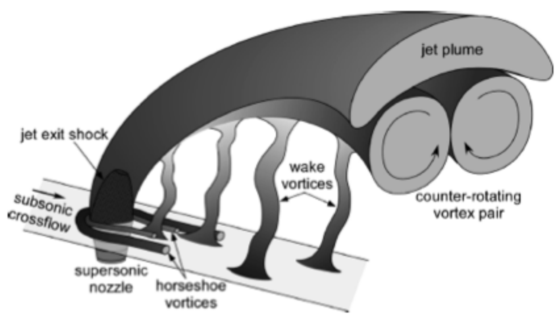
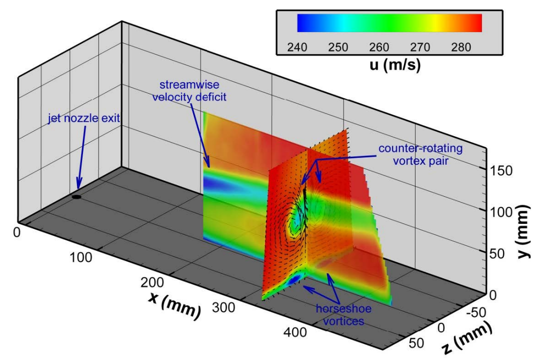
Exceptional service in the national interest

Sandia National Laboratories





Bayesian calibration of a RANS model with a complex response surface – A case study with jet-in-crossflow configuration

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

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

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

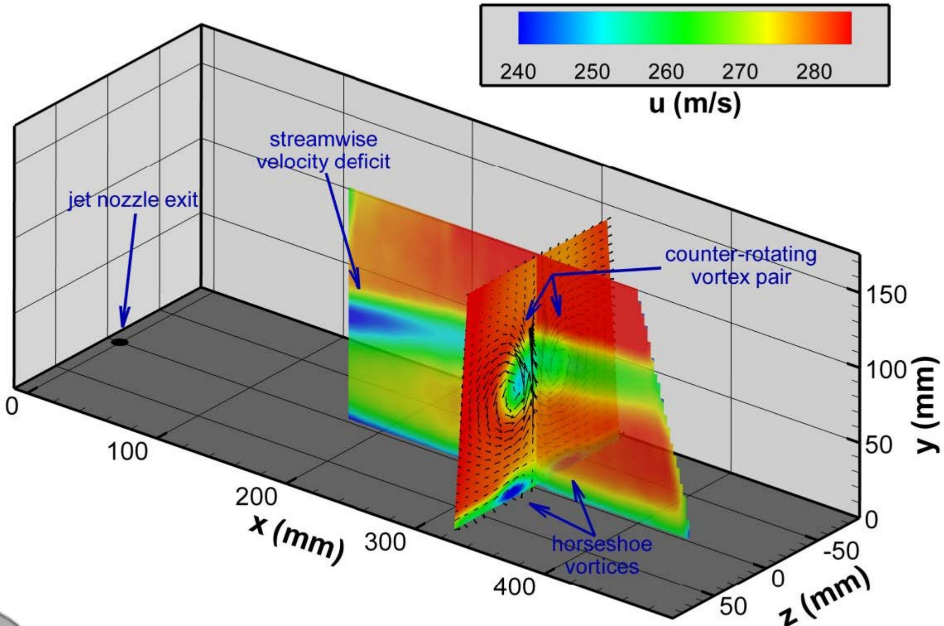
$$\frac{\partial \rho \varepsilon}{\partial t} + \frac{\partial}{\partial x_i}\left[\rho u_i \varepsilon - \left(\mu + \frac{\mu_T}{\sigma_\varepsilon}\right)\frac{\partial \varepsilon}{\partial x_i}\right] = \frac{\varepsilon}{k}\left(C_1 f_1 P_k - C_2 f_2 \rho\varepsilon\right) + S_\varepsilon$$

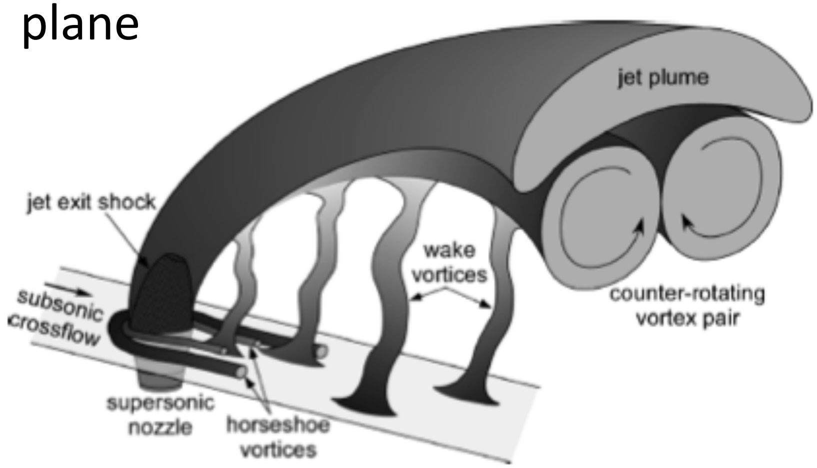
$$\mu_T = C_\mu f_\mu \rho \frac{k^2}{\varepsilon}$$

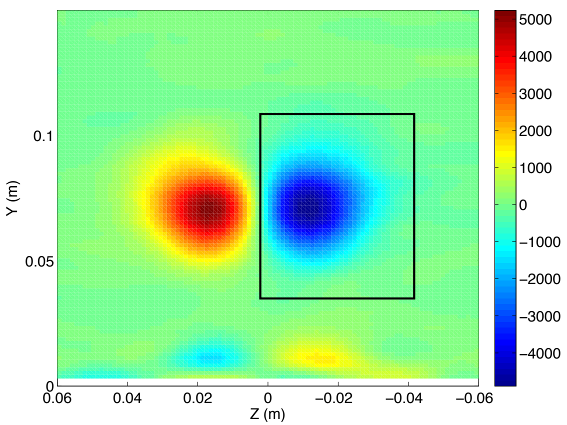
- Calibration parameters
  - **C** = {C_μ, C_1, C_2} ; C_μ: affects turbulent viscosity; C_1 & C_2: affects dissipation of TKE
- Calibration method
  - Pose a statistical inverse problem using experimental data
  - Estimate parameters using Markov chain Monte Carlo (MCMC)
    - 10^4 RANS calls
  - Construct a polynomial surrogate RANS simulations and use them inside MCMC

