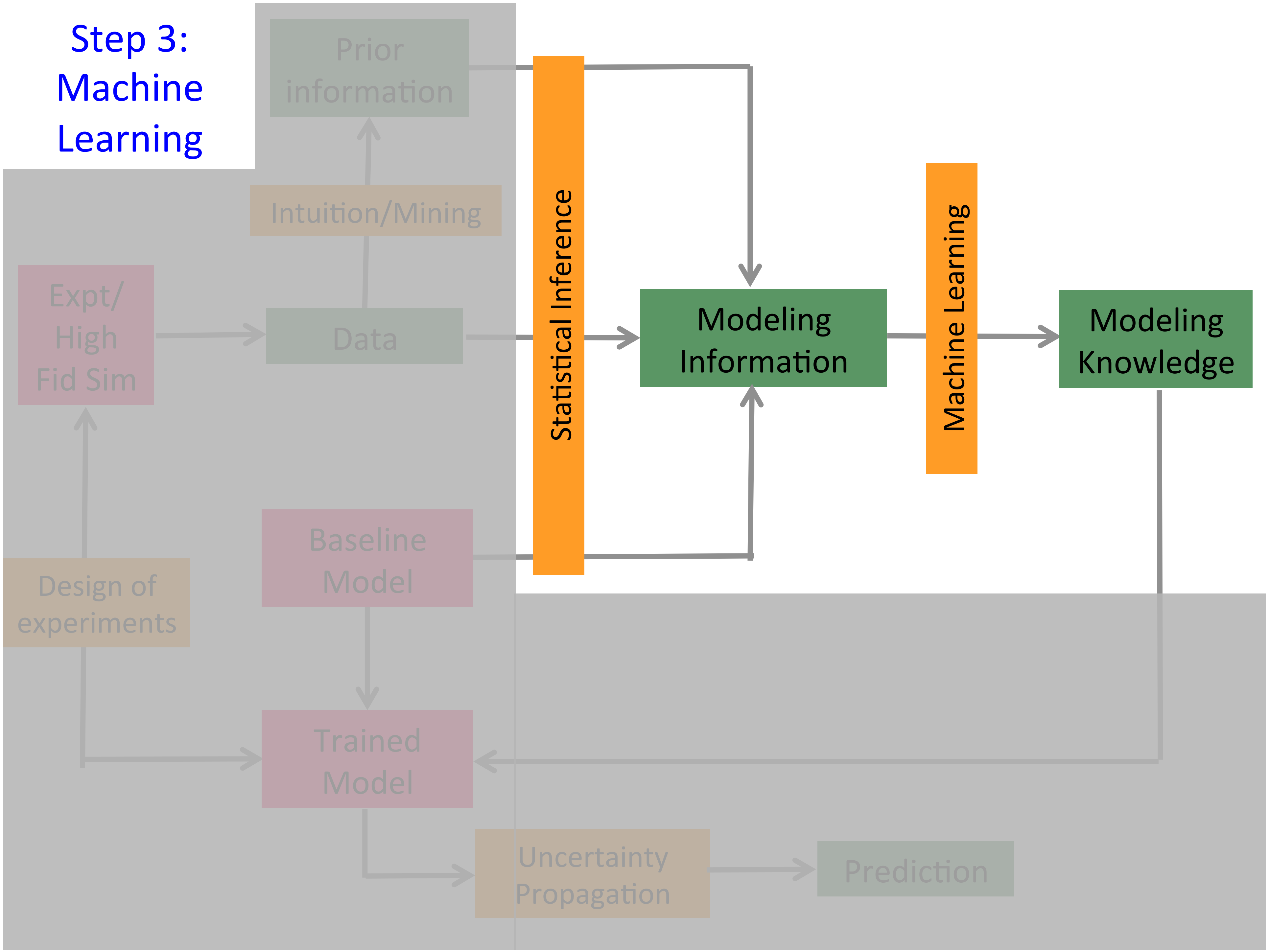
Large parameter sweep of different bump heights and Reynolds numbers

Targeting adverse pressure gradients and separation









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

How to transform information to knowledge?

$\beta^1(x, y)$

$\beta^2(x, y)$

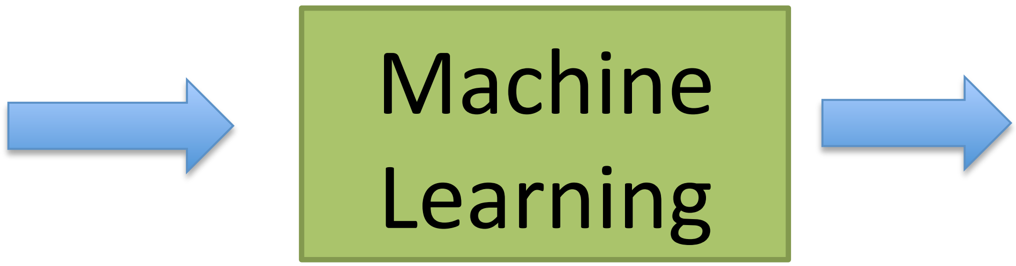
$\beta^3(x, y)$

$\beta^4(x, y)$

.....

.....

