51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition
07 - 10 January 2013, Grapevine (Dallas/Ft. Worth Region), Texas

AIAA 2013-0259

Application of Supervised Learning to Quantify Uncertainties in Turbulence and Combustion Modeling

Brendan Tracey *, Karthik Duraisamy †,
Juan J. Alonso ‡
Department of Aeronautics & Astronautics
Stanford University, Stanford, CA 94305, U.S.A

The accuracy of low-fidelity models of turbulent flow such as those based on the Reynolds Averaged Navier–Stokes (RANS) equations can be questionable, especially when these models are applied in situations different from those in which the models were calibrated. At present, there is no general method to quantify structural uncertainties in such models. Greater accuracy and a reliable quantification of modeling errors is much needed to expand the use of affordable simulation models in engineering design and analysis. In this paper, we introduce a methodology aimed at improving low-fidelity models of turbulence and combustion and obtaining error bounds. Towards this end, we first develop a new machine learning algorithm to construct a stochastic model of the error of low-fidelity models using information from higher-fidelity data sets. Then, by applying this error model to the low-fidelity result, we obtain better approximations of uncertain model outputs and generate confidence intervals on the prediction of simulation outputs. We apply this technique to two representative flow problems. The first application is in flamelet-based simulations to model combustion in a turbulent mixing layer; and the second application is in the prediction of the anisotropy of turbulence in a non-equilibrium boundary layer flow. We demonstrate that our methodology can be used to improve aspects of predictive modeling while offering a route towards obtaining error bounds.

I. Introduction

As computational approaches are increasingly used in analysis and design of engineering systems, quantification of simulation errors and uncertainties are also becoming evermore relevant. These uncertainties can broadly be classified into three categories: i) Randomness in simulation inputs or parameters, ii) Numerical approximations, and iii) Inadequacies of the model in representing physical phenomena. In this work, the third among the above-mentioned set, *i.e.*, errors and uncertainties due to modeling or model-form uncertainties are addressed. While many attempts have been made at investigating modeling uncertainties, most of the work has focused on a) Parameterizing the model and evaluating the bounds in quantities of interest as introduced by the parameters, or b) Bayesian averaging of multiple models or extensions therein. In the present work, we introduce an alternate route and instead employ Machine Learning techniques to learn the relationships between model *prediction* and *reality* under *controlled conditions* and this information is exploited to guide a framework to estimate the modeling error in related applications. This technique is developed and applied to specific applications involving turbulence and combustion.

Turbulence Modeling: Direct numerical simulations (DNS) of the Navier–Stokes equations is infeasible for most flows of practical interest because of the requirement of resolving the wide range of length and time scales of turbulence. Large Eddy Simulations (LES) offer the promise of significantly reduced cost, but the cost benefit (when compared to DNS) is only marginal in wall-bounded flows. In engineering applications, especially in high Reynolds number situations, the Reynolds Averaged Navier–Stokes (RANS) equations are most prevalent, due to their relatively low computational requirements. However, the ensemble averaging

*PhD Candidate, AIAA Member
†Consulting Assistant Professor, AIAA Member
‡Associate Professor, AIAA Associate Fellow

process used to derive the RANS equations introduces additional correlations of fluctuating quantities that have to be modeled. The closure of this system has been a topic of research over the past five decades, and a myriad of turbulence models have been developed to do so. There is much interest in quantifying inaccuracies in the closure of a turbulence model. Practitioners typically validate a specific turbulence model with comparisons to specific flow configurations, and use intuition and experiments to extrapolate the performance of a turbulence model to general flow situations. While often useful, there exists no formal procedure for obtaining error bars around solution results. This problem is exacerbated when the turbulence model is used in applications far from its designed flow regime, as errors can be large and less understood. At present, most work in this area has focused on developing uncertainties in the coefficients of a turbulence model and using Monte Carlo simulations to examine the effects of these uncertainties (for instance, Refs. 1,2), although some recent efforts (Refs. 3, 4) have attempted to address structural uncertainty in turbulence models.

Turbulent Combustion Modeling: DNS of turbulent combustion is an even more expensive proposition because of the additional need to resolve chemical time and length scales. In practice, simpler combustion models are pursued, especially when used in conjunction with RANS models of turbulence. One such model is the Flamelet Progress Variable Approach[5] (FPVA) which assumes that chemical time-scales are shorter than turbulence time-scales and hence the numerical solution of turbulent (and mixing) processes can be performed separately from the chemical processes which are instead pre-computed and tabulated. In comparison to detailed finite rate chemistry calculations which involve the solution of transport equations for many species, the FPVA involves solving just three scalar equations (in addition to the mean flow equations) and is thus computationally both less expensive and less stiff. Though FPVA models have been successfully used in many engineering applications, the flamelet approximation is known to be inaccurate in situations such as those involving auto-ignition and when chemical time-scales are not orders of magnitude faster than the time-scales associated with the flow. There is, hence, great interest in quantifying modeling errors resulting from flamelet approximations and its use as a combustion model. Prior work on uncertainty quantification in combustion modeling has mostly focused on treating parametric errors in the chemical kinetics (for instance, Ref. 6) and using Monte-Carlo or spectral projection techniques to propagate these uncertainties to simulation outputs.

Quantifying turbulence modeling uncertainties using parametric variations has provided useful insight, but ultimately they are limited as they only express the effects of errors in the parameters of the turbulence model while not providing any information about errors in the form of the model. In this work, we use supervised learning techniques to learn the relationships between selected outputs from low and high-fidelity simulations. Using these results, a framework to estimate errors in computed low-fidelity solutions is developed, and we apply our methods to two example problems. The first example learns the error between FPVA solutions and finite rate chemistry solutions, and uses the error model to improve the FPVA solution. The second example uses a DNS flow solution of a flow over a bump to learn errors in the Reynolds stress anisotropy due to the use of a linear eddy viscosity model. In both cases, a stochastic model is used to provide error bars on relevant macroscopic quantities of interest.

The paper is organized into three parts : Section II introduces machine learning (ML) and our specific algorithm; Section III uses the ML algorithm to learn output features in terms of specific input features and Section IV demonstrates application of the trained algorithm in two sample cases involving combustion and turbulence.

II. Supervised Learning Methodology

A. Machine Learning

In *supervised learning,* the algorithm is provided with a set of input feature vectors $x_1, x_2, ...x_n$, and a corresponding set of output feature vectors, $y_1, y_2, ..., y_n$ from which a model can be built. A simple example of supervised learning is least-squares regression, in which a polynomial is fit to a data set. The feature vector is constructed as $1, x, x^2, ...$, and the output, y, is assumed to be a linear combination of the features, *i.e.* $y = \beta_0 + \beta_1 x + \beta_2 x^2 ....$ The values β_i are learned by multiplying the pseudo-inverse of the feature matrix with the output vector. The least-squares method is an example of "parametrized" learning, in that the relationship between the input and the output is assumed to be a function of a set of parameters which are learned from data. In non-parametric regression, no such specific form is assumed, and models freely conform to input data. Non-parametric techniques are much more flexible in the functions they can fit, in

that they make no assumptions about the form of the relationship between the input and output data. A simple example of non-parametric regression is *nearest neighbor regression*, which predicts the output at x to be equal to the output of the closest training data location. It is clear that given enough data, nearest-neighbor regression will provide a prediction close to the truth, though "enough data" may be quite large and it may perform significantly worse if the training data set is small. Other non-parametric techniques are much more sophisticated, such as Gaussian processes,[7] kernel regression,[8] and locally-weighted regression.[9]

B. Use of Machine Learning

Given the results of a high-fidelity model and a low-fidelity model, we apply machine learning algorithms to learn the relationships between the model discrepancy in terms of selected flow features. More specifically, we split this process into two pieces:

1) We obtain an estimate of the error in the low-fidelity model based *solely* on its local features. These features could be, for instance, mass and mixture fractions in a combustion simulation. Applying this model gives an estimate of the error in a quantity of interest (for instance, the enthalpy of the mixture) at each point in the flow.

