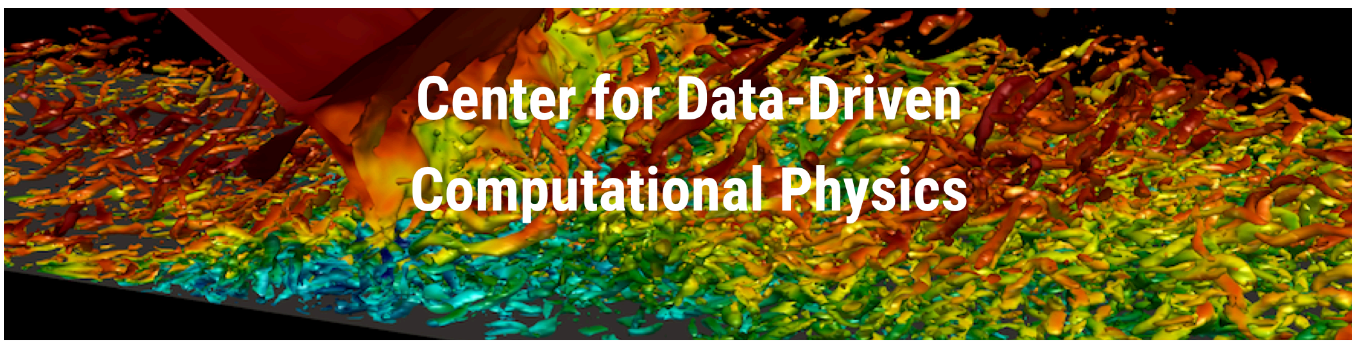
Data-enabled, Physics-constrained Predictive Modeling of Complex Systems

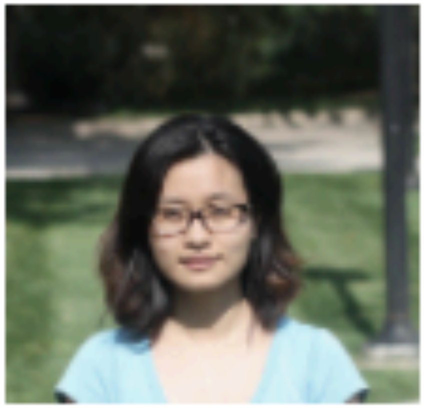
Karthik Duraisamy



AEROSPACE ENGINEERING
UNIVERSITY of MICHIGAN

Future CFD Technologies
Jan 7, 2018

Thanks to…



Helen Zhang

Anand Pratap Singh

Eric Parish

Shaowu Pan

5 "Classes" of Data-driven physics modeling problems

- ***Class A :*** Given a system X, some data d, how best can we explain the data efficiently ➔ Data decomposition, reconstruction

- ***Class B :*** Given a system X, some data d, for some time [0.. T], how well can we make a future state prediction

- ***Class C:*** Given a system X, some data d, and a model $\mathcal{M}$, how well can we predict quantities of interest Q (in the future) ➔ Inference + propagation / Data assimilation

- ***Class D:*** Given a system X(p), some data d(p), and a model $\mathcal{M}$, under some "operating conditions" p, how well can we predict Q(r) where there might or might not be data?

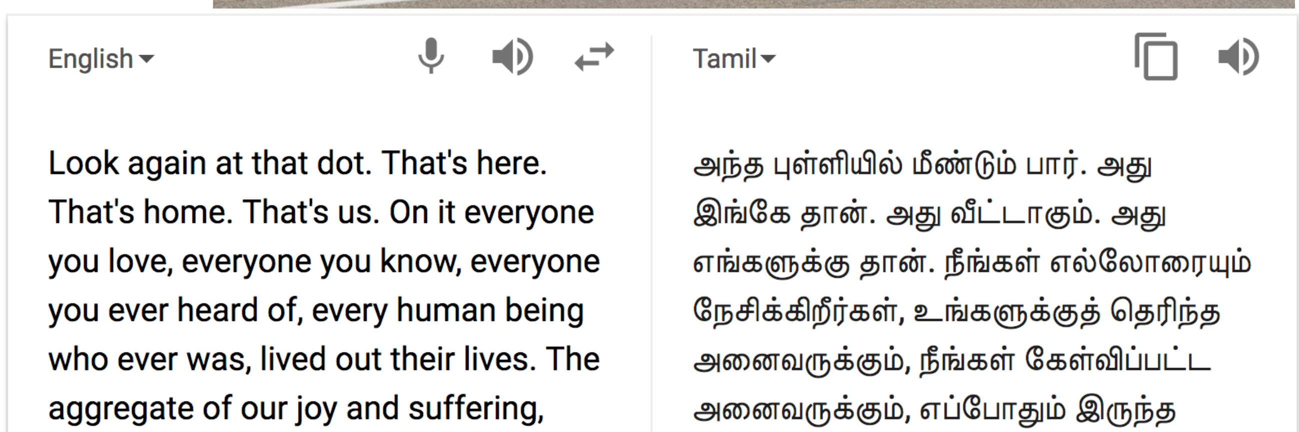
- ***Class E:*** Given systems $X_1(p)$, $X_2(p)$,.. $X_n(p)$, some data $d_1(p)$, $d_2(p)$,.. $d_n(p)$, and a model $\mathcal{M}$, how well can we predict Q(r) in a different system Y(r)?

Spectacular success of Artificial Intelligence

Data

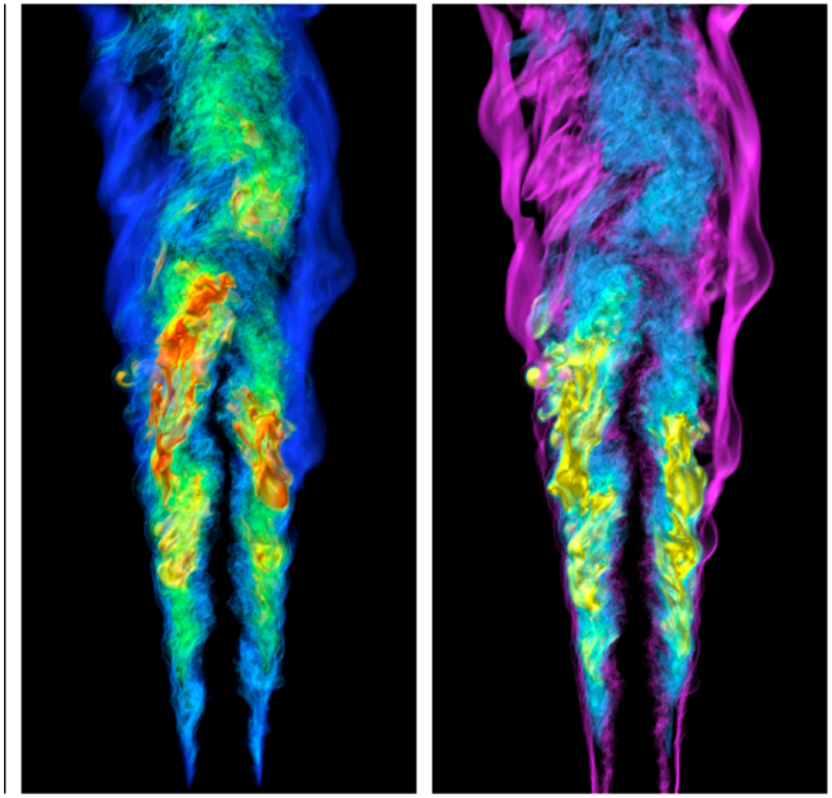
Machine Learning

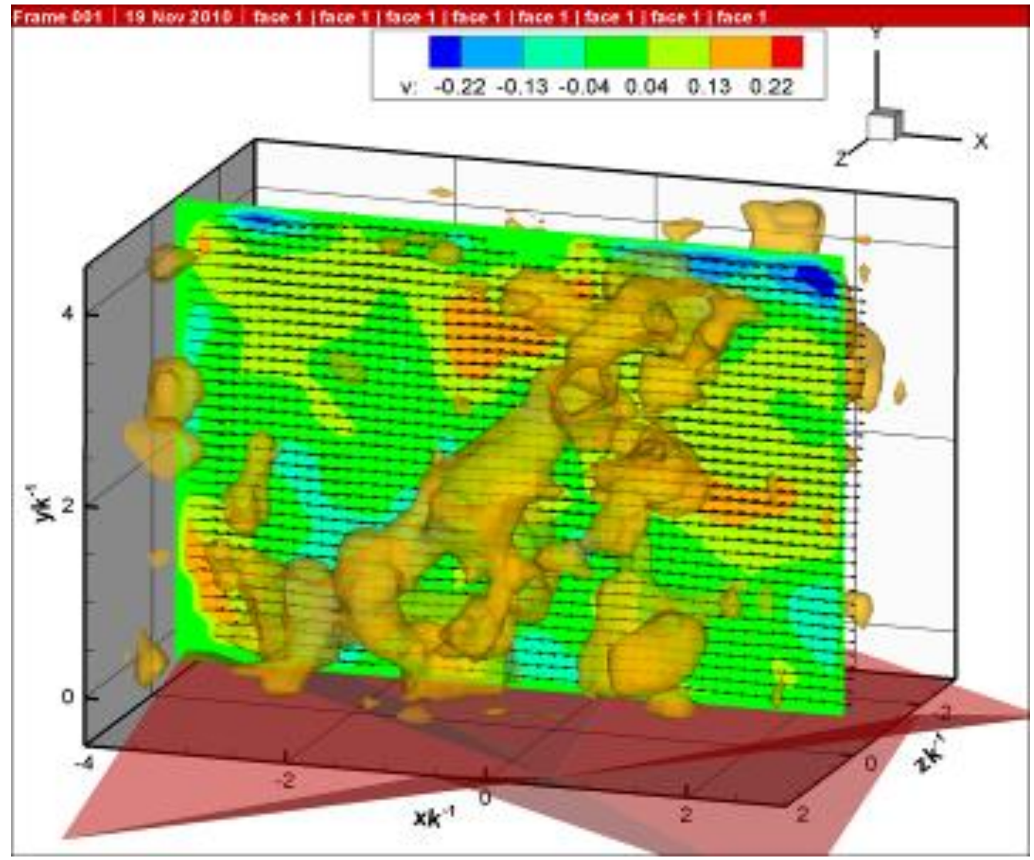
Predictive capability



Combustion simulation
(Jackie Chen, Sandia)

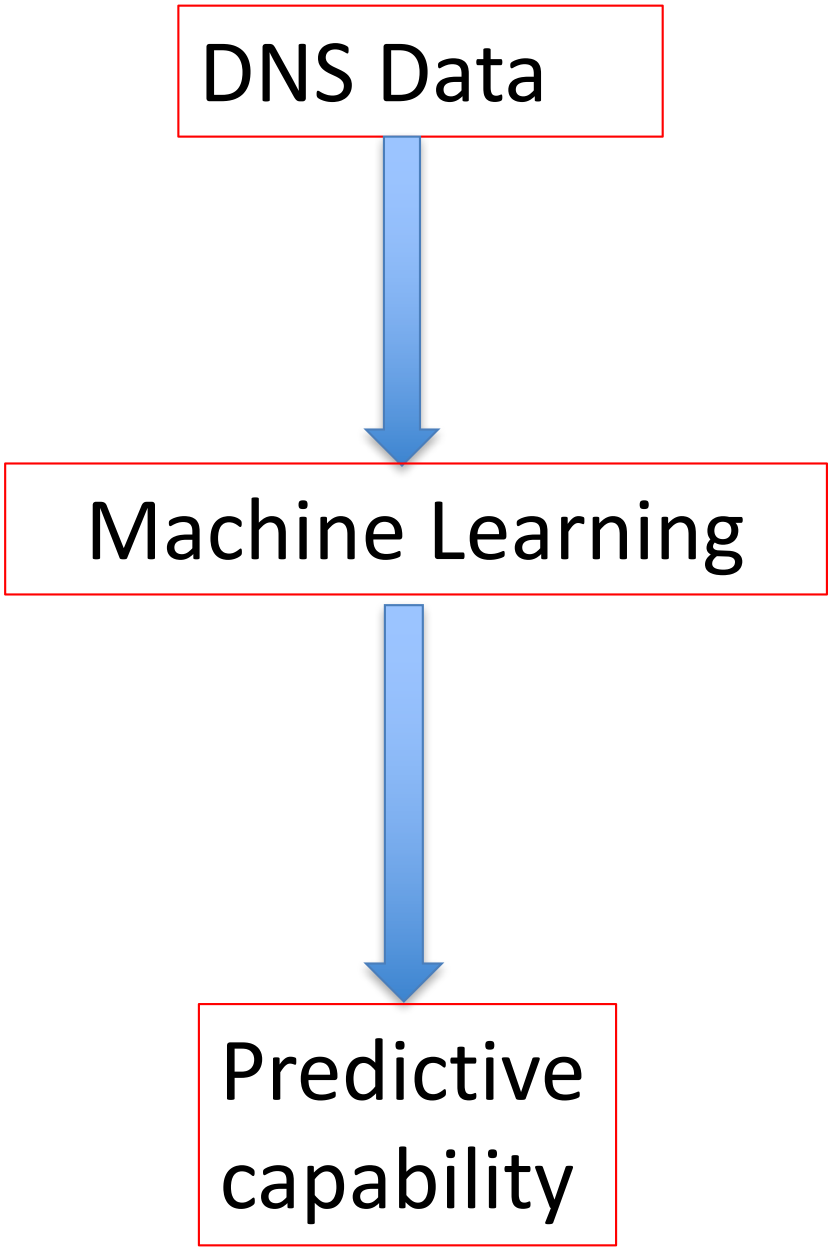
Holographic PIV
(J. Katz, JHU)





1.3×10^9 grid points, 22 species,
140GB / time step,
1 x 10^6 time steps

Machine learning for computational physics ?



Machine learning for computational physics ?

DNS Data

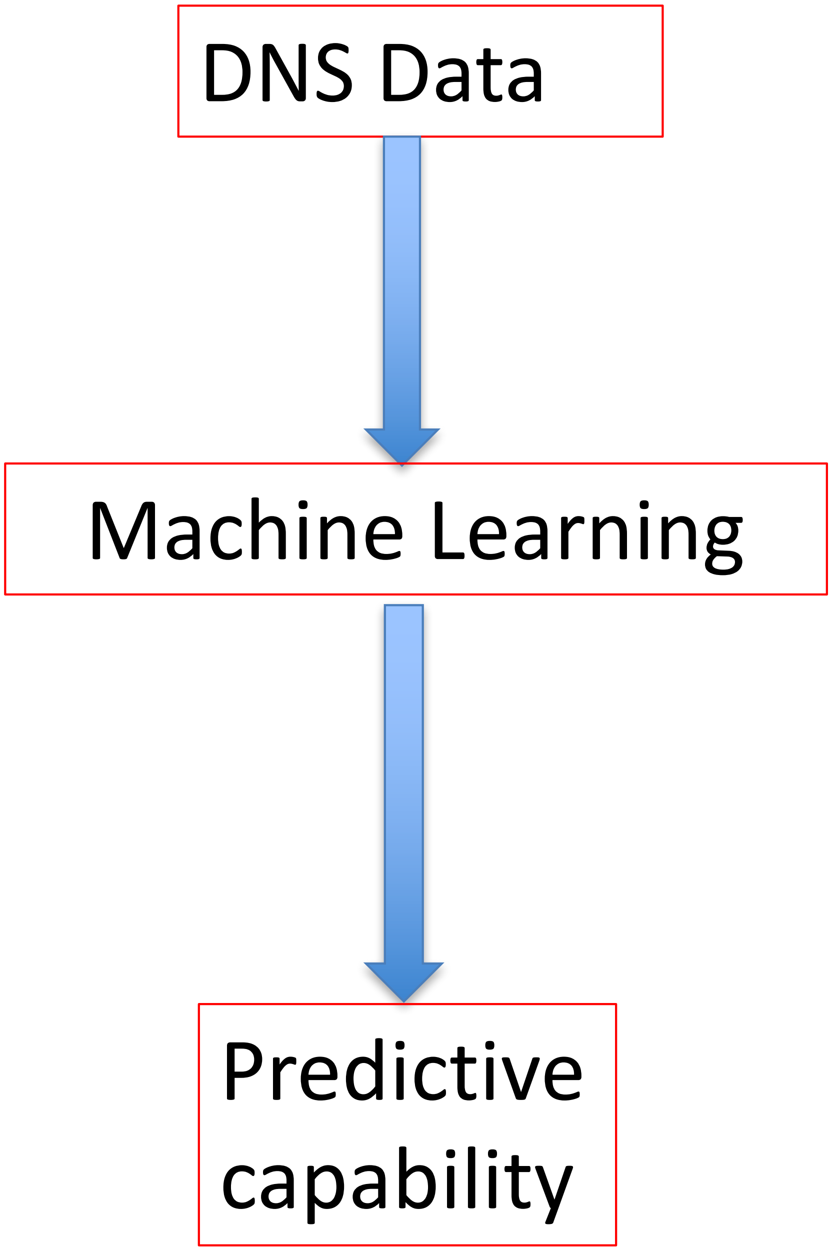
↓

Machine Learning

↓

Predictive capability

Machine learning for computational physics ?