$\beta^n(x, y)$



$\eta_1, \eta_2, \ldots$

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

Selection of Features

Step 1: Look inside the baseline model

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

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

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

Step 2: Look for relevant physics

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

Step 3: Feature-subset selection*

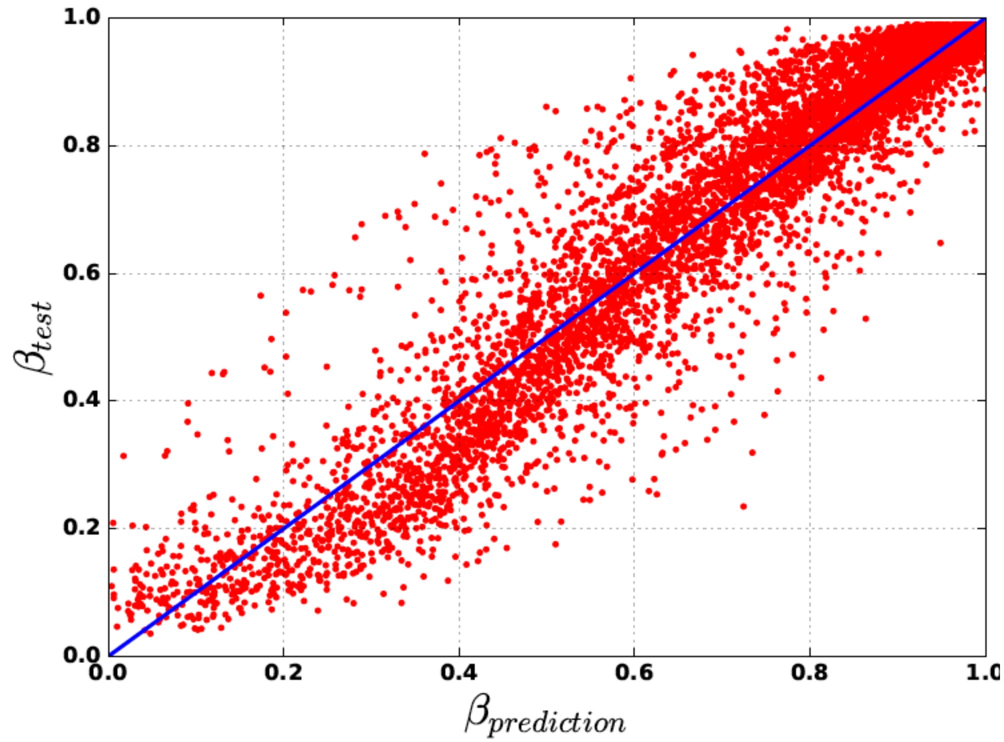
Hill-climbing algorithm

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

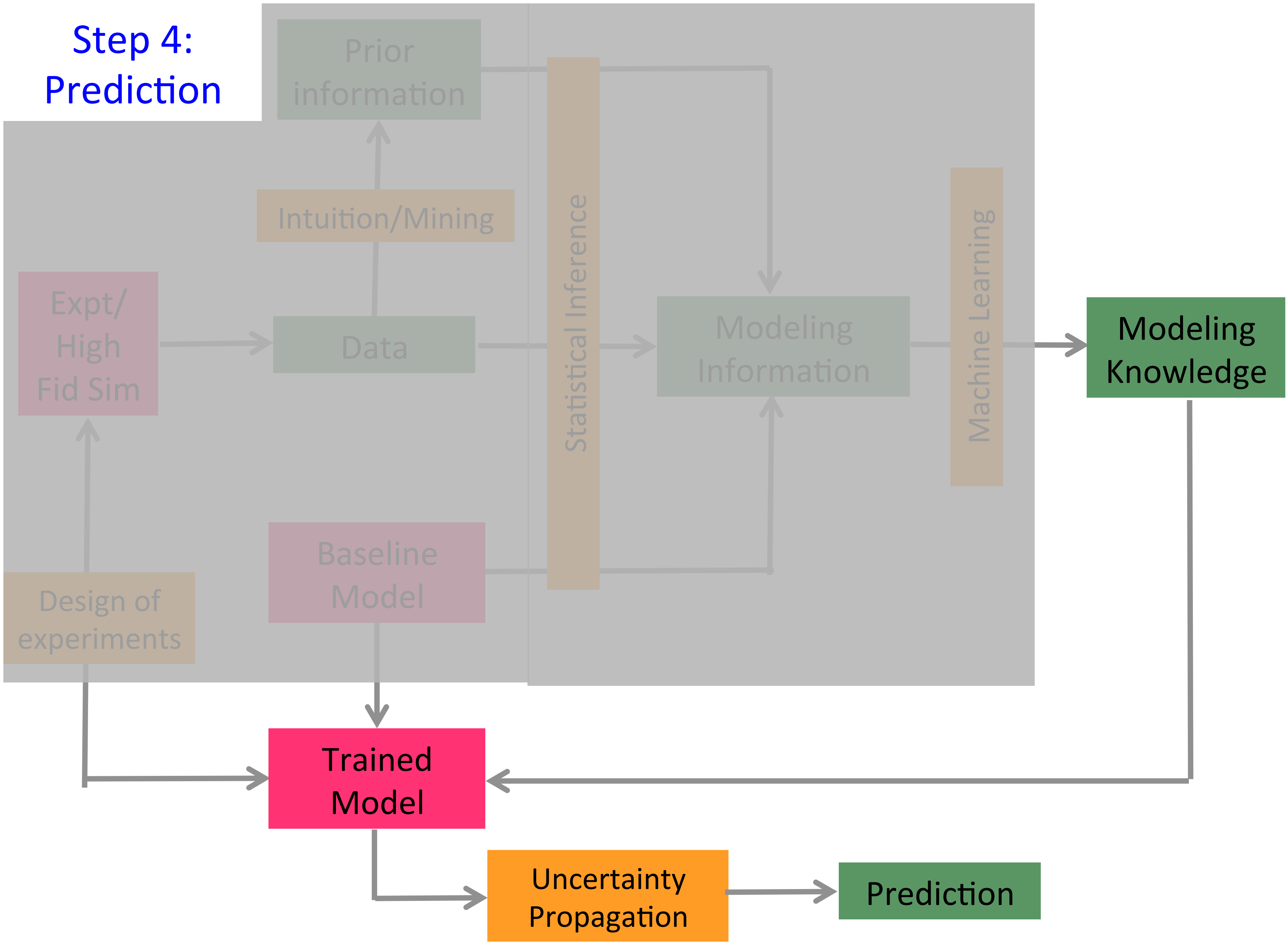
Features locally non-dimensional

Evaluation

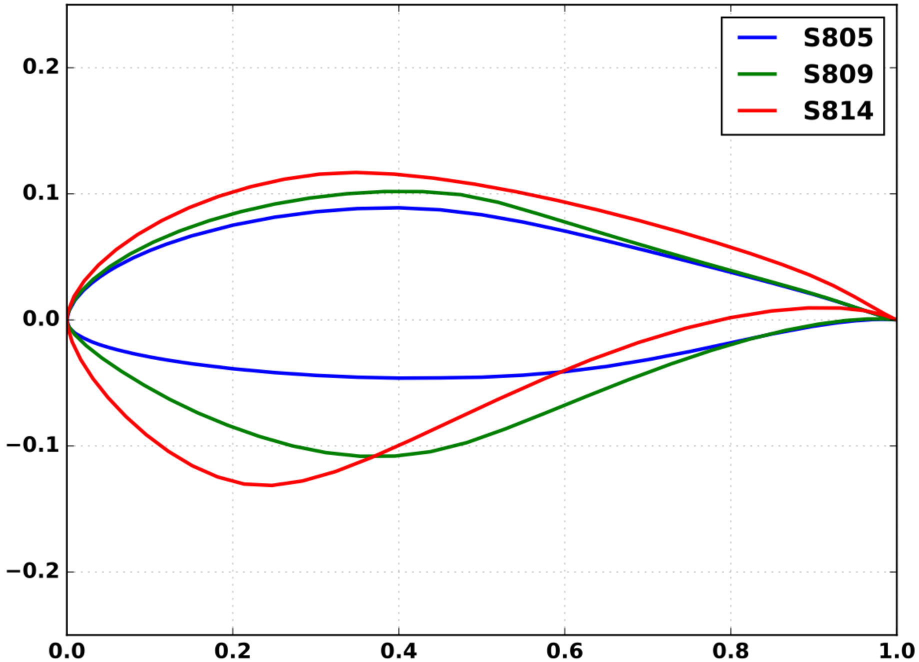
Neural Networks
GP regression
*Multiscale GP regression**
Symbolic regression



**Sparse Multiscale Gaussian Process Regression Using Hierarchical Clustering*, Z. Zhang, K. Duraisamy, N. Gumerov, Applied Numerical Mathematics 2016 (Submitted)



Step 4:
Prediction
Prior information
Intuition/Mining
Expt/ High Fid Sim
Data
Statistical Inference
Modeling Information
Machine Learning
Modeling Knowledge
Design of experiments
Baseline Model
Trained Model
Uncertainty Propagation
Prediction

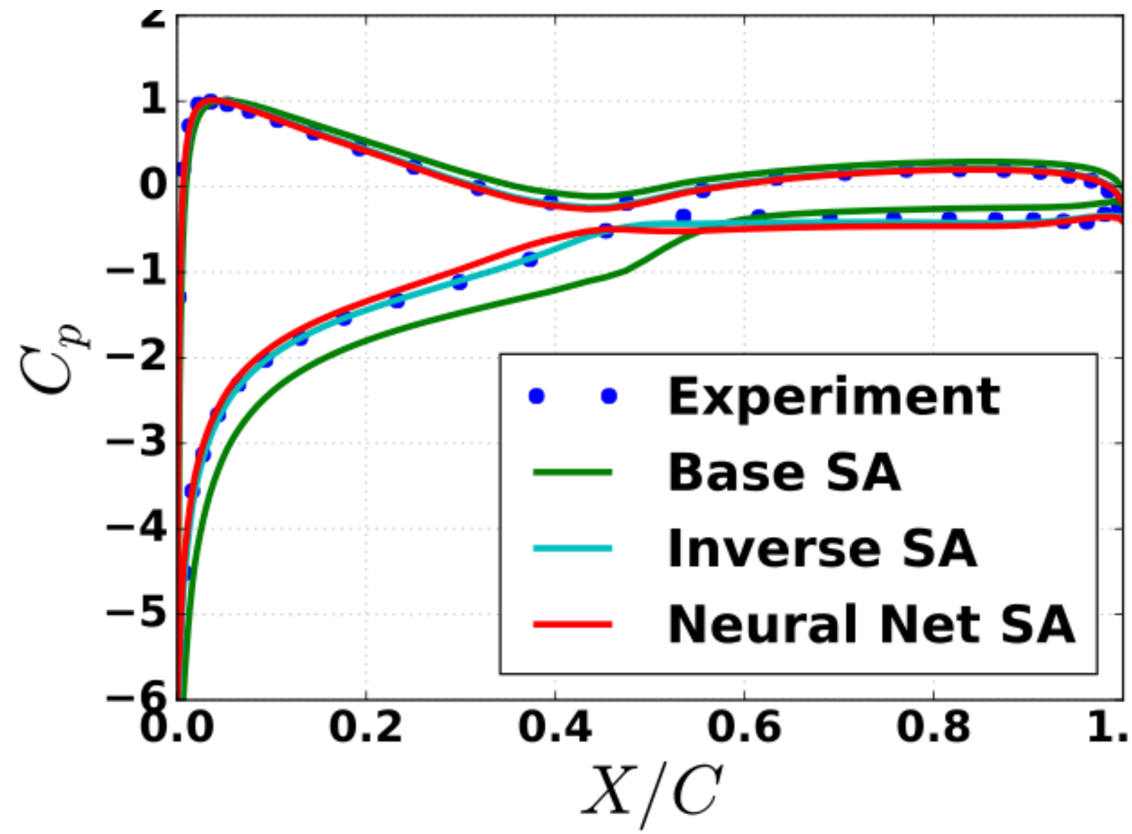


Tests

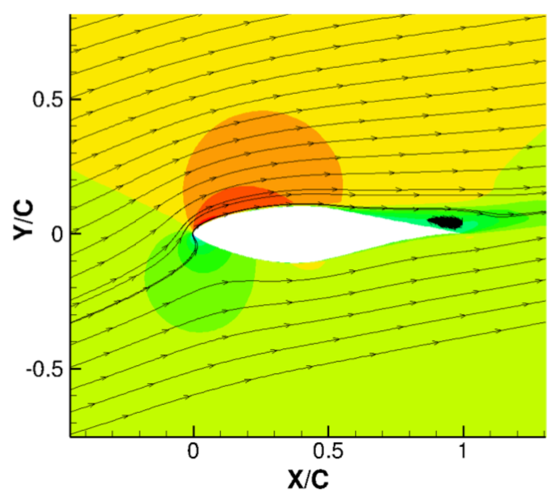
Singh, A., Medida, S. & Duraisamy, K., Data-augmented Predictive Modeling of Turbulent Separated Flows over Airfoils Submitted, AIAA Journal, 2016.

Model label	Training data
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 $Re = 1 \times 10^6, 2 \times 10^6$

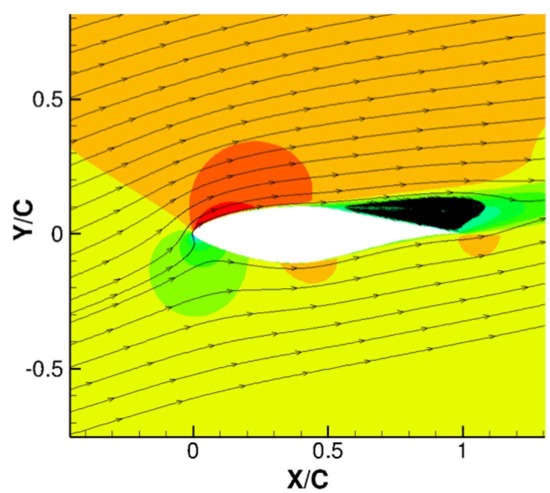
Training set →



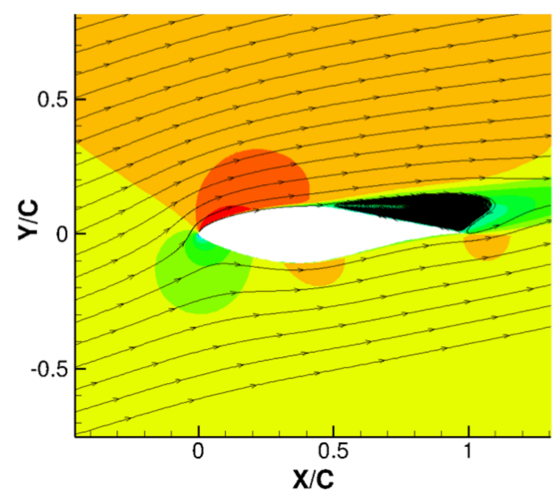
True prediction !