Target problem - jet-in-crossflow

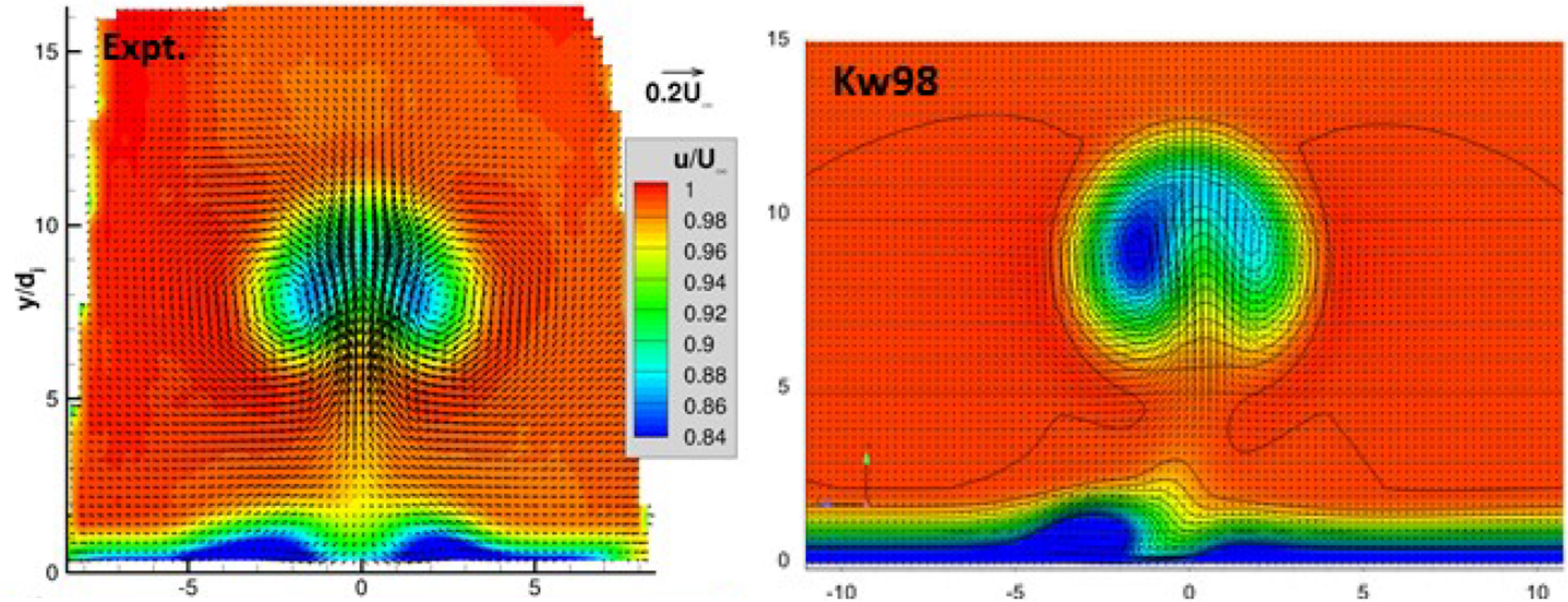
- A canonical problem for spin-rocket maneuvering, fuel-air mixing etc.
- We have experimental data (PIV measurements) on the cross- and mid-plane
- Will calibrate to vorticity on the crossplane and test against mid-plane







RANS (k-ω) simulations - crossplane results



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

Aims of the study

- Aims of the calibration
  - Calibrate to crossplane data but also match the midplane velocity profiles
  - Calibrate to a M = 0.8, J = 10.2 interaction
  - Also check predictive skill for a M = 0.8, J = 16.7 (as a check of accuracy away from calibration points)
- Technical challenges
  - Computational cost of 3D JinC RANS simulation
    - Replace 3D RANS with a surrogate model i.e., model crossplane streamwise vorticity $\omega^{(RANS)}_x(\mathbf{y}) = f(\mathbf{y}; C\mu, C_2, C_1)$, $f(:; \mathbf{C})$ is a curve-fit
  - 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 $f(:; \mathbf{C})$?

The Bayesian calibration problem

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

$$\Lambda\left(\omega_{\mathrm{ex}}^{(j)} \mid C\right) \propto \prod_{j \in \mathcal{P}} \exp\left(-\frac{\left(\omega_{ex}^{(j)} - \omega^{(j)}(C)\right)^2}{2\sigma^2}\right)$$

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

$$P(C,\sigma \mid \omega_{ex}^{(j)}) \propto \Lambda(\omega_{ex}^{(j)} \mid C,\sigma)\, \pi_{\mu}(C_{\mu})\, \pi_2(C_2)\, \pi_1(C_1) \pi_{\sigma}(\sigma)$$

- $P(\mathbf{C}|\omega_{ex})$ is a complicated distribution that has to be described/ visualized 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 out the new parameter combination

Making surrogate models - 1

- Training data
  - Parameter space $\mathfrak{P}$: $0.06 < C\mu < 0.12$; $1.7 < C_2 < 2.1$; $1.2 < C_1 < 1.7$
  - $\mathbf{C}_{nom} = \{0.09, 1.93, 1.43\}$
  - Take 2744 samples in $\mathfrak{P}$ 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 surrogate models 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 surrogate models for these 108 grid cells with large vorticity

Making surrogate models - 2

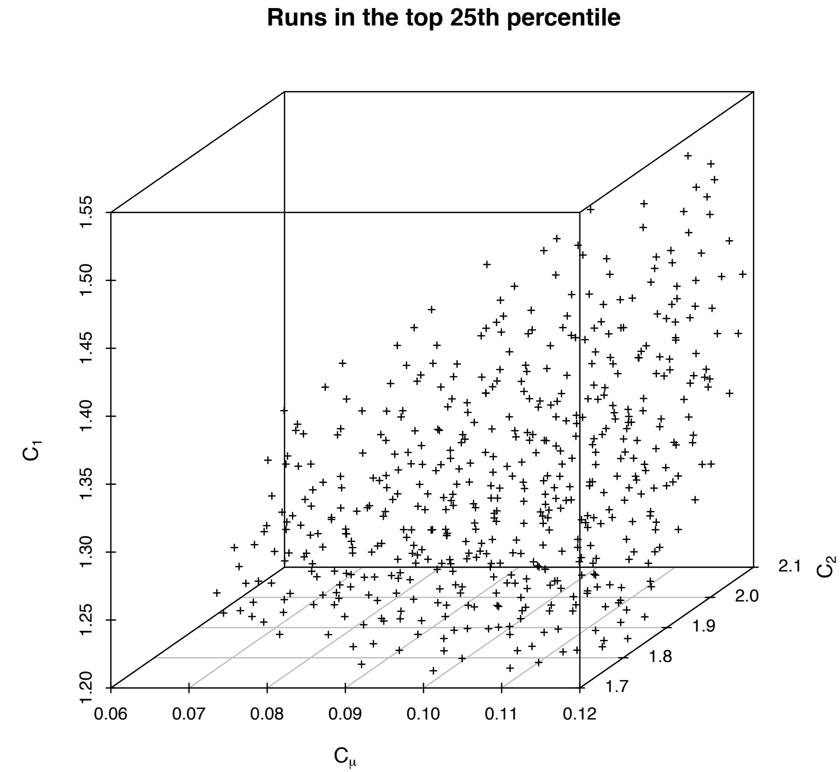
- Model ω_x in grid cell *j* as a function of **C** i.e. $\omega^{(j)}{}_x = f^{(j)}(\mathbf{C})$
  - Approximate this dependence with a polynomial

$$\omega^{(j)} \cong a_0 + a_1 C_\mu + a_2 C_2 + a_3 C_1 + a_4 C_\mu C_2 + a_5 C_\mu C_1 + a_6 C_2 C_1 + \ldots..$$