2) These individual errors may be highly correlated between two locations close to each other in the physical domain, and hence we define a model to describe the correlations among local uncertainties, as ignoring these correlations may significantly under-estimate the true macroscopic error.

Our procedure starts with the identification of a set of local flow features, Y, in the high-fidelity model whose value is to be learned, and a set of local flow features, X, in the low-fidelity model that are relevant to predicting the value of Y. In theory, any number of input or output quantities could be used, but practically as the number of features grows, the training procedure requires more data and becomes computationally expensive. With this in mind, an ideal set of output features will be small in size, highly relevant to the output quantity of interest, and capture the effects of different uncertainties. The input features should have similar qualities.

C. Extended Kernel Regression

With a set of features in hand, and their values extracted from a data set, the next step is to choose a supervised learning algorithm to train the local error model on the collected data. This model is a mapping from the m input features to the n output features ($\mathbb{R}^m \rightarrow \mathbb{R}^n$). We would like our learning algorithm to be non-parametric, as we do not know the form of our uncertainty, and also to be stochastic, thus allowing for the generation of error bars for uncertainty (as opposed to getting a single correction to the low fidelity model). To meet these criteria, we have developed an extended form of kernel regression as our learning algorithm. In kernel regression, one chooses a *kernel function* $K(x_i, x_j)$ over the feature vectors of any two data locations. This function defines the closeness of any two input locations x_i and x_j. In standard kernel regression, the output y for an input location x is computed by a weighted sum over all of the data points, *i.e.*

$$\tilde{w}_i = K(x, x_i)$$

$$w_i = \frac{\tilde{w}_i}{\sum_i^N w_i}$$

$$y_j = \sum_i^N w_i y_{i,j}, \tag{1}$$

$$\tag{2}$$

where $x_1, ..., x_N$ are the input feature vectors for points $1, ..., N$ in the data set, and $y_{i,j}$ is the j^{th} output for the i^{th} data point. Kernel regression has the advantage of being non-parametric, and thus it can represent any smooth function given enough data. Since we do not have to assume a specific form of the function, we can learn the true map from the input to the output.

Kernel regression was originally designed to model deterministic functions, and it gives a single prediction for the output quantity rather than generating a probability distribution over possible values. We do not expect that the mapping between our input and output features to be deterministic, and furthermore we

are interested in learning the probability distribution over the true value of the output given the input. To address this shortcoming, we use a modified version of kernel regression that generates probabilistic outputs. As a simplification, we assume that the output has a multivariate Gaussian distribution given the input. Furthermore, we assume that each output dimension is independent from the others. These assumptions may not be generally true, but they simplify our analysis at the present time and will be relaxed in future work. The normal prediction of kernel regression is treated as the mean of the Gaussian distribution, and a weighted standard deviation of the data samples is treated as the standard deviation of the output. Mathematically,

$$\begin{aligned}
\tilde{w}_i &= K(x, x_i), \\
w_i &= \frac{\tilde{w}_i}{\sum_i^N w_i}, \\
\mu(x)_j &= \sum_i^N w_i y_i, \\
\sigma(x)_j &= \sqrt{\sum_i^N w_i (y_i - \mu_x)^2} \quad .
\end{aligned} \tag{3}$$

where μ and σ denote the mean and standard deviation respectively. For a data point with feature vector x, the above equations provide a prediction for the mean and variance of the output feature y_j at that location. Equations (3) are computed with each feature vector of the flow field which gives a prediction of the true mean and standard deviation of each output feature at every grid point.

To illustrate this regression technique, assume we have the data set represented in Fig. 1 which has one input feature and one output feature. In certain regions of the input feature, the output feature value has high variability while in other regions the output feature is very predictable.

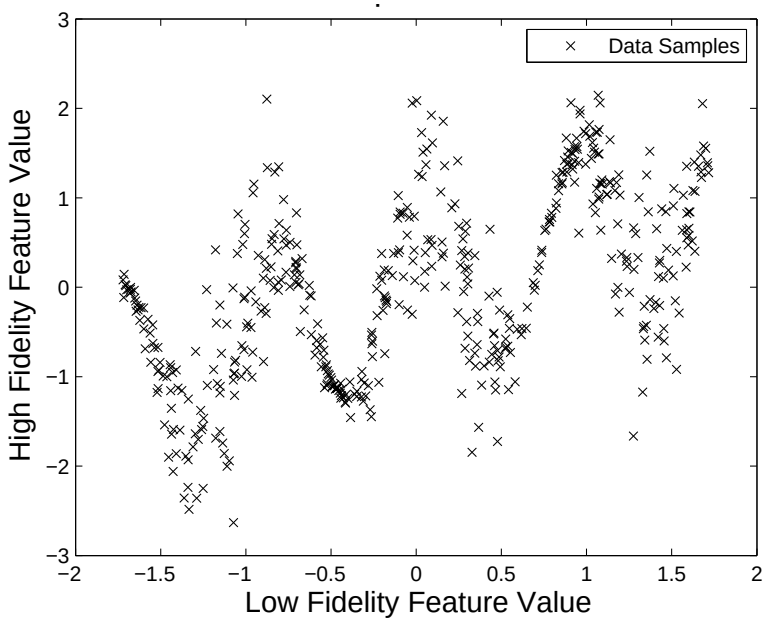


Figure 1: An example data set that may be given to the extended kernel regression technique. The x-axis is the value of the input flow feature, and the y-axis represents the value of the output flow feature.

By using extended kernel regression, this intuition can be made precise. We select a squared-exponential as our kernel function,

$$K(x_i, x_j) = exp(-(x_i - x_j)^2/\tau)$$

where τ is a "bandwidth" parameter representing the length scale of influence. For a specific value of τ, the mean and variance at each location in the input feature space can be computed using eqn. (3). It is

important to choose a suitable value for τ; a τ that is too large may miss local variations in the mean and variance, while a τ that is too small may generate variations that are not present. To balance these trade-offs, the value for τ is chosen by using k-fold cross-validation.[10] Loosely, cross-validation successively leaves out points in the data set, and the mean and variance at the held-out location is predicted. The τ which best predicts these held-out training points (as measured by the sum of the log-likelihoods of the held-out data) is chosen as the best value for τ. At the end of this procedure, we end up with a prediction for the mean and variance of the output flow feature as a function of the input flow feature, as shown in Fig. 2, which is used as the local error model.