(a) Base SA

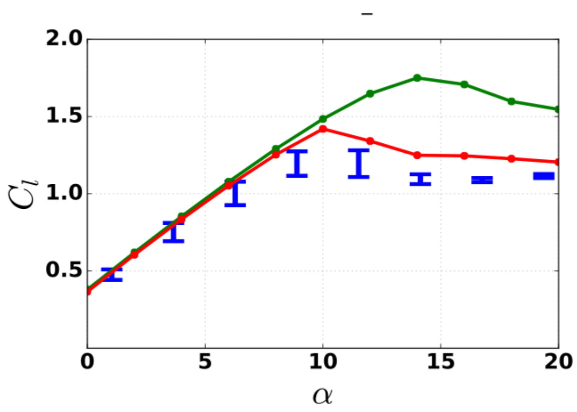


(b) Inverse SA

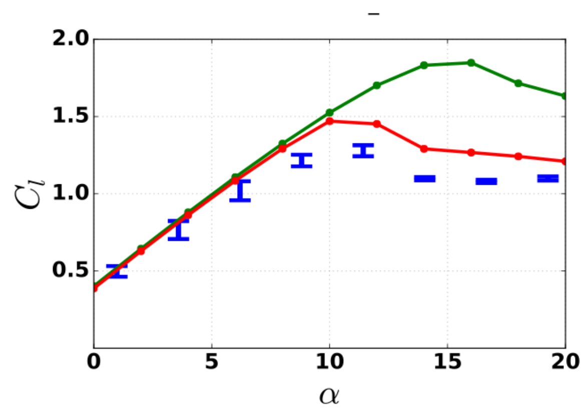


(c) NN-augmented SA (prediction)

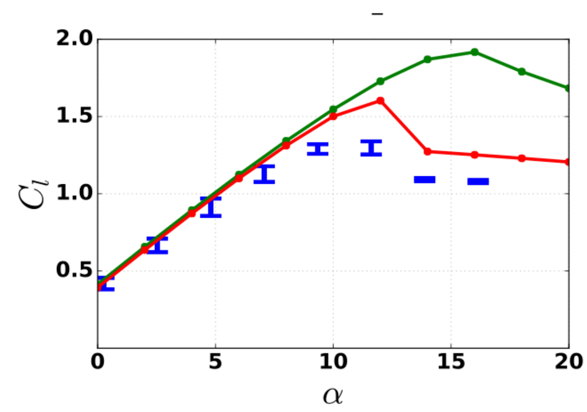
Prediction – S814



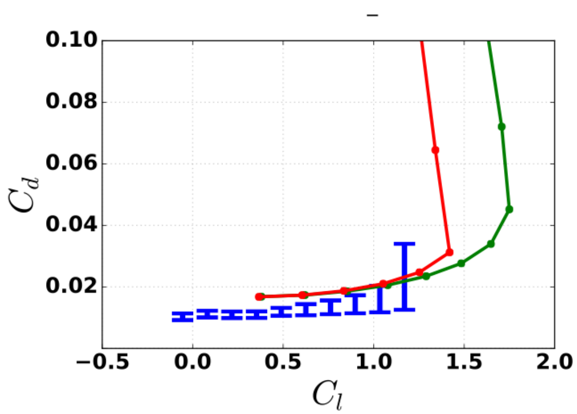
(a) $Re = 1 \times 10^6$



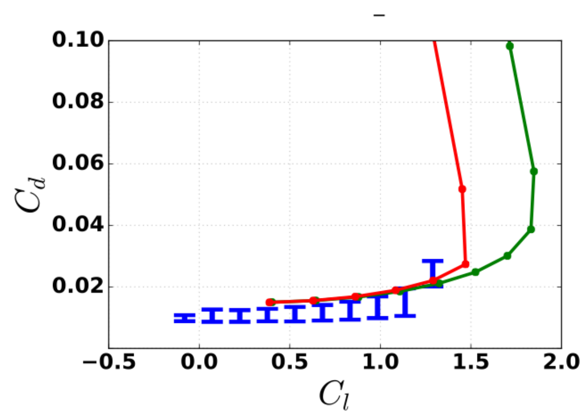
(b) $Re = 2 \times 10^6$



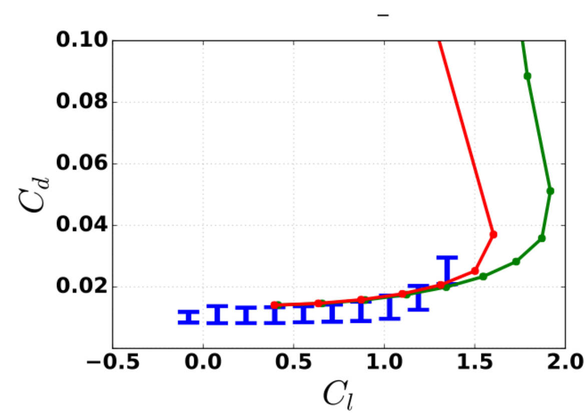
(c) $Re = 3 \times 10^6$



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



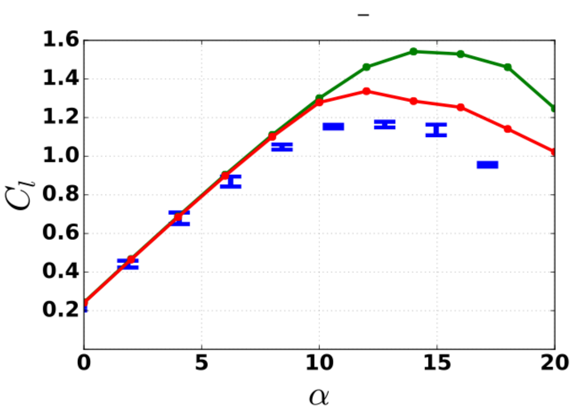
(e) $Re = 2 \times 10^6$



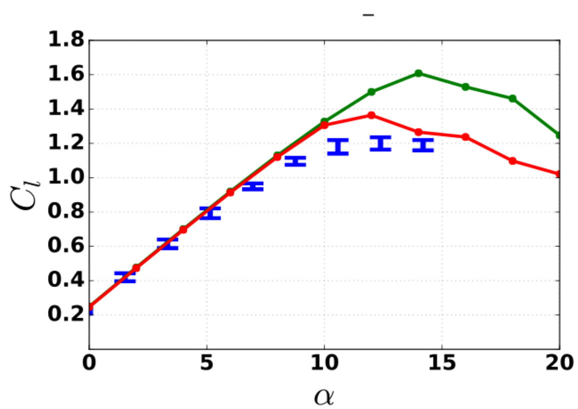
(f) $Re = 3 \times 10^6$

Collaboration with Altair, Inc.

