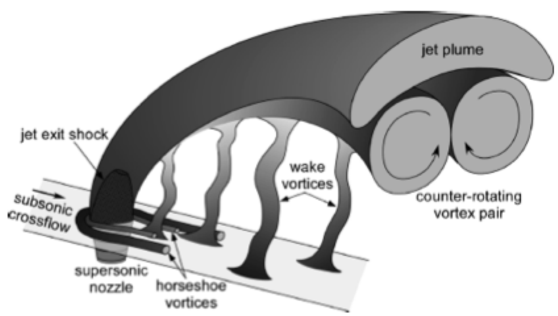
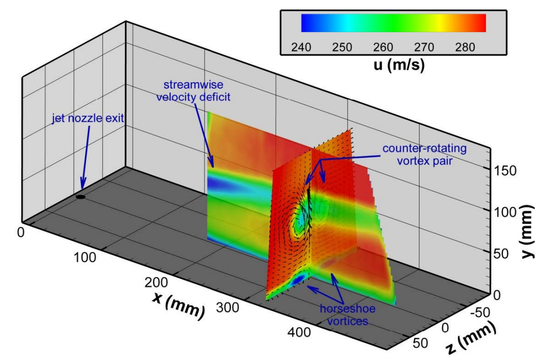
Exceptional service in the national interest

Sandia National Laboratories





Estimating and verifying k-ε model coefficients for jet-in-crossflow simulations

J. Ray, S. Lefantzi, L. Dechant and S. Arunajatesan

Contact: *jairay@sandia.gov*

U.S. DEPARTMENT OF ENERGY

NNSA National Nuclear Security Administration

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. SAND2014-2429C

Introduction

- Aim: Develop a predictive RANS model for transonic jet-in-crossflow (JinC) simulations
- Drawback: RANS simulations are simply not predictive
  - They have "model-form" error i.e., missing physics
  - The numerical constants/parameters in the k-ε model are usually derived from canonical flows
- Hypothesis
  - One can calibrate RANS to jet-in-crossflow experiments; thereafter the residual error is mostly model-form error
  - Due to model-form error and limited experimental measurements, the parameter estimates will be approximate
    - We will estimate parameters as probability density functions (PDF)
  - We hypothesize that most of the error in JinC simulations is parametric, not model-form

The problem

- The model
  - Devising a method to calibrate 3 k-ε parameters **C** = {C_μ, C_2, C_1} from expt. data

$$\frac{\partial \rho k}{\partial t} + \frac{\partial}{\partial x_i}\left[\rho u_i k - \left(\mu + \frac{\mu_T}{\sigma_k}\right)\frac{\partial k}{\partial x_i}\right] = P_k - \rho\varepsilon + S_k$$

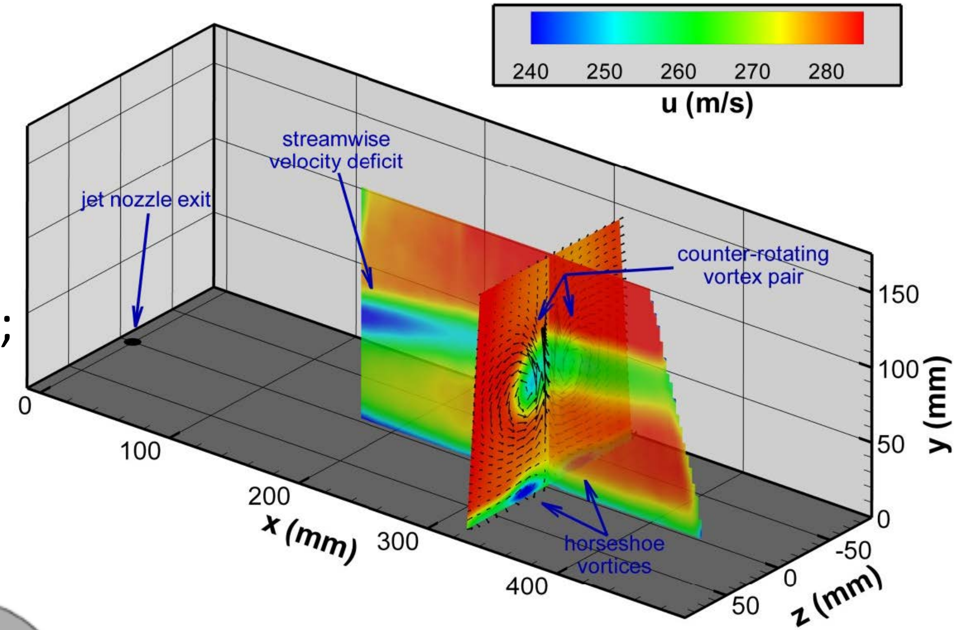
$$\frac{\partial \rho \varepsilon}{\partial t} + \frac{\partial}{\partial x_i}\left[\rho u_i \varepsilon - \left(\mu + \frac{\mu_T}{\sigma_\varepsilon}\right)\frac{\partial \varepsilon}{\partial x_i}\right] = \frac{\varepsilon}{k}\left(C_1 f_1 P_k - C_2 f_2 \rho\varepsilon\right) + S_\varepsilon$$

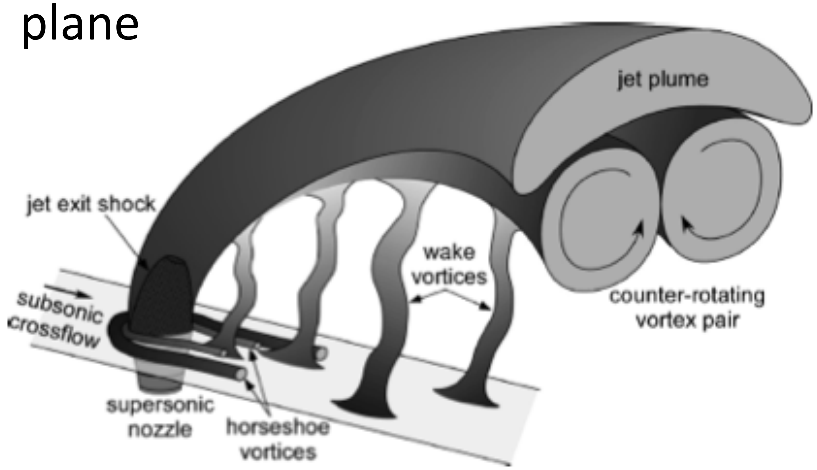
$$\mu_T = C_\mu f_\mu \rho \frac{k^2}{\varepsilon}$$

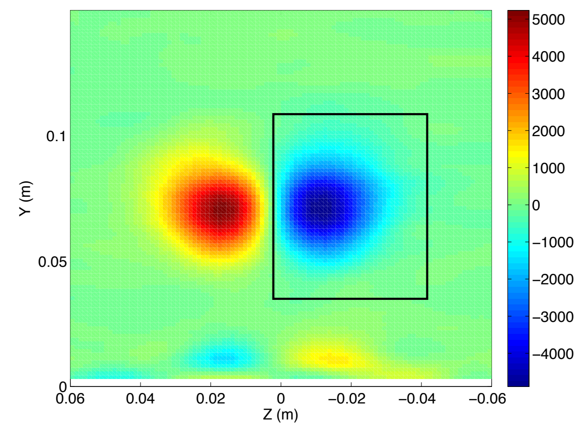
- Calibration parameters
  - **C** = {C_μ, C_1, C_2} ; C_μ: affects turbulent viscosity; C_1 & C_2: affects dissipation of TKE
- Calibration questions
  - Calibrate to 3 different datasets
    - Are the values of {C_μ, C_1, C_2} similar? Are some parameters more sensitive to others? Which ones?