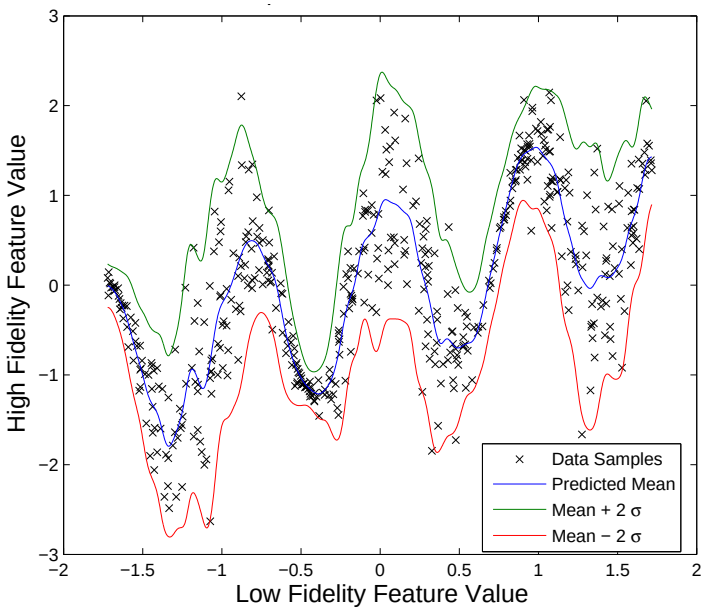


Figure 2: An example fit to the data set The x-axis is the value of the input flow feature, and the y-axis represents the value of the output flow feature. The blue line represents the prediction for the mean of the output flow feature prediction, while the red and green lines represent the mean plus and minus two standard deviations respectively.

D. Spatial Correlation

The next step is to form a spatial correlation model. There are many possible choices for this, and we choose two different methods for the two examples to follow in the next section. In the combustion example of the following section, we use a Gaussian Process-type construction to form a covariance matrix $\tilde{\Sigma}$ over 50 evenly spaced locations in the 1-D flow field. This covariance matrix is then scaled by the individual standard deviations to create a joint covariance matrix for the full flow field, *i.e.*

$$\Sigma_{i,j} = \sigma_i \sigma_j \tilde{\Sigma}_{i,j}, \tag{4}$$

where, i and j are the i^{th} and j^{th} locations in the flow field. This joint distribution has marginal uncertainties which match the individual uncertainties, but locations close to each other in the flow field are more likely to deviate in the same direction. A vector of output features for the flow field is constructed by sampling from this multi-variate Gaussian and error bars in the combustion simulation are generated by Monte-Carlo sampling in this joint distribution.

For the RANS case which involves a 2-D flow field, sampling from a covariance matrix will be expensive. Instead, we generate two standard normal random variables, and shift each point in the barycentric map by that number of standard deviations relative to its own uncertainty. Each such realization gives a new output map, and again, uncertainties in macroscopic quantities of interest are generated by sampling from this set of maps.

III. Application of Machine Learning to training data sets

To evaluate the feasibility and effectiveness of the techniques described in the previous sections, two sample applications are attempted.

Example 1: Source term of reaction progress variable in FPVA

In the FPVA, the flame is modeled as an ensemble of one-dimensional laminar flames (or flamelets). This allows for the thermo-chemical properties to be pre-computed using the solution of differential equations, the results of which are stored in a tabulated form as a function of a limited number of scalars. In our ML approach, we extract these scalars from a high-fidelity data set and use extended kernel regression to find the appropriate output quantities as a function of those inputs. Using these "experimentally" generated values for the table, inaccuracies in the FPVA table can be quantified. The high-fidelity data is a temporal DNS of a reacting mixing layer using finite-rate chemistry (Ref. 11) Using the full data set to learn itself would not be informative, and so rather than utilizing the full data set, a random subset of five thousand data points were used as a training data set.

As an output quantity, we consider the rate of generation $\tilde{\dot{\omega}}_C$ (see Appendix A) of the mass fraction of H_2O (the reaction progress variable) in the mixture. The FPVA looks up $\tilde{\dot{\omega}}_C$ in terms of the mean mixture fraction, the variance of the mixture fraction and the reaction progress variable ($\tilde{Z}$, $\tilde{Z}''^2$, and $\tilde{C}$, respectively). Applying eqn. (3) using the training data gives an updated value for the table. We choose an anisotropic squared-exponential as the kernel function,

$$K(x_i, x_j) = exp(-\sum_k (x_{i,k} - x_{j,k})^2/\tau_k) \quad ,$$

where $x_{i,k}$ represents the k^{th} element of the input vector x_i, and τ_k is the bandwidth parameter for the k^{th} input variable (a total of three in this case). We use cross-validation to choose τ_k, and then apply eqn. (3) to construct a look-up table using the mean prediction of the source term.

The extended kernel regression algorithm also gives a measurement of the uncertainty of each element in the table, which is used to generate estimates of uncertainty in the prediction of the source term. Naively, one may sample the Gaussian distribution for each element in the table and re-run the FPVA. However, this may result in a noisy (spatially uncorrelated) prediction of the source term. In reality, the true source term can be expected to be spatially smooth.

We account for these correlations by introducing a global covariance model superimposed on the individual uncertainties. We assume that the source term for the flow field is generated from a multi-variate Gaussian, where the marginal variance at each location is the uncertainty in the table value, but spatially closer locations are more highly correlated than locations that are far from each other. We assume that any two locations in the flow field are correlated according to

$$\rho_{i,j} = exp(-(y_i - y_j)^2/b) \tag{5}$$

where y is the spatial location and b is a bandwidth parameter representing the length scale of correlation. From eqn. (5), it can be seen that two locations very close together are almost perfectly correlated, while locations far apart in the flow field are uncorrelated from one another. The full covariance matrix for the flow field is computed as

$$\Sigma_{i,j} = \rho_{i,j}\sigma_i\sigma_j \quad , \tag{6}$$

where σ_i is the individual standard deviation determined using extended kernel regression. When coupled to the FPVA simulations as described in the next section, samples from this multivariate Gaussian distribution are used to generate the source term over the entire domain. This covariance matrix is formed over multiple locations in the flow field, and a spline fit is used to interpolate the source term onto the rest of the grid cells. Fig. 3 shows an example of this sampling process. In this figure, the dashed black line is the original prediction for the source term from the FPVA table, and the solid black line is the mean of the prediction from DNS data. The thin colored lines represent different samples from the multivariate Gaussian distribution over the flow field. As will be explained in the next section, these different draws are communicated back to the flow solver.

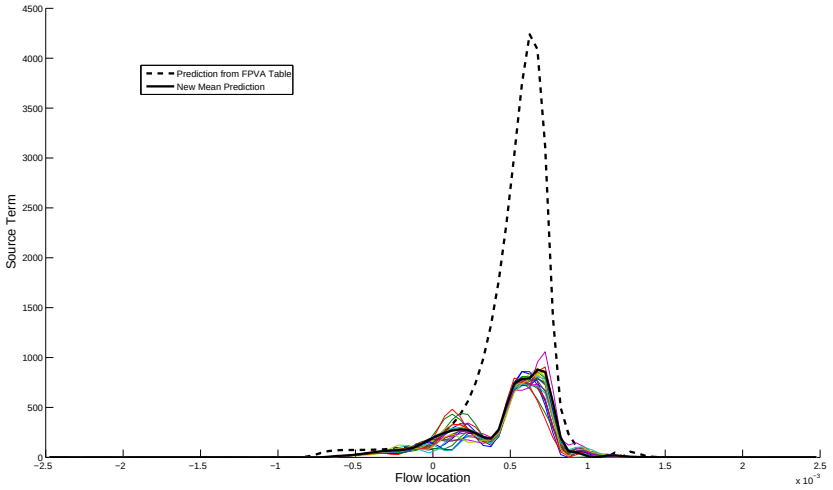


Figure 3: Sample predictions of the chemical source term in a flamelet approach.