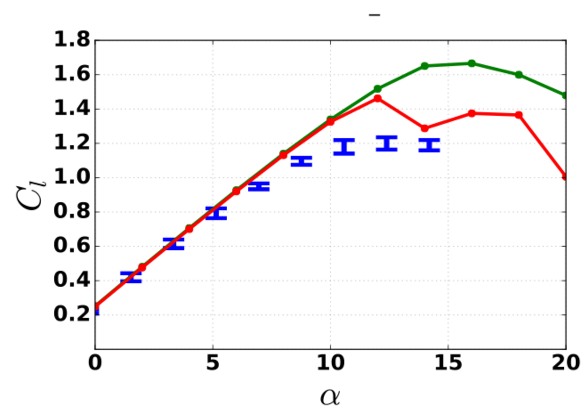
Prediction – S805



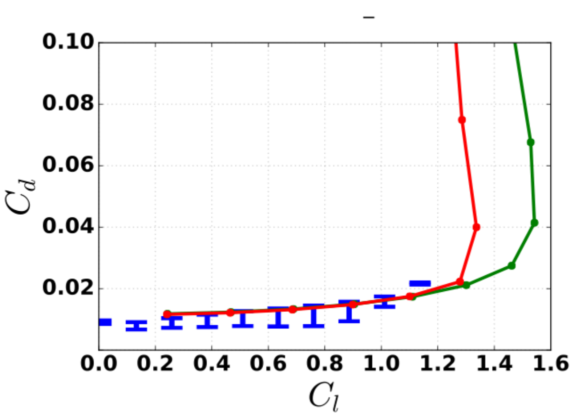
(a) $Re = 1 \times 10^6$



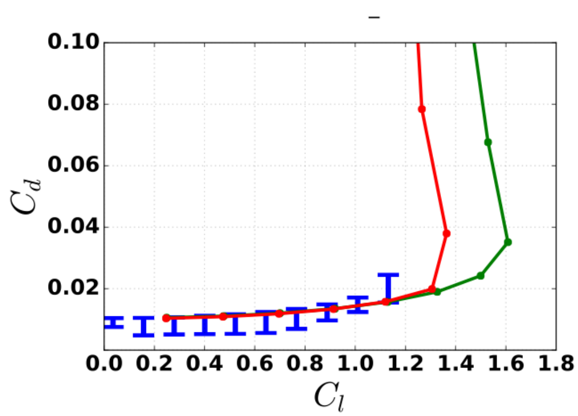
(b) $Re = 2 \times 10^6$



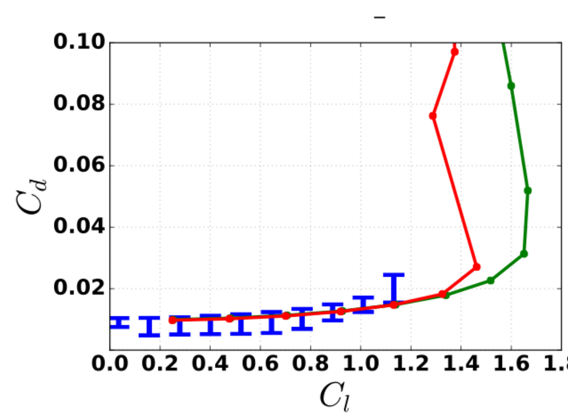
(c) $Re = 3 \times 10^6$



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



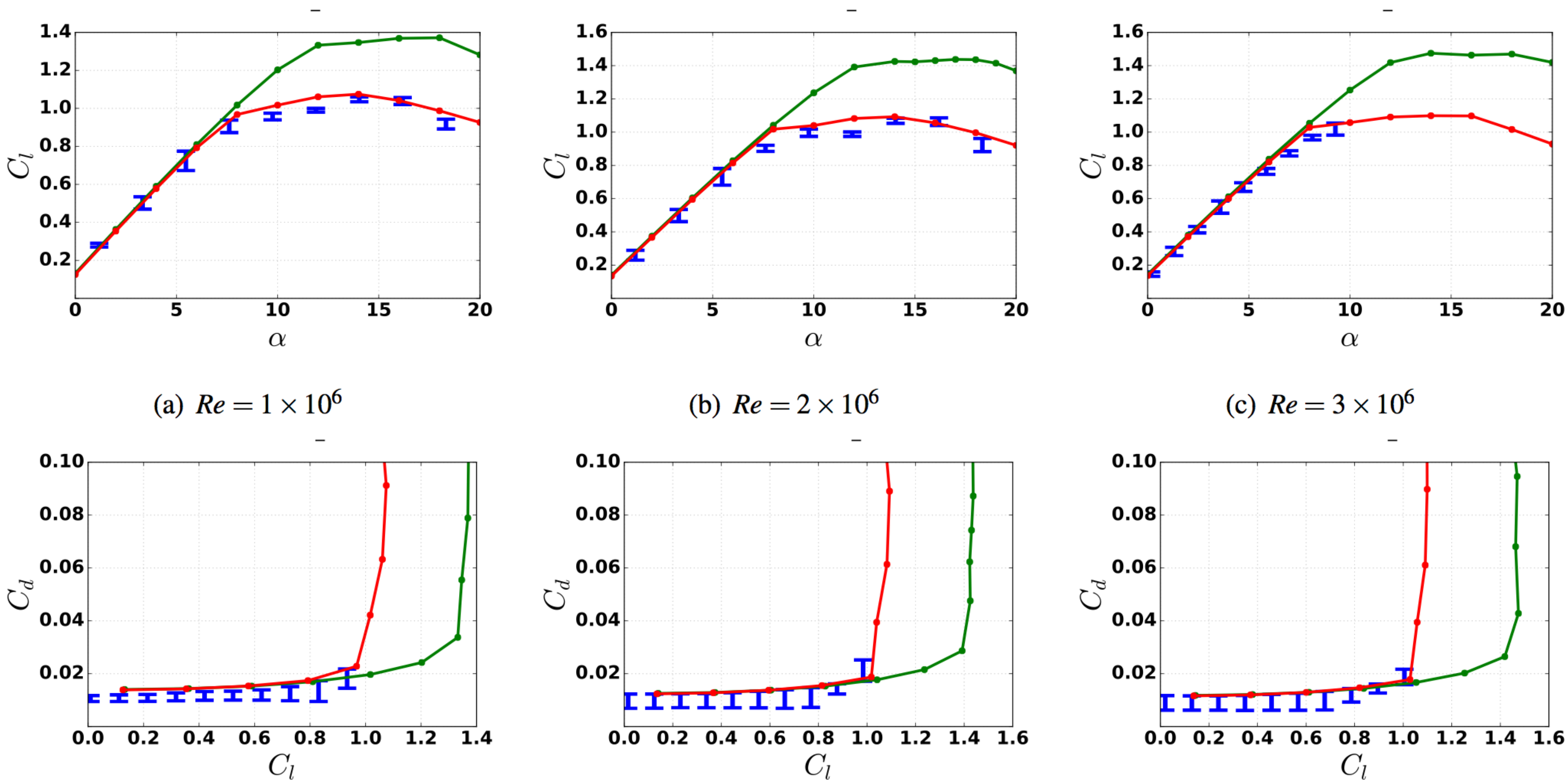
(e) $Re = 2 \times 10^6$



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

Collaboration with Altair, Inc.

Prediction – S809



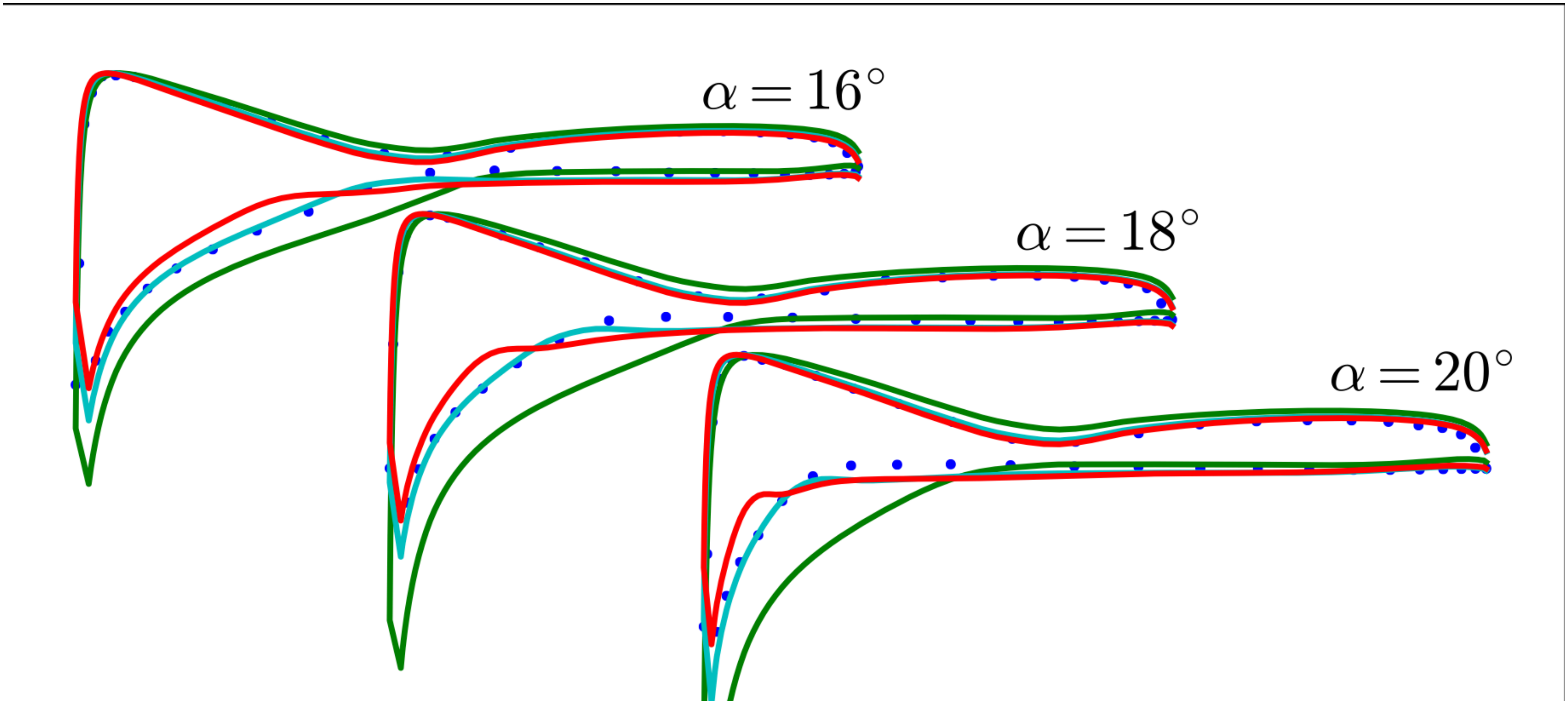
(a) $Re = 1 \times 10^6$ (b) $Re = 2 \times 10^6$ (c) $Re = 3 \times 10^6$

(d) $Re = 1 \times 10^6$ (e) $Re = 2 \times 10^6$ (f) $Re = 3 \times 10^6$

Collaboration with Altair, Inc.

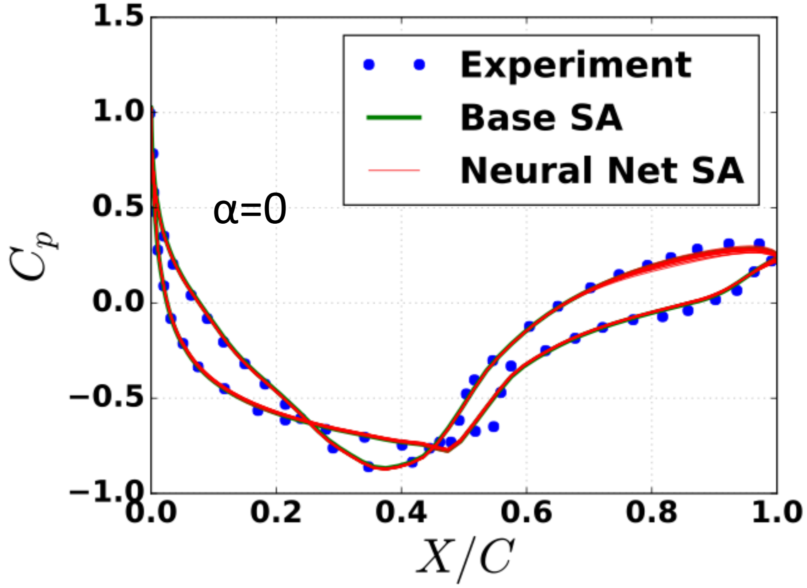
True prediction !

S 809, Re=2 Million

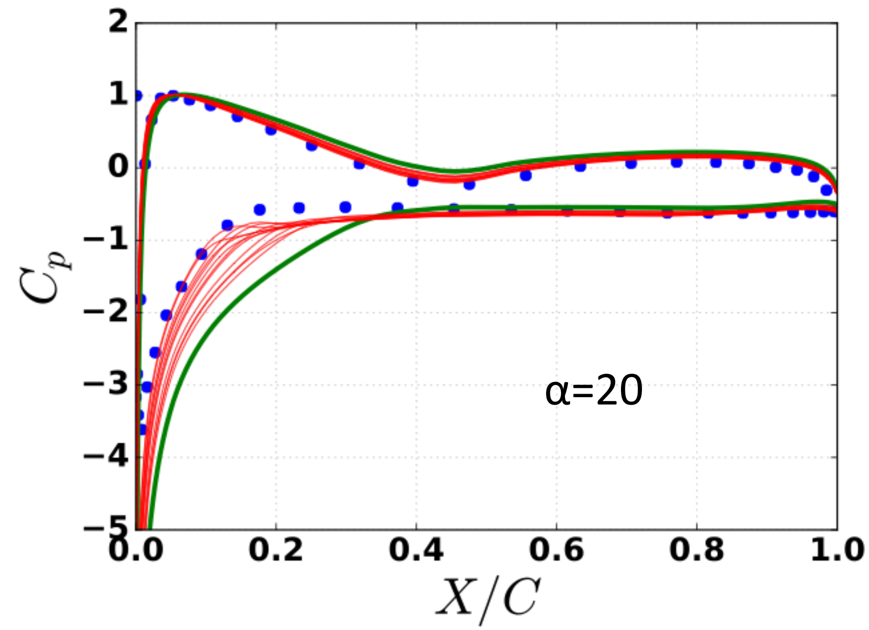


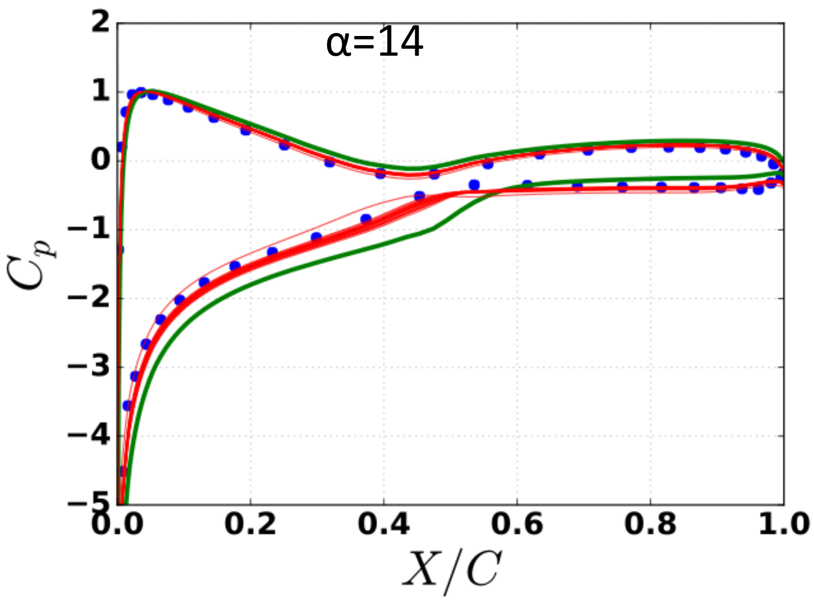
Inference used only CL data, NN-augmented model provides considerable predictive improvements of Cp