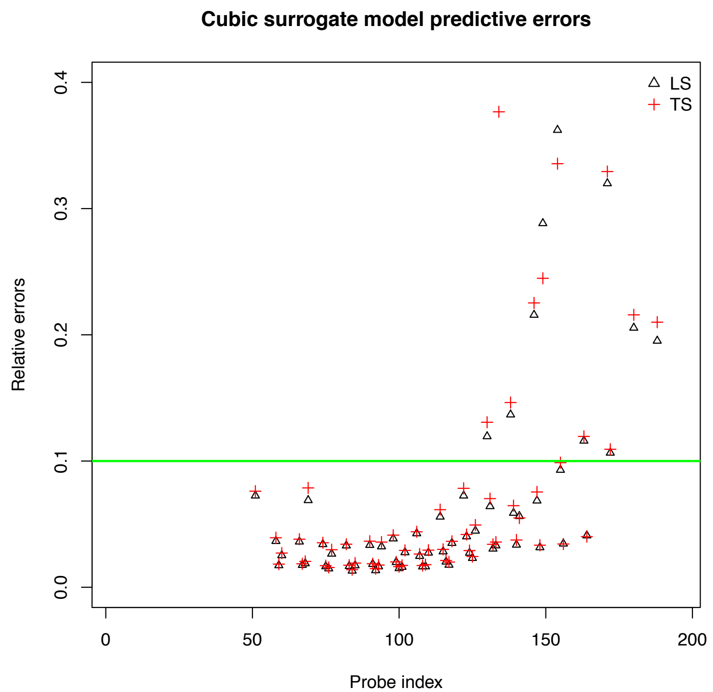
- But how to get (a_0, a_1, ….) for each of the probe locations to complete the surrogate model for each probe?
  - Divide training data in a Learning Set and Testing Set
  - Fit a full cubic model for to the Learning Set via least-squares regression; sparsify using AIC
  - Estimate prediction RMSE for 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

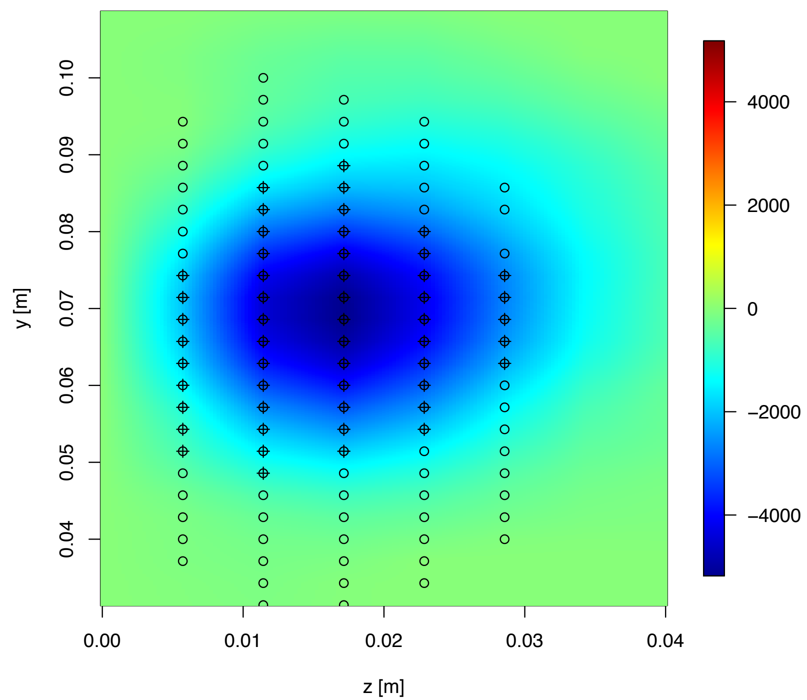
Making surrogate models - 3

- Choosing $\mathcal{R}$
  - Surrogates 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 surrogate models - 4



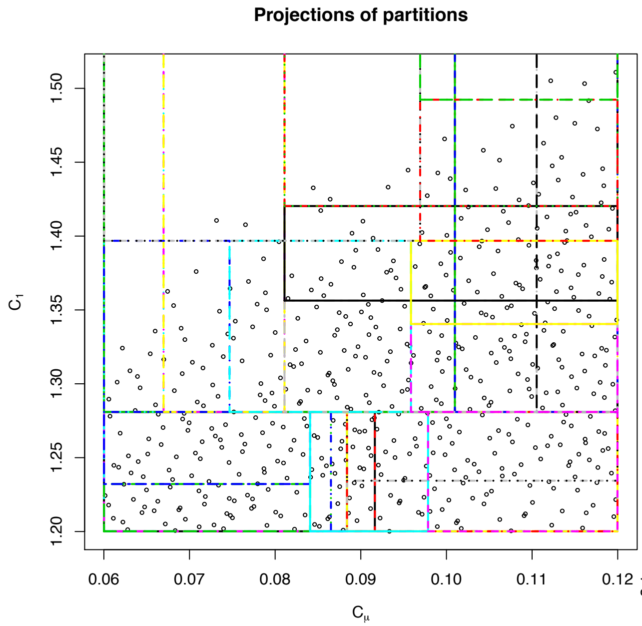


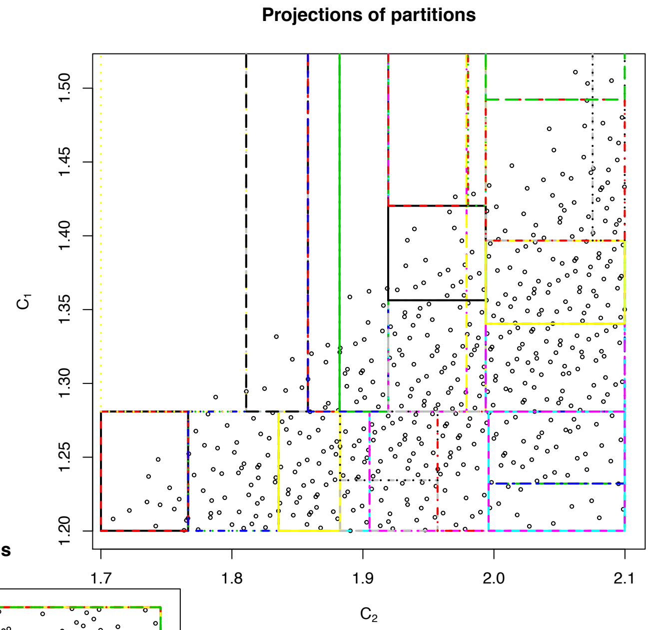
- Attempted to fit cubic surrogates to all 108 grid cells
  - Managed to achieve < 10% error at 52 / 108 grid cells
  - These are our "probes" where we will try to match experimental vorticity by optimizing **C** = $\{C\mu, C_2, C_1\}$