- Data contains “real” quantities; Model contains “modeled” quantities
  ➔ Loss of consistency is especially a problem in turbulence models)

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

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

+ Machine learning reasons

Data

+

Inference

+

Physical modeling

+

Machine Learning

+

Theoretical insight

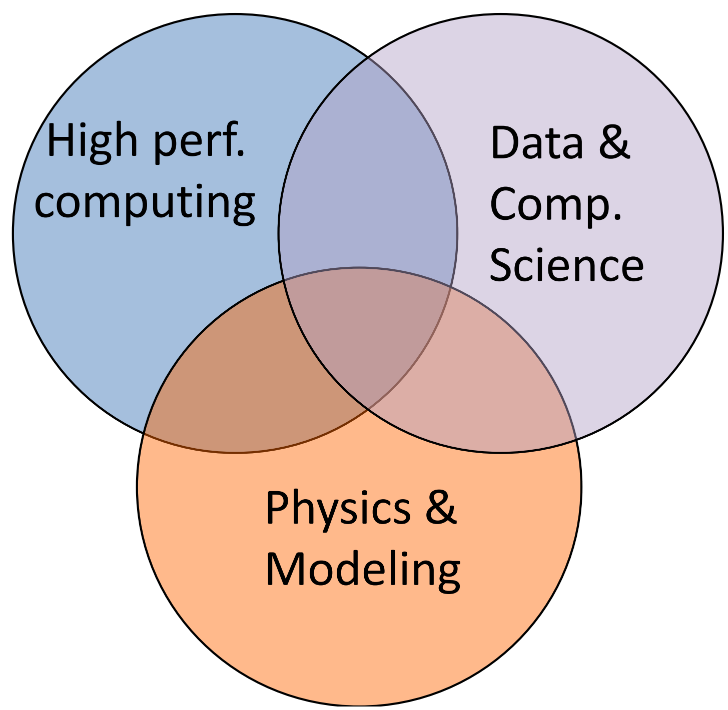
+

Problem-specific thought process

+

Computer science

= Useful solution



“Data-driven discovery”: Blind Examples

Example: Data-driven decomposition/discovery

True process

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2} \quad ; \quad u(x,0) = g(x)$$

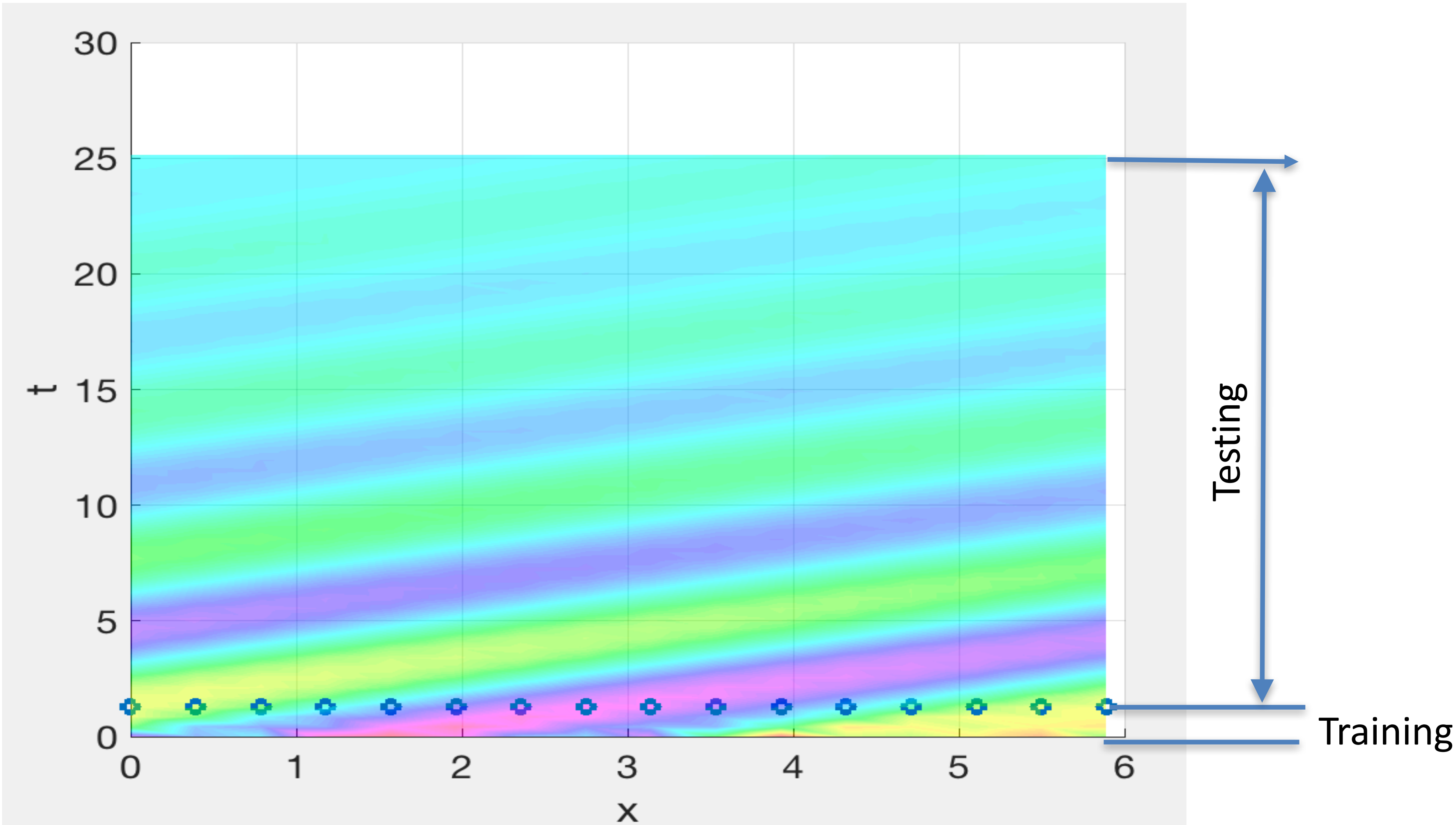
$$u(x_j,t) = \sum_{k=-\frac{m}{2}}^{\frac{m}{2}-1} \alpha_k(t)\mathrm{e}^{\mathrm{i}kx_j} \rightarrow \quad \frac{d\alpha_k}{dt} + \mathrm{i}ak\alpha_k = -\nu k^2\alpha_k$$

$$\mathbf{X} = [\boldsymbol{\alpha}(0) \;\; \boldsymbol{\alpha}(\Delta t) \;\; \boldsymbol{\alpha}(2\Delta t) \ldots.], \qquad \mathbf{X}' = [\boldsymbol{\alpha}(\Delta t) \;\; \boldsymbol{\alpha}(2\Delta t) \;\; \boldsymbol{\alpha}(3\Delta t) \ldots.],$$

$$\mathbf{X}' = \hat{\mathbf{A}}\mathbf{X}$$

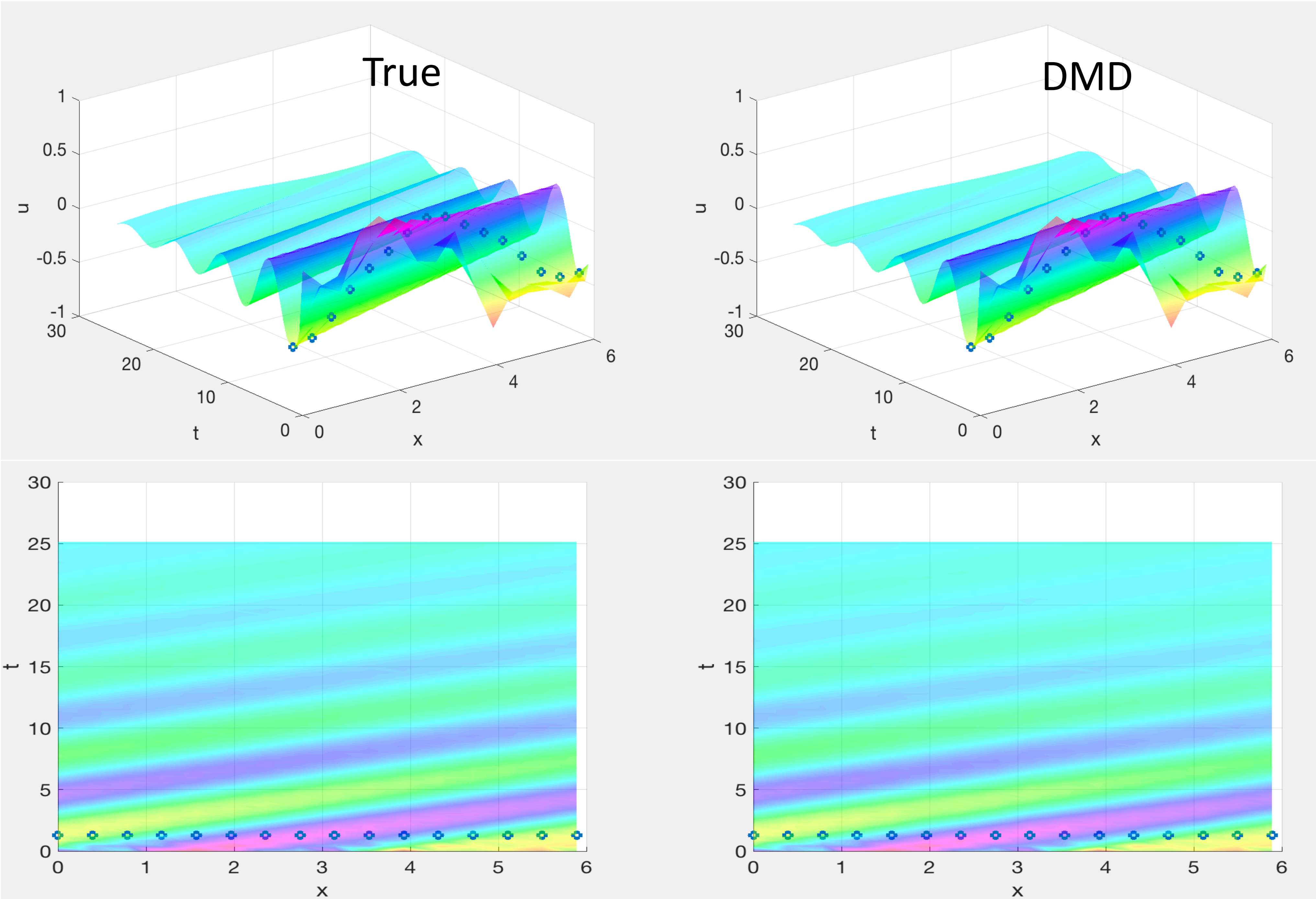
Data-driven model (DMD)

Example: Data-driven discovery

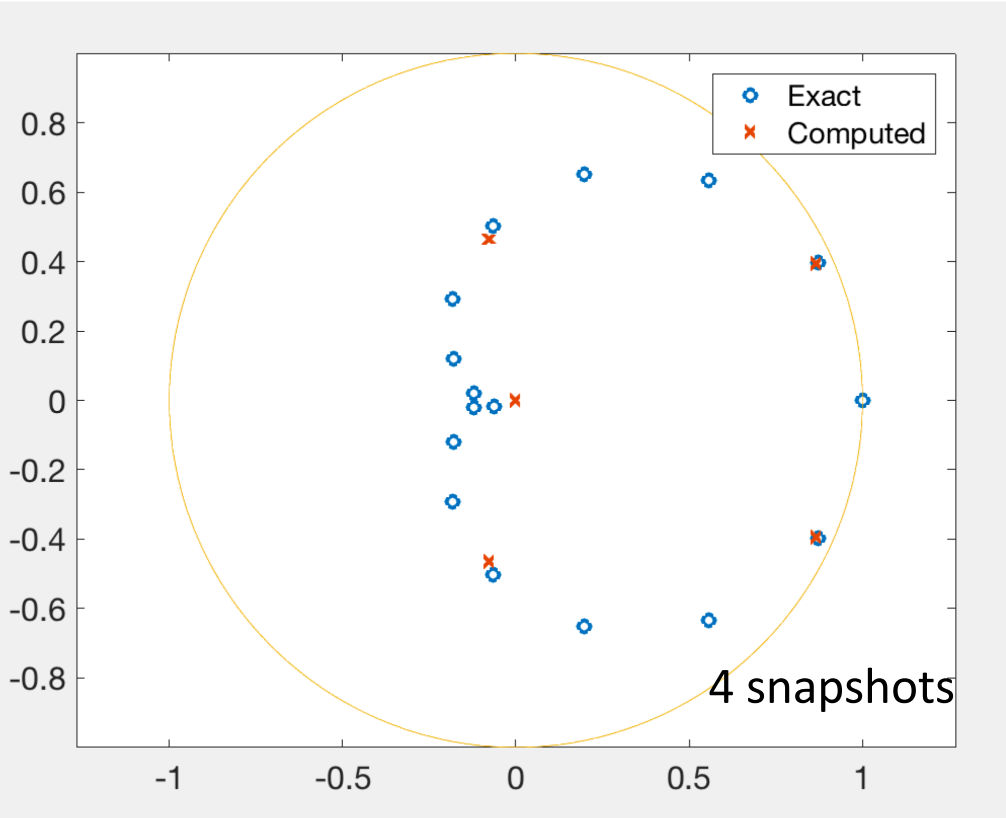


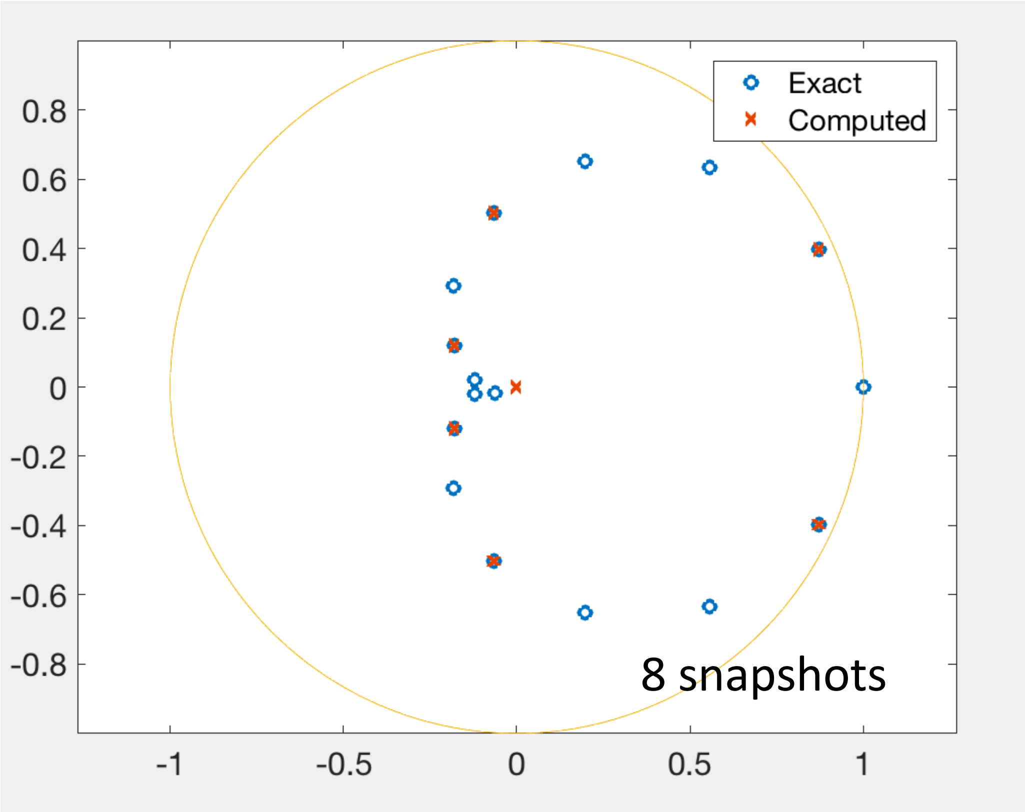
Initial condition $u(x,0) = \frac{\sin x}{2} + \frac{\cos 4x}{4} + \frac{\cos[6x - \frac{1}{2}]}{10}$

Example: Data-driven discovery



Example: Data-driven discovery





$$u(x,0) = \frac{\sin x}{2} + \frac{\cos 4x}{4} + \frac{\cos[6x - \frac{1}{2}]}{10}$$

DMD only discovers parts of the model that can be “seen”

Discovering equations from data

Unknown dynamical system

$$\mathbf{x}^{n+1} = \mathbf{f}(\mathbf{x}^n)$$

Define features

$$\mathbf{g}(\mathbf{x})$$

Example:

$$\mathbf{g}(\mathbf{x}) = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_1 x_2 \\ x_1^2 \\ x_2^2 \\ .. \\ .. \\ .. \end{bmatrix}$$

Discovering equations from data

Unknown dynamical system

$$\mathbf{x}^{n+1} = \mathbf{f}(\mathbf{x}^n)$$

Define features

$$\mathbf{g}(\mathbf{x})$$

Approximate evolution in feature space

$$\mathbf{g}(\mathbf{x}^{n+1}) = \mathbf{K}\mathbf{g}(\mathbf{x}^n)$$

Define state to features map

$$\mathbf{x} = \mathbf{C}\mathbf{g}(\mathbf{x})$$

Discovering equations from data

Unknown dynamical system

$$\mathbf{x}^{n+1} = \mathbf{f}(\mathbf{x}^n)$$

Define features

$$\mathbf{g}(\mathbf{x})$$

Approximate evolution in feature space

$$\mathbf{g}(\mathbf{x}^{n+1}) = \mathbf{K}\mathbf{g}(\mathbf{x}^n)$$

Define state to features map

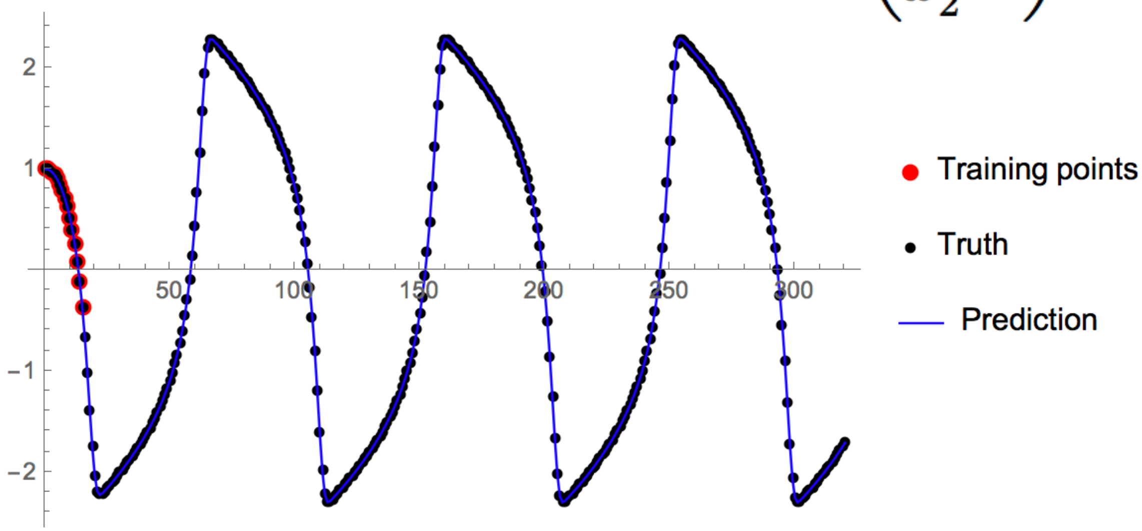
$$\mathbf{x} = \mathbf{C}\mathbf{g}(\mathbf{x})$$