Variability



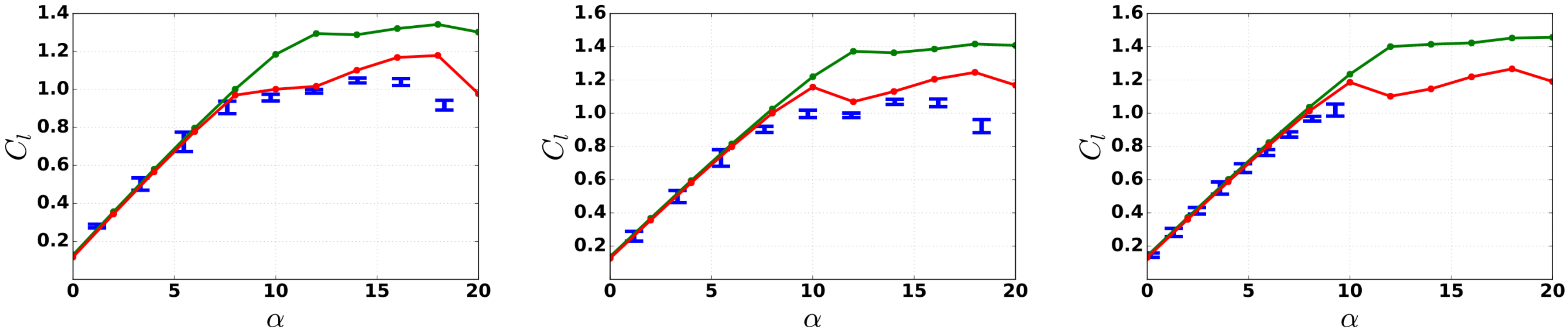
S 809, Re=2 Million



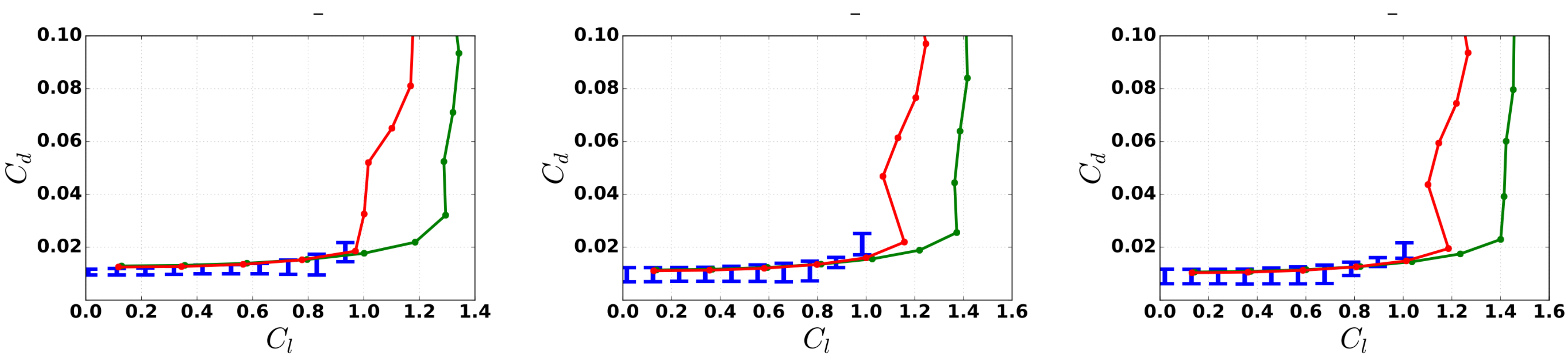


Training from different sets

Portability : Implementation in AcuSolve

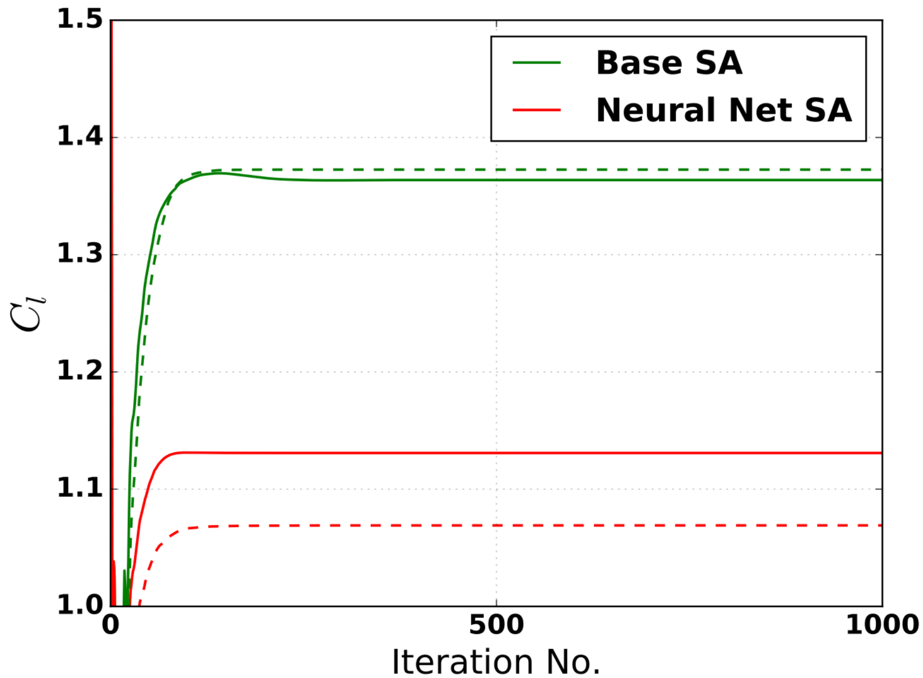


(a) $Re = 1 \times 10^6$ (b) $Re = 2 \times 10^6$ (c) $Re = 3 \times 10^6$

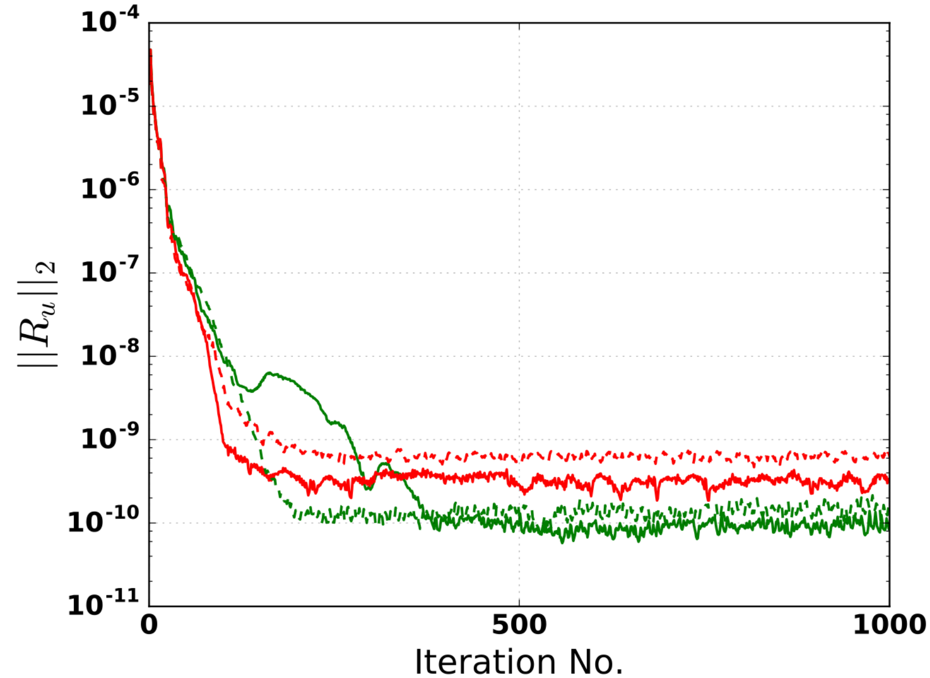


S809 Airfoil : Predictive results in Commercial CFD solver

Robustness:
Implementation in AcuSolve



(a) Lift coefficient



(b) L2 norm of solver residual

S809 Airfoil : Predictive results in Commercial CFD solver

Summary of predictions

- NN-augmented S-A model shows significant improvement over the baseline model in predicting CL , CD and stall onset angles.
- Predictions confirmed to be excellent in shapes and flow conditions that were not part of the training set.
- No deterioration of accuracy noticed in situations in which original model was accurate
- Inference used only CL but NN-augmented model provides considerable predictive improvements of Cp

• Ensemble of predictions used to assess the sensitivity of the model outputs to training