Target problem - jet-in-crossflow

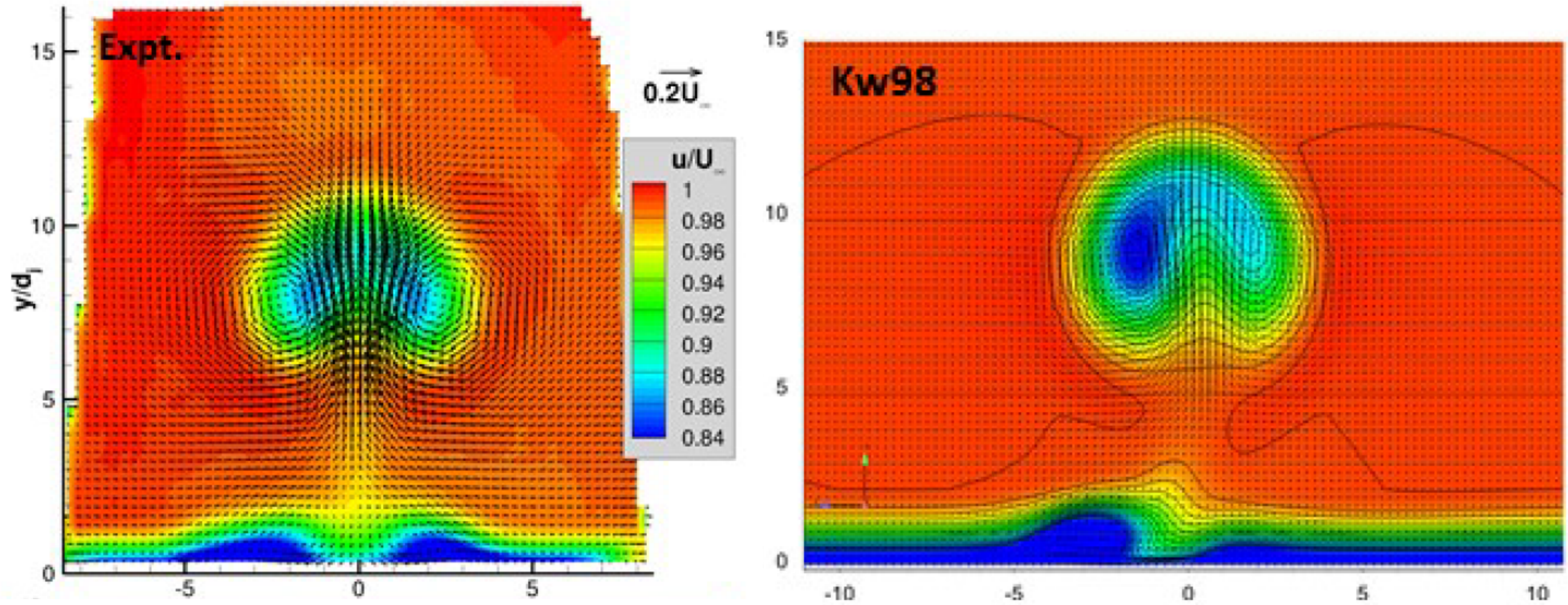
- A canonical problem for spin-rocket maneuvering, fuel-air mixing etc.
- We have experimental data (PIV measurements) on the mid-plane; also crossplane for M = 0.8
- Will calibrate to velocity on the midplane and test against cross-plane







RANS (k-ω) simulations - crossplane results



- Crossplane results for streamwise velocity deficit
- Computational results (SST) are too round; Kw98 doesn't have the mushroom shape; non-symmetric!
- Less intense regions; boundary layer too weak

Details of the study

- Aims of the calibration
  - Calibrate to velocity measurements on the midplane
    - Check against measurements on crossplane (when possible)
  - Calibrate to a M = {0.6, 0.7, 0.8}, J = 10.2 interactions
    - Check if they yield similar estimates
  - Perform calibration by posing & solving a Bayesian inverse problem
    - Estimate **C** as a joint PDF, using a Markov chain Monte Carlo (MCMC) method
- Technical challenges
  - MCMC requires $O(10^4)$ calls to the RANS simulator
    - 3D JinC RANS simulation expensive; replace with an emulator **v** = f(; **C**)
  - Arbitrary combinations of ($C\mu$, C_2, C_1) may be nonphysical
    - How to build surrogates when ($C\mu$, C_2, C_1) are nonsensical?
  - What functional form to use for f(:; **C**)?

The Bayesian calibration problem

- Model experimental values at probe j as $u^{(j)}_{ex} = u^{(j)}(\mathbf{C}) + \varepsilon^{(j)}$, $\varepsilon^{(j)} \sim N(0, \sigma^2)$

$$\Lambda(\mathbf{v_{ex}}|\boldsymbol{C}) \propto \sum_{j\epsilon\mathcal{P}} \exp\left(-\frac{\left(u^{(j)} - u^{(j)}(\mathbf{C})\right)^2}{2\sigma^2}\right)$$

- Given prior beliefs π on **C**, the posterior density (‘the PDF’) is

$$P(C|\mathbf{v}_{ex}) \propto \Lambda(\mathbf{v}_{ex}|\mathbf{C})\pi_\mu(C_\mu)\pi_2(C_2)\pi_1(C_1)\pi_\sigma(\sigma)$$

- $P(\mathbf{C}|u_{ex})$ is a complicated distribution that has to be described by drawing samples from it
- This is done by MCMC
  - MCMC describes a random walk in the parameter space to identify good parameter combination
  - Each step of the walk requires a model run to check the new parameter