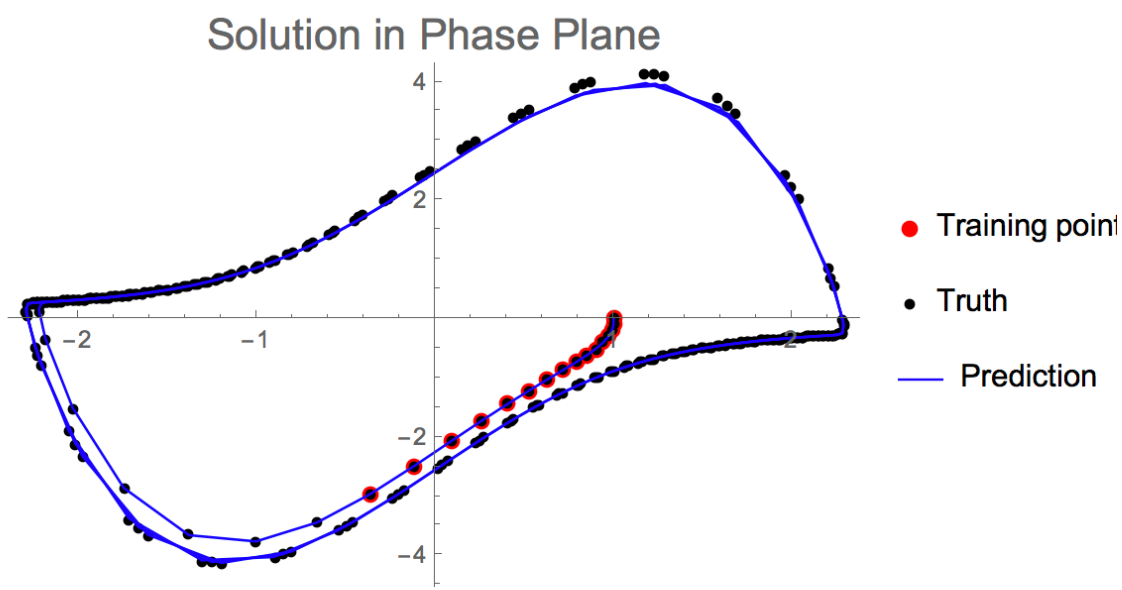
Extract governing equations

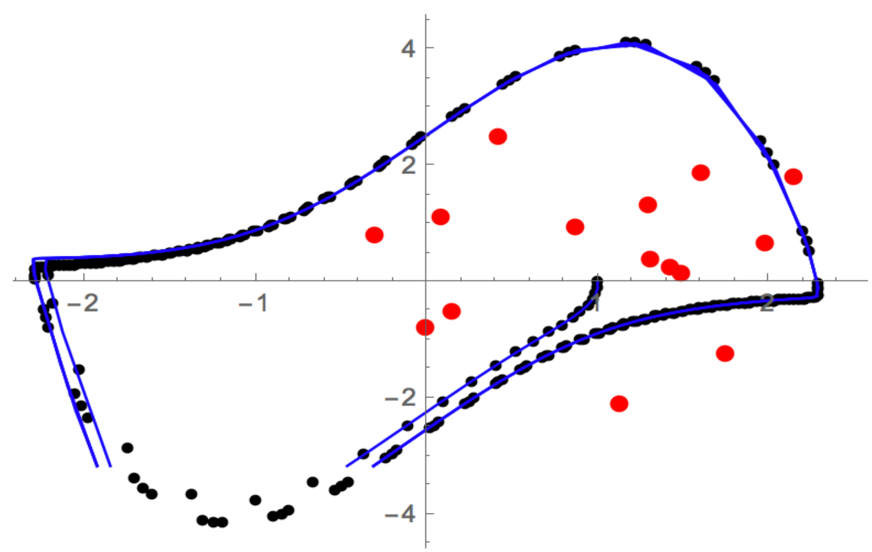
$$\mathbf{x}^{n+1} = \mathbf{C}\mathbf{K}\mathbf{g}(\mathbf{x}^n)$$

Discovering equations from data (global polynomial features)

$$\begin{pmatrix} x_1^{n+1} \\ x_2^{n+1} \end{pmatrix} = \begin{pmatrix} x_1^n + \Delta t x_2^n \\ x_2^n + \Delta t(\mu(1 - x_1^{2n}) x_2^n - x_1^n) \end{pmatrix}$$



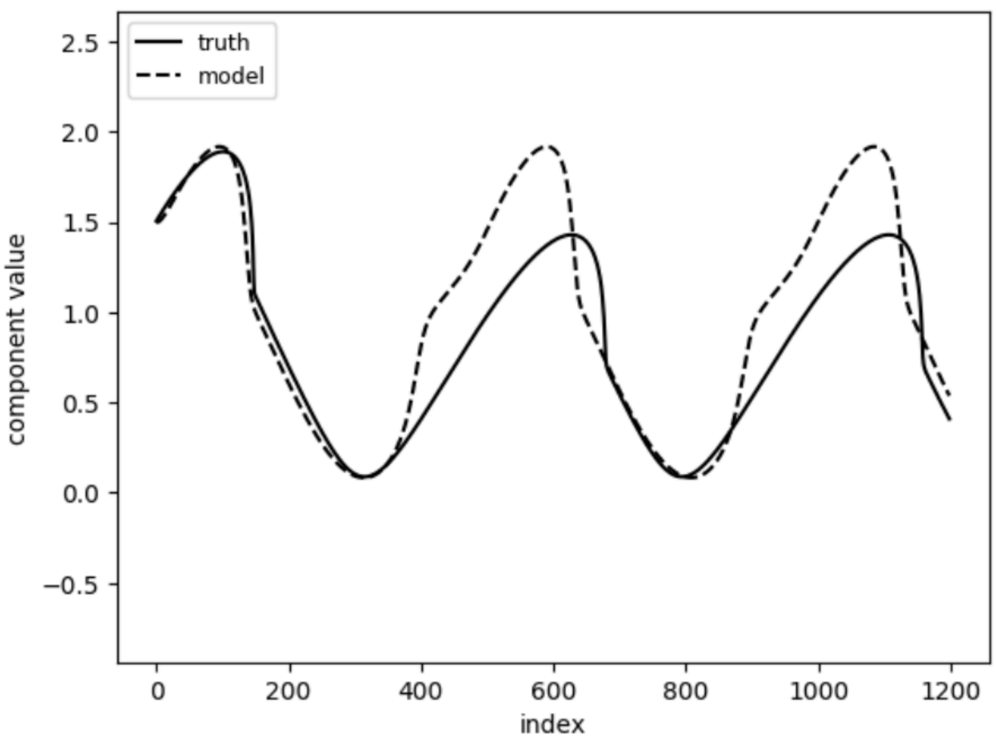


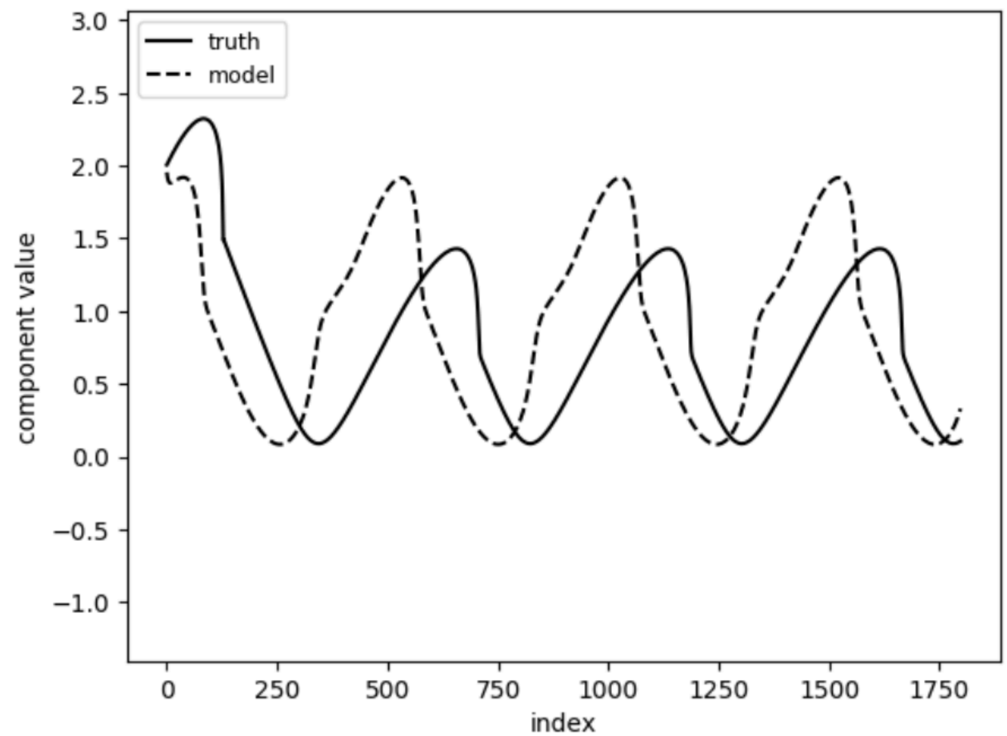


Discovering equations from data (global polynomial features)

)04:

$$\begin{pmatrix} x_1^{n+1} \\ x_2^{n+1} \end{pmatrix} = \begin{pmatrix} x_1^{n} \\ x_2^{n} \end{pmatrix} + \Delta t \begin{pmatrix} 2.5 - 100 \dfrac{x_1^n x_2^n}{1 + (x_2^n/0.52)^4} \\ -200 \dfrac{x_1^n x_2^n}{1 + (x_2^n/0.52)^4} + 9.2 - 2.3 x_2^n - 1.28 |x_2^n|^{3/2} \end{pmatrix}$$





Neural Networks / Deep-learning (local features)

Implicit feature selection

$$\boldsymbol{y} = f(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{b}) = W_3 \sigma(W_2 \sigma(W_1 \boldsymbol{x} + b_1) + b_2) + b_3$$

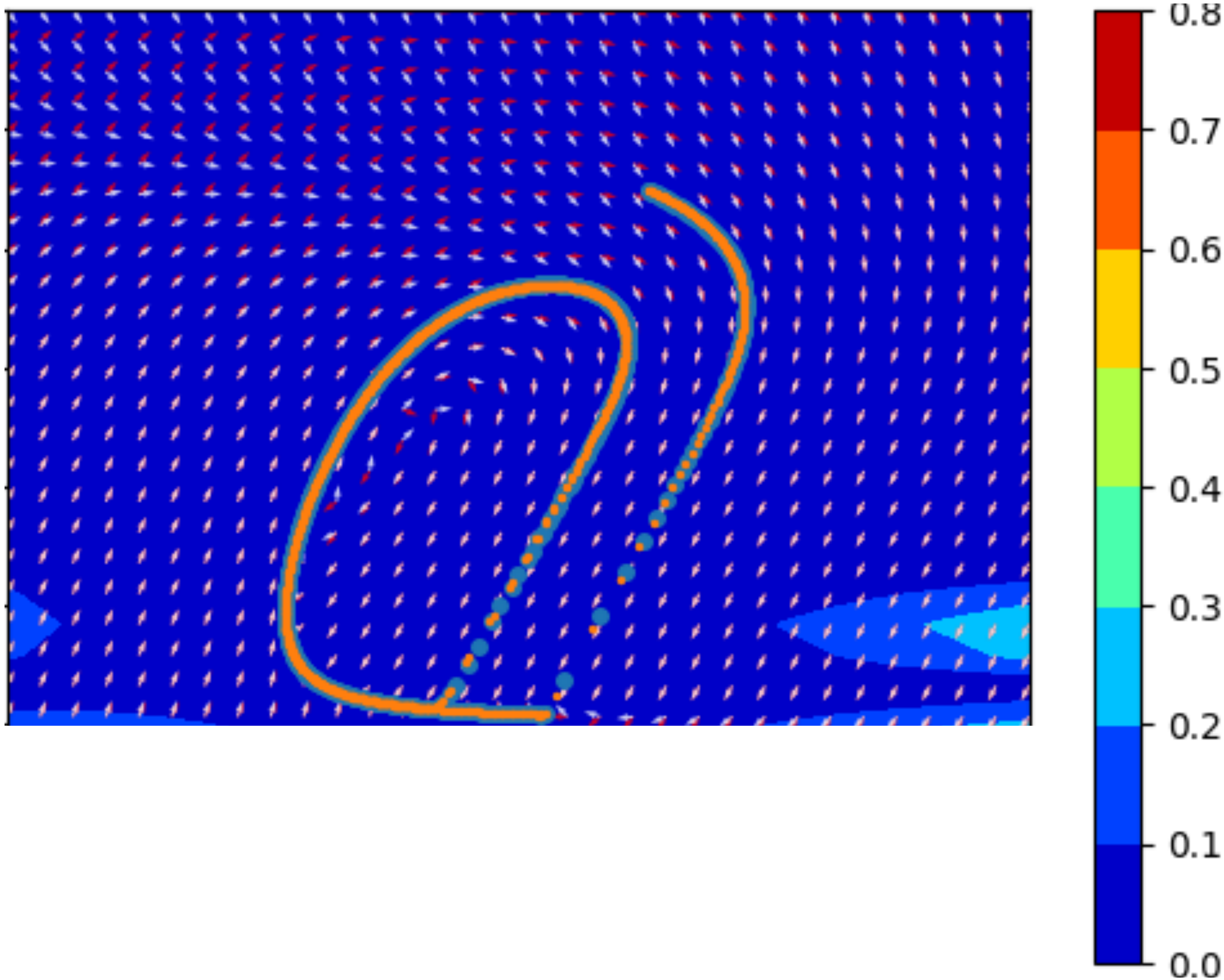
Minimize cost

$$\min_{\boldsymbol{W}^3, \boldsymbol{b}^3} \frac{1}{N_{train}} \sum_{i \in I_{train}} \| f(\boldsymbol{x}^i; \boldsymbol{W}^3, \boldsymbol{b}^3) - \boldsymbol{y}^i \|_2^2$$

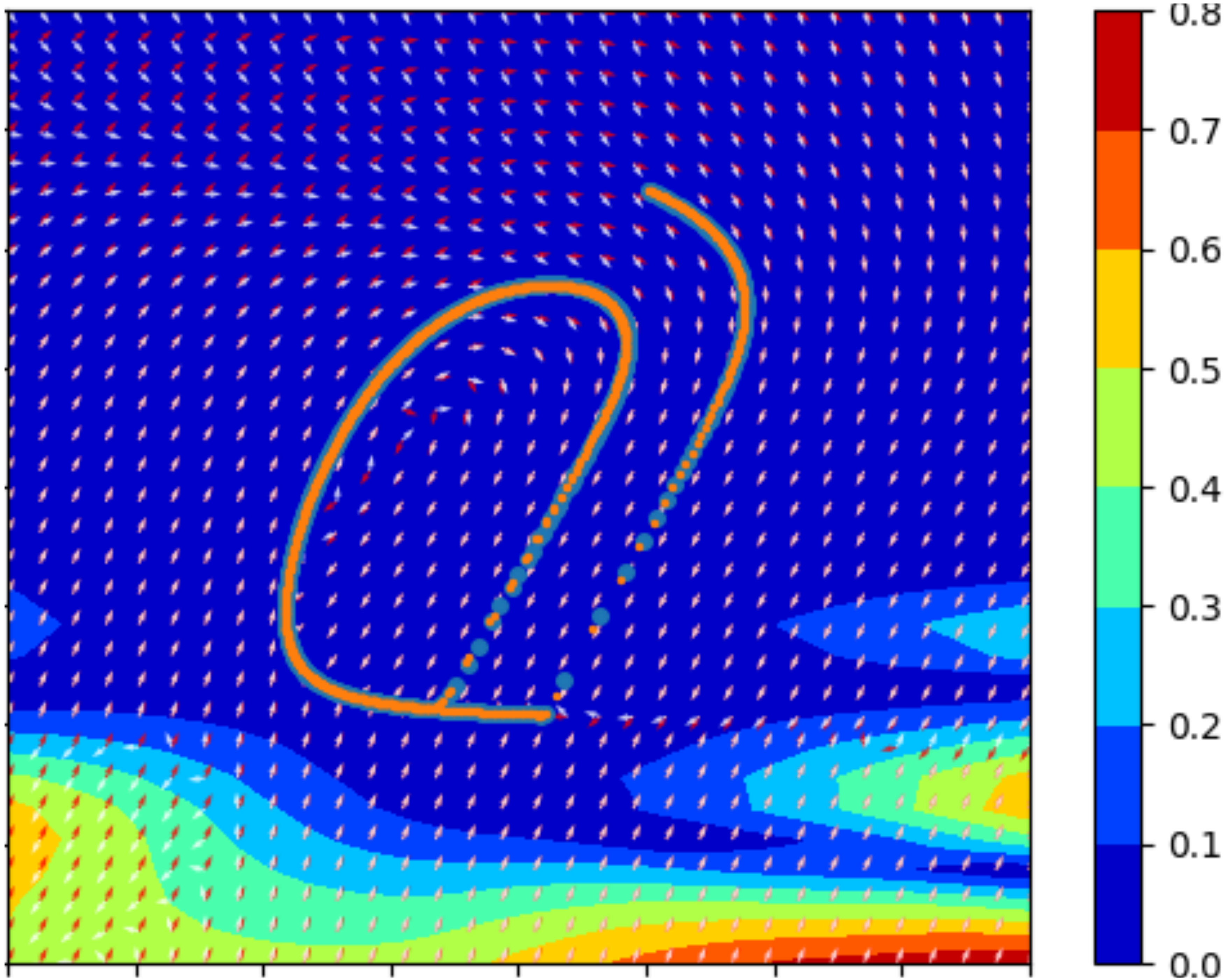
New idea : Local + Global

$$\min_{\boldsymbol{W}, \boldsymbol{b}} \frac{1}{N_{train}} \sum_{i \in I_{train}} \| f(\boldsymbol{x}^i; \boldsymbol{W}, \boldsymbol{b}) - \boldsymbol{y}^i \|_2^2 + \lambda \| J(\boldsymbol{x}^i; \boldsymbol{W}, \boldsymbol{b}) \|_F^2$$

Application of Deep learning



Application of Deep learning



Perspectives

Direct data-driven modeling / machine learning can work if we find magic basis or small number of features (ML can also help us find reduced dimensions)

Data driven discovery : Challenges

- Feature space has to be small ➔ If not, we will need too much data
- Tendency to overfit to data is very high, even with “best-practices”
- Perhaps a good interpolatory tool (in space of features)

Complex problems : Physics-constrained Data-augmented models

- There is a great amount of domain expertise, intuition

- Combine with physics based model & physics constraints, “Discover” model discrepancies and embed in a predictive setting

  ➔ Implicit dimension reduction. Machine learning can be actually powerful

  ➔ Optimal model, conditional on data and assumptions possible

  ➔ Most sensible thing to do if we care about specific problems (Lots of data for a class of problems, Lots of expertise/knowhow)

Physics-constrained Data-augmented modeling:
The setting

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

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

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

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

Approximate Model

Inferring, embedding discrepancies

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

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

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

$$\frac{\partial \bar{u}}{\partial t} + \mathcal{R}(\bar{u}) = \mathcal{M}^{\gamma}(\bar{u}, \delta(\bar{u}), \alpha)$$

Operators (γ) — Functions (δ) — Parameters (α)

Outline

- Introduction
- How do we setup the data-driven modeling problem?
- What are the components?
- Demonstration
  - ➔ Predictions in Airfoil flows
- Scaling / computer science, etc.
- Vision / Perspectives

Field Inversion & Machine learning (FIML)

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

Field Inversion

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

Information

Spatial discrepancy

Field Inversion & Machine learning (FIML)

