Field Inversion and Machine learning for Predictive Turbulence Modeling

Symposium, University of Michigan

Anand Pratap Singh[1] **Karthik Duraisamy**[1]

[1]Computational Aerosciences Lab
University of Michigan, Ann Arbor, MI

Outline

- ~~Motivation~~
- Full–field Inversion and Machine Learning (FIML) framework
- Application to augment $k - \omega$ model for APG flow over a bump
  - Compare augmentation using sparse surface data vs full field velocity data
  - Variability/uncertainty in the augmentation
  - Constrained augmentation
- Application to augment SA model for separated flows over airfoil
  - Portability

Objective: Present FIML with some examples and variations.

Challenges with data-driven approach

Data is not available in a form that is directly useful

Surface pressure, lift, drag, velocity ....

$$\vdots$$

Predictive capabilities

Spalart Allmaras, $k - \omega$,

$$\frac{D\tilde{\nu}_t}{Dt} = P(\tilde{\nu}_t, Q) - D(\tilde{\nu}_t, Q) + T(\tilde{\nu}_t, Q)$$

Even if relevant data is available from DNS, it will not be consistent with the RANS model.

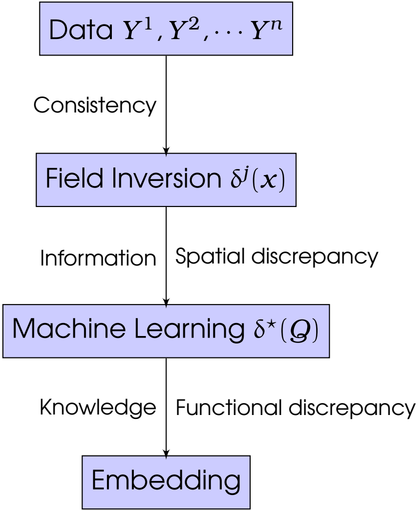
Need to transform data into useful information

via

Full–field Inversion

Field Inversion and Machine Learning (FIML)

for consistent data driven physics based augmentation



Prediction: Injection into solver

Duraisamy et al. (2015)

Q : flow variable, $\tilde{\nu}$: model working variable

$$\frac{D\tilde{\nu}}{Dt} = R(Q,\tilde{\nu}) + \delta^j(x) : \min_{\delta^j(x)} \|Y^j - Y^j(Q)\|$$

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

$$\delta^\star(f(Q,\tilde{\nu}))$$

Call to machine learning function

$$\frac{D\tilde{\nu}}{Dt} = R(Q,\tilde{\nu}) + \boxed{\delta^\star(f(Q,\tilde{\nu}))}$$

Solving the Inverse Problem

Problem specification for the $k - \omega$ model

$$\frac{D\omega}{Dt} = R(Q, k, \omega) + \delta(x) \tag{1}$$

Modify the $k - \omega$ model,

$$\frac{D\omega}{Dt} = P(Q, k, \omega) - D(Q, k, \omega) + T(Q, k, \omega) + \delta(x) \tag{2}$$

rewrite as,

$$\frac{D\omega}{Dt} = \beta(x)P(Q, k, \omega) - D(Q, k, \omega) + T(Q, k, \omega) \tag{3}$$

$$\delta(x) = (\beta(x) - 1)P(Q, k, \omega) \tag{4}$$

$\beta(x_k) \equiv \beta_k \equiv \beta$ at grid point k in the discretized setting.
Number of unknowns = number of cell volumes/nodes.

Next, we need to infer β_k's

Solving the Inverse Problem

Tikhonov regularization

$$\beta_{inverse} = arg\min_{\beta} \frac{1}{2}\left(\mathfrak{J}_1(\beta) + \lambda \mathfrak{J}_2(\beta)\right)$$
$$= arg\min_{\beta} \frac{1}{2}\left[(d - h(\beta))^T(d - h(\beta)) + \lambda(\beta - \beta_{prior})^T(\beta - \beta_{prior})\right]$$

d = data
$h(\beta)$ = prediction given β, RANS solve
$\beta_{prior} = 1$

Alternate approaches..