Example 2: Turbulence anisotropy

For our second example problem, we examine a non-reacting turbulent flow over a curved surface in a channel. A major source of error in an eddy-viscosity model of turbulence is a result of the inability to account for the anisotropy in the Reynolds stresses. For our high-fidelity data, we use a DNS data set at $Re_\tau = 395$ computed by Marquille and Laval.[12] Our low-fidelity model is an *apriori* calculation of the Reynolds stresses (using the DNS mean-field and turbulent time-scales) using Menter's SST $k-\omega$ turbulence model.[13]

To visualize the anisotropy of the turbulence field, we use the *Barycentric map* proposed by Banerjee et al.[14] and used by Emory and Iaccarino.[4] This mapping represents the Reynolds stress tensor as a point within an equilateral triangle, the corners of which correspond to limiting states of anisotropy and the interior of the triangle represents a realizable Reynolds stress. This representation starts with the computation of the eigenvalues λ_i of the anisotropy tensor (with k being the turbulent kinetic energy)

$$a_{ij} = \frac{\overline{u_i' u_j'}}{2k} - \frac{1}{3}\delta_{ij} \tag{7}$$

These eigenvalues can then be used to construct the quantities

$$C_{1c} = \lambda_1 - \lambda_2 \tag{8}$$
$$C_{2c} = 2(\lambda_2 - \lambda_3) \tag{9}$$
$$C_{3c} = 3\lambda_3 + 1 \tag{10}$$

The subscripts $1c, 2c, 3c$ represent the one, two and three component limits[a] of the turbulence. Once these quantities are determined, the map can be visualized via the transformation

$$x_{bary} = C_{1c}x_{1c} + C_{2c}x_{2c} + C_{3c}x_{3c} \tag{11}$$
$$y_{bary} = C_{1c}y_{1c} + C_{2c}y_{2c} + C_{3c}y_{3c} \quad , \tag{12}$$

where x_{1c},y_{1c}, etc. are the locations of the corners of the barycentric triangle. Figure 4 shows the three corners of the triangle and a color map which will be used to identify locations in the barycentric map. Figures 5, 6 and 7 compare the anisotropy in the DNS solution to that of the RANS model. The DNS results contain a rich variation of anisotropy in the flow field, while the eddy-viscosity based RANS model (as expected in 2d incompressible flow) constrains the anisotropy along a line in the barycentric map.

[a]For instance, isotropic turbulence represents the three component limit and near-wall regions of boundary layers will exhibit strong one component tendencies

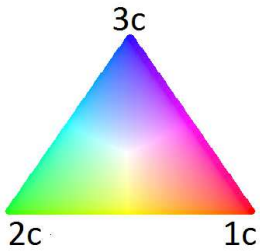


Figure 4: Color map representing location in barycentric map.

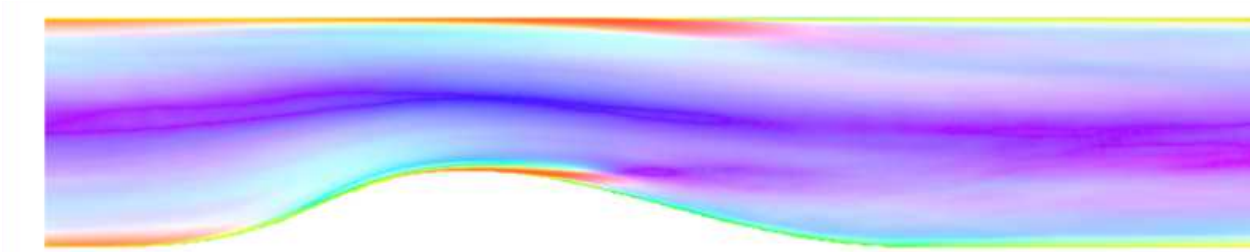


Figure 5: Anisotropy in the DNS flow field.

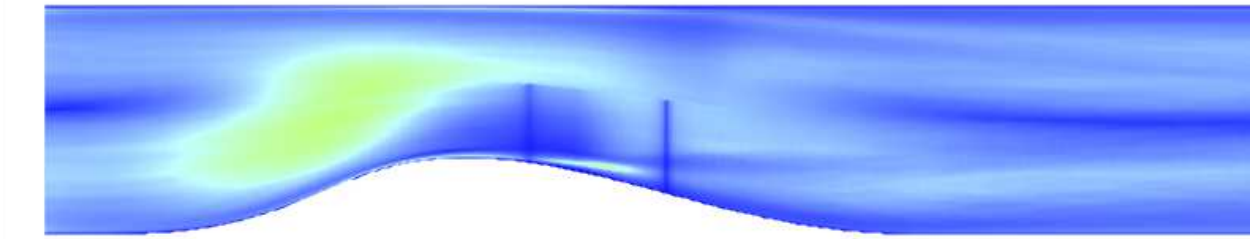
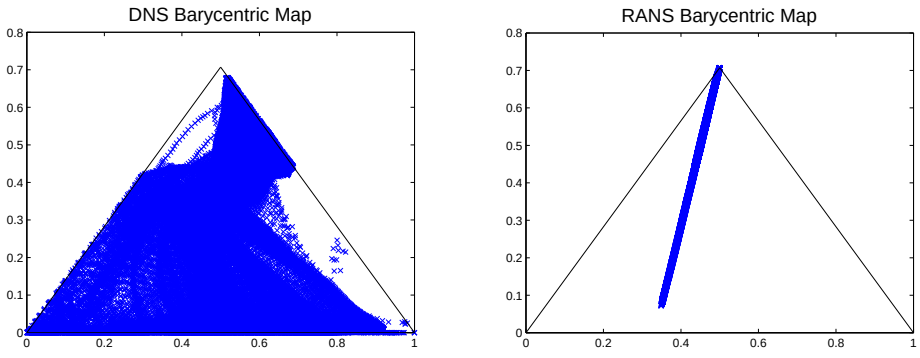


Figure 6: Anisotropy in the RANS flow field.



(a) DNS (b) RANS

Figure 7: Barycentric coordinates for entire flow field.

We again use extended kernel regression to learn the true anisotropy of the turbulent stresses given a set of flow features from the low-fidelity model. As input features, we choose four quantities: x_{bary}, y_{bary} predicted by the RANS, a so-called marker function m which provides a measure of the departure of the local flow condition from parallel shear flow (see Appendix B) and the ratio of production of turbulent kinetic energy to its dissipation. The desired output feature is the 'true' barycentric coordinate location. As in the previous example, we use a squared-exponential kernel, and learn the four bandwidth parameters using cross-validation. Unlike the first example, however, we have a two dimensional output and for convenience, we assume that the two output dimensions are uncorrelated. Specifically, when extended kernel regression is used, each training point in the data set is assigned a weight set by the single kernel, and given these weights, a prediction consisting of a mean vector and a diagonal covariance matrix is generated. Instead of using the entire DNS data set to train the model, we take a random sampling of 5000 flow locations with a limit on the near-wall points.

Given the DNS mean-field and training data at 5000 locations, the anisotropy predicted by our model is shown in Fig. 8. Substantial improvement over the SST model is noticeable. We can generate uncertainties in this new prediction by using the predicted variances. Choosing an appropriate model for the spatial correlation is, in general, difficult for turbulent flow. As a simplification, we instead sample from two normally-distributed random variables; the first is the number of standard deviations to shift each barycentric location in the x_{bary}-dimension, the second is the number of standard deviations to shift in the y_{bary}-dimension. The same shift, in terms of number of standard deviations, is applied to each location in the flow field, though the actual amount shifted depends on the local uncertainty. This shift can result in locations crossing the boundary of the barycentric triangle (thus representing unrealizable Reynolds stress anisotropies). In such cases, a projection to the nearest point on the boundary is used. Each specific shift creates a different barycentric map, and thus results in a family of anisotropy models with differing likelihoods. Two example draws from this distribution are shown in Fig. 9.