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

Field Inversion

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

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

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

Information — Spatial discrepancy

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

Machine Learning

Knowledge — Functional discrepancy

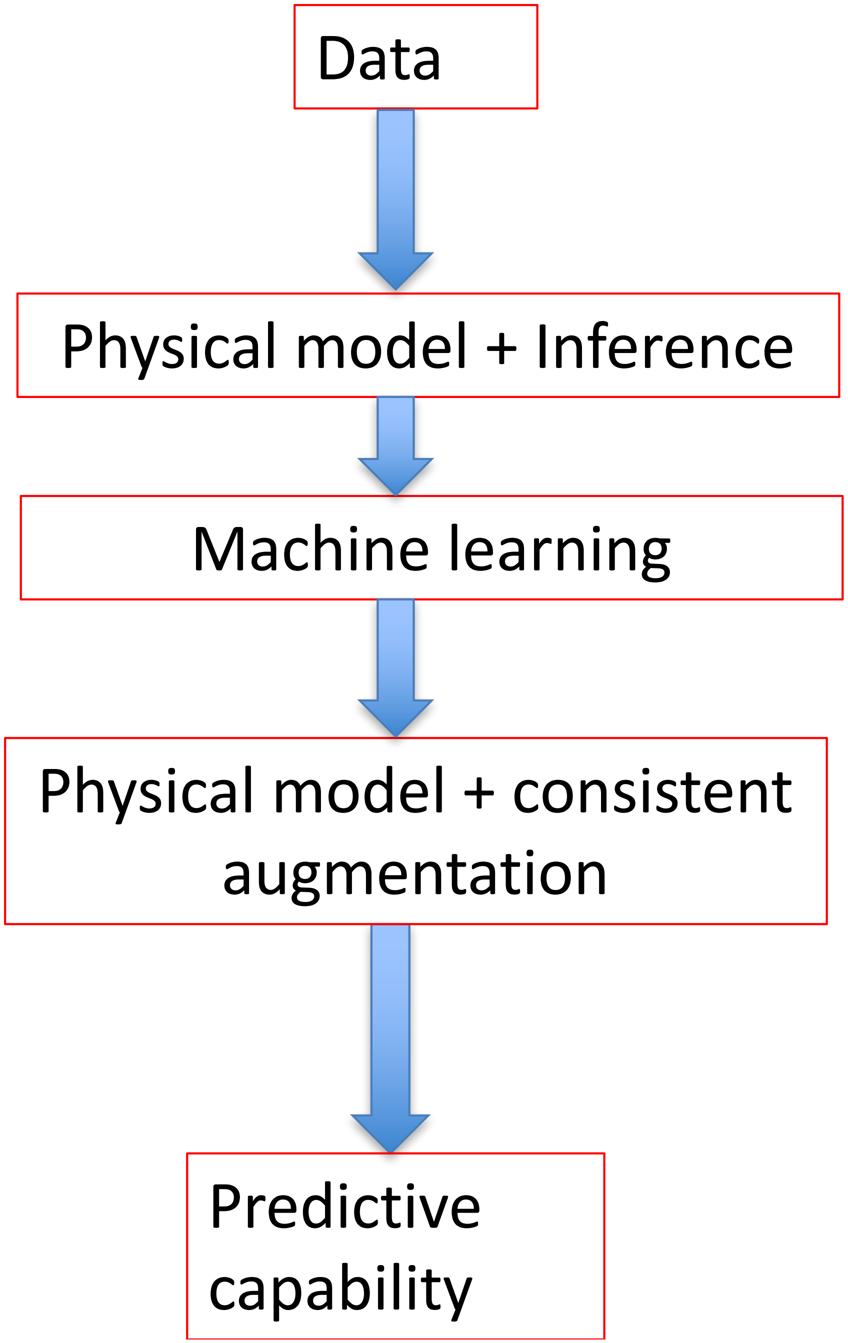
$$\hat{\delta}(\, f(Q) \,)$$

Embedding

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

Prediction : Injection into solver

How does it address the challenges?



- 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 quantities

- Data will be noisy and of variable quality, inherent uncertainty
  ➔ Probabilistic casting of inference and learning

Outline

- Introduction
- How do we setup the data-driven modeling problem?
- What are the components?
- Demonstration
    - ➔ Predictions in Airfoil flows
- Scaling / computer science, etc.
- Vision / Perspectives

1) Inference

3) Machine Learning

2) Design of Experiments

4) Prediction

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,

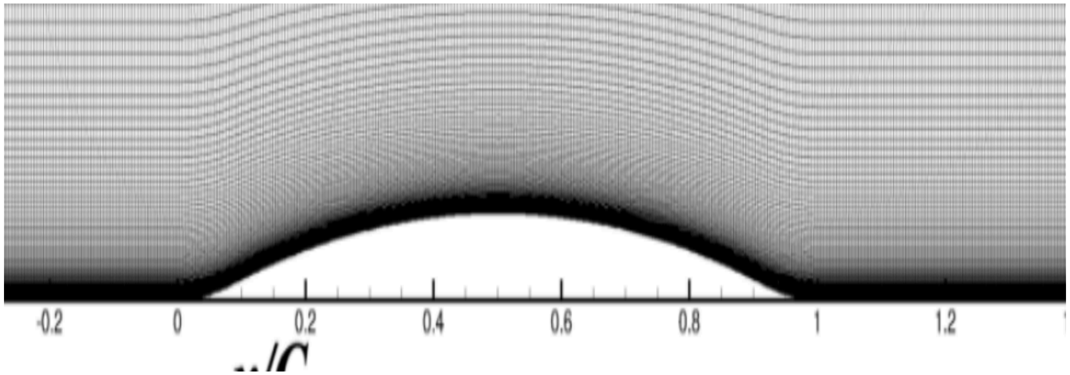
$$\nu_{i,n}\frac{\partial R_m}{\partial u_n} = -\frac{\partial R_m}{\partial\boldsymbol{\beta}_i}$$

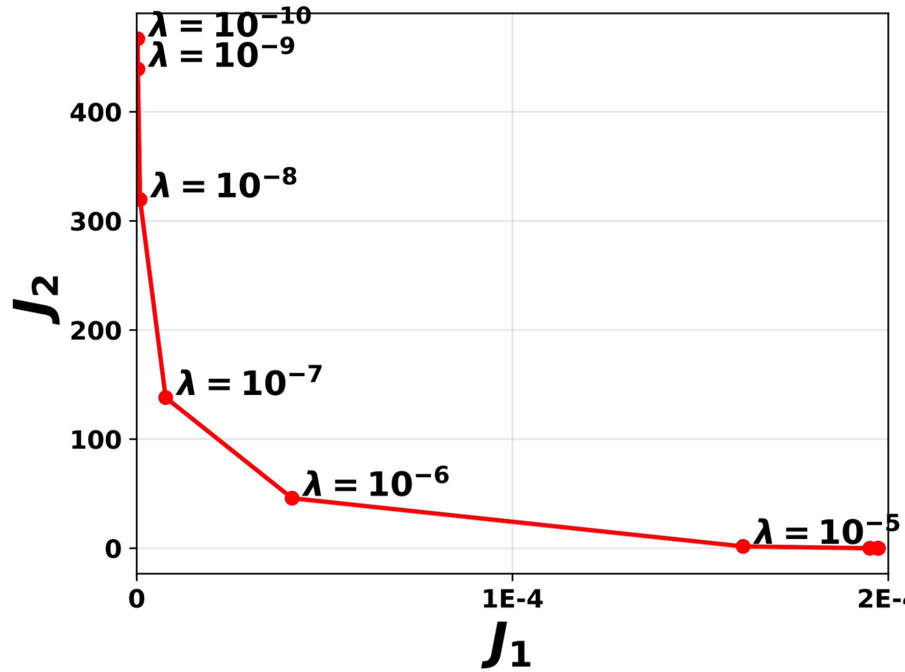
$$\mu_{i,m}\frac{\partial R_m}{\partial u_k} = -\frac{\partial^2 F}{\partial\boldsymbol{\beta}_i\partial u_k} - \psi_m\frac{\partial^2 R_m}{\partial\boldsymbol{\beta}_i\partial u_k} - \nu_{i,n}\frac{\partial^2\mathfrak{J}}{\partial u_n\partial u_k} - \nu_{i,n}\psi_m\frac{\partial^2 R_m}{\partial u_n\partial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

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

More complete PDFs with accelerated MCMC (with P. Constantine U. Colorado)

Example : Flow over a bump – Field inversion

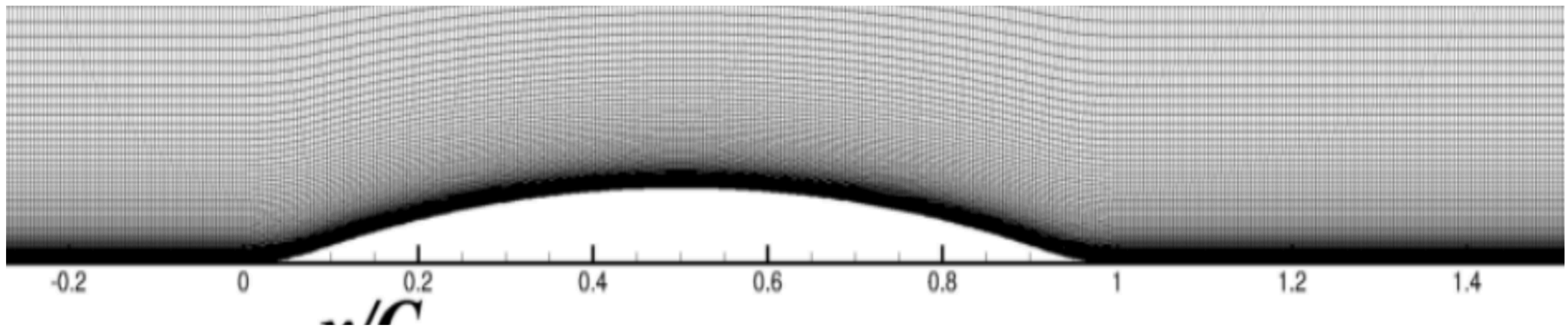


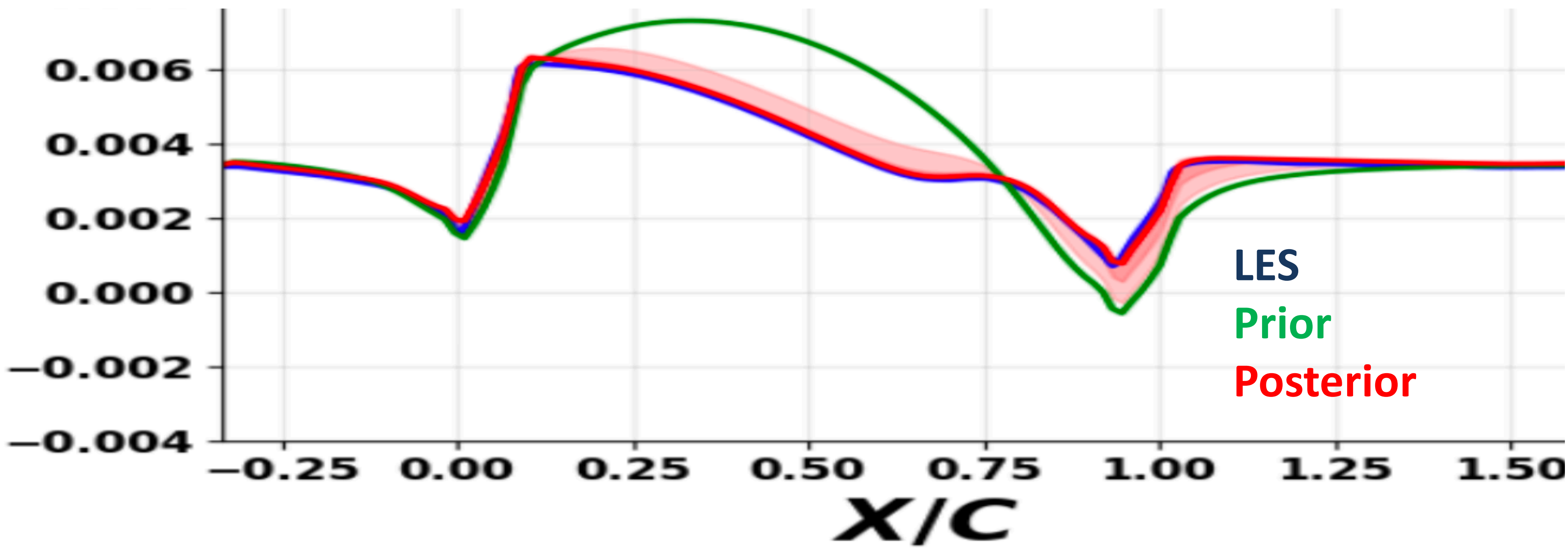


$$\frac{D\omega}{Dt} = \beta(\boldsymbol{x})P(k,\omega,\mathbf{U}) - D(k,\omega,\mathbf{U}) + T(k,\omega,\mathbf{U}).$$

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

Inferred quantity - Cf

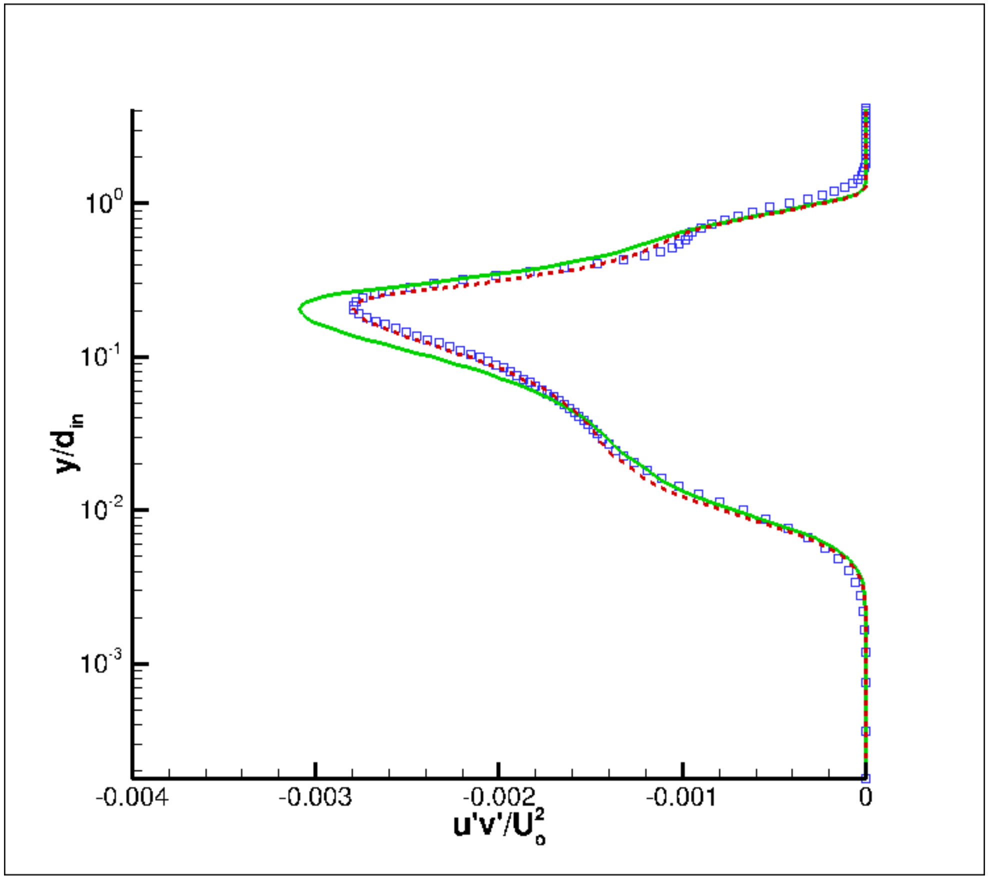


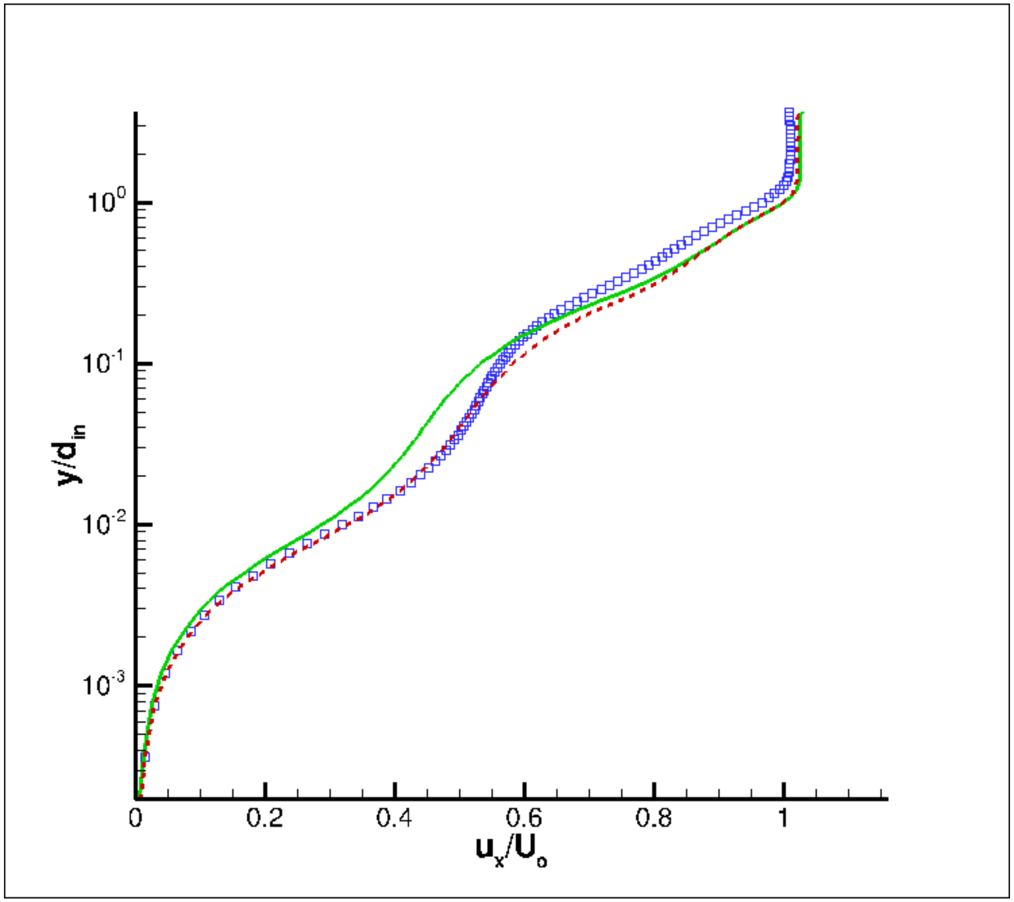


Secondary quantities

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



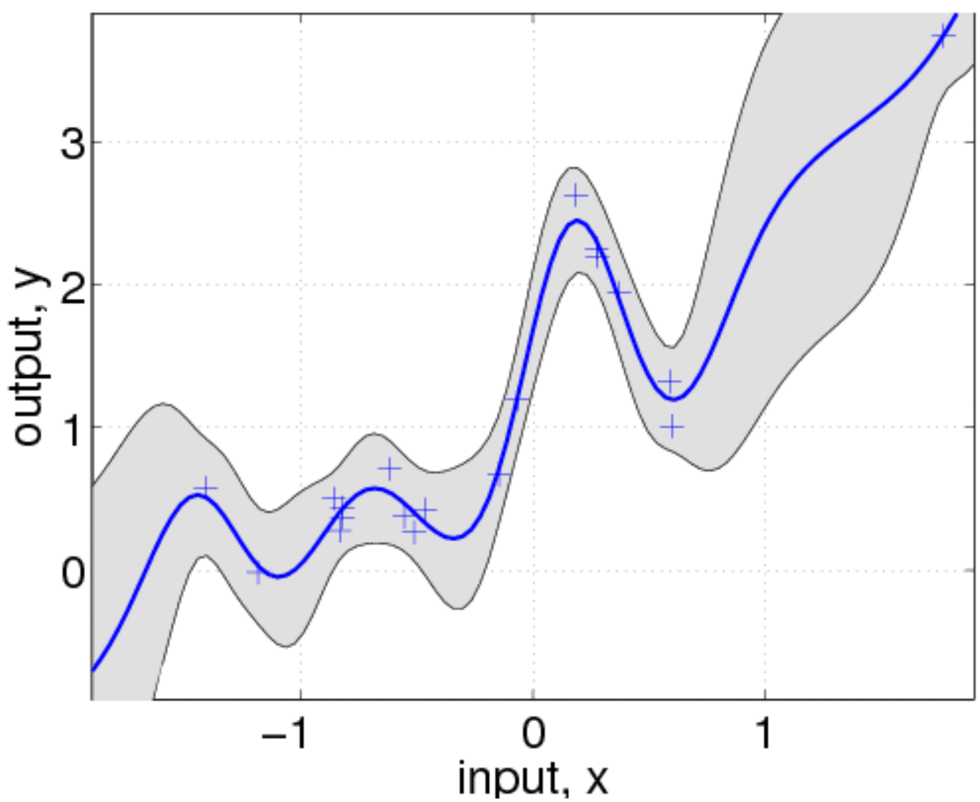


Machine Learning

Supervised Learning: Given a set of labeled data $\{x_i, y_i\}$, learn the mapping y(x)

Unsupervised Learning: Given data, discover patterns and groupings

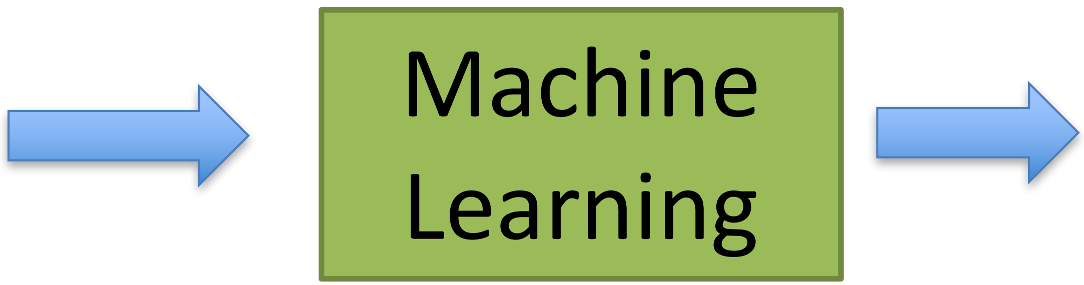
Typically cast in a probabilistic framework ➔ Deep connections with statistical mechanics



GPML

How to transform information to knowledge?