Making emulators - 1

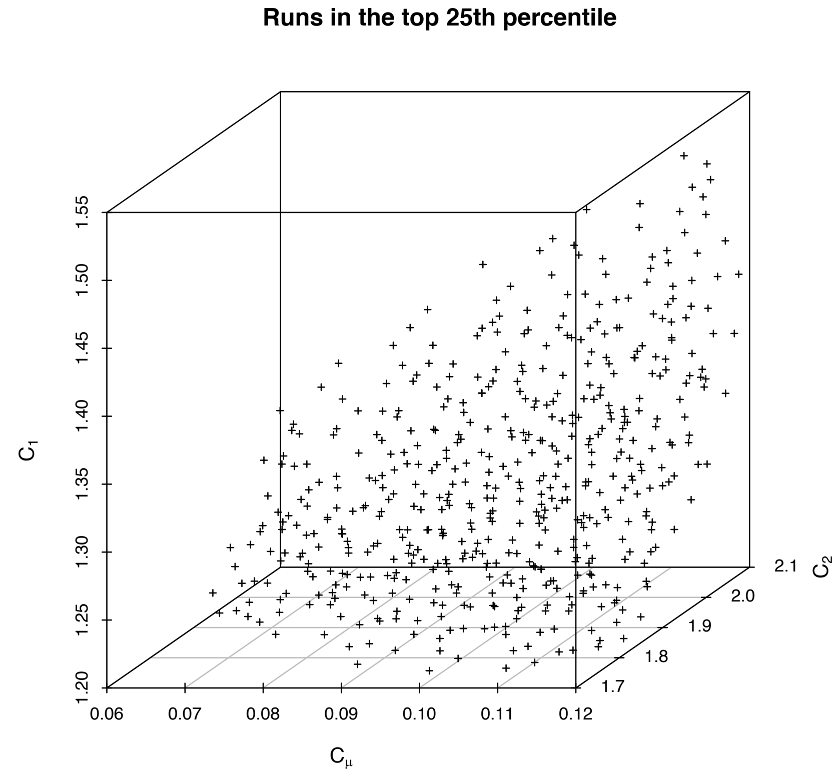
- Training data
  - Parameter space $\mathcal{C}$: $0.06 < C\mu < 0.12$; $1.7 < C_2 < 2.1$; $1.2 < C_1 < 1.7$
  - $\mathbf{C}_{nom} = \{0.09, 1.93, 1.43\}$
  - Take 2744 samples in $\mathcal{C}$ using a space-filling quasi Monte Carlo pattern
    - Save the streamwise vorticity field $\omega_x(\mathbf{y}; \mathbf{C})$
- Choosing the "probes"
  - Will try to create emulators for each grid cell on the crossplane
  - Most grid cells have lots of numerical noise
  - For a given run, choose the grid cells with vorticity the top 25% percentile (56 grid cells)
  - Take the union of such grid cells, union over the 2744 members of the training set (comes to 108 grid cells)
    - We will try to make emulators for these 108 grid cells with large vorticity

Making emulators - 2

- Model **v** in grid cell *j* as a function of **C** i.e. $\mathbf{v}^{(j)} = f^{(j)}(\mathbf{C})$
  - Approximate this dependence with a polynomial
  - $\mathbf{v}^{(j)} = a_0 + a_1 C_\mu + a_2 C_2 + a_3 C_1 + a_4 C_\mu C_2 + a_5 C_\mu C_2 + a_6 C_2 C_1 + \cdots$
- But how to get $(a_0, a_1, \ldots.)$ for each of the probe locations to complete the surrogate model for each probe?
  - Divide training data in a Learning Set and Testing Set
  - Fit a full cubic model for to the Learning Set via least-squares regression; sparsify using AIC
  - Estimate prediction RMSE for Learning & Testing sets; should be equal
- Final model tested using 100 rounds of cross-validation
- 10% error threshold was used to select models for the probes

Making emulators - 3

- Choosing $\mathcal{R}$
  - emulators failed – we could not model any surrogates to within 10% accuracy
  - This is because many **C** = {$C\mu$, C_2, C_1} combination are nonphysical
  - We compute the RMSE vorticity difference between the training set RANS runs and experimental observations
    - We retain only the top 25 percentile of the runs (using RMSE) as training data ($\mathcal{R}$)

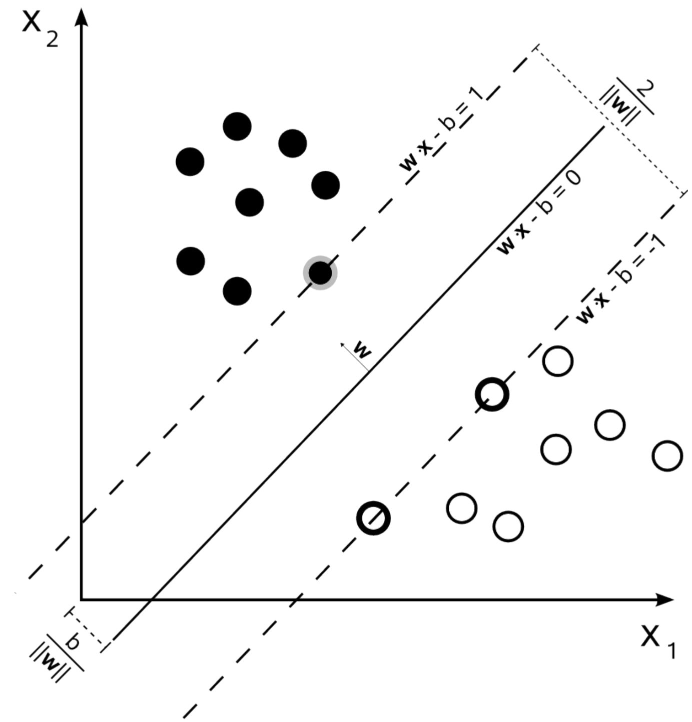


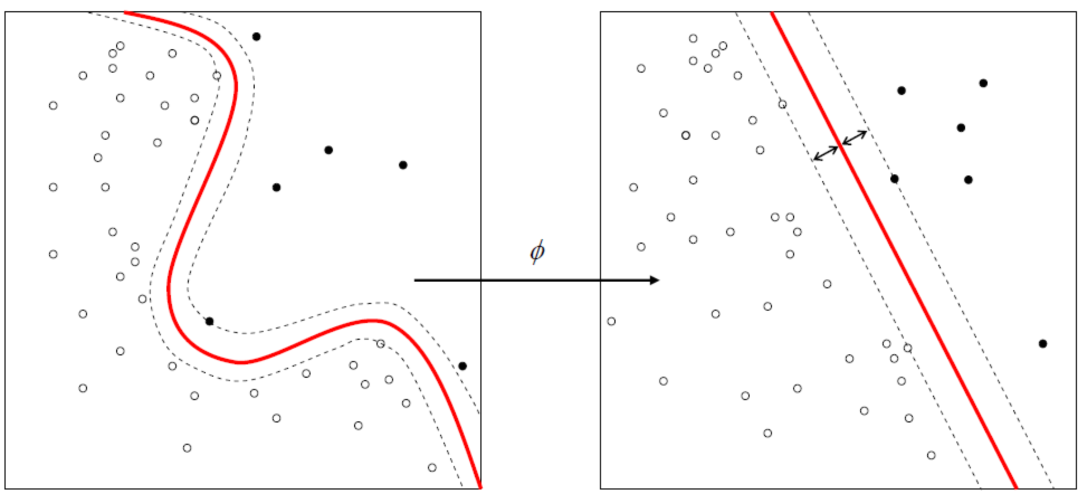
Making the informative prior

- Our emulators are valid only inside $\mathcal{R}$ in the parameter space C
- During the optimization (MCMC) we have to reject parameter combinations outside $\mathcal{R}$ (this is our prior belief $\pi_{prior}(\mathbf{C})$)
  - We define $\zeta(\mathbf{C}) = 1$, for **C** in R and $\zeta(\mathbf{C}) = -1$ for **C** outside $\mathcal{R}$
  - Then the level set $\zeta(\mathbf{C}) = 0$ is the boundary of $\mathcal{R}$
- The training set of RANS runs is used to populate $\zeta(\mathbf{C})$
- We have to "learn" the discriminating function $\zeta(\mathbf{C}) = 0$
  - We'll do that using support vector machine (SVM) classifiers

What is a SVM classifier?