➔ Currently introducing uncertainties via GPs

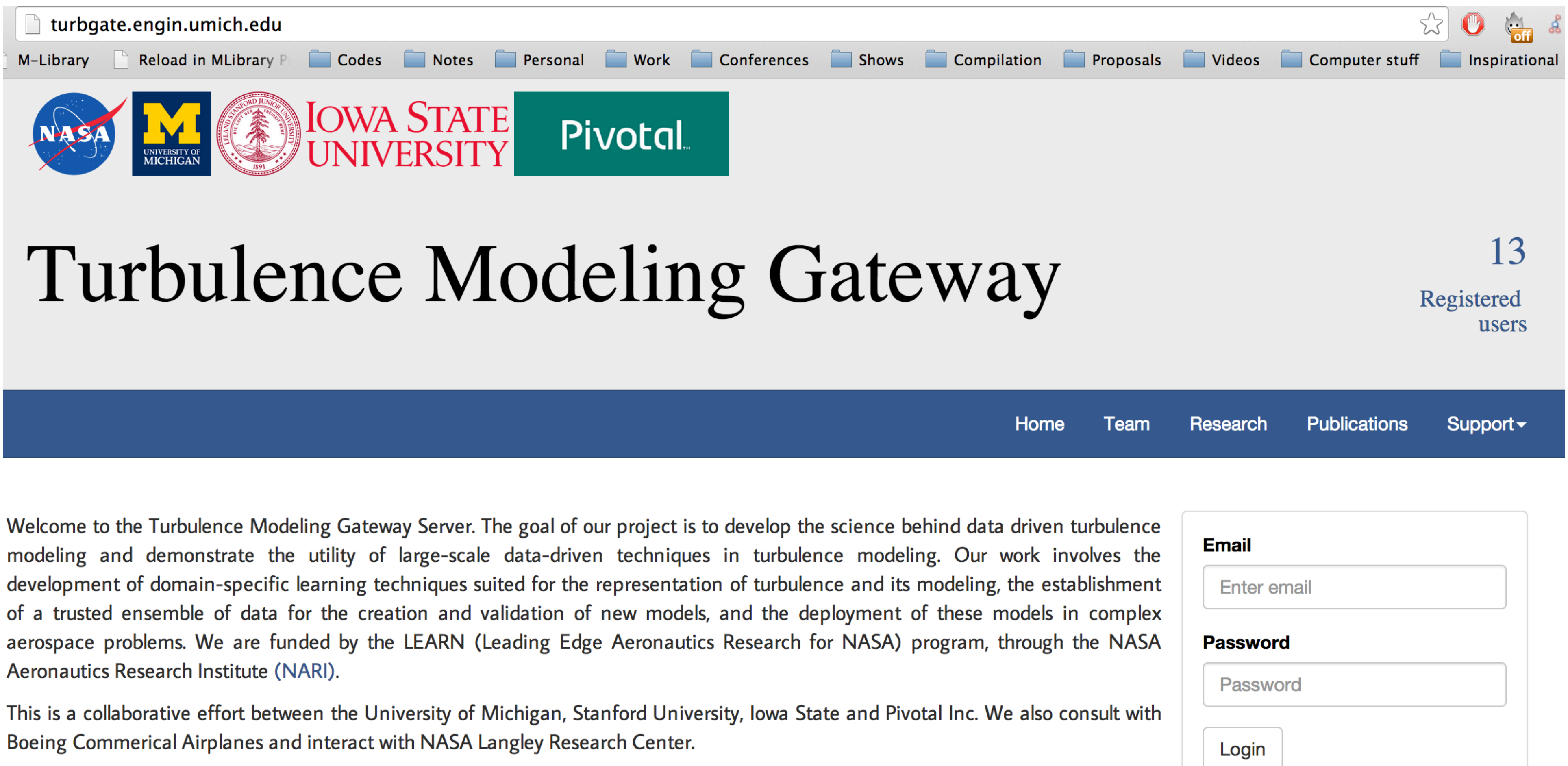
• Solver convergence was assessed and typically the cost overhead for the NN augmentations ~ 10-30%

• Portability addressed by developing framework in one solver and injecting in a commercial solver (Acusolve)

Vision for the future

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

Users upload/download/process data, generate maps.



Growing community for data-driven turbulence modeling

2013- onwards: Duraisamy et al
2015- onwards: Ling et al. (apriori ➔ modeling)
2015- onwards: Weatheritt et al (apriori)
2016- onwards: Xiao et al. ("open box UQ" ➔ data-driven)
2016- onwards: Dwight et al. (parametric ➔ non-parametric)
2016- onwards: Wang et al. (UQ ➔ closure improvement)

Others: Moser, Edeling, Cinella, etc.

Companies: Altair, Inc. ; CD-Adapco; GE; Boeing

Thanks to NASA for getting it started

Challenges

Assembly → Collection of relevant / necessary data

Data

Inverse modeling → Extreme-scale optimization

Information

Machine Learning → Noisy, Complex, Extreme-scale data, feature set selection

Modeling Knowledge

Embedding → UQ and Computational Efficiency

Predictive capability

Implications & Impact

Data: Very limited experimental data (more DNS/LES data when available)

Modeling insight : Modeler can understand what the model lacks to match data. This is done within the context of the model

Improved models : Can learn the missing components of model and generate improved models

Model credibility: Can validate/invalidate model structures

Uncertainty quantification: Can obtain modeling error bounds → Demonstrated in 1D probem

We are applying this framework within the context of turbulent flow, materials modeling and astrophysics.

References

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

Zhang, Z. & Duraisamy, K. & Gumerov, N. Sparse Multiscale Gaussian Process Regression using Hierarchical Clustering, Submitted, Applied Numerical Mathematics, 2016

Singh, A. & Duraisamy, K. Using Field Inversion to Quantify Functional Errors in Turbulence Closures, Physics of Fluids, 2016

Parish, E. & Duraisamy, K., A paradigm for data-driven predictive modeling using field inversion and machine learning, Journal of Computational Physics, 2016

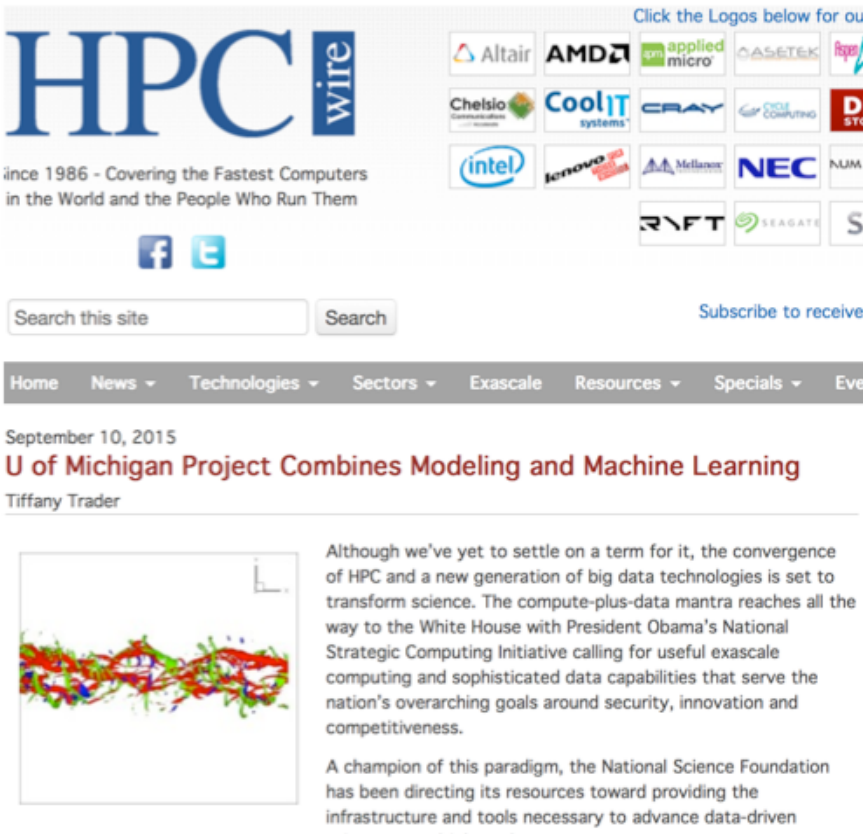
Zhang, Z. & Duraisamy, K., Machine Learning Methods for Data-Driven Turbulence Modeling, AIAA Aviation, 2015.

Tracey, B. & Duraisamy, K., & Alonso, J. A Machine Learning Strategy to Assist Turbulence Model Development, AIAA SciTech, 2015.

Duraisamy, K.; Zhang, Z. & Singh, A, New Approaches in Turbulence and Transition Modeling Using Data-driven Techniques, AIAA SciTech, 2015

Duraisamy, Karthik & Durbin , P.A., Transition modeling using data driven approaches, Proc. of the CTR Summer Program, 2014.

Looking for post-docs and grad students


Click the Logos below for ou

HPC wire

ince 1986 - Covering the Fastest Computers in the World and the People Who Run Them

Search this site | Search

Subscribe to receive

Home | News | Technologies | Sectors | Exascale | Resources | Specials | Eve

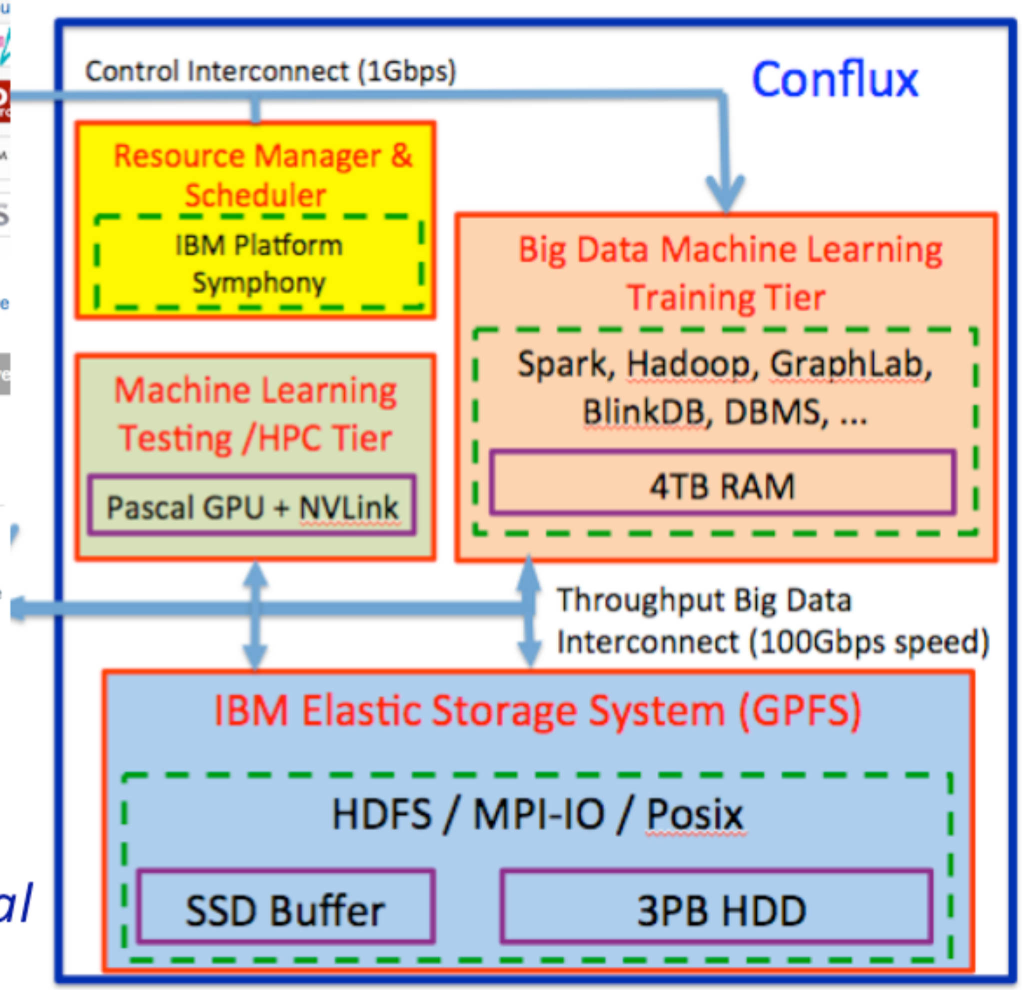
September 10, 2015

U of Michigan Project Combines Modeling and Machine Learning

Tiffany Trader

Although we've yet to settle on a term for it, the convergence of HPC and a new generation of big data technologies is set to transform science. The compute-plus-data mantra reaches all the way to the White House with President Obama's National Strategic Computing Initiative calling for useful exascale computing and sophisticated data capabilities that serve the nation's overarching goals around security, innovation and competitiveness.