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



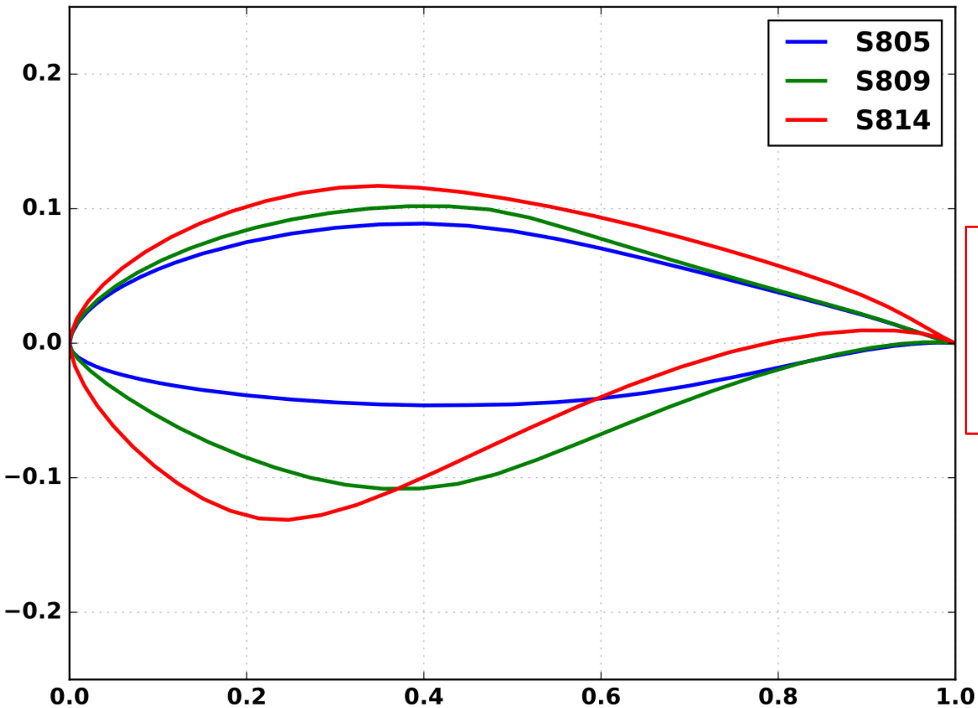
$$\eta_1, \eta_2, \ldots$$

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

Outline

- Introduction
- How do we setup the data-driven modeling problem?
- What are the components?
- Demonstration
  - ➔ Predictions in Airfoil flows
- Scaling / computer science, etc.
- Vision / Perspectives

Prediction in Airfoil flows



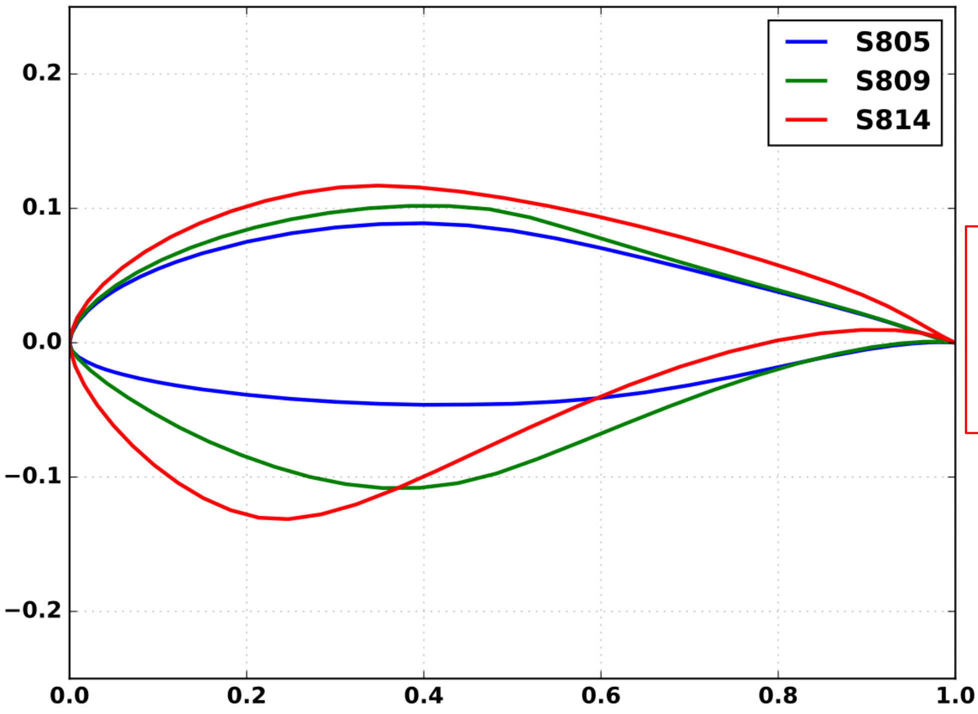
Singh, A., Medida, S. & Duraisamy, K., Data-augmented Predictive Modeling of Turbulent Separated Flows over Airfoils, AIAA Journal, 2017.

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

$$-\rho \overline{u_i' u_j'} = 2\mu_t \overline{S}_{ij} \qquad \mu_t = \rho \hat{\nu} f_{v1}$$

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

Prediction in Airfoil flows

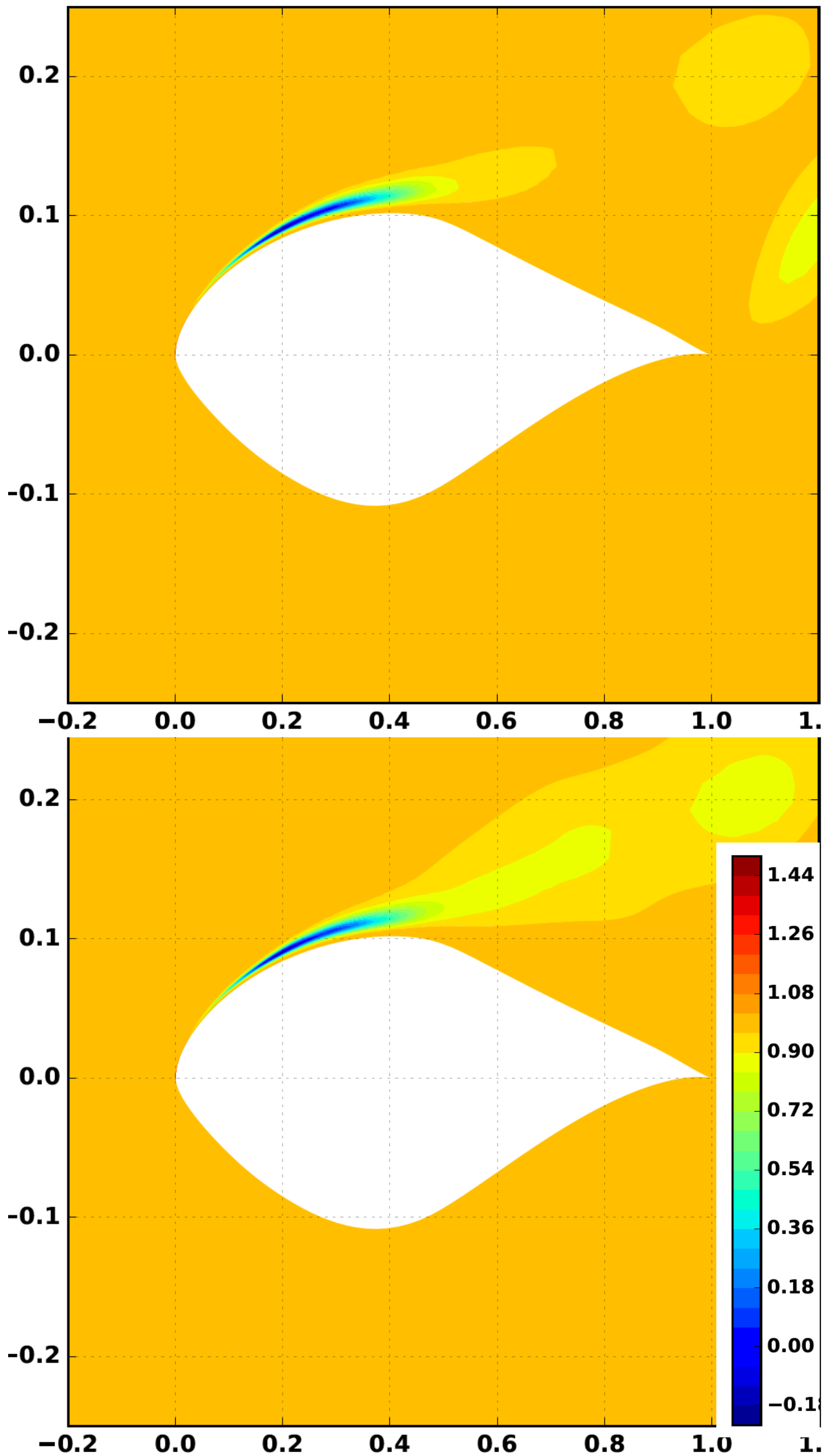


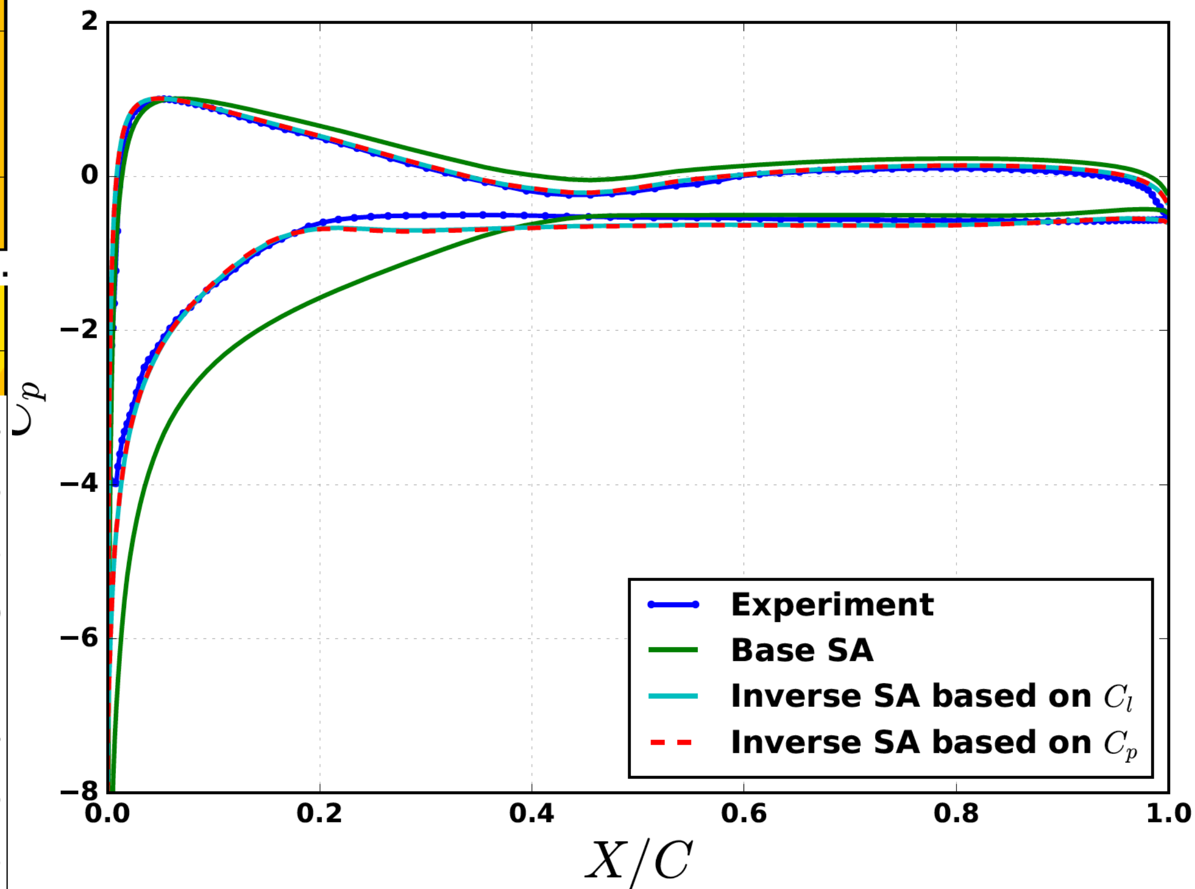
Singh, A., Medida, S. & Duraisamy, K., Data-augmented Predictive Modeling of Turbulent Separated Flows over Airfoils, AIAA Journal, 2017.

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 $Re = 1 \times 10^6, 2 \times 10^6$

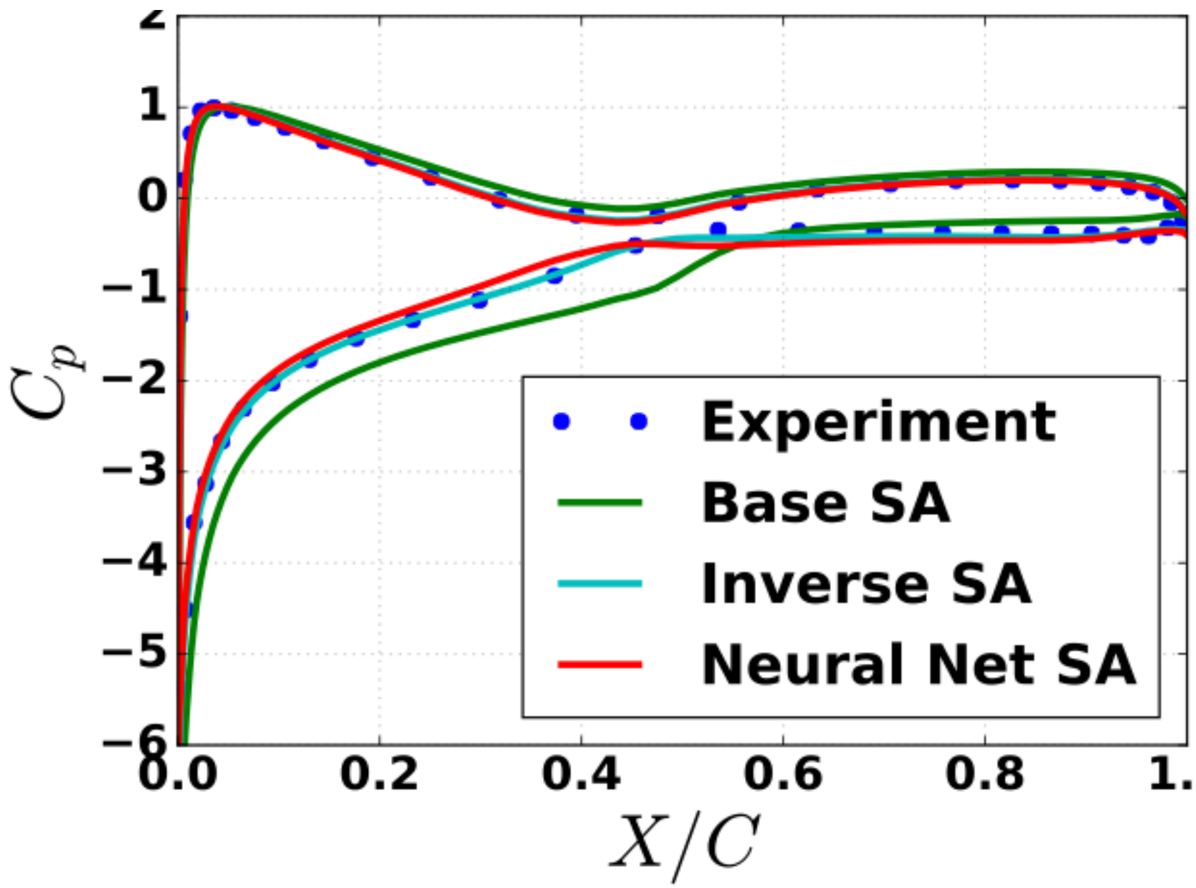
Training set ➔

Inversion based on Pressures
vs
Inversion based on LIFT!