- Given a binary function y = f(**x**) as a set of points (y_i, $\mathbf{x}_i$), y_i = (0, 1)
  - Find the hyperplane y + Ax = 0 that separates the x-space into y = 0 and y = 1 parts
- Posed as an optimization problem that maximizes the margin

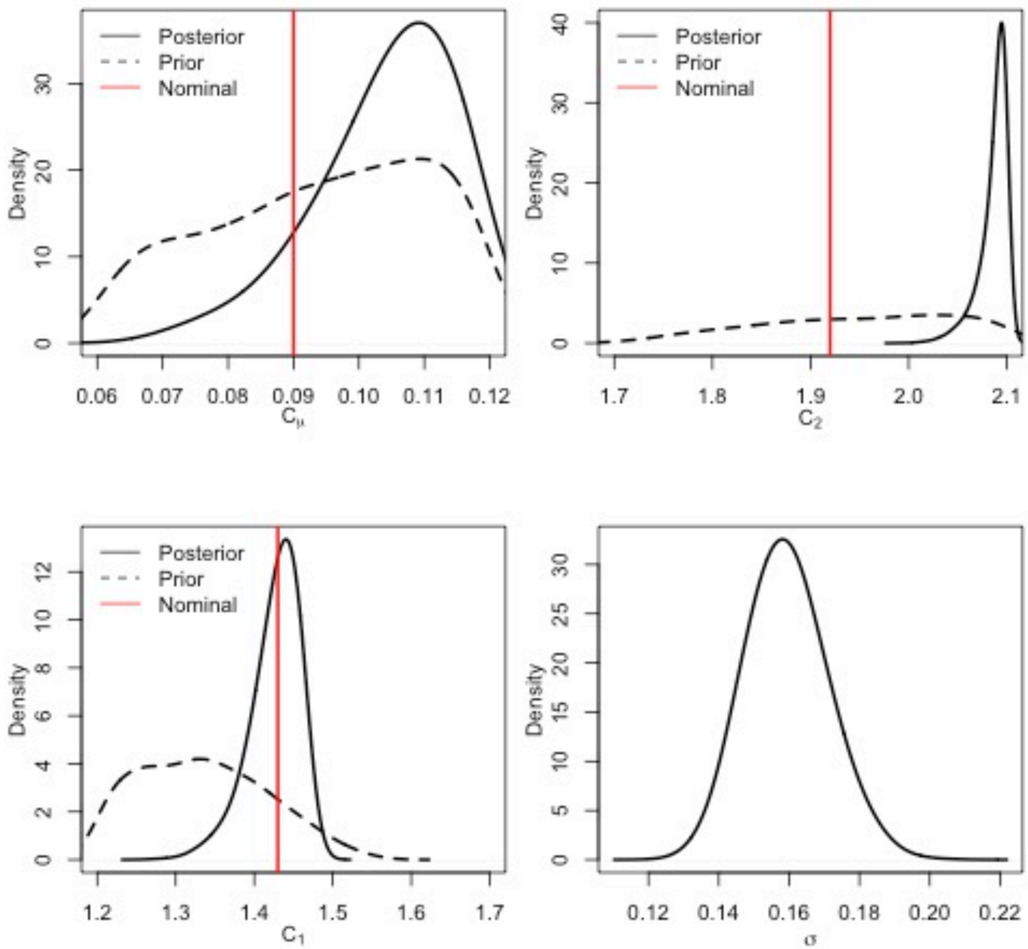




- In case of a curved discriminator, need a transformation first
  - Achieved using kernels
  - We use a cubic kernel

Solution of the inverse problem

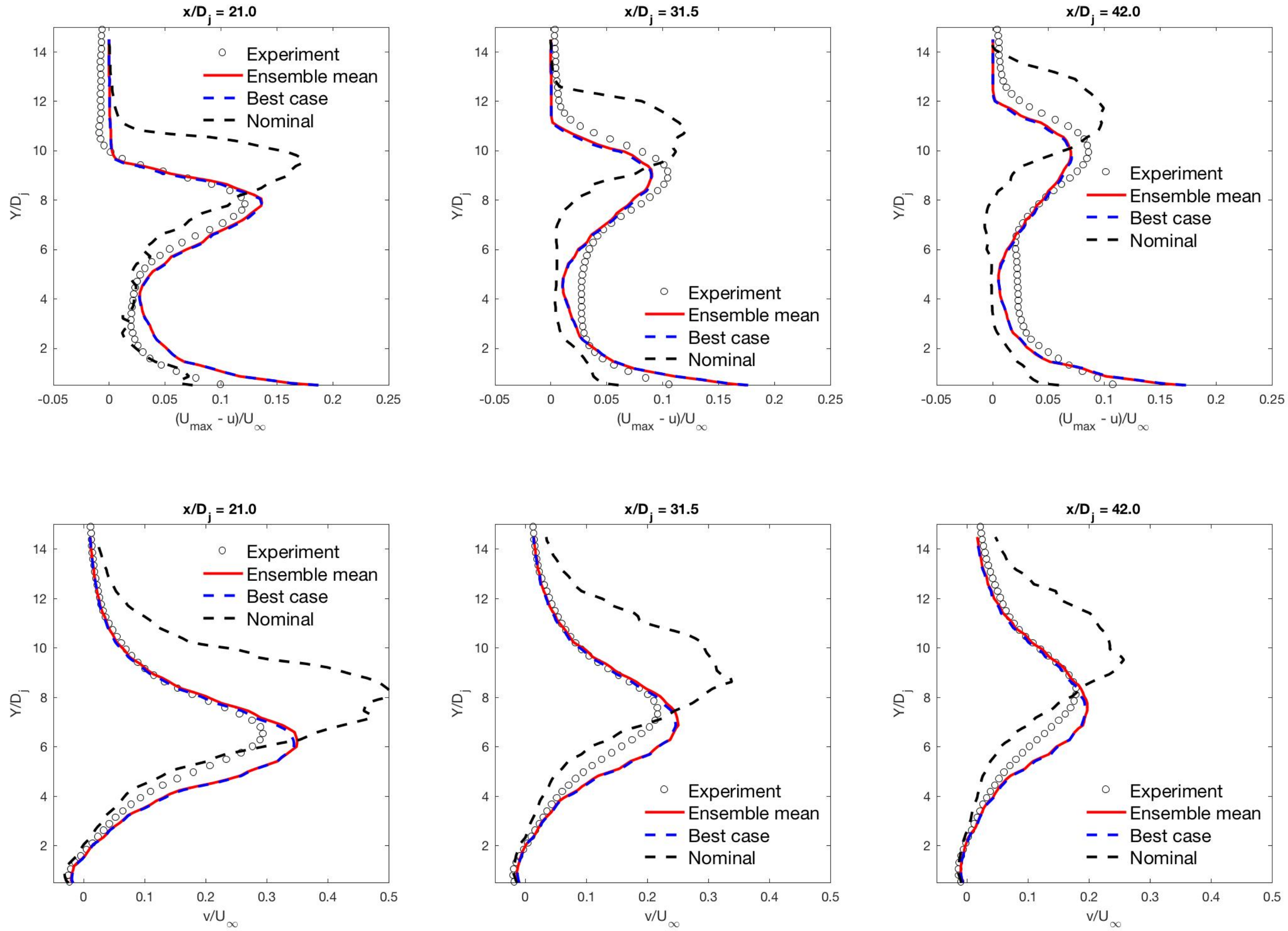
- We solve the calibration problem with MCMC (DRAM)
  - The treed classifier imposes the prior $\pi_{prior}(\mathbf{C})$
  - About 25,000 MCMC steps need to reach converged 4-dimemsional ($C\mu$, C_2, C_1, σ^2) PDFs
- We test the 4-D PDF by:
  - Taking 100 ($C\mu$, C_2, C_1) samples from the PDF
  - Running the RANS simulator
  - Checking the flowfield
- This manner of prediction is called a 'pushed forward posterior'