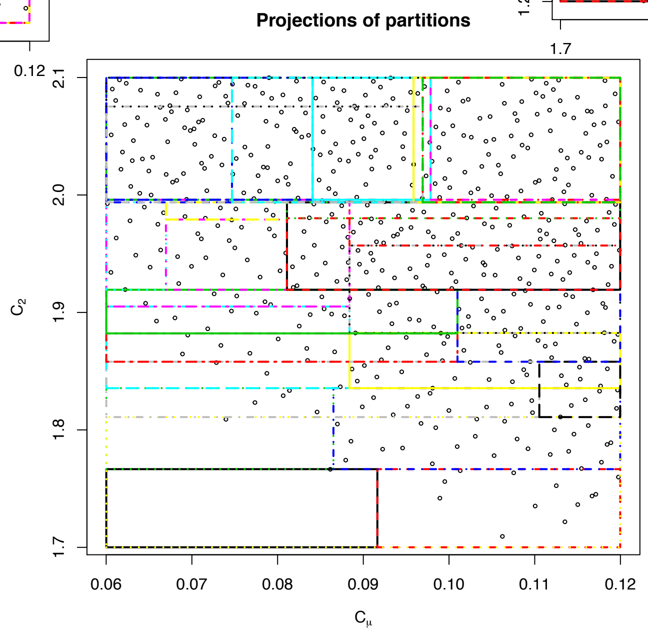
Making the informative prior - 1

- Our surrogate models are valid only inside $\mathcal{R}$ in the parameter space $\mathfrak{P}$
- During the optimization (MCMC) we have to reject parameter combinations outside $\mathcal{R}$ (this is our prior belief $\pi_{prior}(\mathbf{C})$)
  - We design a classifier based on treed linear models
  - We define $\zeta(\mathbf{C}) = 1$, for $\mathbf{C}$ in R and $\zeta(\mathbf{C}) = -1$ for $\mathbf{C}$ 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 $\zeta(\mathbf{C})$
- Treed models
  - Divides $\mathfrak{P}$ into boxes of equal variances; the recursively divides the boxes till the boxes are too small
  - Fits a linear model $\zeta(\mathbf{C})$ inside the leaf nodes
  - Allows a quick evaluation of $\zeta(\mathbf{C})$ for arbitrary $\mathbf{C}$