$$\frac{D\omega}{Dt} = P_\omega - \beta(x) D_\omega + T_\omega$$ Singh & Duraisamy, PoF 2016

$$\frac{DR_{ij}}{Dt} = C_{ij} + P_{ij} + T_{ij} + \Pi_{ij} + D_{ij} + \beta(x)_{ij} \epsilon_{ij}$$ Parish & Duraisamy, Aviation 2014

$$\frac{DR_{ij}}{Dt} = \beta(x)_{ij} a_0 \omega \left(R_{ij,eq} - R_{ij}\right)$$ Singh & Duraisamy, Scitech 2016

$$R_p = 2k\left[\frac{I}{3} + V\left(\Lambda + \beta(\vec{x})\right) V^T\right]$$ Duraisamy, SIAM 2016

Machine Learning

Algorithm & Selection of features

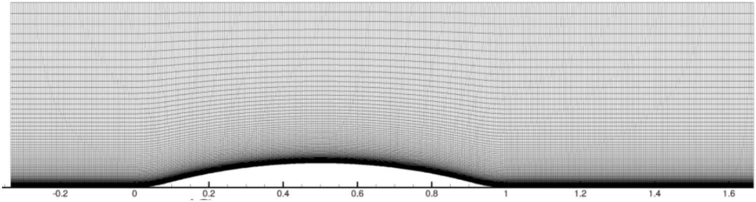
$\beta^1(x)$
$\beta^2(x)$
$\beta^3(x)$
$\cdots$
$\cdots$
$\beta^n(x)$

$\longrightarrow$ Machine Learning (Training) $\longrightarrow$ $\eta_1, \eta_2, \cdots$
$\beta^\star(\eta_1, \eta_2, \cdots)$

– Selection of non dimensionalized η_i's are critical
– Currently based on understanding of the Physics
– $\rho|S|d_{wall}^2/\mu$, $\mu_T|S|/\tau_{wall}$, and d_{wall} are used
– Adaboost algorithm is used to build the regression model

Flow over a bump

Setup

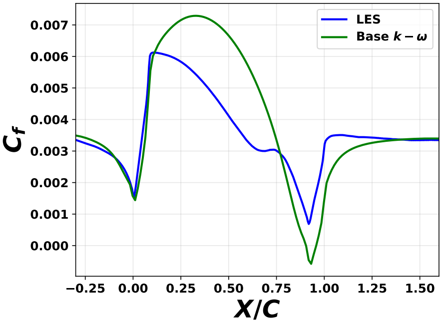


- Five different heights and varying inlet momentum thickness
- Flow is incompressible with $Re_\theta \approx 2500$
- LES performed at Iowa State (Matai & Durbin)

Flows involve varying degree of adverse pressure gradients, curvature, and separation in some cases.

Flow over a bump

Setup

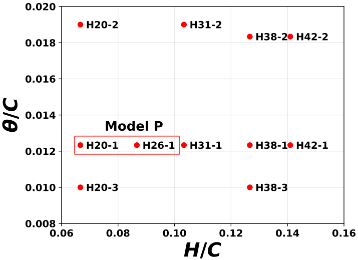


- Five different heights and varying inlet momentum thickness
- Flow is incompressible with $Re_\theta \approx 2500$
- LES performed at Iowa State (Matai & Durbin)

Flows involve varying degree of adverse pressure gradients, curvature, and separation in some cases.

Flow over a bump

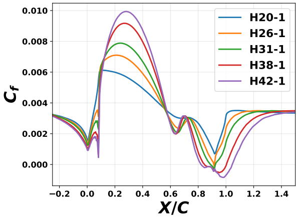
Matrix of cases



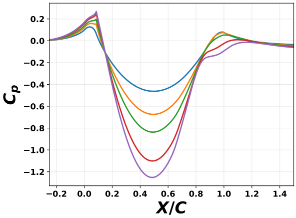
LES data for 11 cases are available.