Results for the M = 0.8, J = 10.2 case

Check # 1 – mid-plane comparisons

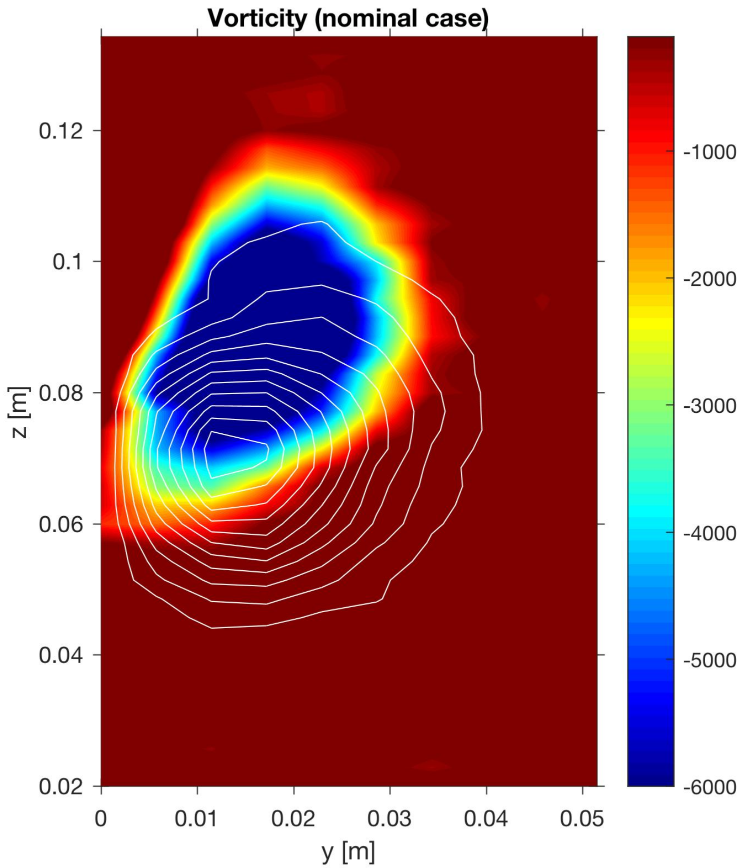
Sandia National Laboratories



Results for the M = 0.8, J = 10.2 case

- Good match; but this was what we calibrated to

Check # 2 – the vorticity field

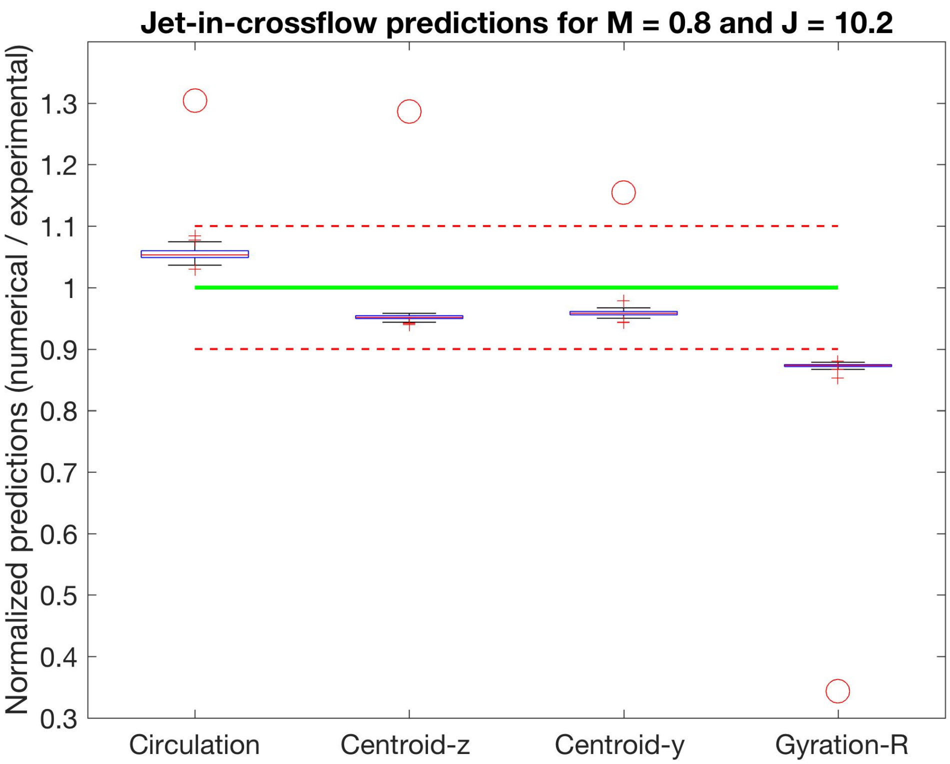


Results for the M = 0.8, J = 10.2 case

- The improvement is significant. And this was NOT the calibration variable

Check # 3 – point vortex summary

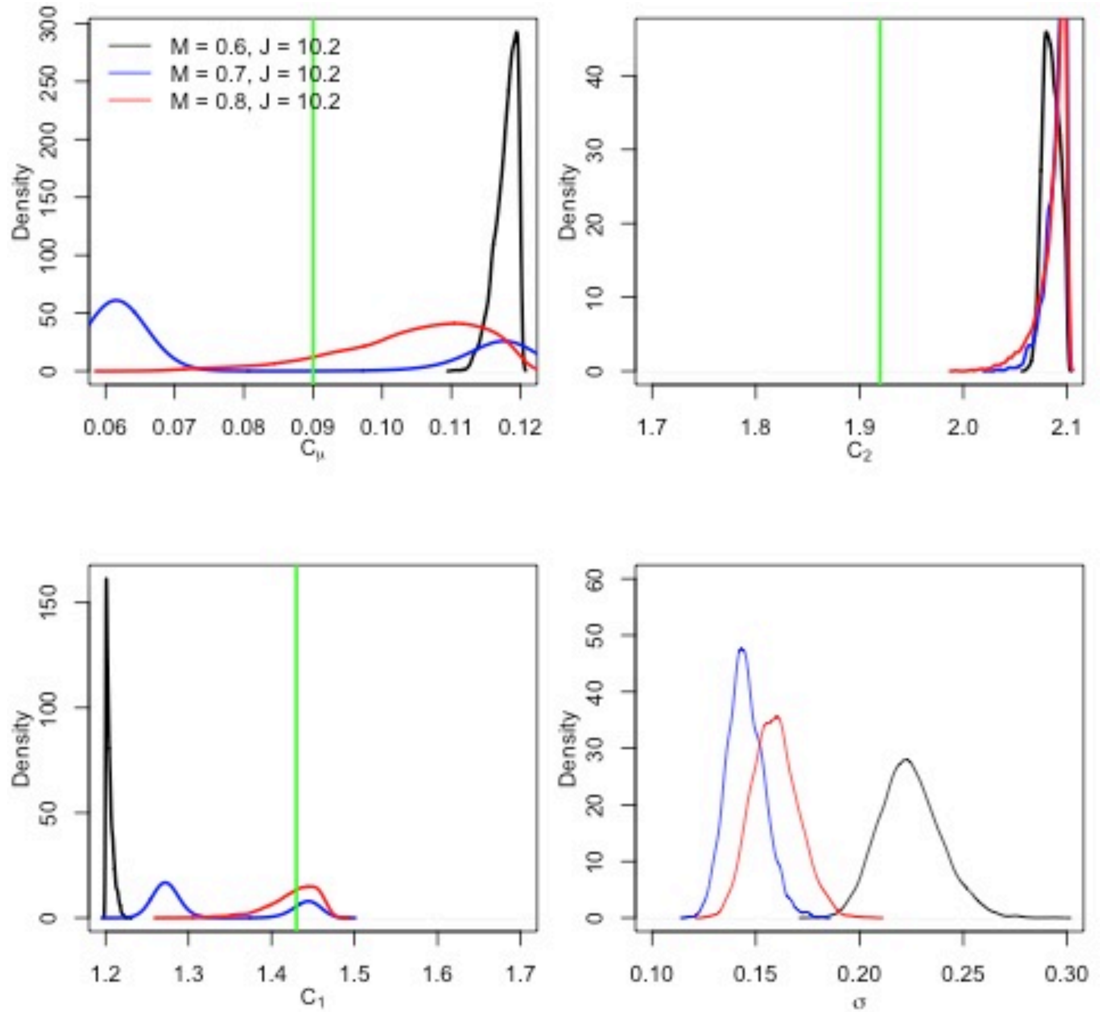
- Use the crossplane vorticity fields from the 100 RANS runs ('pushed forward posterior') to compute