Making an informative prior - 2



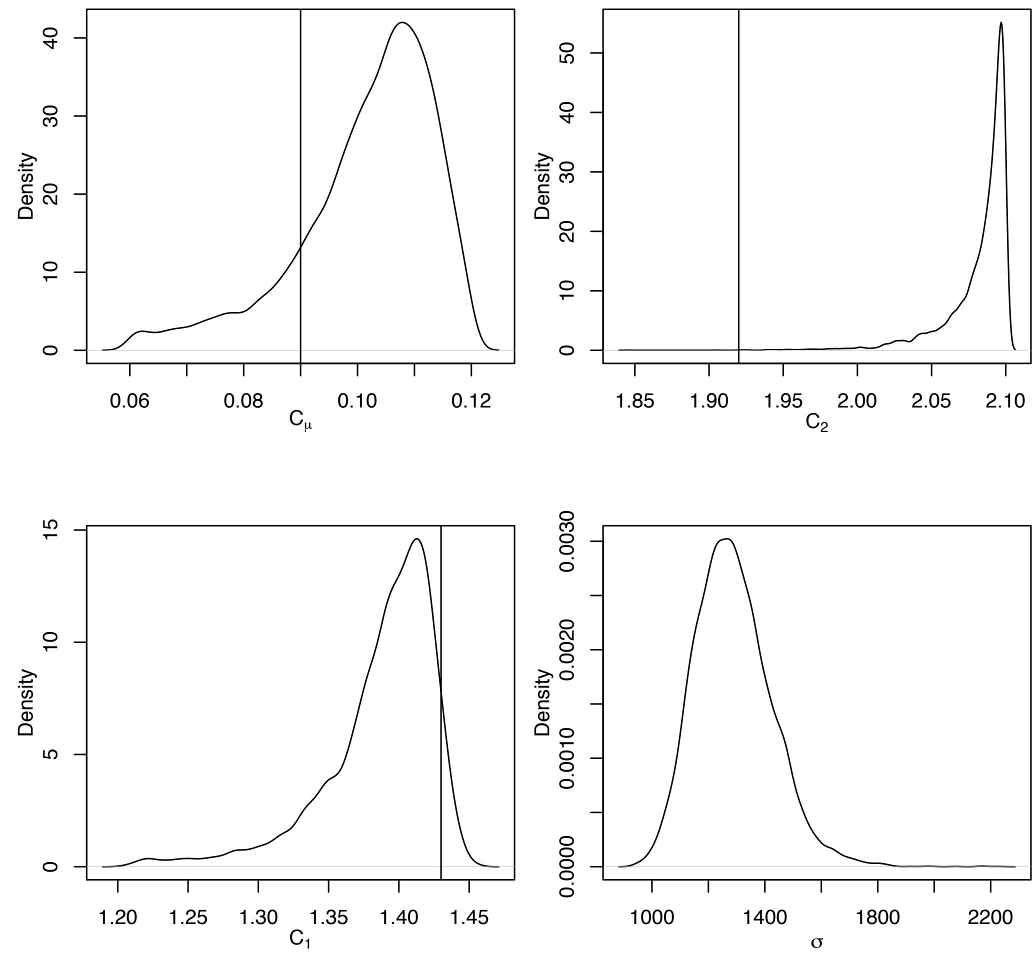




- Projections of boxes
- Dots are parameter combinations inside $\mathcal{R}$

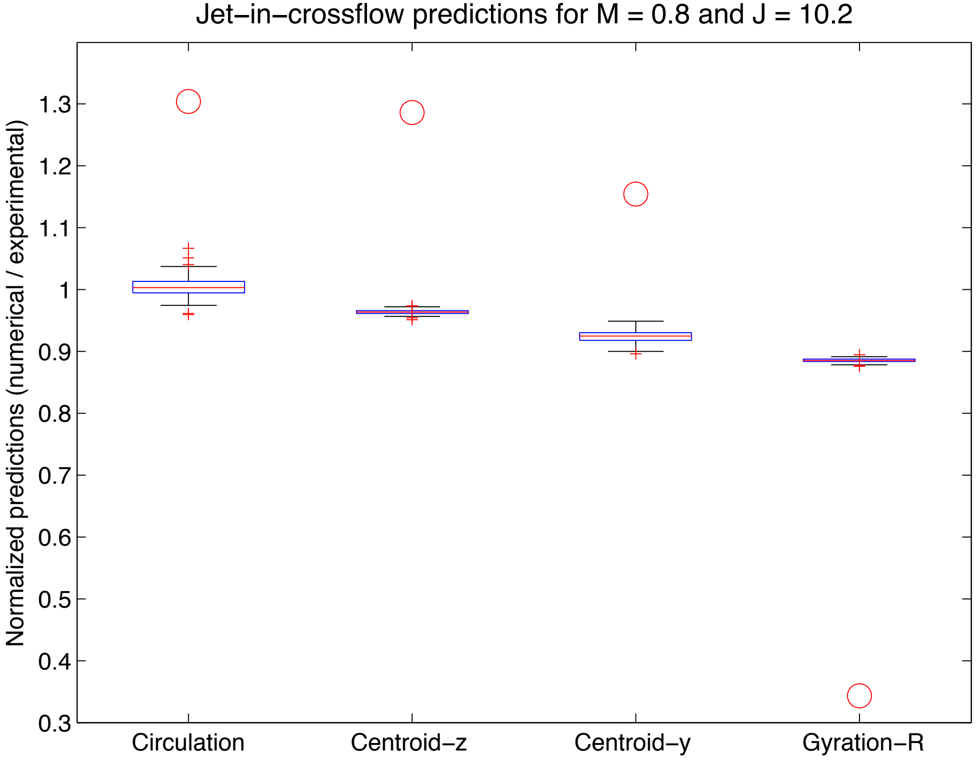
Solution of the inverse problem

- 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'



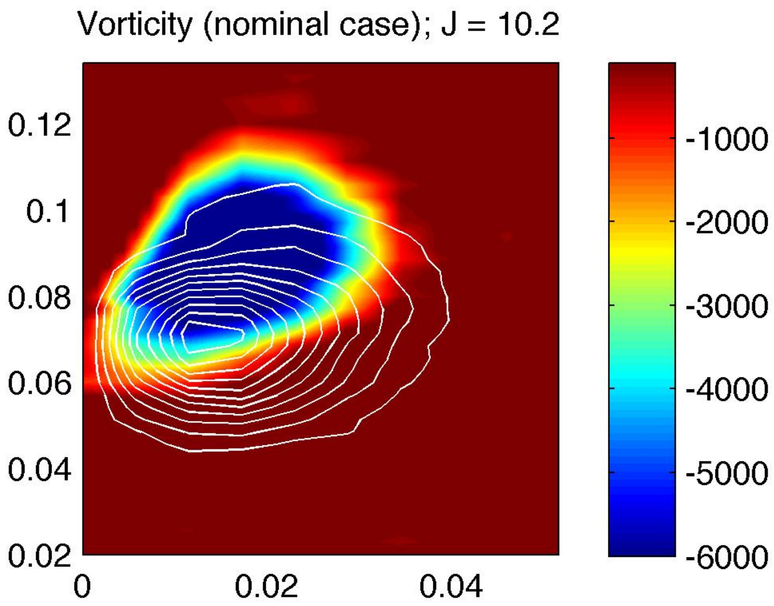
Check # 1 – point vortex summary

- 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
  - We also find a **C**$_{opt}$ = {0.1025, 2.09, 1.42} that provides the best predictions

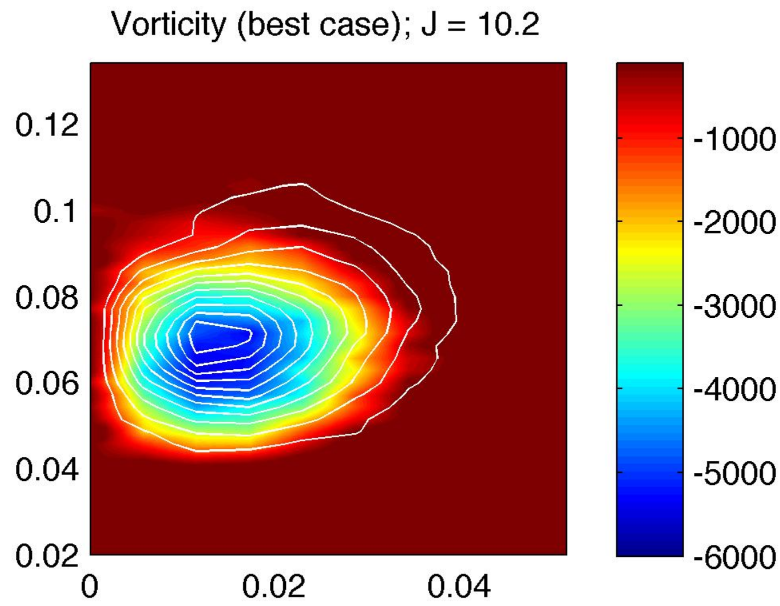


The spread of point vortex summaries are tightly distributed around 1. The red circles are the predictions from the nominal values of **C**

Check # 2 – the vorticity field



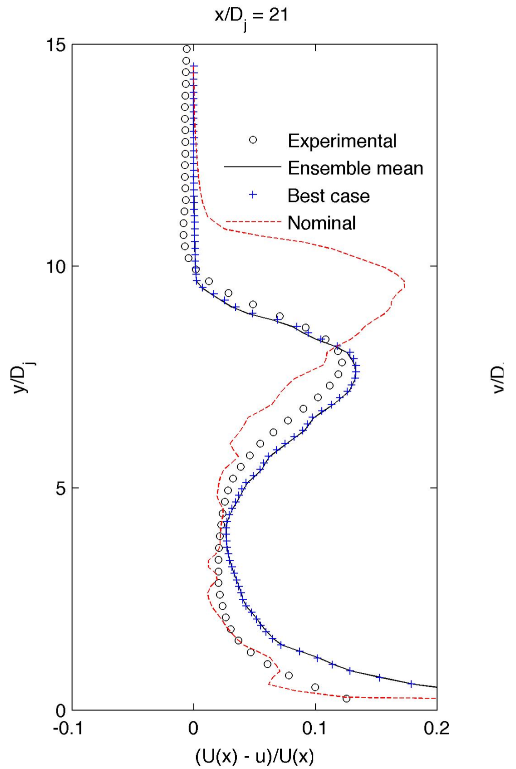
RANS predictions with $\mathbf{C}_{nom}$