Flow over a bump

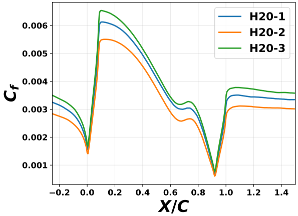
LES solution



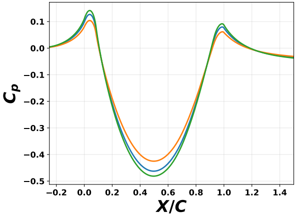
(a) Skin friction



(b) Pressure

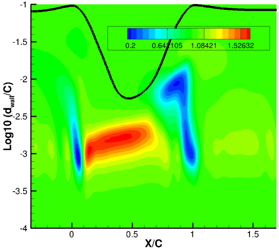


(a) Skin friction

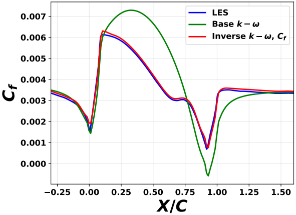


(b) Pressure

Inverse solution



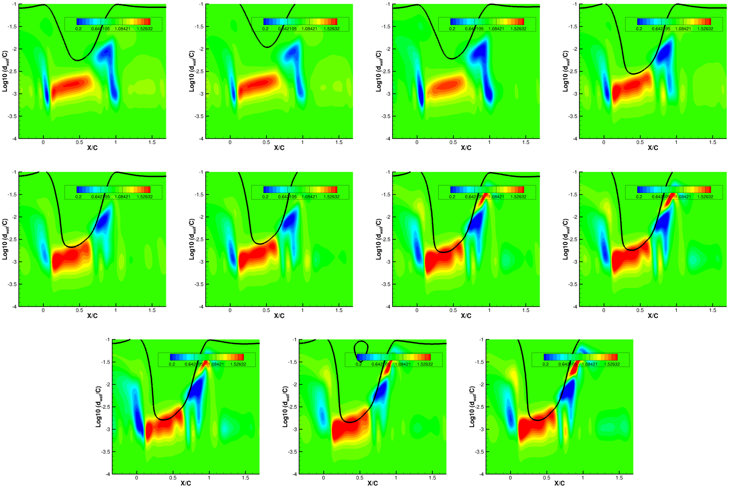
(a) $\beta(x)$



(b) Skin friction

- Skin friction data (surface quantity) is used
- Gradient based algorithm is used
- Gradients are calculated using adjoints

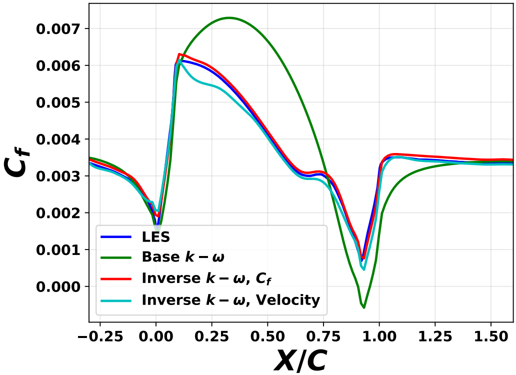
Inverse solution



Universality in the discrepancy?

Inverse solution

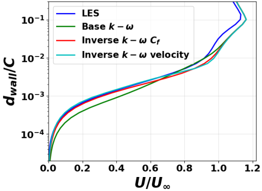
Data: C_f vs Velocity



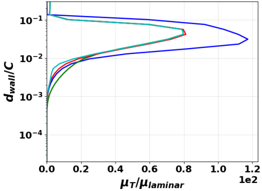
Inversion using data for skin friction or using velocity results in similar solution.

Inverse solution

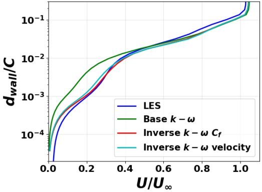
Data: C_f vs Velocity



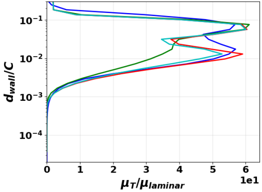
(a) location 1



(b) location 1



(c) location 2

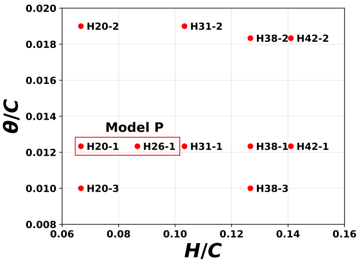


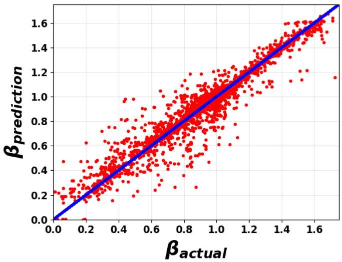
(d) location 2

Inversion using data for skin friction or using velocity results in similar solution.