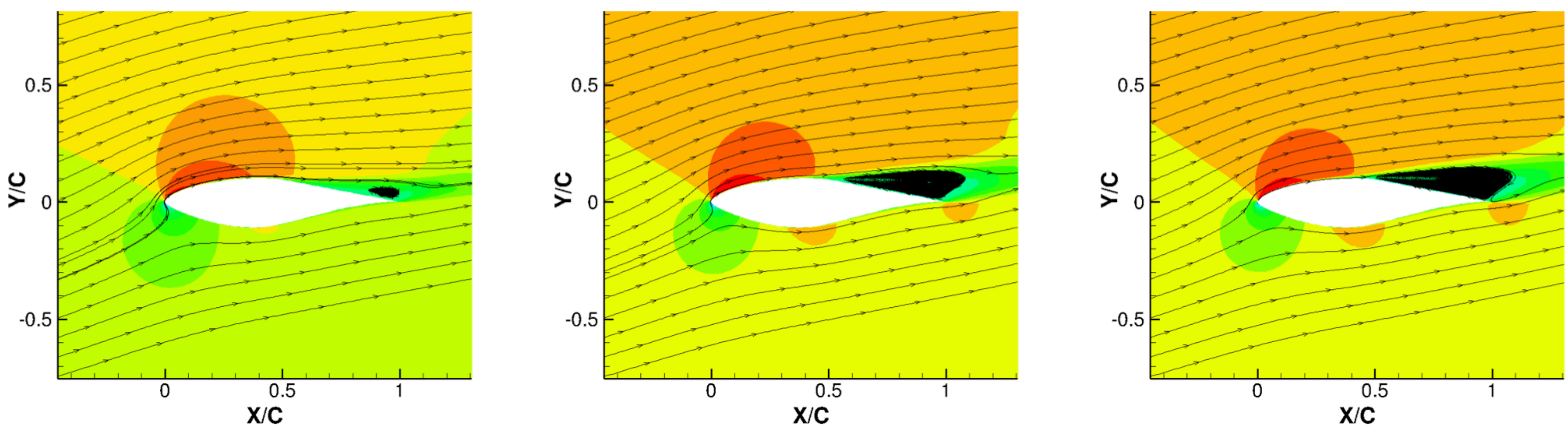


Ability to work on sparse amount of data is critical



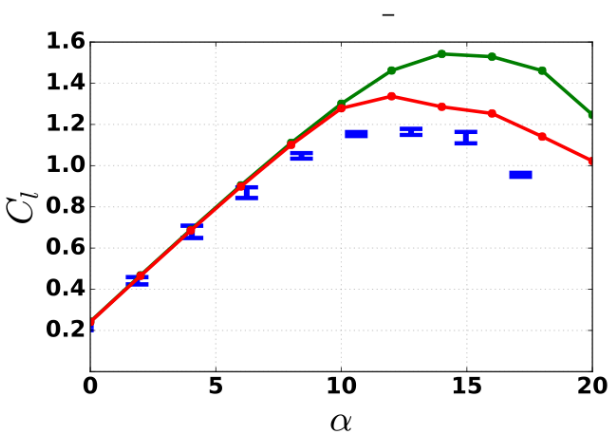
True prediction !

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

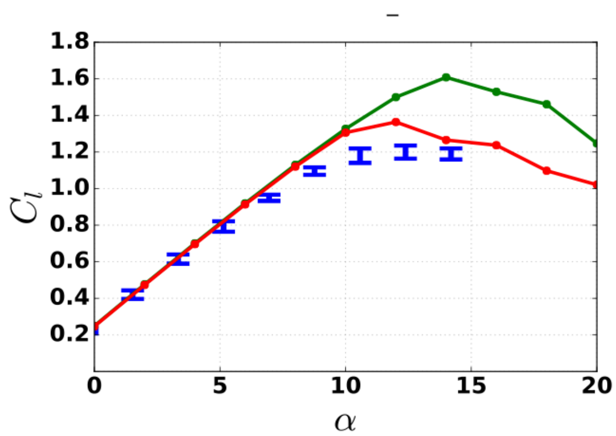


(a) Base SA (b) Inverse SA (c) NN-augmented SA (prediction)

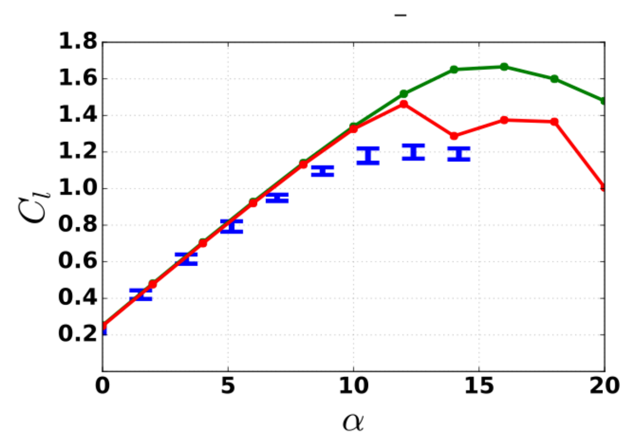
Prediction – S805



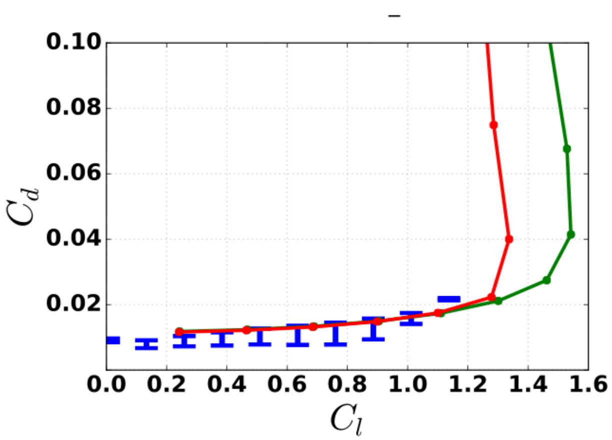
(a) $Re = 1 \times 10^6$



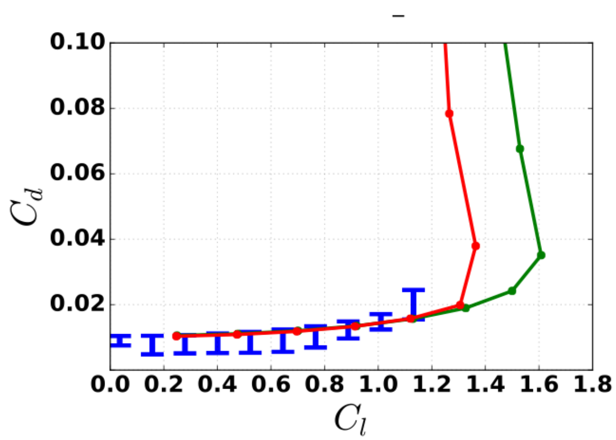
(b) $Re = 2 \times 10^6$



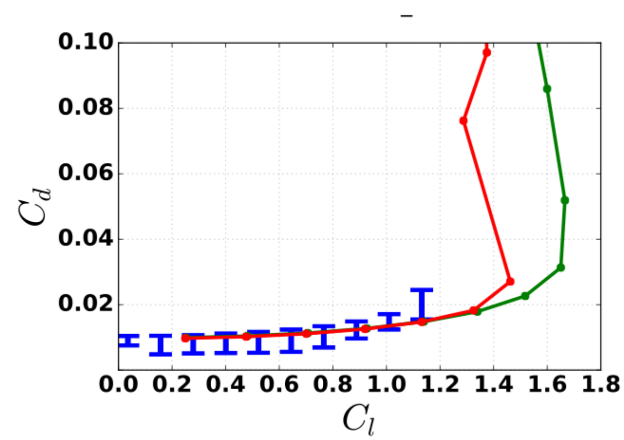
(c) $Re = 3 \times 10^6$



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



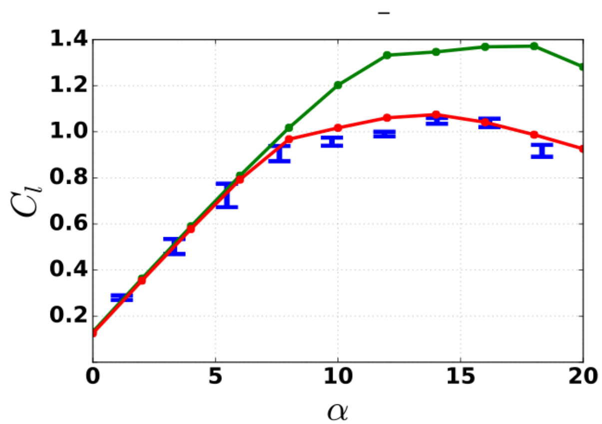
(e) $Re = 2 \times 10^6$



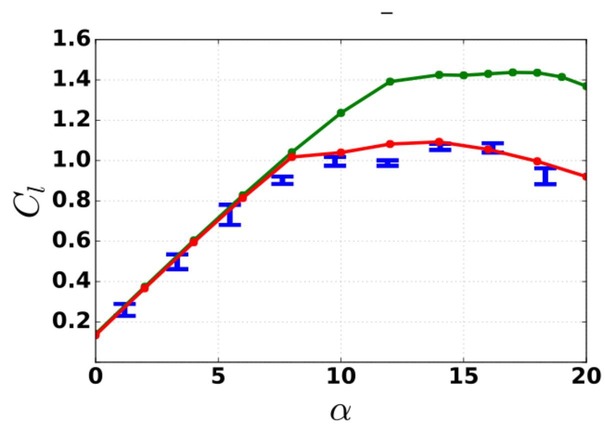
(f) $Re = 3 \times 10^6$

Collaboration with Altair, Inc.

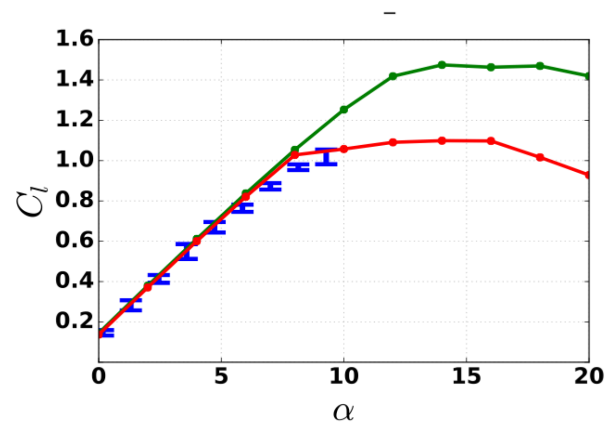
Prediction – S809



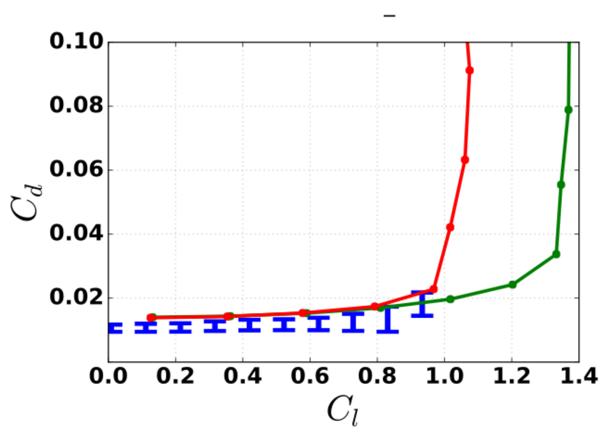
(a) $Re = 1 \times 10^6$



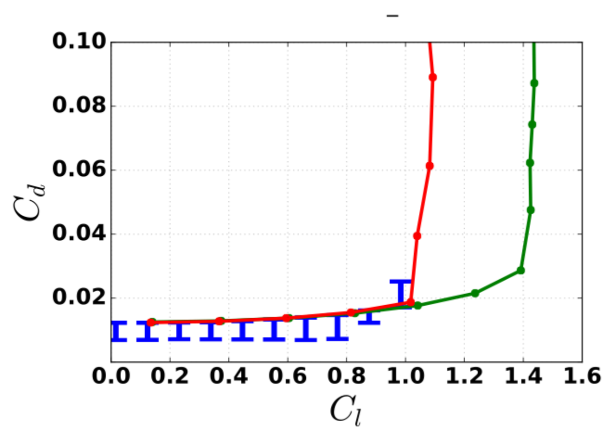
(b) $Re = 2 \times 10^6$



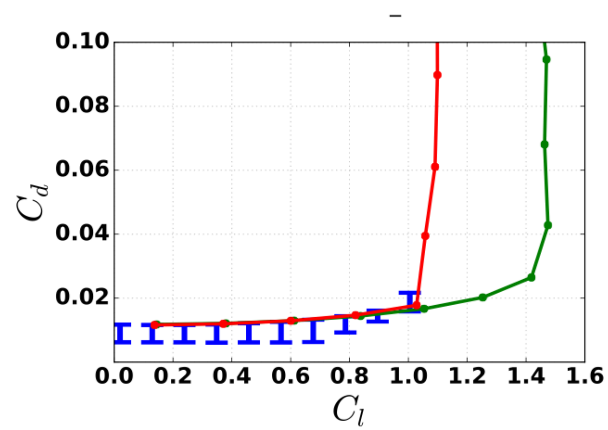
(c) $Re = 3 \times 10^6$



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



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

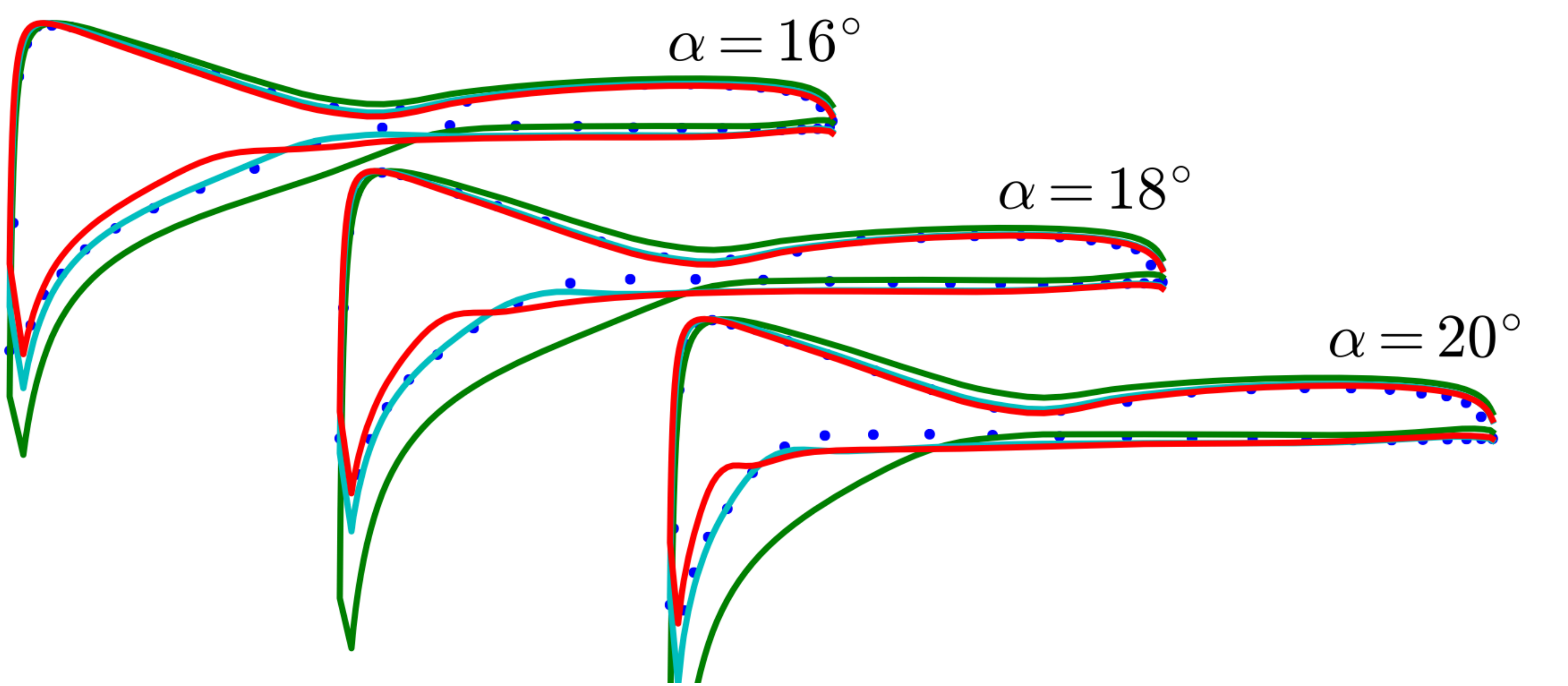


(f) $Re = 3 \times 10^6$

Collaboration with Altair, Inc.

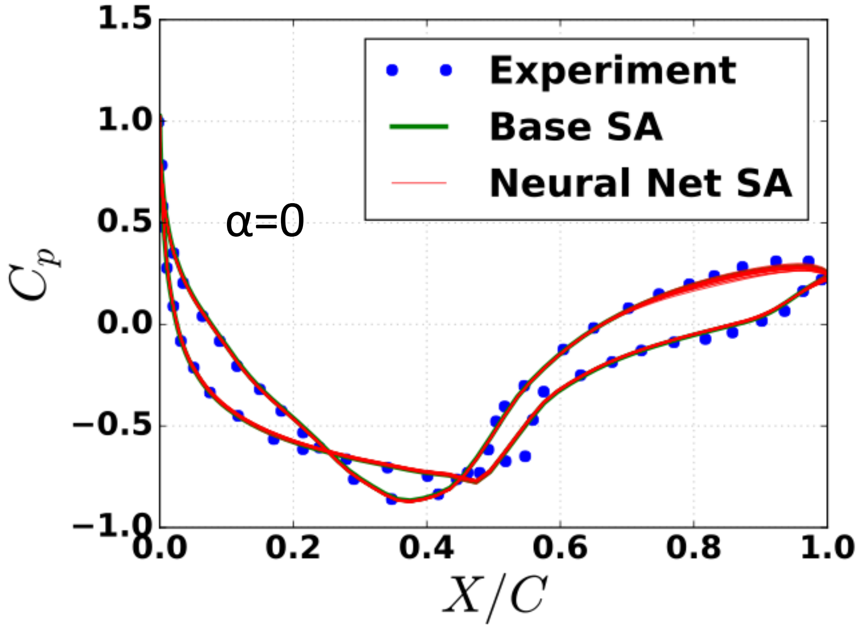
True prediction !