  - Total circulation
  - Centroid of vorticity field
  - Radius of gyration of vorticity field
  - Normalize each by their experimental counterpart
- We expect to get an ensemble of values for each metric around 1



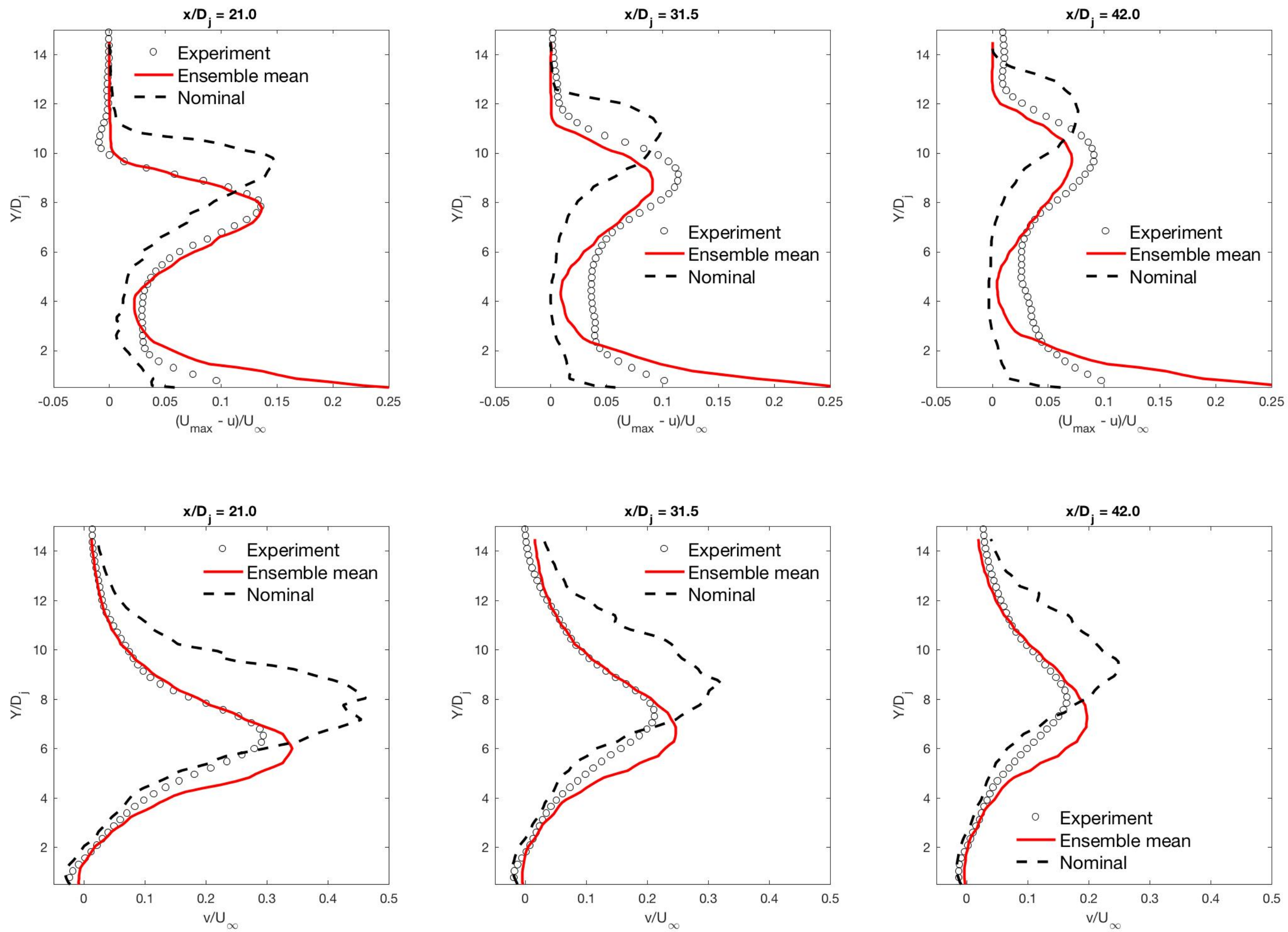
Results for the M = 0.8, J = 10.2 case

Comparison of all runs

- Repeated the calibration for M = 0.6 and M = 0.7
- Takeaways
  - C_2 seems to be consistently high
  - $C\mu$ seems to be clustered around the higher end
  - C1 seems to be either near nominal or at the lower bound
    - M = 0.6 run has the work model-data mismatch