Machine learning

Training, reconstructing β





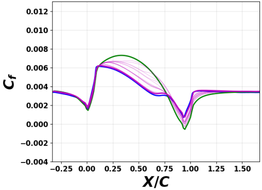
Model P is trained on the inverse solution for H20-1 and H26-1 and then used for prediction.

$$\text{Training} \equiv \beta(x) \to \beta(\eta(Q))$$

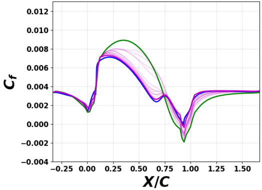
Prediction

Skin friction

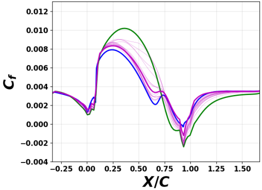
Legend: **LES**, **Base** $k-\omega$, **Adaboost** $k-\omega$



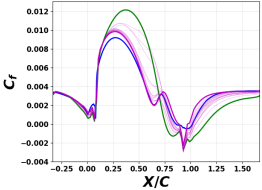
(a) H20-1



(b) H26-1



(c) H31-1



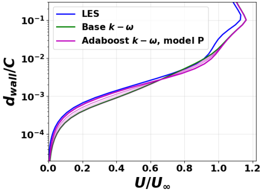
(d) H38-1

Model P improves the solution for all the bump heights and mometum thickness.

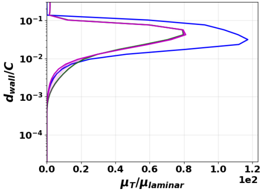
Prediction

Field quantites

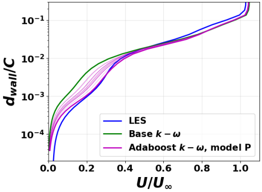
Legend: **LES**, **Base** $k-\omega$, **Adaboost** $k-\omega$



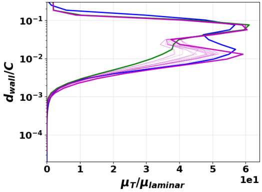
(a) location 1



(b) location 1



(c) location 2



(d) location 2

Improved prediction of field quantities using surface data!

A constrained approach to modifying the SA model via f_w

$$D = c_{w1} f_w \left[\frac{\tilde{\nu}}{d}\right]^2 .$$

The function f_w is defined as:

$$\begin{aligned} f_w &= g\left[\frac{1 + c_{w3}^6}{g^6 + c_{w3}^6}\right]^{\frac{1}{6}} \\ g(r) &= r + c_{w2}(r^6 - r) \\ r &= \frac{\tilde{\nu}}{\tilde{\Omega}\kappa^2 d^2} \end{aligned}$$

Modify f_w as,

$$g(r) = r + \beta(x) c_{w2}(r^6 - r).$$

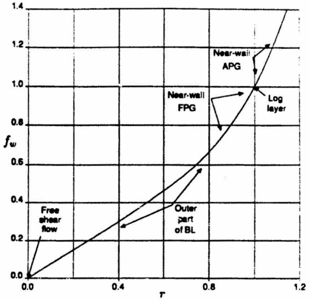


Figure 3: f_w function involved in the destruction term.

Figure: Spalart & Allmaras (1992)

Satisfies physical constraint on f_w; $f_w(r = 1) = 1$.