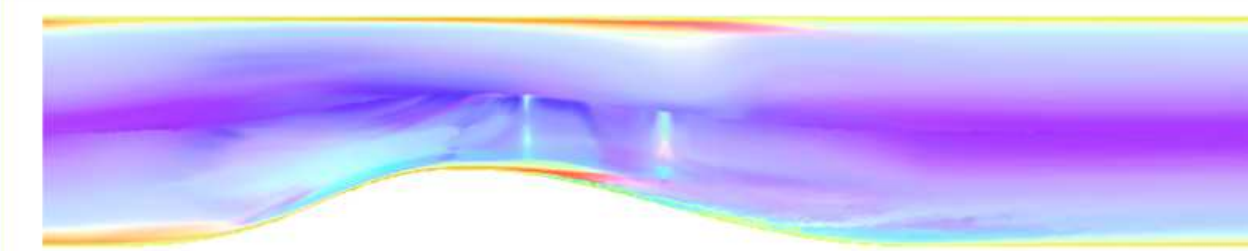


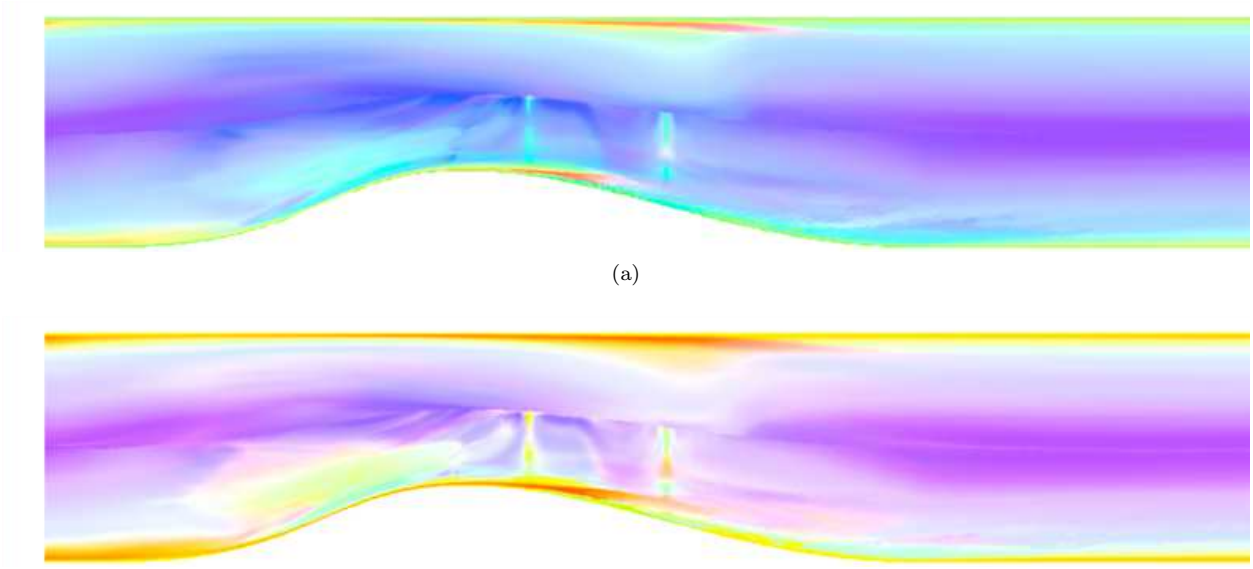
Figure 8: New mean prediction of anisotropy.

IV. Application in Simulations

Once the training model is used to build a probabilistic map, this information can be utilized in a flow simulation as schematized in Fig. 10. The ML process can, in principle, operate on multiple high/low fidelity data sets to build relationships between the input features from the low fidelity model X' to the true output feature Y. As explained in the previous section, our algorithm generates several *realizations* of the output feature (we refer to this as $\underline{Y}$) at every location in the flow, and these realizations (for instance, the barycentric coordinate) can be used to build the modeling term (for instance, the Reynolds stresses) under question. Thus, at every time-step (or iteration), the flow solver sends input features and receives modeling terms which it can use in the solution. Bounds for the prediction can be obtained by running an ensemble of these simulations and constructing an envelope of the outputs. Two examples using the maps generated in the previous section are detailed in this section.

Example 1: Reacting Mixing Layer simulation

Our first example problem is to predict the temporal evolution of a reacting turbulent mixing layer. The flow configuration is the same one used in the training model in the previous section, however, the training model operated on a few randomly selected points in the flow evolution. The details of the mixing streams are shown in Table 1. As a low-fidelity model, the present work adapts the version of the FPVA model developed



(a)

(b)

Figure 9: Sample realizations of predicted anisotropy.

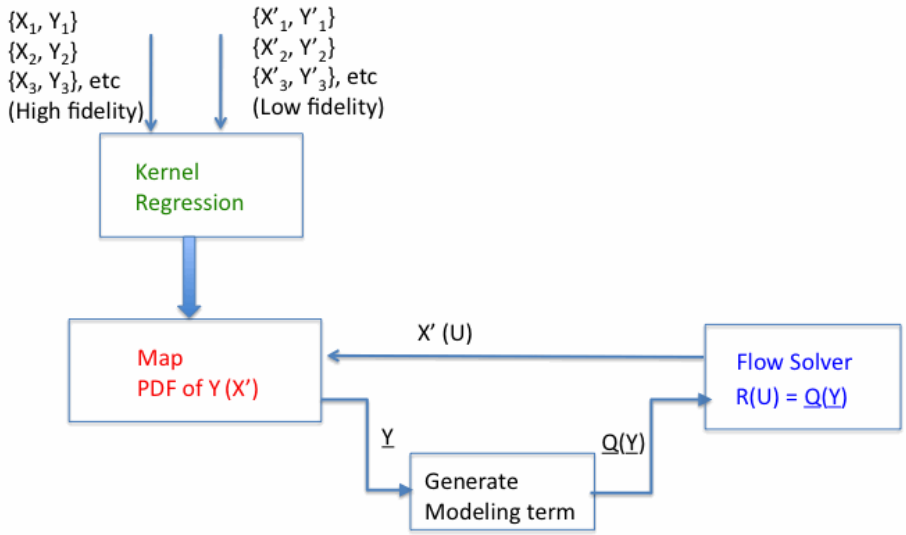


Figure 10: Schematic showing the construction of the machine learning map and coupling to a flow simulation. Note that the training is performed off-line.

by Terrapon et al.[15] In this approach, transport equations are solved for the Favre-averaged variables

$$\mathbf{Q} = (\bar{\rho}, \bar{\rho}\tilde{u}, \bar{\rho}\tilde{e}_t, \bar{\rho}k, \bar{\rho}\omega, \bar{\rho}\tilde{Z}, \bar{\rho}\tilde{Z}''^2, \bar{\rho}\tilde{C}) \tag{13}$$