S 809, Re=2 Million

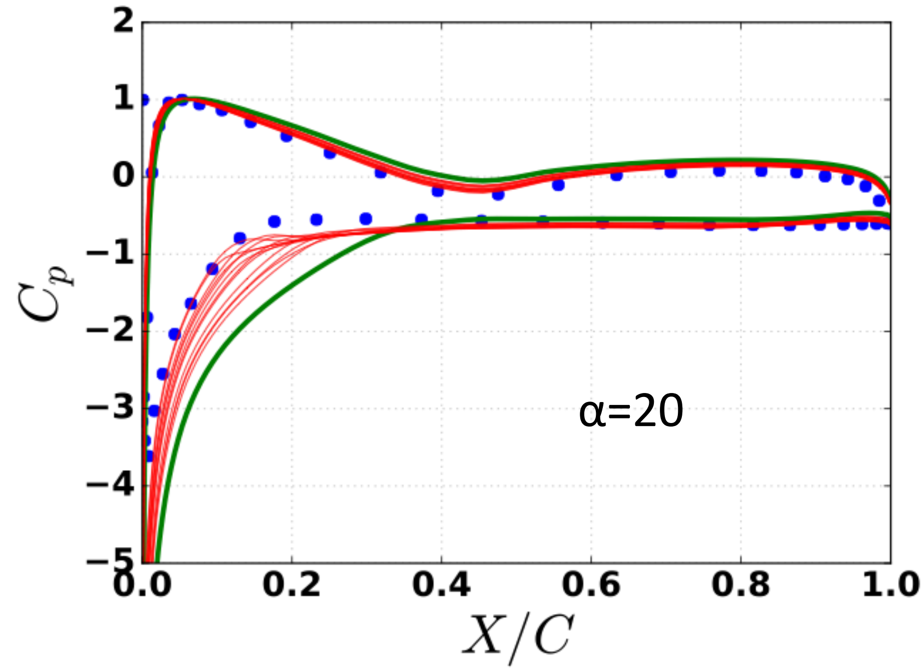


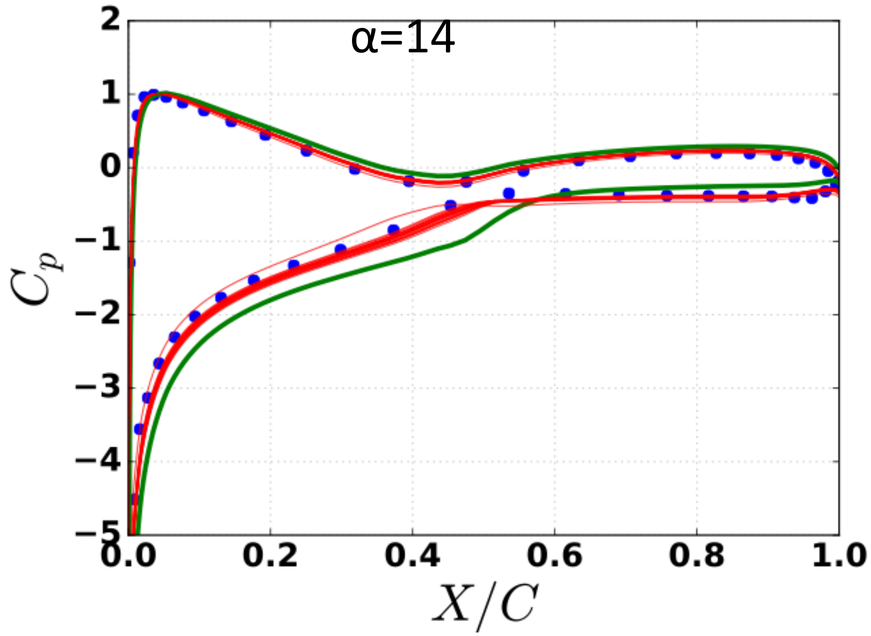
Inference used only CL data, NN-augmented model provides considerable predictive improvements of Cp

Variability



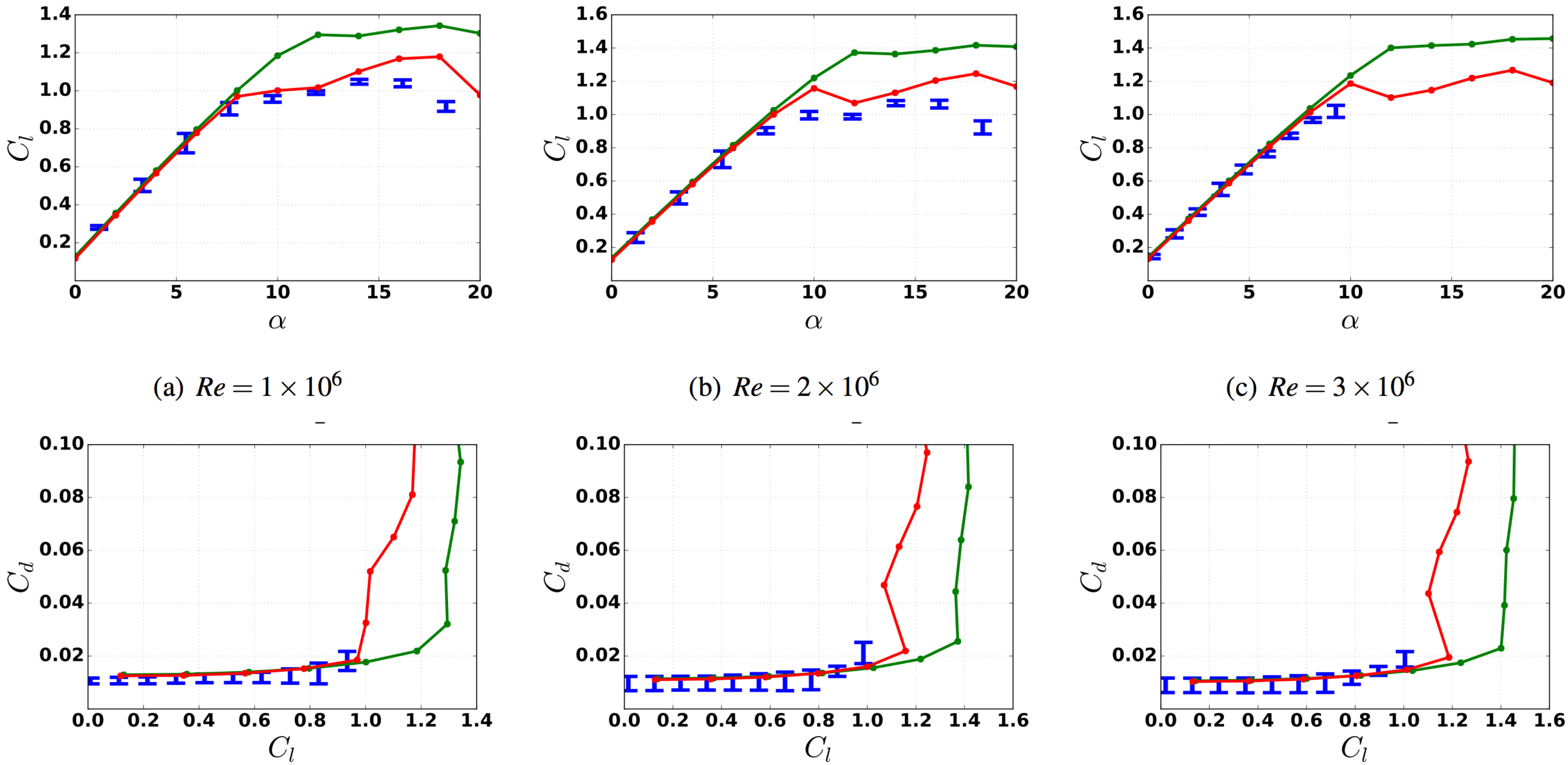
S 809, Re=2 Million





Training from different sets

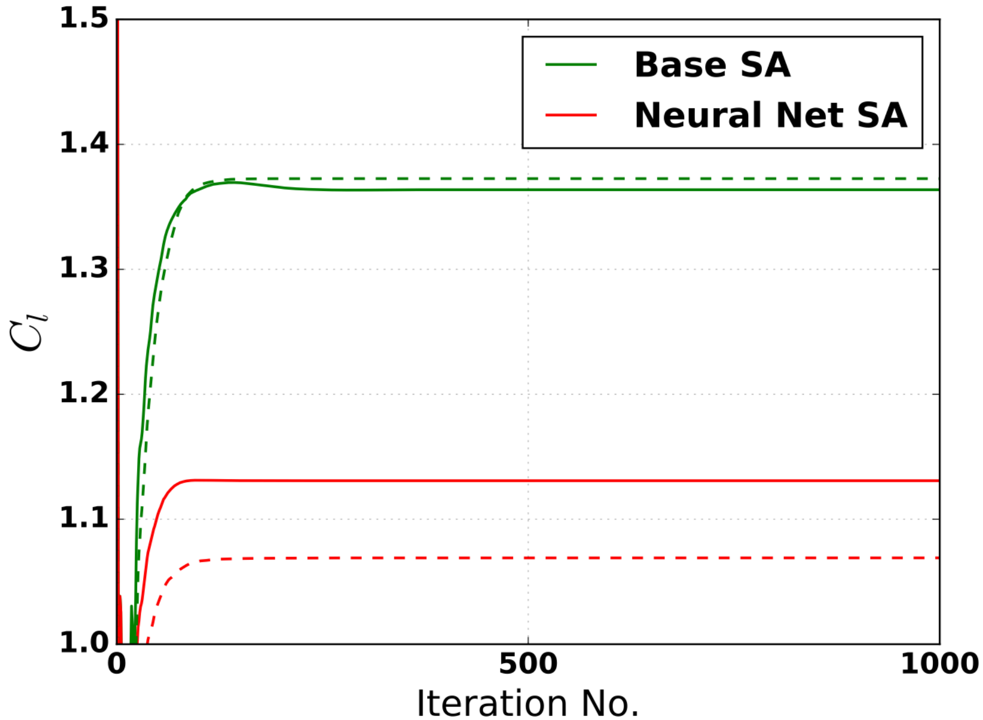
Portability :
Implementation in AcuSolve



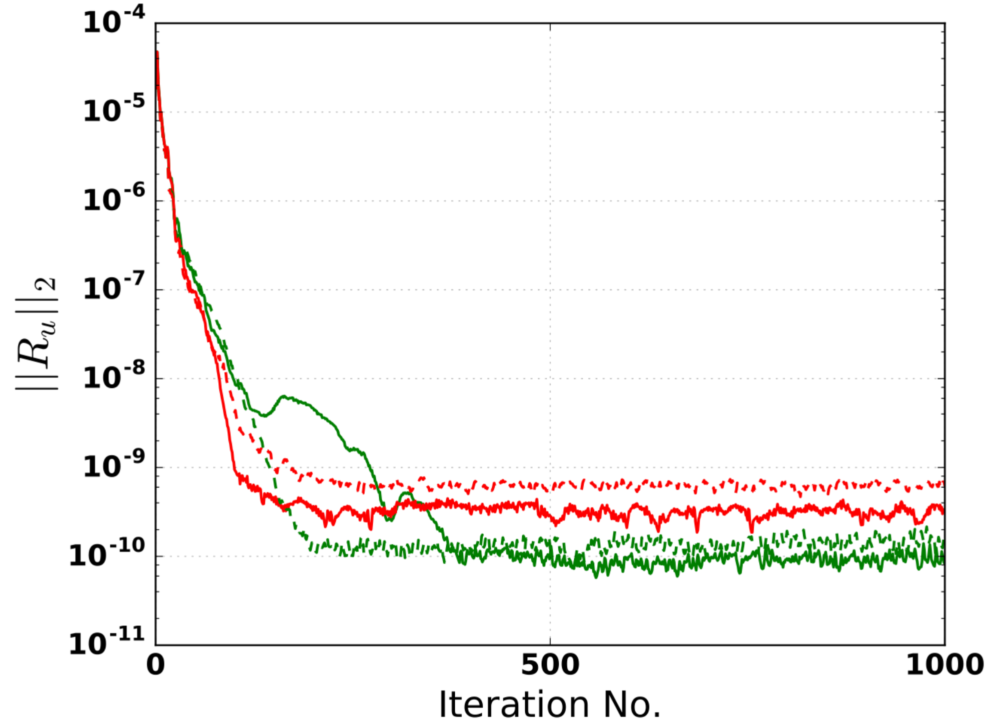
(a) $Re = 1 \times 10^6$ (b) $Re = 2 \times 10^6$ (c) $Re = 3 \times 10^6$

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

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.

Turbulence Modeling Gateway

43 Registered users

Home | Team | Research | Publications | Symposium 2017 | Support

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

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

Some more opinions

- Duraisamy, K. Data-enabled, Physics-constrained Predictive Modeling of Complex Systems, SIAM News July 2017
- Duraisamy, K., Spalart, P., Rumsey, C., Status, Emerging Ideas and Future Directions of Turbulence Modeling Research in Aeronautics, NASA TM 2017-0011477
- Duraisamy, K., Xiao, H., & Iaccarino, G., Turbulence Modeling in the Age of Big Data, Annual Review of Fluid Mechanics, 2019.

High level takeaways

"Machine learning for modeling" is like saying "Integration-based CFD" to define a finite volume solver.

➔ Machine learning is just one tool and a small part of data-driven modeling

Applying Machine learning on CFD data directly is pointless. Don't do it.

➔ Can apply machine learning on CFD information. Generate information via statistical inference and/or human inference

Once the problem has been reduced to something that Machine learning can process, we can achieve success

➔ Example: Low-dimensional feature space, appropriate basis, etc.

Physics, expert knowledge and math can help this reduction

Finer-grained Perspectives 1/2

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

Acknowledgements

NASA (Gary Coleman);

DARPA (Fariba Fahroo);

NSF (Ron Joslin)

Outline

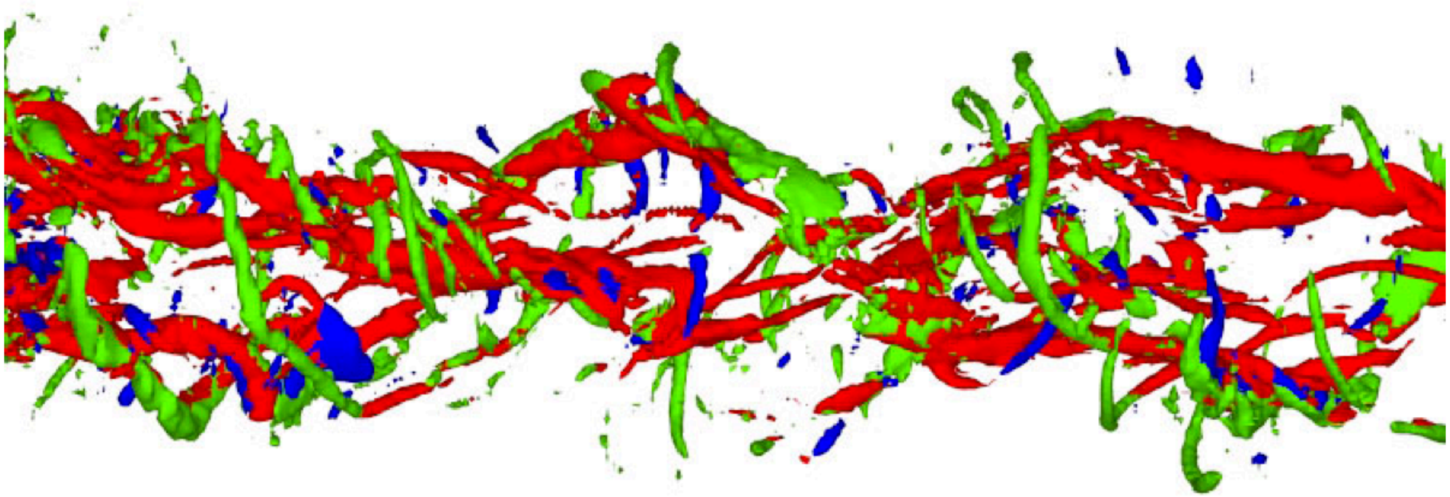
- Introduction
- How do we setup the data-driven turbulence modeling problem?
- What are the components?
- Demonstration
  - ➔ Predictions in Airfoil flows
- Scaling / computer science, etc.
- Vision / Perspectives

How can we scale up these problems?

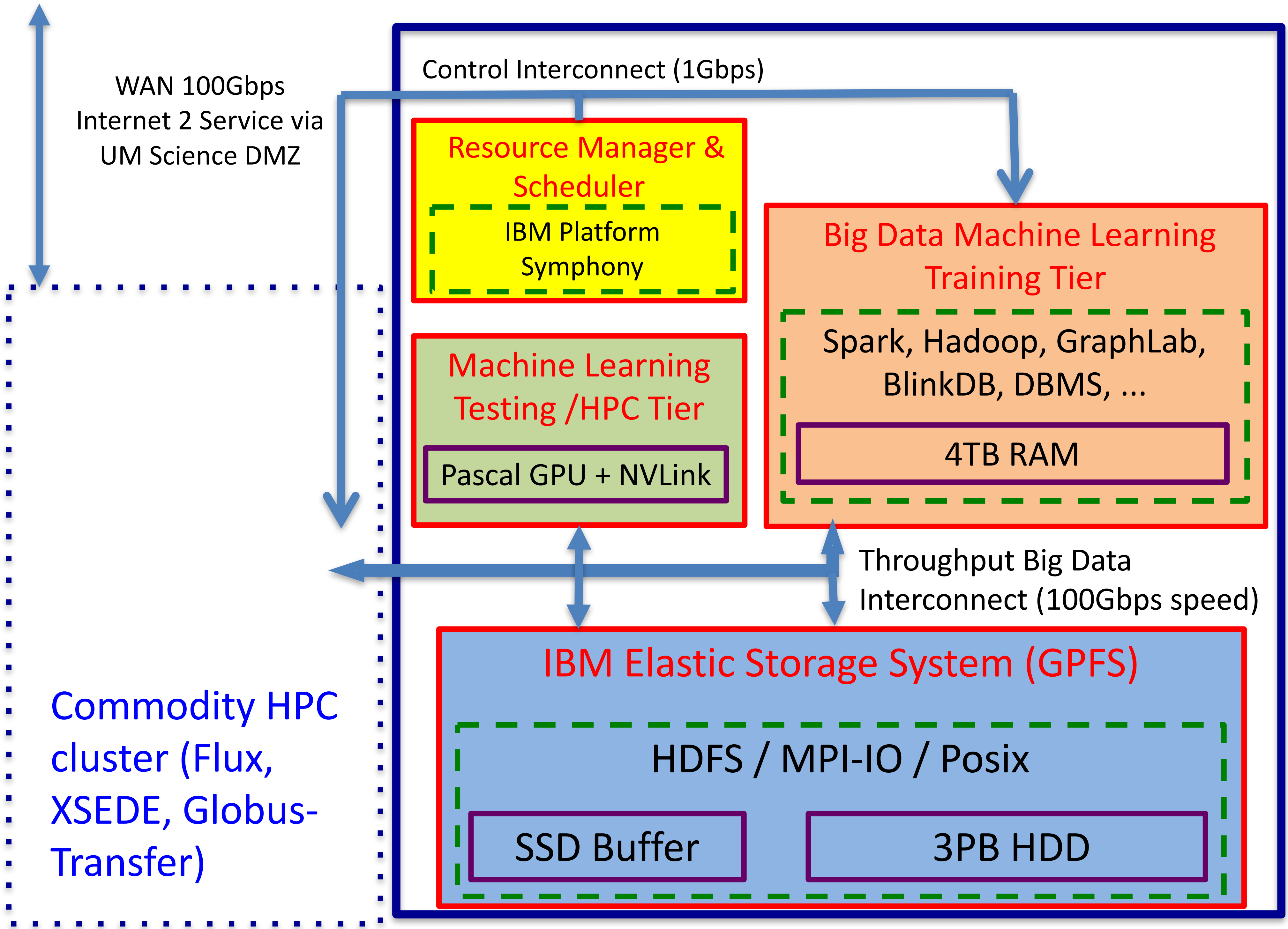
$3.46M to combine supercomputer simulations with big data

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)