A constrained approach to modifying the SA model via f_w

Machine learning reconstruction

Legend: **Base SA**, **Inverse**, **ML reconstruction**

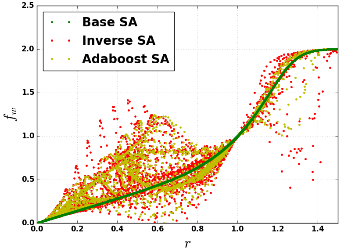


Figure: f_w vs r

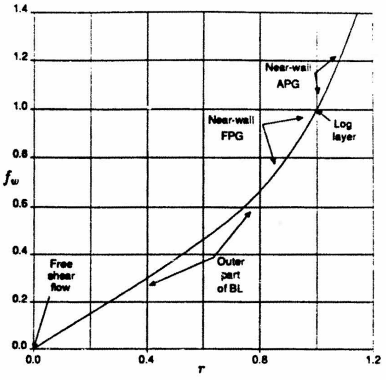
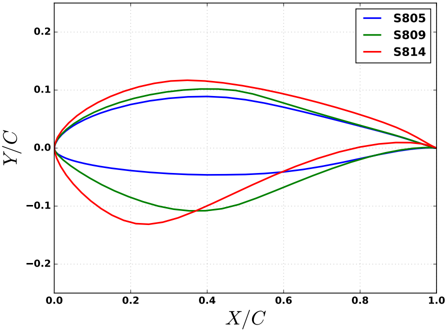


Figure 3: f_w function involved in the destruction term.

Figure: Spalart & Allmaras (1992)

Machine learning reconstruction captures the major features in f_w

FIML: Wind Turbine Airfoils



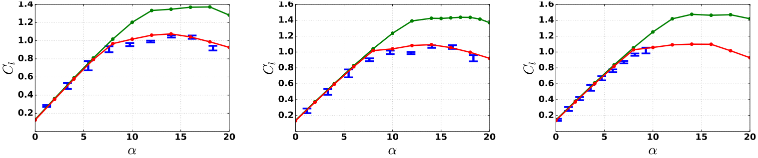
Training data: S814 at $Re = 1 \times 10^6, 2 \times 10^6$

1. Inversion for range of AoAs
2. Neural network to build new model
3. Freeze the model and test!

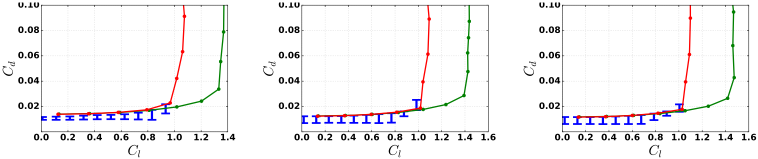
Collaboration with Altair Engineering Inc.

Prediction - S809

Legend: **Expt.**, **Base SA**, **NN SA**



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

Lift and drag improvement for geometry not used for training.

Prediction - S809

Legend: **Expt.**, **Base SA**, **NN SA**

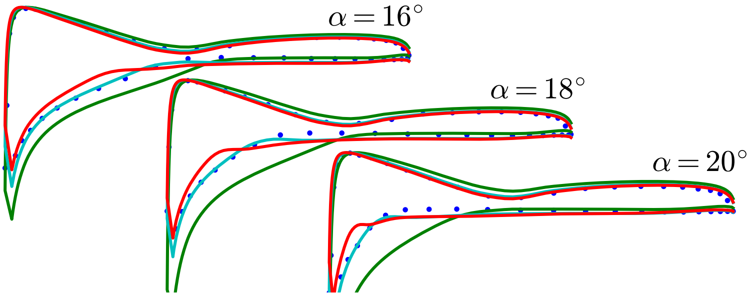
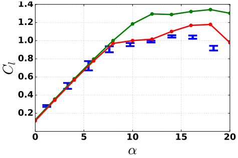


Figure: C_p for S809 at $Re = 2 \times 10^6$, $\alpha = \{16^\circ, 18^\circ, 20^\circ\}$.

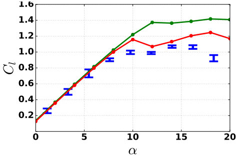
Inference using only C_l, NN-augmented model provides considerable predictive improvements of C_p.