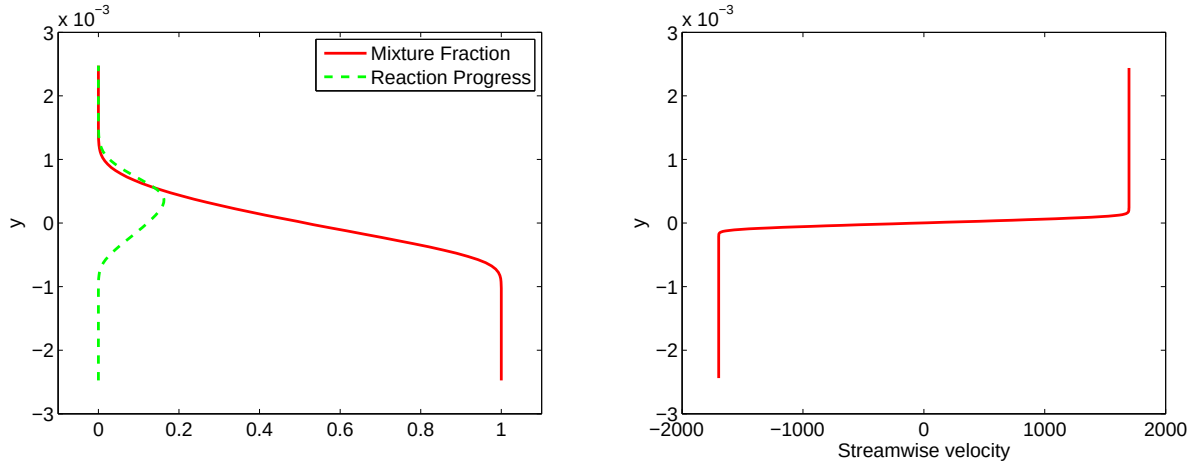
where $\bar{\rho}$, $\tilde{u}$ and $\tilde{e}_t$ are the density, velocity and total energy of the flow, k, ω represent the kinetic energy of the turbulence and its specific dissipation rate, and $\tilde{Z}, \tilde{Z}''^2, \tilde{C}$ represent the mean mixture fraction, variance of the mixture fraction and mean progress variable, respectively. The governing equations and state relationships are detailed in Appendix A. The chemistry look-up table is a function of $\tilde{Z}, \tilde{Z}''^2$ and $\tilde{C}$. The extent of the computational domain in the direction normal to the stream (y) is -0.0025 m to 0.0025 m and is discretized by 100 mesh cells.

Table 1: Properties of fuel and oxidizer streams in reacting mixing layer.

Stream	Temperature (K)	Pressure (N/m^2)	Velocity (m/s)	Y_{N_2}	Y_{O_2}	Y_{H_2}
Oxidizer	1500	2×10^5	1696.8	0.767	0.233	0.000
Fuel	500	2×10^5	-1696.8	0.933	0.000	0.067

The initial velocity distribution is taken to be $U(y) = \pm\frac{\Delta U}{2} tanh\left[\frac{y}{0.5\Delta w_0}\right]$ where $\Delta U = 2 \times 1696.8$ and $y \geq 0$ is the oxidizer stream and $y < 0$ is the fuel stream. The initial vorticity thickness is $\Delta w_0 = 2.38 \times 10^{-5} m$. In the DNS calculations involving finite-rate chemistry,[11] these initial conditions were propagated to a time $t_s = 0.21ms$, at which time turbulent perturbations were superimposed on the flow. The turbulence is allowed to develop in a DNS simulation until a time $t_e = 0.223ms$. As a starting point for the RANS calculations, the DNS solution corresponding to $t_i = 0.214ms$ is used as an initial condition. These conditions are shown in Fig. 11.

To use the ML model, the input features are chosen to be $\tilde{Z}, \tilde{Z}''^2$ and $\tilde{C}$ and the output feature is the source term of the progress variable equation as in the previous section, which is relayed to the flow solver at the beginning of every simulation time-step. At $t = t_e$, Fig. 12 shows the source term interpolated from the FPVA table and two realizations generated by the learning model. The predicted mass fraction of H_2O at the final simulation time $t = t_e$ is shown in Fig. 13. Despite the simplicity of our approach, the additional information offered by the extended kernel regression improves the quality of the FPVA solution. At the present time, only the source term of the H_2O mass fraction equation has been supplemented with the ML information, and a detailed set of thermo-chemical output features could be constructed to further improve the low-fidelity model. Furthermore, a more comprehensive description of the joint distributions of the looked-up quantities should cause a more accurate description of the error in the FPVA solution.



(a) Mean Mixture Fraction and Progress variable

(b) Mean Streamwise velocity

Figure 11: Initial conditions for reacting mixing layer simulation.

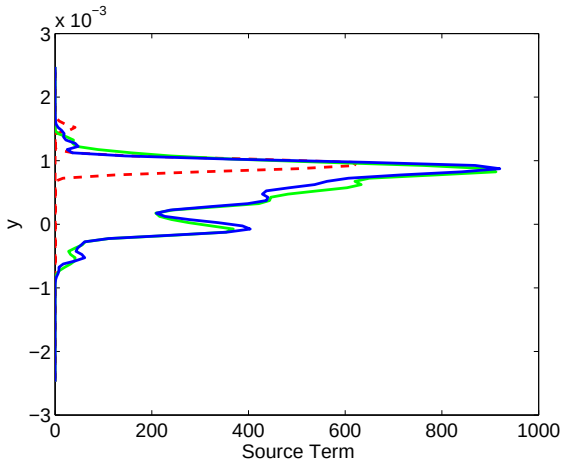


Figure 12: Source term at $t = t_e = 0.223ms$. Dashed line: FPVA table prediction; Solid lines: Two different realizations of ML prediction.

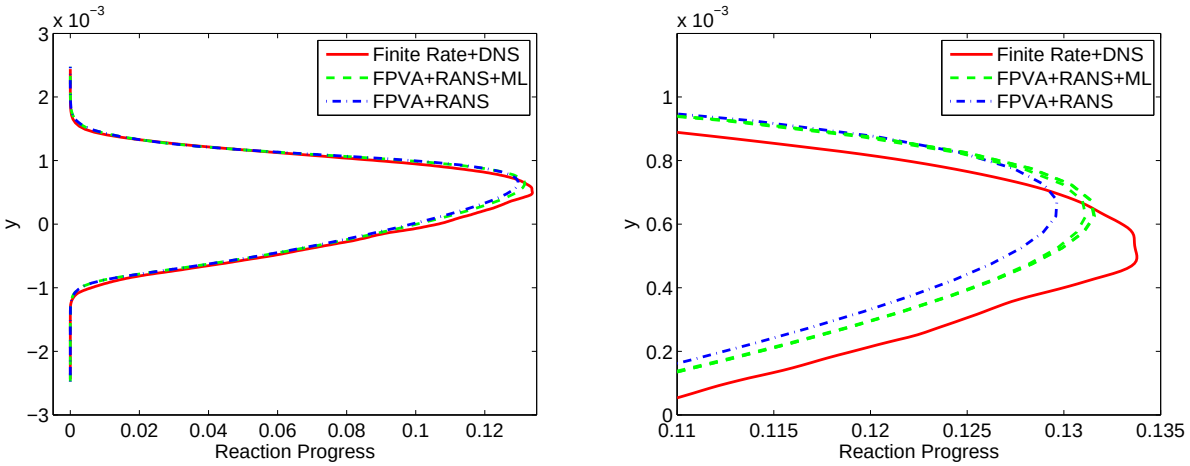


Figure 13: Predicted H_2O mass fraction at final time instant.

Example 2: Non-equilibrium Boundary Layer simulation

In the previous section, the DNS data set for the separated flow over a converging-diverging channel (CDC) was used to learn the relationship between the 'real' turbulence anisotropy as a function of the anisotropy predicted by the RANS model and two other input features. In the present exercise, the applicability of the probabilistic map is explored in two *different* channel flow configurations simulated using Menter's SST model.[13] The training data set is still obtained from the CDC case and applied here directly.