Velocities for M = 0.7

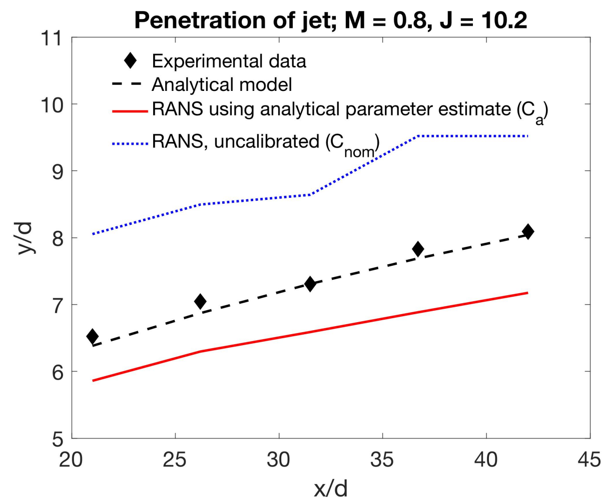


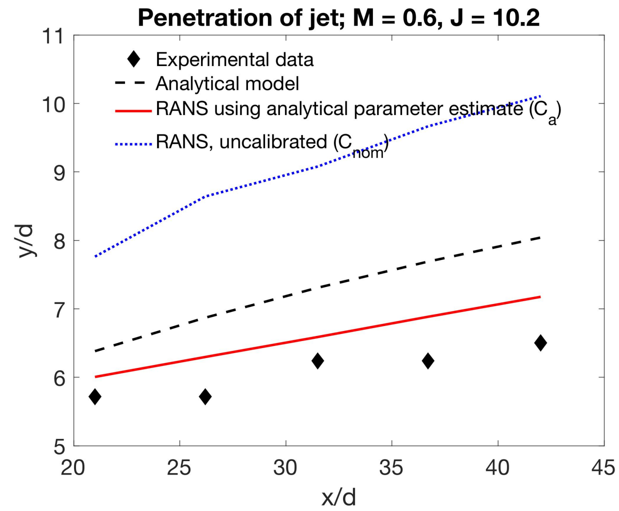
Results for the M = 0.7, J = 10.2 case

- Improvement in fit over the nominal case

An analytical model

- Jet modeled as evolution of a counter-rotating vortex pair
  - The velocity deficit in the core modeled as a wake
  - RANS simplified and cast in a self-similar form
    - $U_s \sim x^{n-1}$, $y_{jet} \sim x^n$
- 2 solutions: near-field (n = ½) and far-field (n=1/3)
  - Near-field solution leads to estimates of ($C\mu$, C_2, C_1) = (0.1, 2.1, 1.34)
  - Far-field solution provides the trajectory of the jet
- Details: DeChant et al, "Κ-ε Turbulence Model Parameter Estimates Using an Approximate Self-Similar Jet-in-Crossflow Solution", submitted AIAA Journal.