Portability: Implementation in Altair's AcuSolve

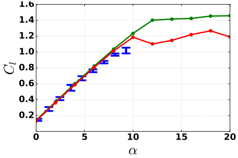
Legend: Expt., Base SA, NN SA



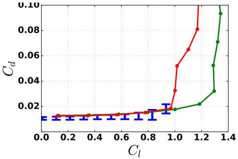
(a) $Re = 1 \times 10^6$



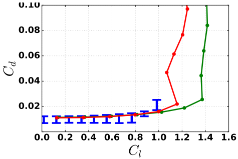
(b) $Re = 2 \times 10^6$



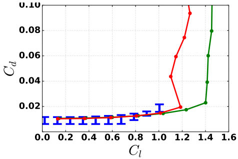
(c) $Re = 3 \times 10^6$



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



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



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

Model built using in-house code and embedded in a commercial solver

Summary

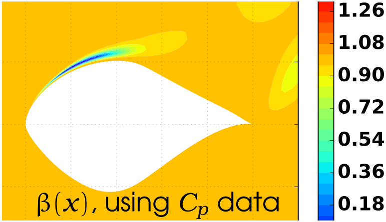
- Full field Inversion establishes consistency between data and model (Data $\rightarrow$ Information)
- Machine learning on a number of inverse solutions discovers functional form of model correction (Information $\rightarrow$ knowledge)
- Machine-learned corrections augments solver on-line.
- Machine learning-augmented model significantly improves the RANS prediction of friction, velocity, etc.
- The utility of the FIML framework has been demonstrated for a variety of problems.
- Portability of the FIML augmented model is demonstrated by embedding the model in Altair's AcuSolve.

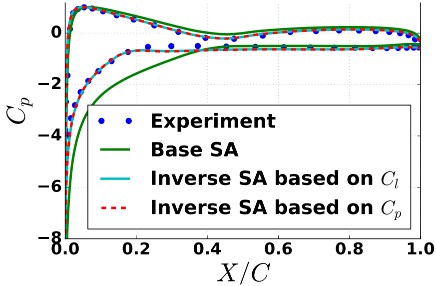
Acknowledgments

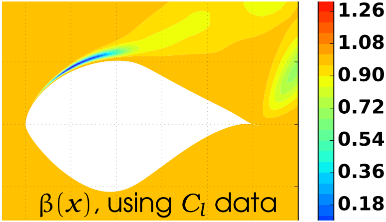
This work is supported by:

- DARPA under the EQUiPS project (Technical Monitor: Dr Fariba Fahroo).
- NASA LEARN Project No. NNX15AN98A (Technical Monitor: Gary Coleman).

Inversion based on Pressure vs Lift







- Ability to work on sparse amount of data is critical

- Experimental lift coefficient is used for all the following results.

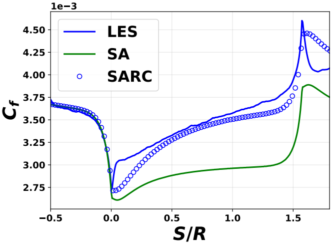
one data point!

$$\mathfrak{J}(\beta) = \min_{\beta} \left[\left[C_{l,exp} - C_l(\beta) \right]^2 + \lambda \sum^{N_{cells}} [\beta(x_n) - 1]^2 \right]$$

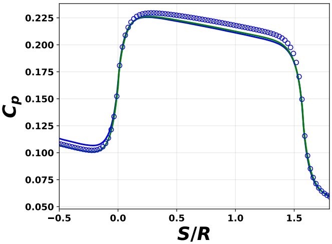
Curved Channel

- Eddy viscosity based models are insensitive to curvature.
- Several analytical curvature corrections to such models exists, SA–RC for SA (Spalart & Shur).
- Assume SA–RC to be truth and reconstruct it purely using data. Data generated by running a RANS simulation with SA–RC.

Curved Channel - Baseline solutions



(a) Skin friction



(b) Surface pressure

Figure: LES vs SA vs SA–RC

SA–RC is assumed to be the truth for this exercise. No uncertainty related to flow setup, mesh, numerics, boundary conditions etc.