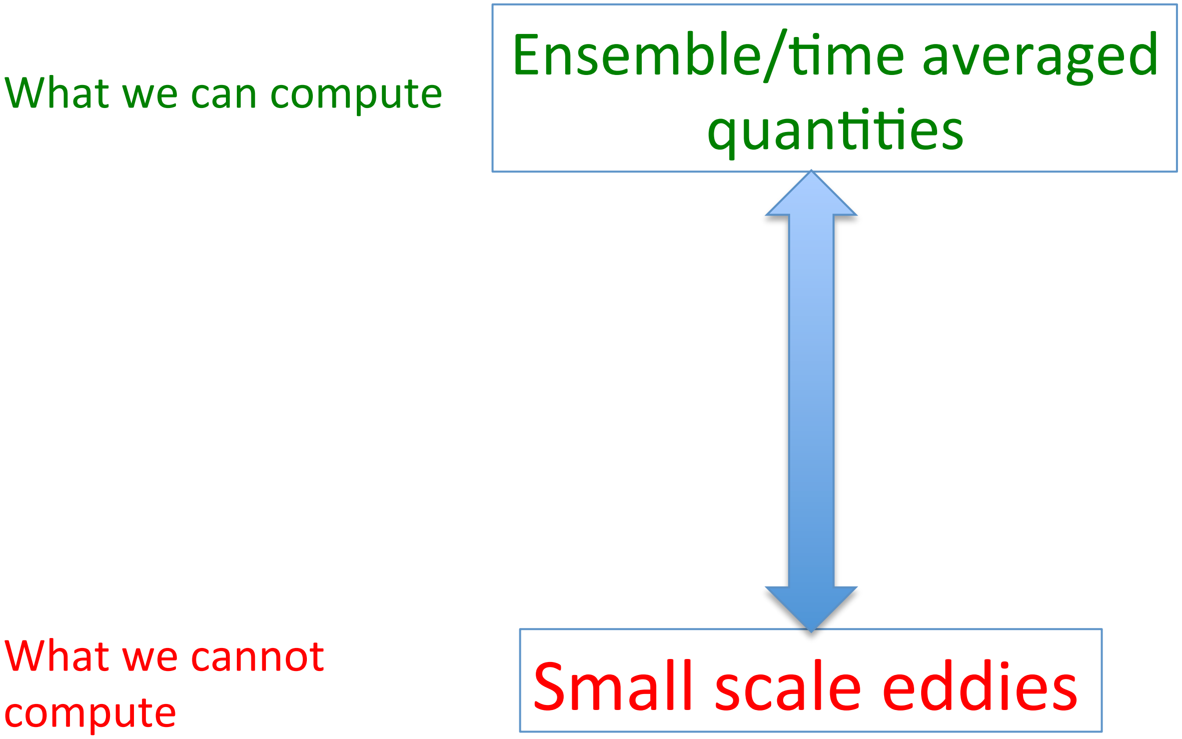
A champion of this paradigm, the National Science Foundation has been directing its resources toward providing the infrastructure and tools necessary to advance data-driven science at multiple scales.

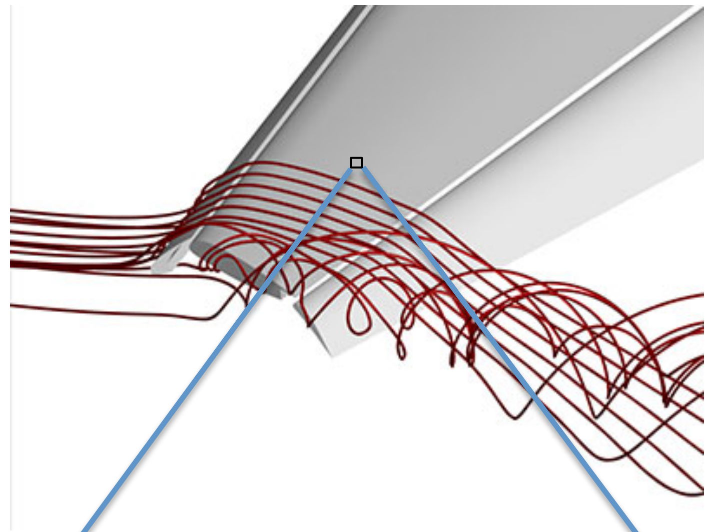


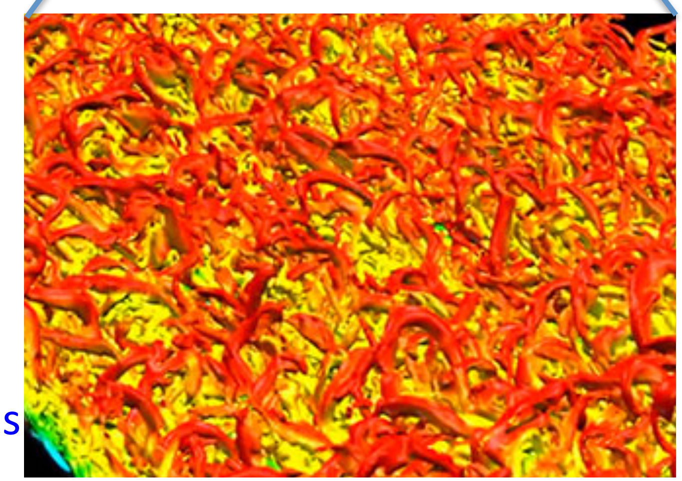
Center for Data-driven Computational Physics, University of Michigan

Backups

Engineering simulation of turbulent flows







Closure problem

➔ Averaged model can only involve averaged quantities

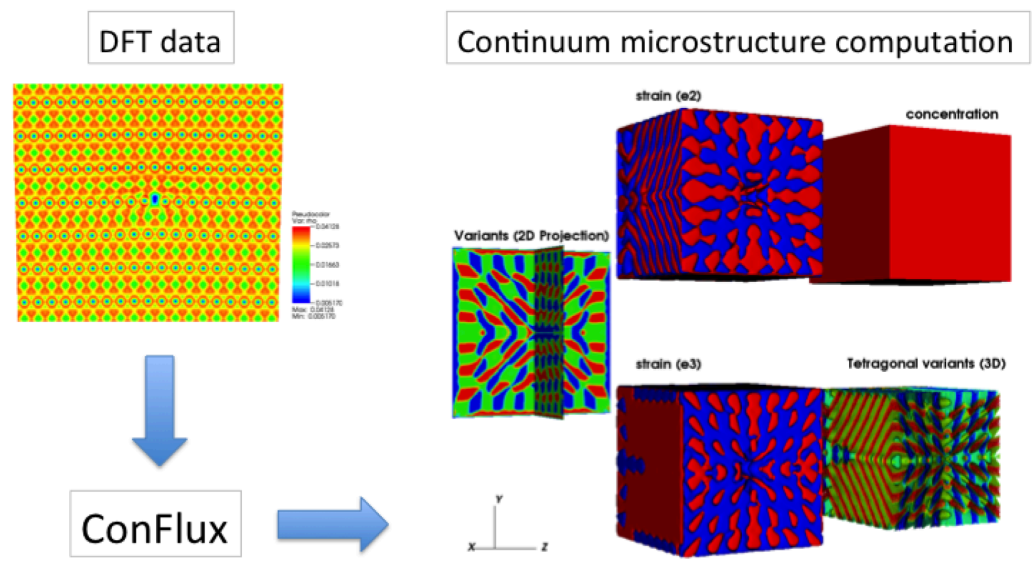
Materials Modeling

The goal is to identify, explain, predict and ultimately to design the properties and responses of these materials.

Hierarchical models have been developed at several scales

➔ These methods have thus far provided insight and qualitative connections to parameters and phenomena from lower scales, but have not been predictive

Quantum Monte Carlo ⇔ Density Functional Theory ⇔ Continuum physics



With Profs. Vikram Gavini and Krishna Garikipati (Mech Engineering and Materials Science)

Subject-specific blood flow modeling

Biggest challenges
➔ lack of physiologic data to inform the boundary conditions
➔ lack of data on mechanical properties of the vascular model

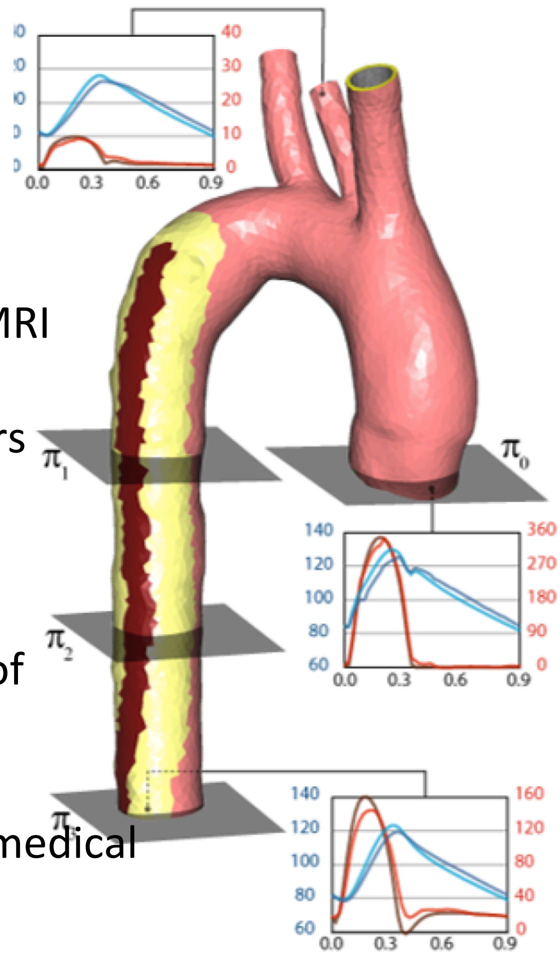
Obtain data from tomography and MRI

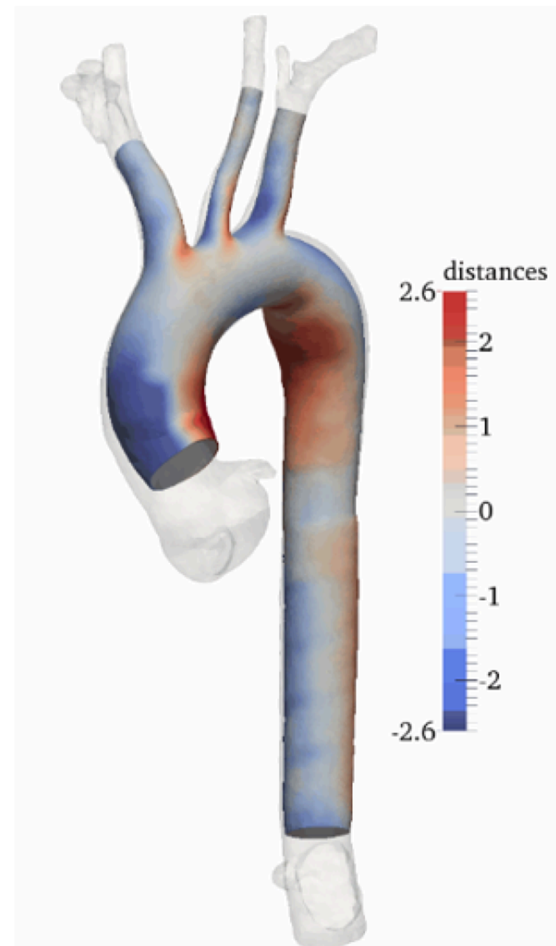
Solve inverse problem for parameters

Massive data size

On-the-fly Lagrangian computation of Motion

Evaluation of arterial stiffness from medical Images !