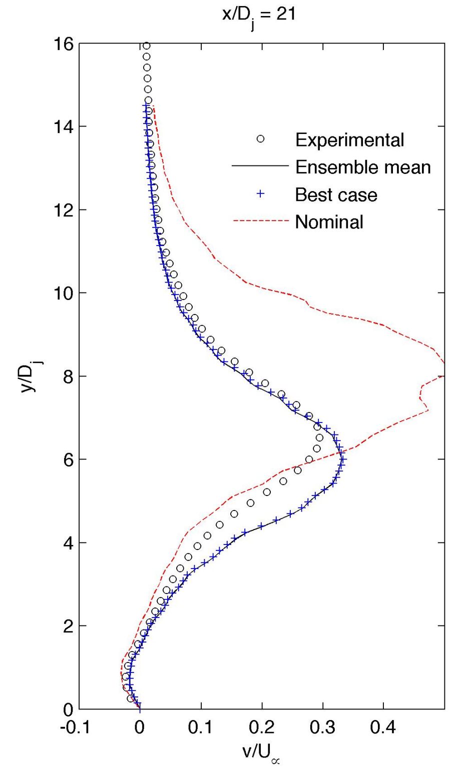


RANS predictions with $\mathbf{C}_{opt}$

- Contours are plotted using the experimental measurements
- The improvement is significant

Check # 3 – mid-plane comparisons



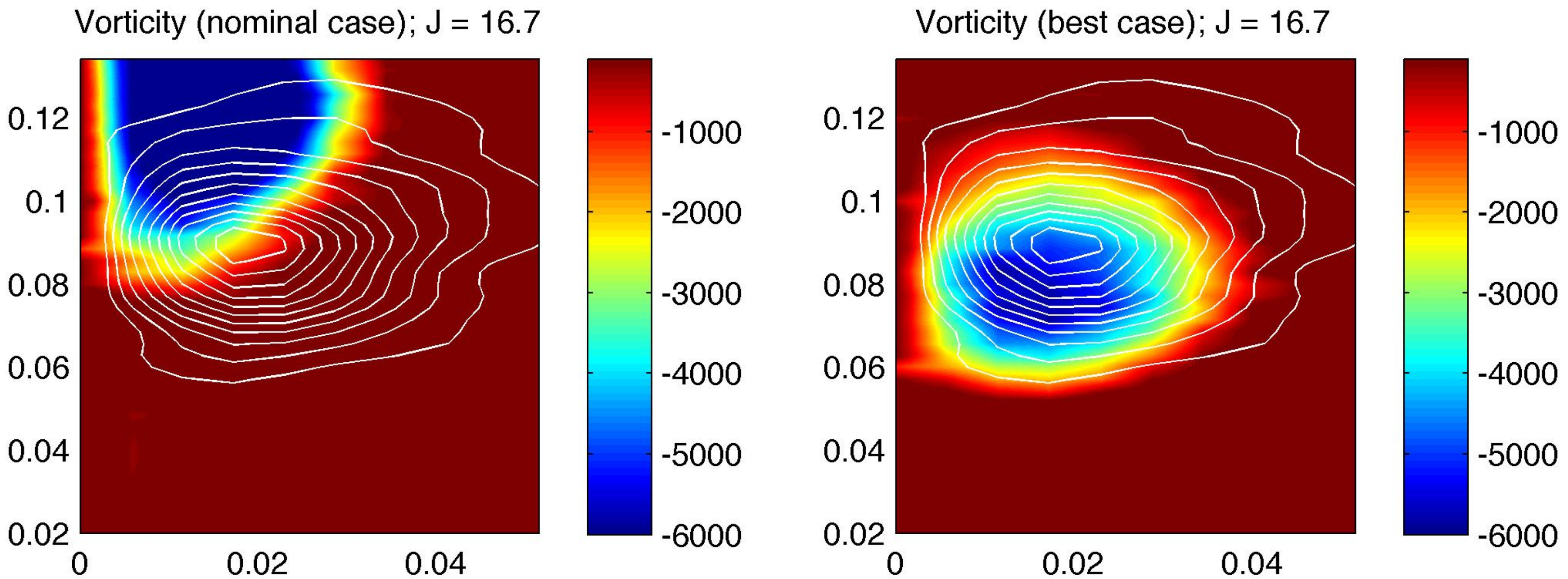


Streamwise velocity deficit at x/D = 21

Vertical velocity at x/D = 21

- Flow quantities on the mid-plane were not used in the calibration

Check at an off-calibration point

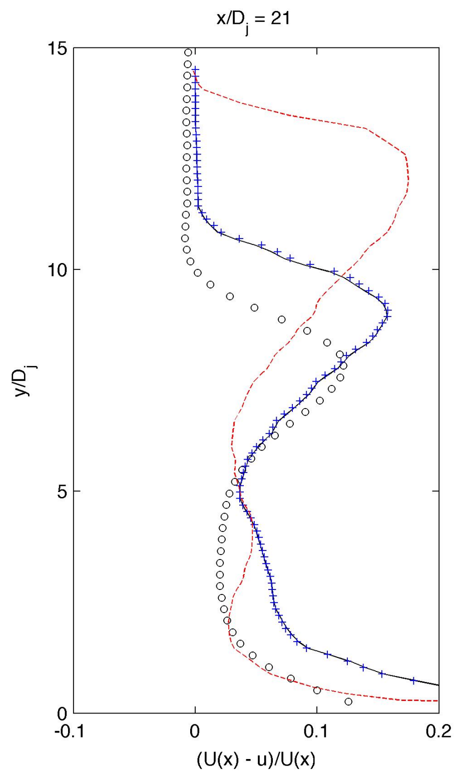


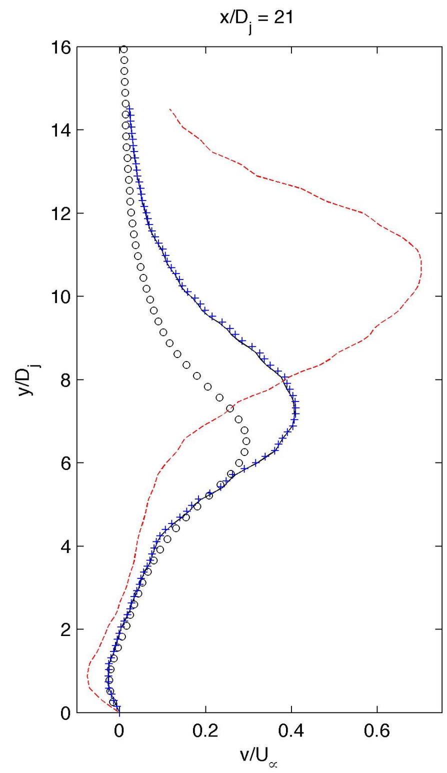
RANS predictions with $\mathbf{C}_{nom}$