Curved Channel - Inverse Problems

Problem specification for the SA model

$$\frac{D\tilde{\nu}}{Dt} = R(Q,\tilde{\nu}) + \delta(x) \tag{5}$$

Modify the SA model,

$$\frac{D\tilde{\nu}}{Dt} = P(\tilde{\nu},Q) - D(\tilde{\nu},Q) + T(\tilde{\nu},Q) + \delta(x) \tag{6}$$

rewrite as,

$$\frac{D\tilde{\nu}}{Dt} = \beta(x)P(\tilde{\nu},Q) - D(\tilde{\nu},Q) + T(\tilde{\nu},Q) \tag{7}$$

$$\delta(x) = (\beta(x) - 1)P(\tilde{\nu},Q) \tag{8}$$

SA–RC has an equivalent term, f_{r1}, which is multiplied to the Production term.

$$\frac{D\tilde{\nu}}{Dt} = f_{r1}(Q)P(\tilde{\nu},Q) - D(\tilde{\nu},Q) + T(\tilde{\nu},Q) \tag{9}$$

Curved Channel - Inverse Problems

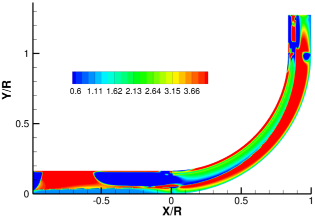
$$\mathfrak{J}_{I-U}(\beta) = \sum_{i=1}^{N_{cells}} [U_i - U_{i,SA-RC}]^2 + \lambda \sum_{j=1}^{N_{cells}} [\beta_j - 1]^2 \quad (10)$$

$$\mathfrak{J}_{I-C_f}(\beta) = \sum_{i=1}^{N_{wall}} \left[C_{f,i} - C_{f,i,SA-RC}\right]^2 + \lambda \sum_{j=1}^{N_{cells}} [\beta_j - 1]^2 \quad (11)$$

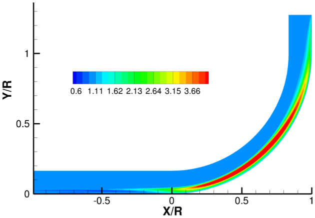
$$\mathfrak{J}_{I-C_p}(\beta) = \sum_{i=1}^{N_{wall}} \left[C_{p,i} - C_{p,i,SA-RC}\right]^2 + \lambda \sum_{j=1}^{N_{cells}} [\beta_j - 1]^2 \quad (12)$$

Three different inverse problems are solved using different type and quantity of data.

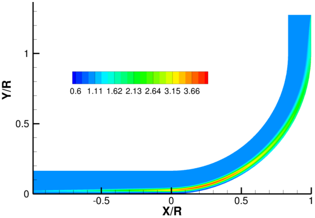
Curved Channel - Inferred Solutions



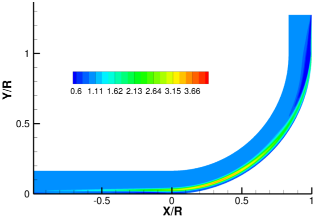
(a) f_{r1} from SA–RC



(b) $\beta(x)$ from $I-U$

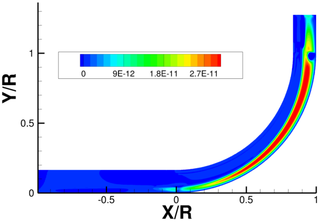


(c) $\beta(x)$ from $I-C_f$

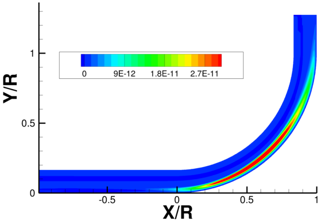


(d) $\beta(x)$ from $I-C_p$

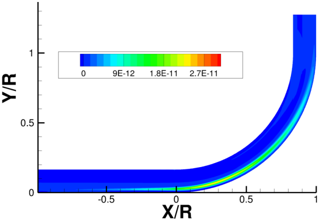
Inferred solution



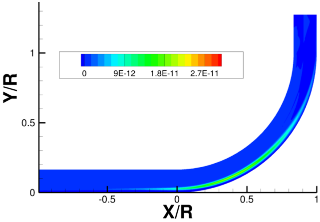
(a) $(f_{r1} - 1) \times Prod$ from SA–RC