The computational domain is shown in Fig. 14 and corresponds to the geometry used in the experimental measurements of Webster, Degraaf and Eaton.[16] All length scales are non-dimensionalized by the chord of the bump. The top and bottom boundaries are treated as viscous walls. The left boundary is a turbulent BL with specified properties, while the right boundary is a subsonic characteristic outflow. We consider two simulations, corresponding to inlet boundary layers at $Re_\theta = 4000$ and $Re_\theta = 12170$. We compare the outputs of our model to experimental measurements of Webster, Degraaf and Eaton[16] for the low Re_θ case and to the wall-modeled LES (WMLES) computed by Radhakrishnan and Piomelli[17] for the high Re_θ case.

As observed from Figs. 5 and 6, the RANS model can be expected to be especially poor in predicting the one component nature of the anisotropy. This is confirmed in Fig. 16, in which the C1c component of the anisotropy tensor is shown along the $x/c = 0.5$ line. The mean of the ML-corrected results shows a better prediction[b] when compared to the post-processed WMLES results. A generally more improved prediction of the anisotropy is also suggested by Fig. 16 which shows the barycentric coordinates in the entire domain.

The ML model is then coupled to the flow-solver in an iterative fashion. At the beginning of each flow solver iteration, four features $(x_{bary}, y_{bary}, m, P/\epsilon)$ are used to obtain realizations of $\underline{x}_{bary}, \underline{y}_{bary}$ which can be used to obtain new eigenvalues $\underline{\lambda}_i$ from an inverse of eqns. 8-12. The corrected eigenvalues are injected into the Reynolds stresses in a fashion similar to that used by Emory et al.[4]

$$\overline{u_i' u_j'} = \frac{2}{3} k \delta_{ij} + k(v_i \underline{\lambda}_i v_j) \tag{14}$$

where k is predicted by the baseline SST model and v is the right eigenvector of the anisotropy tensor. Thus, the corrected barycentric coordinates are used to merely *redistribute* the fluctuation energy among the various components of the Reynolds stress tensor. Fig. 17 shows the skin friction coefficient predicted by the SST model and the SST model supplemented with the ML corrections. The ML corrections result in a lower value of the skin friction over the curved wall. The ML model is, as expected, not very accurate because the model that has been used in this study *only accounts for errors introduced by the inability of the model to predict the correct structure of the anisotropy tensor*. There still remain various aspects of the turbulence model which may result in additional errors. For instance, in the present work, the redistributed Reynolds stresses are not used in computing the production of turbulent kinetic energy, a term that is important in flows with streamline curvature and one that eddy viscosity models routinely mis-predict in the absence of empirically added sensitization.[18] Also, only one set of high and low fidelity data has been used to create the training model. Nevertheless, even based on this very limited information, the departures of the ML corrections from the baseline predictions can serve as an *indicator* of uncertainty. This exercise should be considered as a proof-of-concept that the methodology can be applied in real problem.

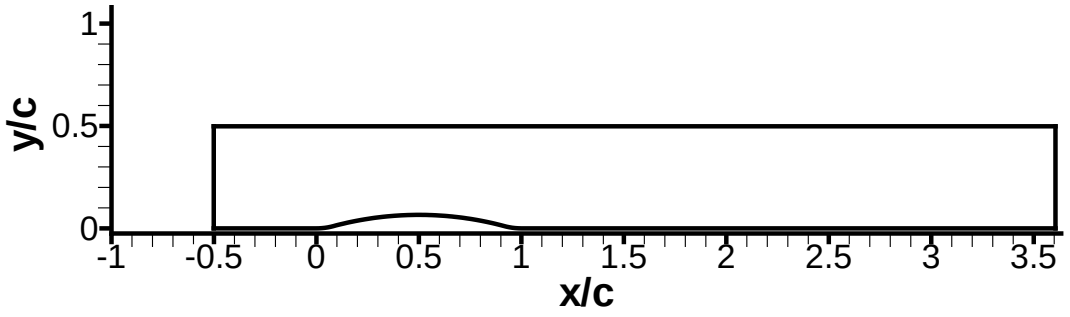


Figure 14: Computational Domain.

[b]The disagreement between the WMLES and the ML prediction is pronounced only in the potential part of the flow.

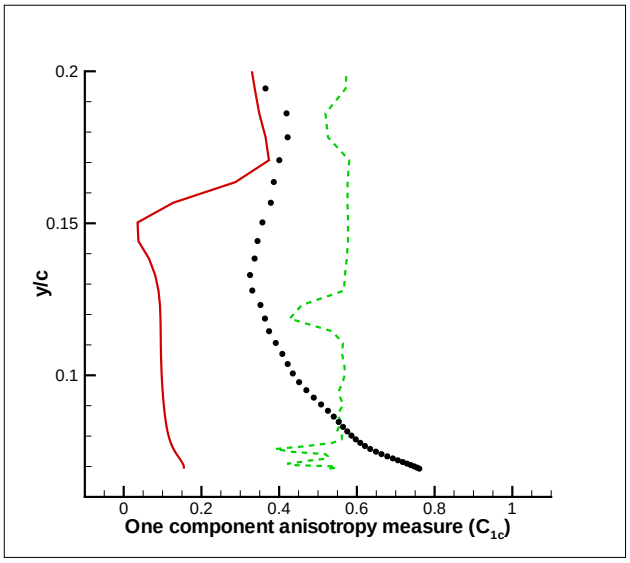
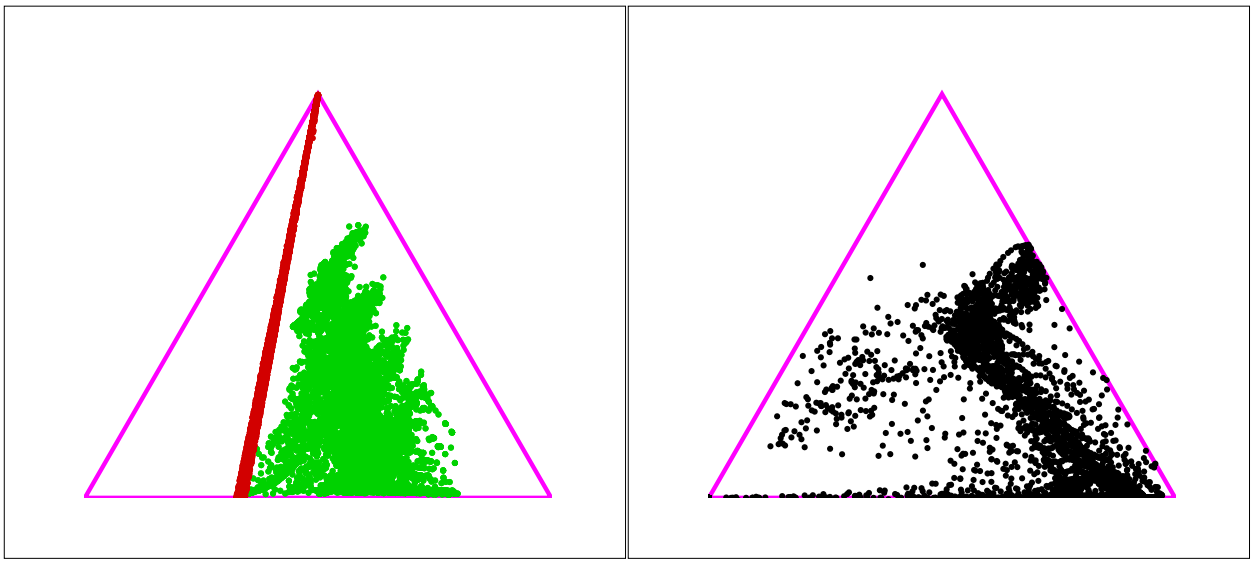