With Prof. Alberto Figueroa (Biomedical Engineering & Surgery)

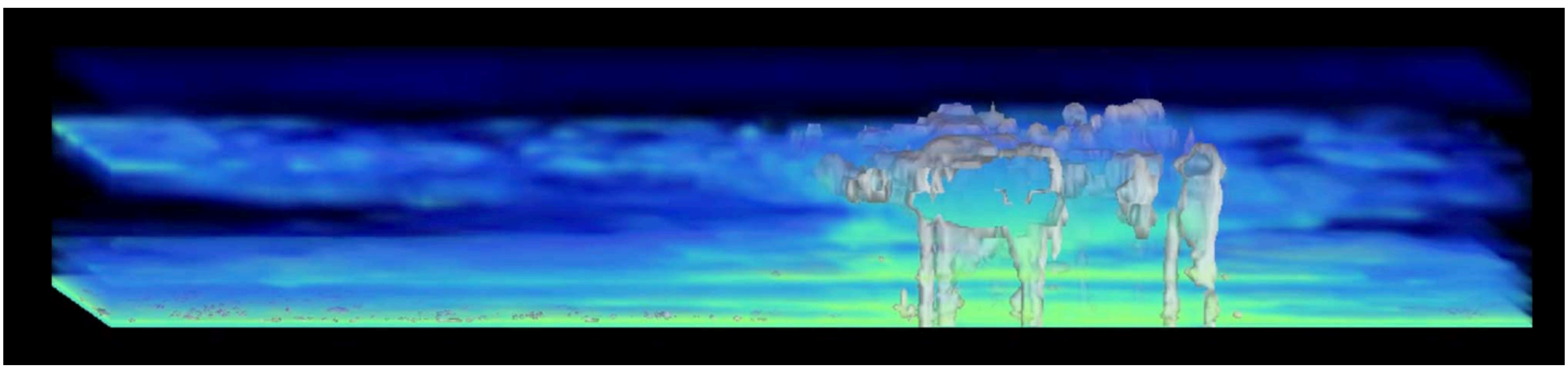
Climate system interactions

The Earth's climate system is composed of multiple interacting components that span spatial scales of 13 orders of magnitude and temporal scales that range from microseconds to centuries.

➔ key responses and feedbacks in the system are not well characterized

Understanding how clouds interact with the larger scale circulation, thermodynamic state, and radiative balance is one of the most challenging problems

We use statistical inversion and machine learning to explore the interaction between changes in the Earths climate system and the radiative fluxes, circulation, and precipitation generated by large scale organized cloud systems.



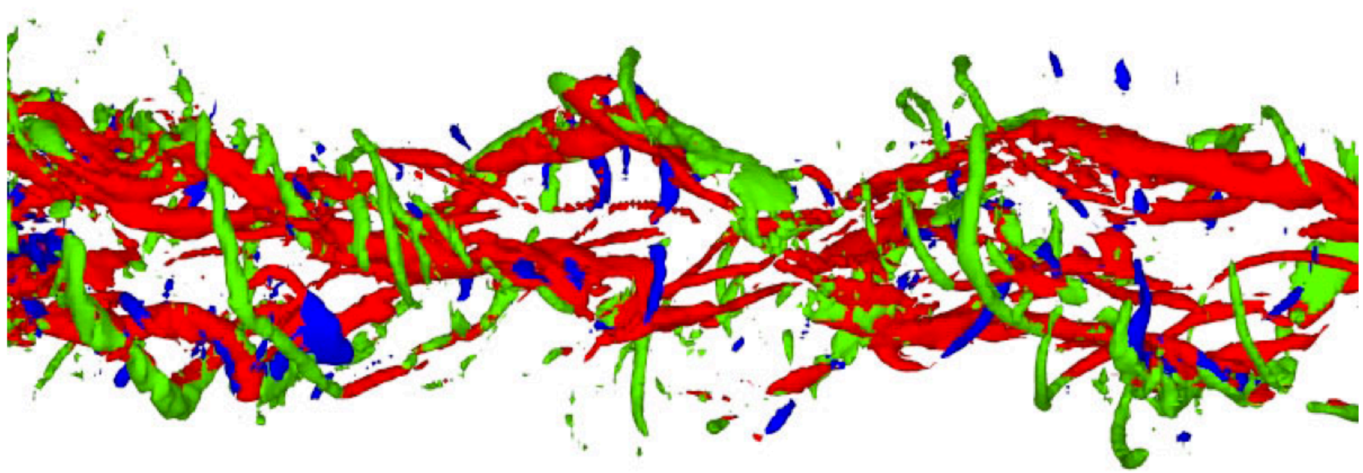
With Prof. Derek Posselt (Atmospheric Oceanic & Space Sciences)

How can we scale up these problems?

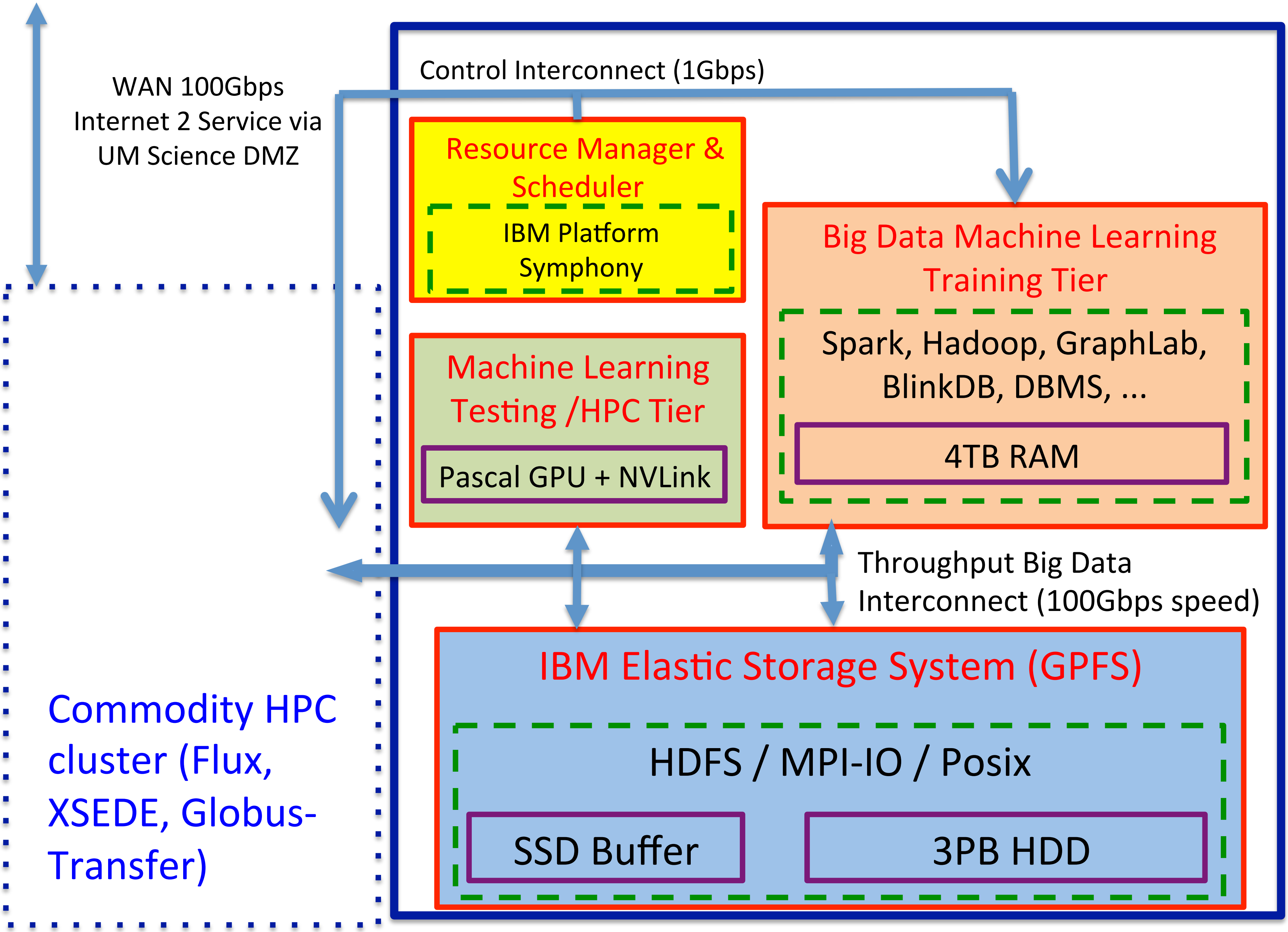
$3.46M to combine supercomputer simulations with big data

9/3/2015

From: **Kate McAlpine**
Michigan Engineering



A new way of computing could lead to immediate advances in aerodynamics, climate science, cosmology, materials science and cardiovascular research. The National Science Foundation is providing $2.42 million to develop a unique facility for refining complex, physics-based computer models with big data techniques at the University of Michigan, with the university providing an additional $1.04 million.


WAN 100Gbps
Internet 2 Service via
UM Science DMZ
Control Interconnect (1Gbps)
Resource Manager &
Scheduler
IBM Platform
Symphony
Big Data Machine Learning
Training Tier
Spark, Hadoop, GraphLab,
BlinkDB, DBMS, ...
4TB RAM
Machine Learning
Testing /HPC Tier
Pascal GPU + NVLink
Throughput Big Data
Interconnect (100Gbps speed)
IBM Elastic Storage System (GPFS)
HDFS / MPI-IO / Posix
SSD Buffer
3PB HDD
Commodity HPC
cluster (Flux,
XSEDE, Globus-
Transfer)