(b) $\delta(x)$ from $I - U$



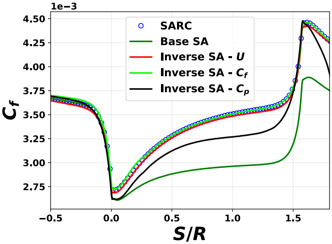
(c) $\delta(x)$ from $I - C_f$



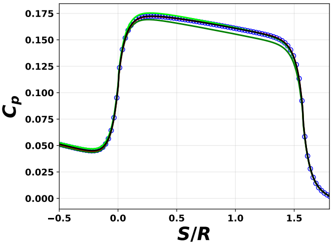
(d) $\delta(x)$ from $I - C_p$

Inferred corrections are quantitatively similar, velocity based inference being closest to the $SA - RC$.

Curved Channel - Inverse Solutions



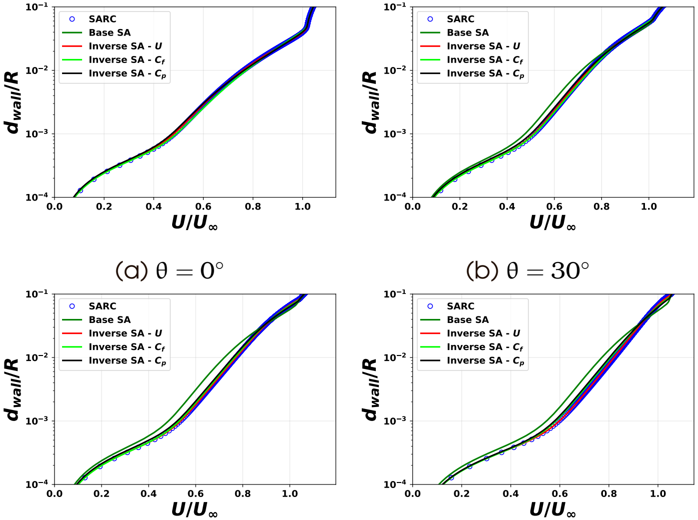
(a) Skin friction



(b) Surface pressure

Both velocity and skin friction based inference predicts better friction and pressure.

Curved Channel - Inverse Solutions



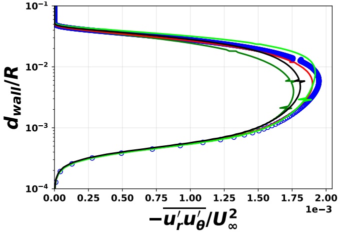
(a) $\theta = 0^\circ$

(b) $\theta = 30^\circ$

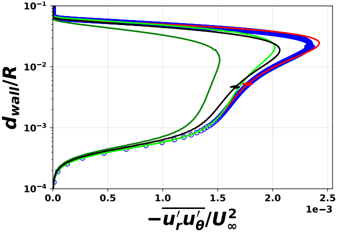
(c) $\theta = 60^\circ$

(d) $\theta = 90^\circ$

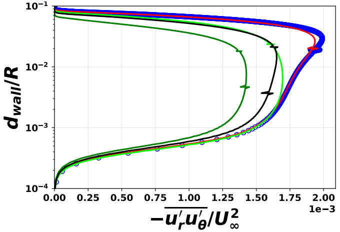
Curved Channel - Inverse Solutions



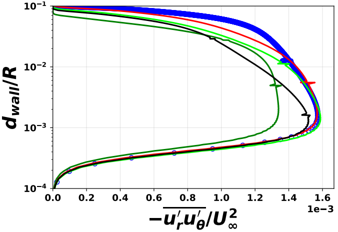
(a) $\theta = 0^\circ$



(b) $\theta = 30^\circ$



(c) $\theta = 60^\circ$



(d) $\theta = 90^\circ$

Pressure based inversion does not improve Reynolds stress in the log layer, possible resulting in the discrepancy in skin friction