RANS predictions with $\mathbf{C}_{opt}$

- Use the PDF from M = 0.8, J = 10.2 to predict a M = 0.8, J = 16.7 flow
- The improvement is significant

Checking at off-calibration point





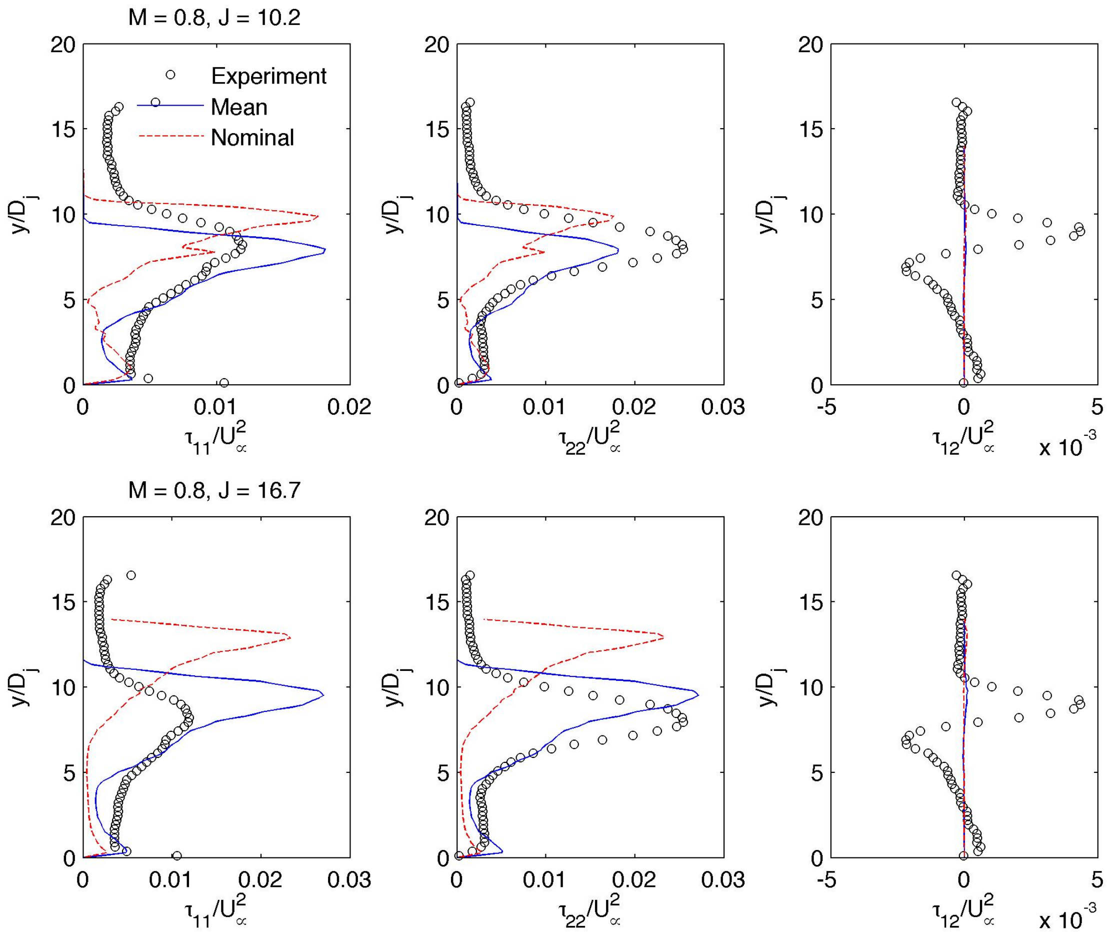
Streamwise velocity deficit at x/D = 21

Vertical velocity at x/D = 21

- Improvement over C_{nom} is substantial

Model-form error

- Shear stress completely off
- The TKE term *k* dominates in τ_{xx}, τ_{yy}
- So numerical τ_{xx}, τ_{yy} are almost equal
- For J = 16.7 predictions, post-calibration, are better



Conclusions

- The errors in RANS simulations of JinC are mostly due to the use of wrong parameters
  - Can be correct via calibration
  - Bayesian calibrations allows one to accommodate the uncertainty in {$C\mu$, C_2, C_1} estimates
  - Calibration to a M = 0.8, J = 10.2 interaction improved the flowfield's match to experiments (including for flow variables not used in the calibration)
  - The improvement in predictive skill carried over to a stronger jet (J = 16.7)
- Post calibration, the error is due to model-form error
  - Much smaller than the error due to wrong parameters
  - Makes itself felt most strongly in the prediction of turbulent stresses

BACKGROUND SLIDES

What is MCMC?

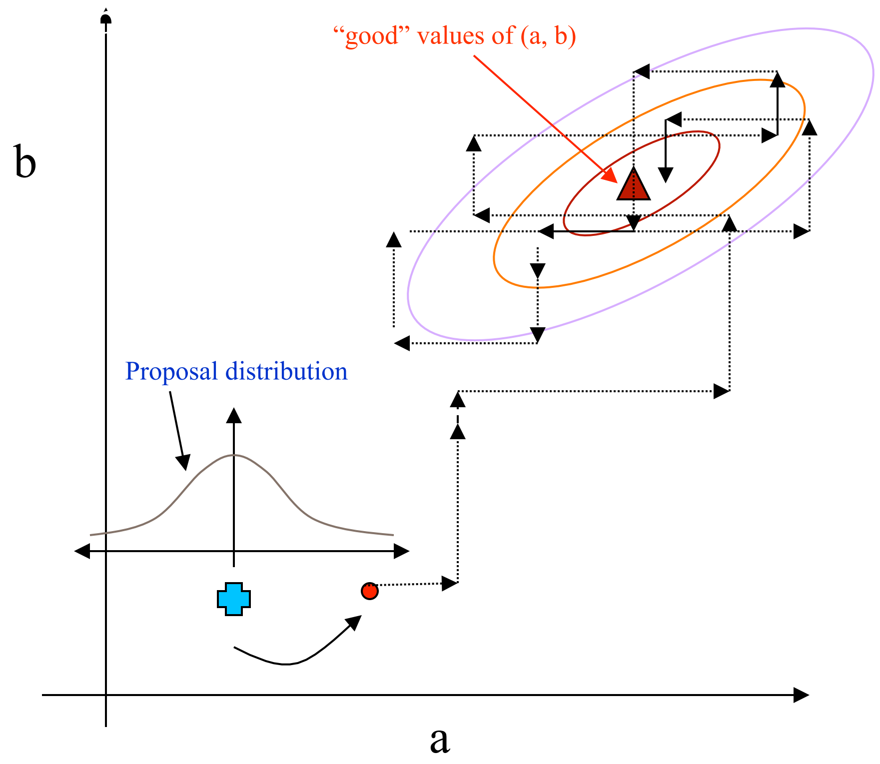
- A way of sampling from an arbitrary distribution
  - The samples, if histogrammed, recover the distribution
- Efficient and adaptive
  - Given a starting point (1 sample), the MCMC chain will sequentially find the peaks and valleys in the distribution and sample proportionally
- Ergodic
  - Guaranteed that samples will be taken from the entire range of the distribution
- Drawback
  - Generating each sample requires one to evaluate the expression for the density π
  - Not a good idea if π involves evaluating a computationally expensive model