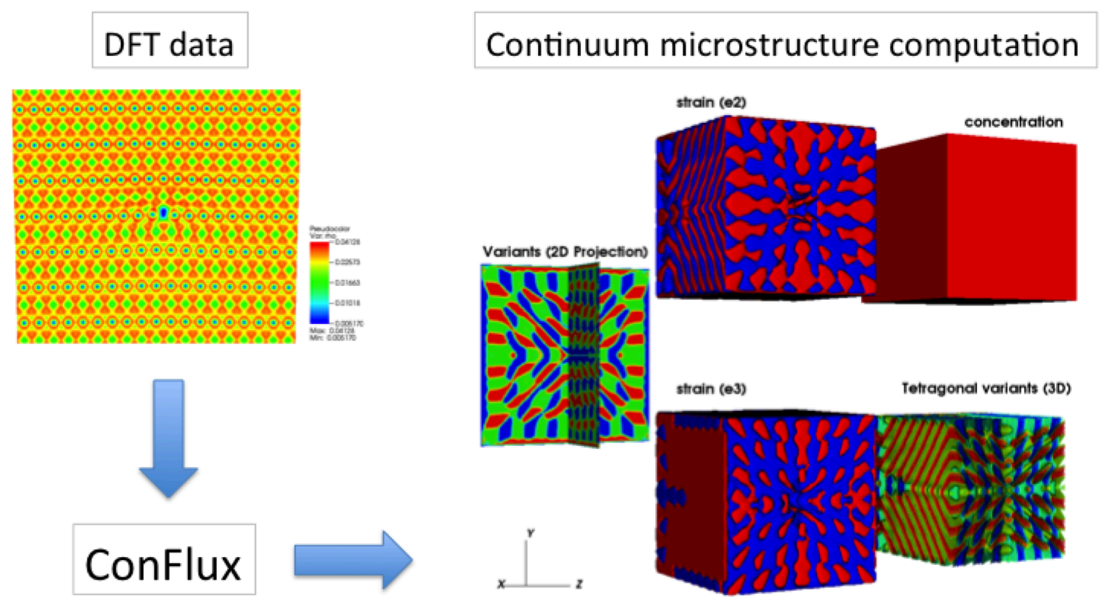
Materials Modeling

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

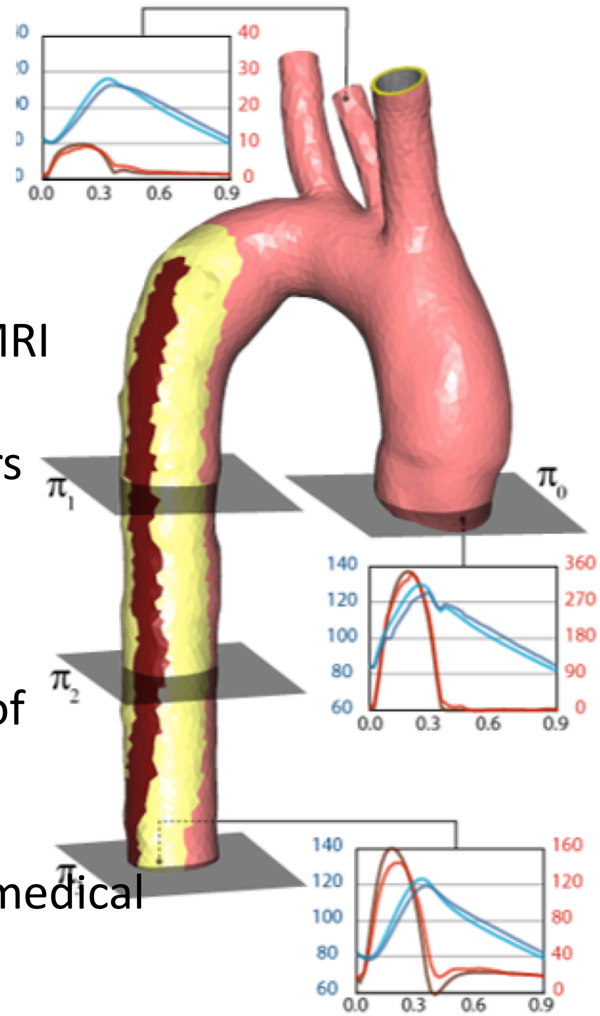
Obtain data from tomography and MRI

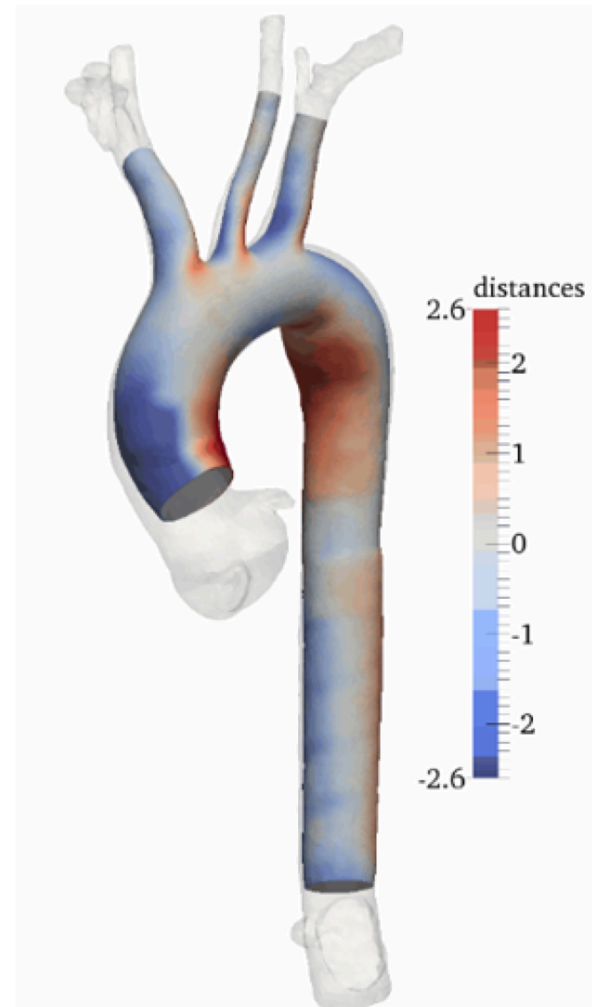
Solve inverse problem for parameters

Massive data size

On-the-fly Lagrangian computation of Motion

Evaluation of arterial stiffness from medical Images !





With Prof. Alberto Figueroa (Biomedical Engineering & Surgery)

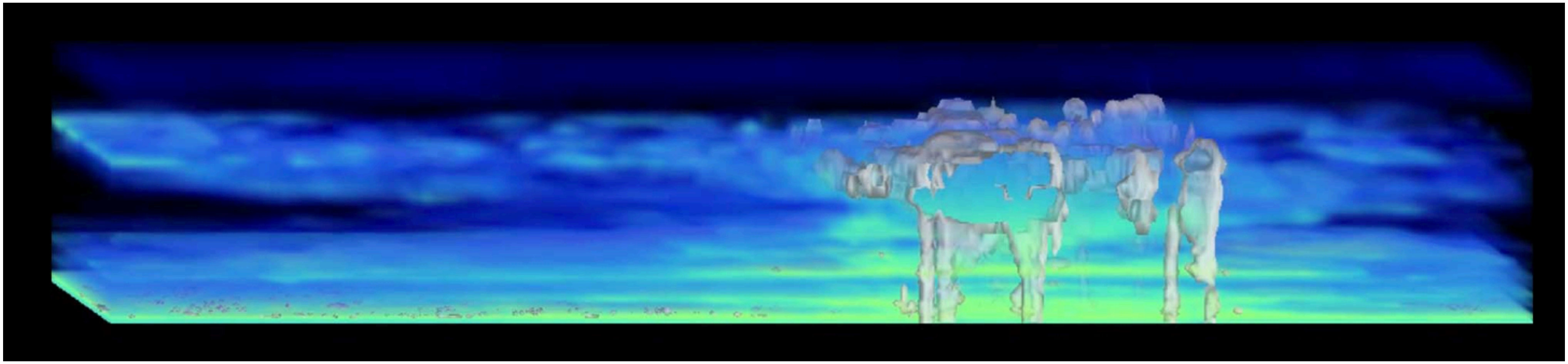
Climate system interactions

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)

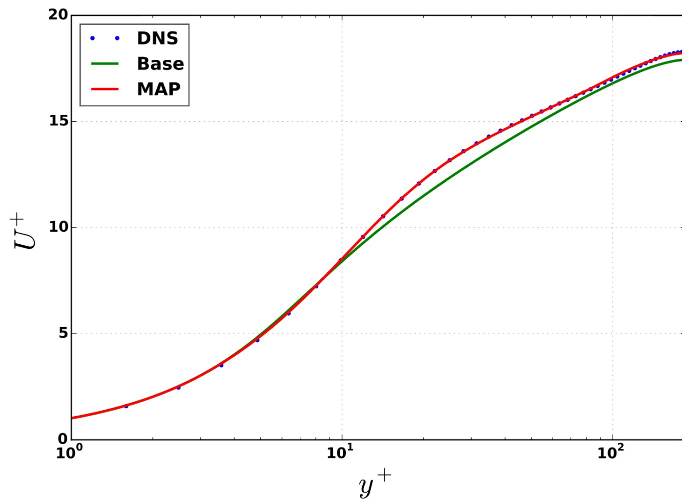
Some topics for discussion

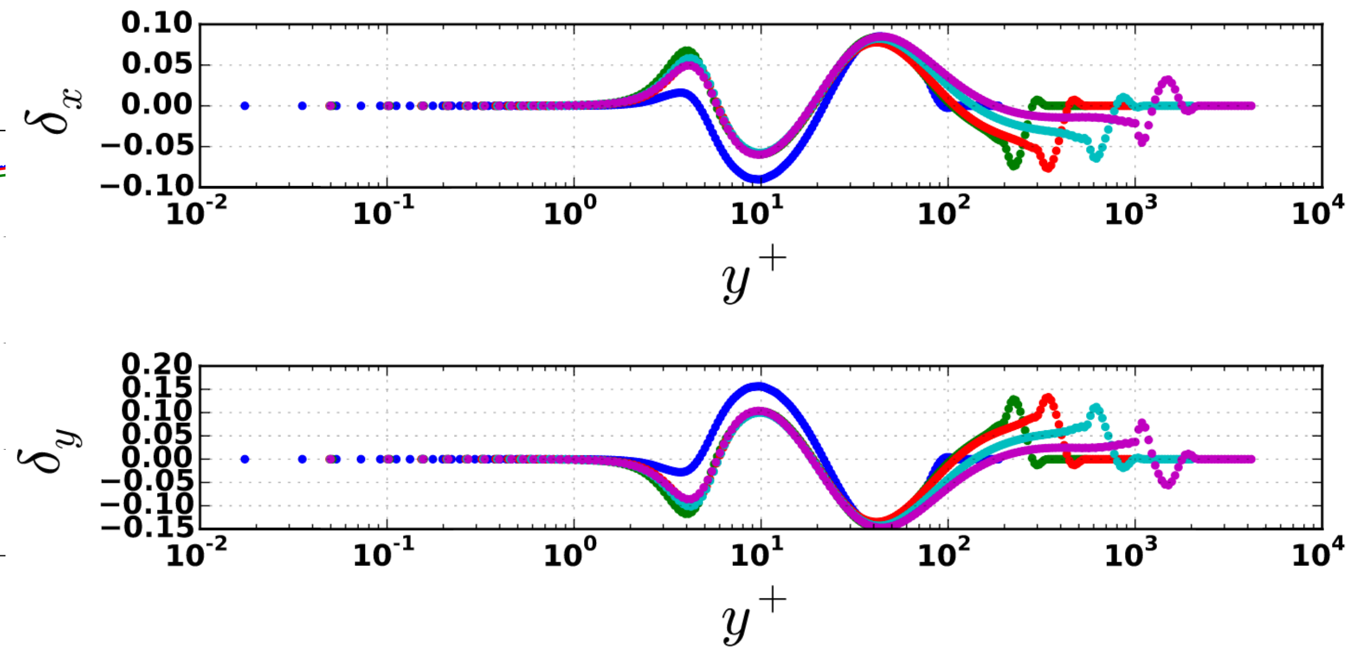
- How best to combine physics-based models and data?
- How best to embed physics constraints in machine learning outputs?
- Under what situations can we use completely data-driven models?
- How can we ensure generality of machine learning models?
- Can we make deep neural networks interpretable?
- What are the best techniques to solve high dimensional inverse problems?

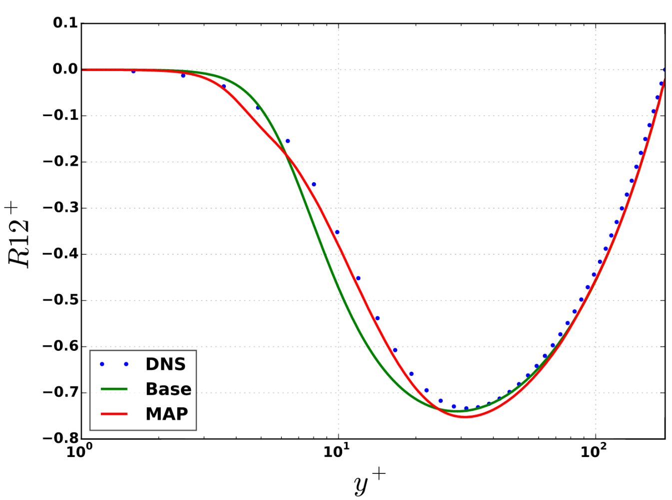
Introducing discrepancies in stress perturbations

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

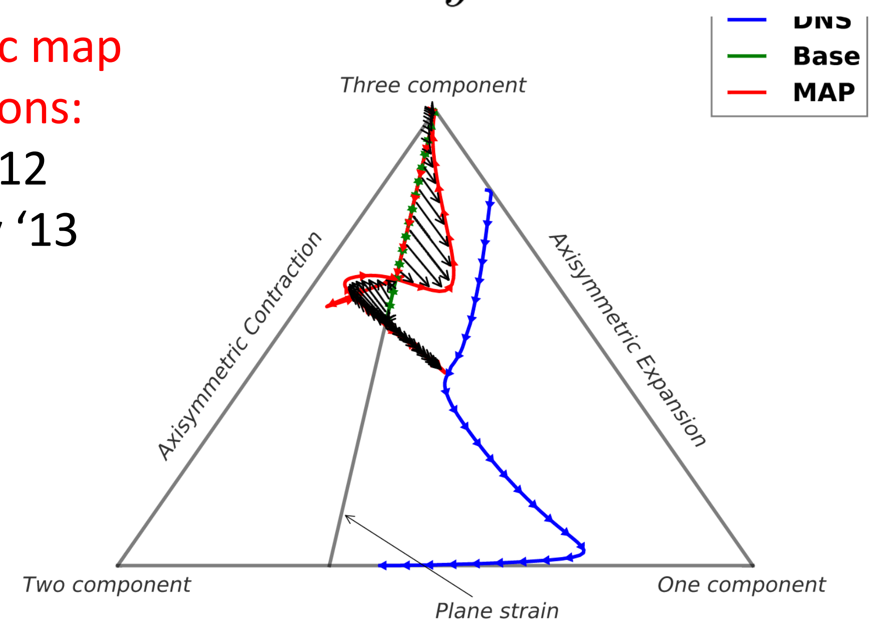
k-ω model



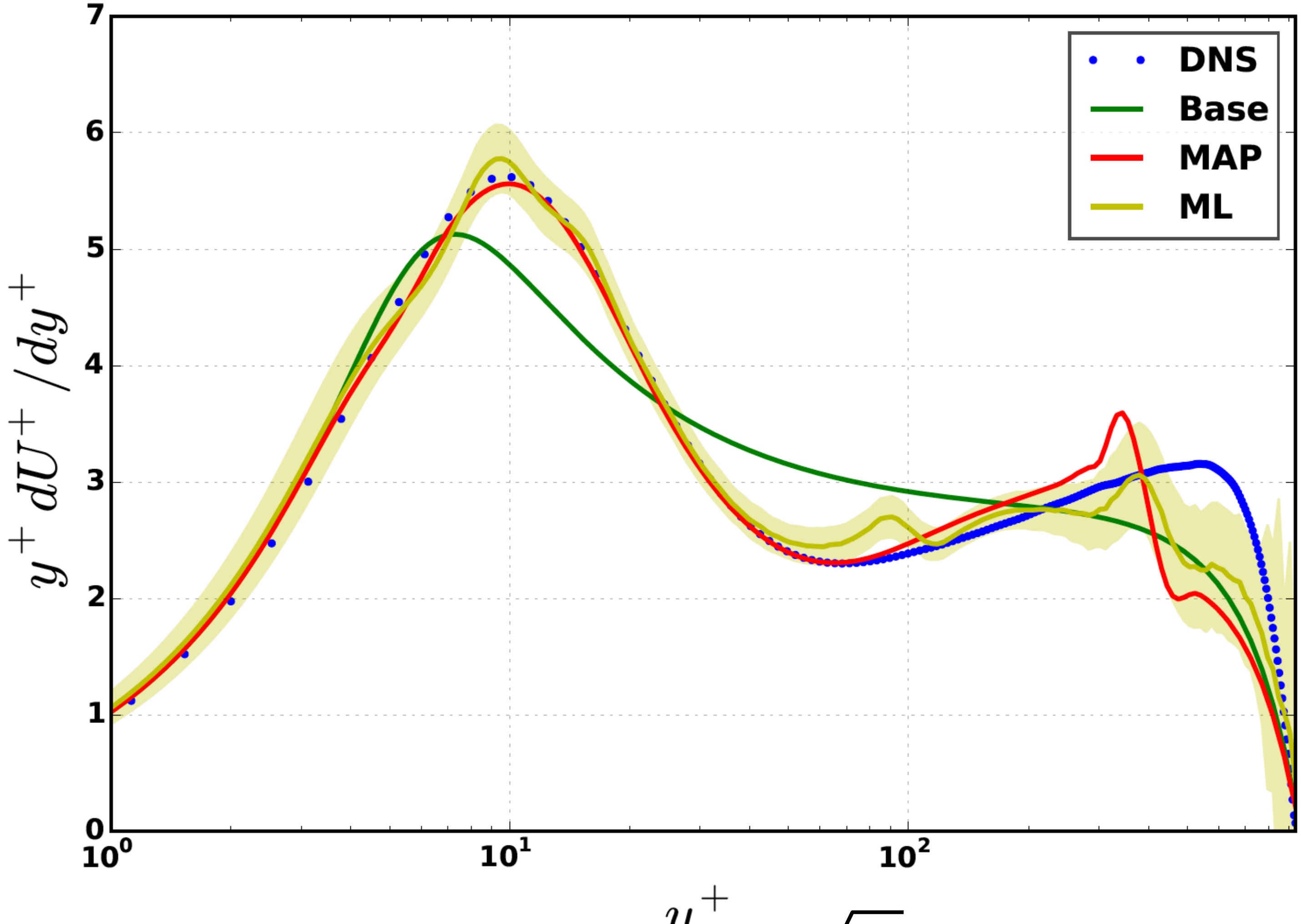




Barycentric map perturbations:
Iaccarino '12
Duraisamy '13
Xiao '15



Prediction with Machine-Learning Injection (Re_τ = 950)



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

Turbulence models

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

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

- 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