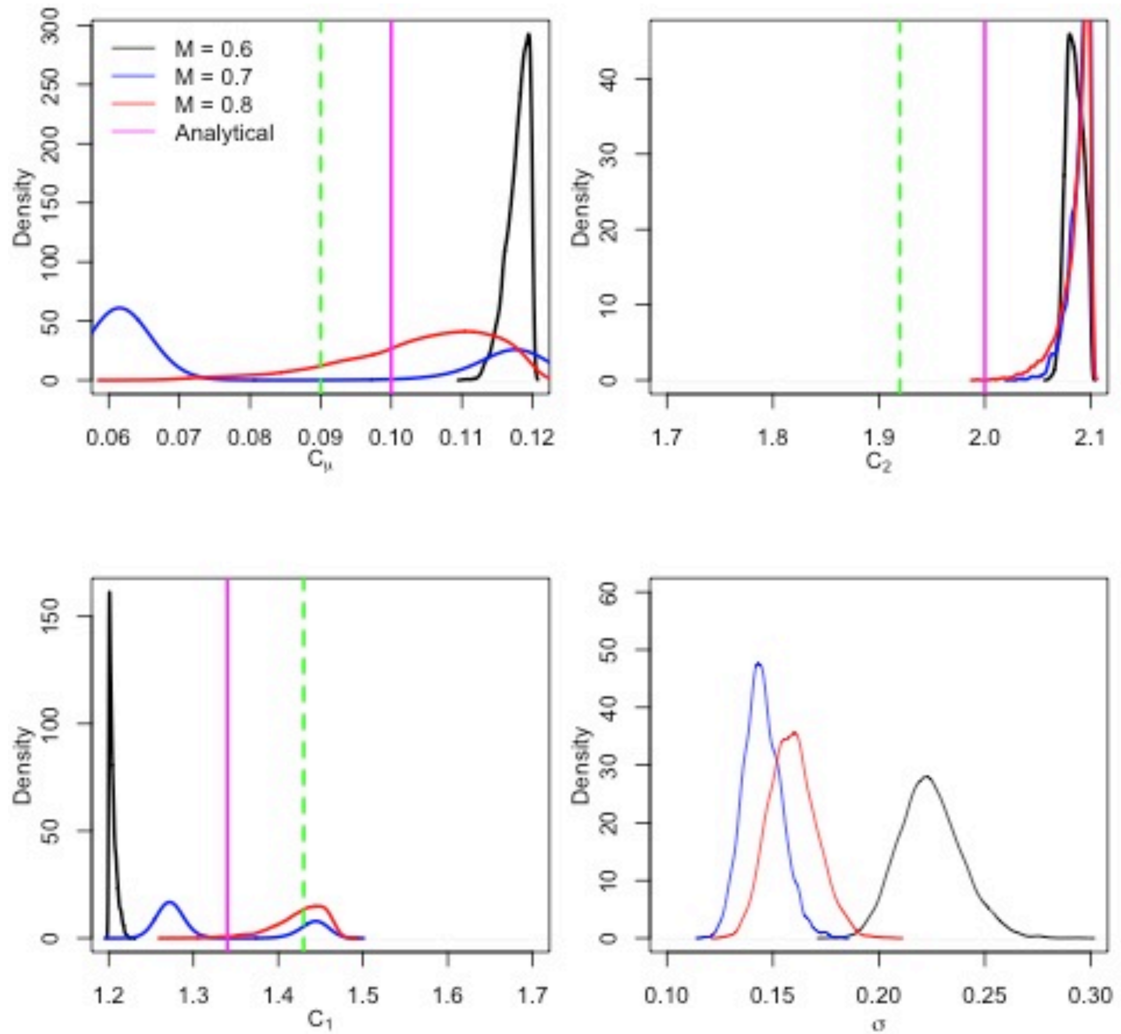




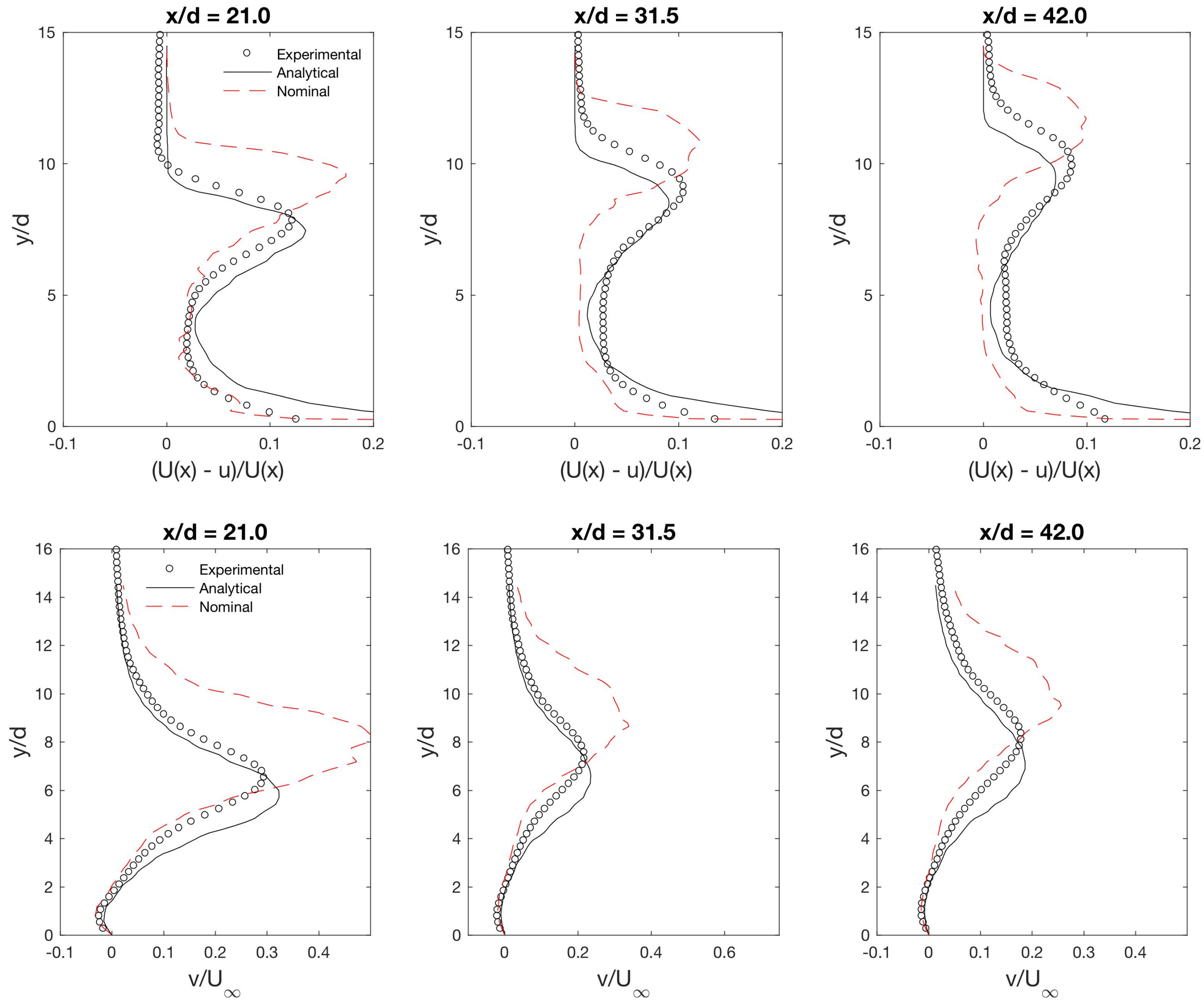
uturb/docs/aiaaj03/figs/Fig08

Comparison – analytical v/s PDFs

- Analytical values show a shift in the right direction
- The calibration was *not* sweeping model-form errors "under the carpet"



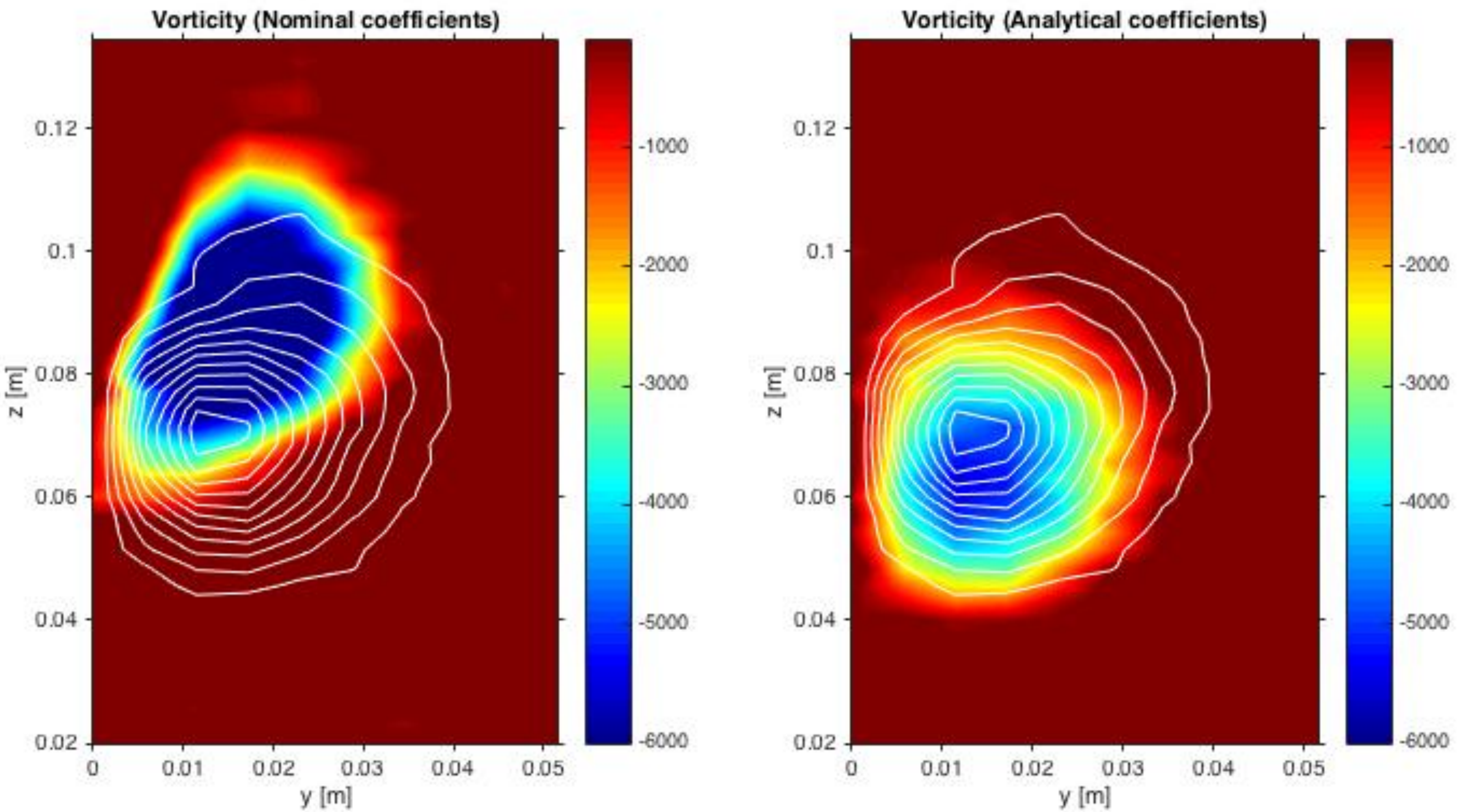
Validate model v/s experiment



Results for the M = 0.8, J = 10.2 case

uturb/docs/aiaaj03/figs/Fig02/

Comparing vorticity



Results for the M = 0.8, J = 10.2 case

- The analytical coefficients are quite predictive

Conclusions

- We performed a Bayesian calibration for jet-in-crossflow for 3 different Mach numbers
  - Calibration changed the predictive skill immensely
  - The PDFs for the 3 cases were NOT identical, but …
  - All indicated that the 2/3 calibrated parameters should be changed, and in the same direction
  - We worried that the calibration was simply compensating for model-form errors,
    - and not indicating that the nominal values of the parameters ($\mathbf{C}_{nom}$)were inappropriate
- So, we developed an analytical model – no fitting to data was performed – and obtained estimates of **C**
  - Which agreed with our calibrated values
  - And also had better predictive skill than $\mathbf{C}_{nom}$