An example, using MCMC

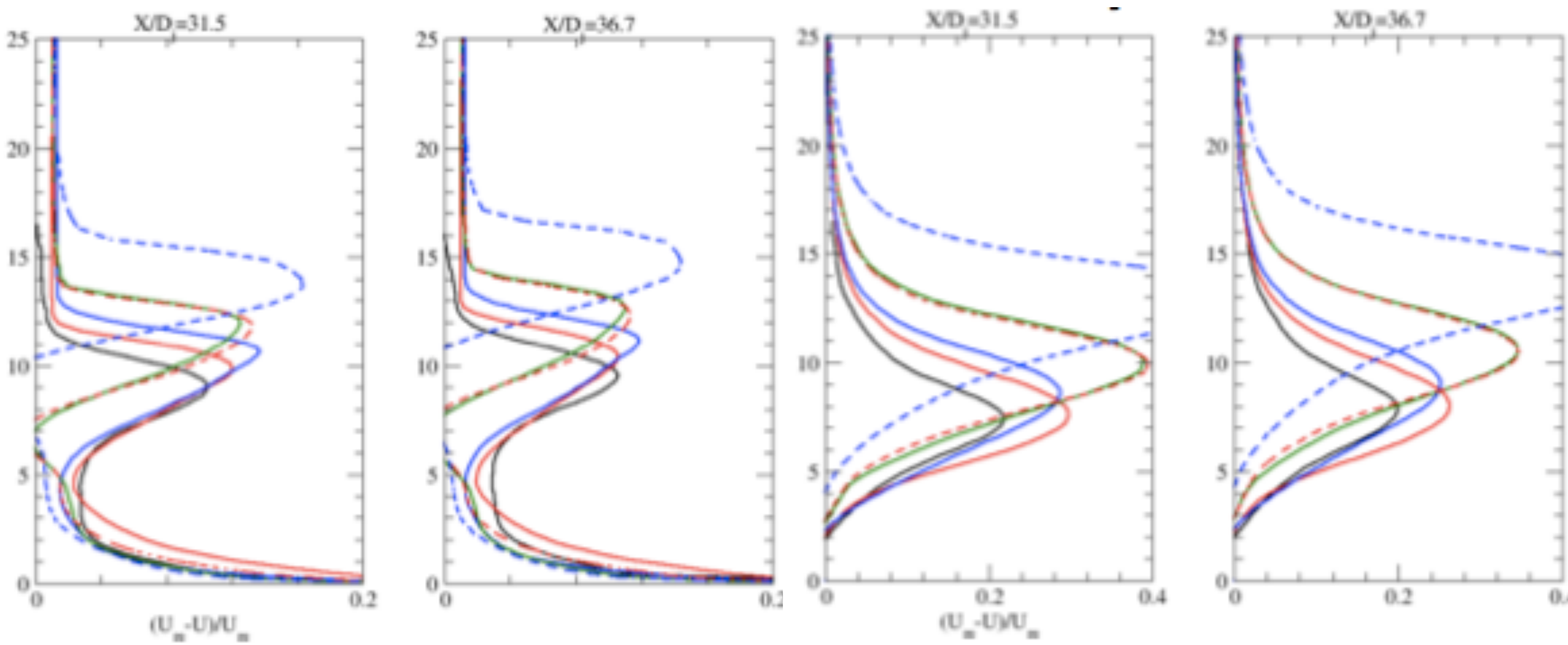
- Given: (Y^{obs}, X), a bunch of n observations
- Believed: y = ax + b
- Model: $y_i^{obs} = ax_i + b_i + \varepsilon_i$, $\varepsilon \sim \mathcal{N}(0, \sigma)$
- We also know a range where a, b and σ might lie
  - i.e. we will use uniform distributions as prior beliefs for a, b, σ
- For a given value of (a, b, σ), compute "error" $\varepsilon_i = y_i^{obs} - (ax_i + b_i)$
  - Probability of the set (a, b, σ) = $\Pi \exp(- \varepsilon_i^2/\sigma^2)$
- Solution: $\pi (a, b, \sigma \mid Y^{obs}, X) = \Pi \exp(- \varepsilon_i^2/\sigma^2)$ * (bunch of uniform priors)
- Solution method:
  - Sample from $\pi (a, b, \sigma \mid Y^{obs}, X)$ using MCMC; save them
  - Generate a "3D histogram" from the samples to determine which region in the (a, b, σ) space gives best fit
  - Histogram values of a, b and σ, to get individual PDFs for them
  - Estimation of model parameters, with confidence intervals!

MCMC, pictorially

- Choose a starting point, $P^n = (a_{curr}, b_{curr})$
- Propose a new a, $a_{prop} \sim \mathcal{N}(a_{curr}, \sigma_a)$
- Evaluate $\pi (a_{prop}, b_{curr} \mid \ldots) / \pi (a_{curr}, b_{curr} \mid \ldots) = m$
- Accept a_{prop} (i.e. $a_{curr} \leftarrow a_{prop}$) with probability $\min(1, m)$
- Repeat with b
- Loop over till you have enough samples



RANS (k-ω) simulations – midplane results



U-defect

V - velocity

- Experimental results in black
- All models are pretty inaccurate (blue and red lines are the non-symmetric results)

Model-form error - 1

- Calibration obtains good values of the parameter **C**
- Any error or mismatch with experiments that remains should be largely due to model-form error or missing physics
- One of the largest modeling assumption in RANS is the Boussinesq assumption
  - The turbulent stresses are a linear function of the strain rate
  - So the chances are that the largest error, post calibration, should be in the turbulent stresses
  - Luckily we have experimental measurements τ_{xx}, τ_{yy}, τ_{xy} on the midplane