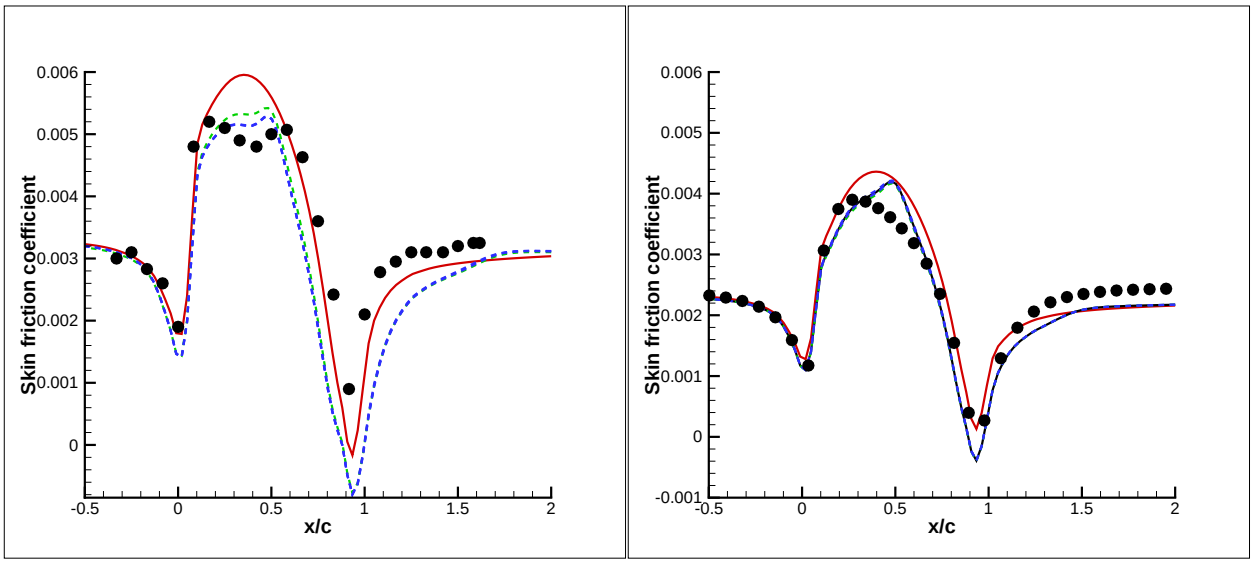
Figure 15: Barycentric coordinate at $x = 0.5c$. Dots: WMLES; Solid line: SST; Dashed line: SST+ML model (Different realizations are shown).



(a) SST: Red; SST+ML: Green

(b) WMLES

Figure 16: Barycentric map in the entire domain showing anisotropy of Reynolds stress tensor.



(a) $Re_\theta = 4000$ (b) $Re_\theta = 12170$

Figure 17: Predicted Skin friction coefficient. Dots: Measured ($Re_\theta = 4000$), WMLES ($Re_\theta = 12170$); Solid line: SST model; Dashed lines: SST+ML model.

V. Perspectives and Conclusions

We have presented a machine learning-based technique to improve low-fidelity models and obtain error bounds by using information from related high-fidelity data sets. We introduced a *new supervised learning technique* which extends kernel regression to create a local error model. Given data from a more accurate model, we create a probabilistic mapping between a set of input features from the low-fidelity model and a set of desired output features in the high-fidelity model. Our technique uses a kernel function to weight the relevance of each training point, and using these weights, sets a Gaussian probability on the set of output features. We then combine these local errors with a global correlation model, and sample the resulting stochastic model to obtain error bars on the solution output. We apply this methodology to two illustrative problems in combustion and turbulence. In the combustion example, we use extended kernel regression to replace the thermo-chemical source terms in a flamelet model by a model that is informed by data from a finite-rate chemistry numerical simulation. We also learn a global correlation model that smooths out the spatial variability in the source term. Applying this model, we see a better prediction of the source term compared with the standard flamelet model. The second example problem uses a DNS simulation to learn the anisotropy of the Reynolds stress tensor as a function of several relevant mean flow parameters. We apply this model to two different channel flow problems, and demonstrate an improved prediction of the turbulence anisotropy.

It is important to mention that this paper presents a proof-of-concept on how machine learning *could* be used in uncertainty quantification. A simplistic version of the methodology (in terms of number of flow-features, training data-sets, spatial correlations, etc) has been implemented, and, despite this simplicity, encouraging initial results have been obtained. Much improvement is required before this technique can be used routinely in model correction/error quantification, especially in the latter aspect. An underlying assumption of universality is presumed as model errors are related (in a stochastic fashion) to a specific set of input flow features, whereas in reality, the physics could be different across problems. However, the authors do not consider this to be a serious limitation as the original models themselves are built and calibrated under specific (and often controlled) conditions and thus assume some form of universality. In addition, if the training model can be developed in regimes that are relevant to the problem of interest, the accuracy of the correction/error models can be improved. At the very least, this technique can be used as an *indicator* of error in the base model.

There are a number of potential directions of moving beyond the simple implementations in this paper.

First and foremost is the choice of input and output features, where engineering insight should be a good initial guide. There also exist specific machine learning techniques for feature selection, such as Principal Component Analysis,[19] which can help down-select from a number of relevant features. Another important improvement will be to consider error models that rely on distributions other than Gaussians. While Gaussians have a number of desirable mathematical properties, they often underestimate the probability of rare events and can thus underestimate the true uncertainties. Replacing Gaussian uncertainties with, for instance, Laplace or Gamma distributions may be appropriate in many situations. Furthermore, much investigation remains in creating appropriate global error models. In this work, we used two very different, but simplistic global error models. The first assumed that correlations could be represented by correlated Gaussian random variables, while the second assumed that all locations in the field were equally biased. These assumptions can be relaxed by segmenting the flow into different regions (such as laminar/transitional/turbulent regions or attached/separating boundary layers, etc) and adopting different correlations in each region of the flow. Nevertheless, the initial results in this paper are sufficiently encouraging to merit future exploration and development in the use of machine learning for epistemic uncertainty quantification.

Acknowledgment

This work has been supported in part by the Predictive Science Academic Alliance Program (PSAAP) at Stanford University funded by the U.S. Department of Energy. The authors are grateful to Amirreza Saghafian for providing the extensive data sets used in the combustion simulations and his help in setting up the RANS simulations.

References

[1]Cheung, S. H., Oliver, T. A., Prudencio, E. E., Prudhomme, S., and Moser, R. D., "Bayesian Uncertainty Analysis with Applications to Turbulence Modeling," *Reliability Engineering and System Safety*, Vol. 96, 2011, pp. 1137–1149.

[2]Oliver, T. and Moser, R., "Bayesian uncertainty quantification applied to RANS turbulence models," *Journal of Physics: Conference Series*, Vol. 318, IOP Publishing, 2011, p. 042032.

[3]Dow, E. and Wang, Q., "Uncertainty Quantification of Structural Uncertainties in RANS Simulations of Complex Flows," *AIAA Paper*, Vol. 3865, 2011.

[4]Emory, M., Pecnik, R., and Iaccarino, G., "Modeling Structural Uncertainties in Reynolds-Averaged Computations of Shock/Boundary Layer Interactions," *AIAA Paper*, Vol. 479, 2011.

[5]Peters, N., "Laminar flamelet concepts in turbulent combustion," *Symposium (International) on Combustion*, Vol. 21, Elsevier, 1988, pp. 1231–1250.

[6]Reagan, M., Najm, H., Debusschere, B., Le Maitre, O., Knio, O., and Ghanem, R., "Spectral stochastic uncertainty quantification in chemical systems," *Combustion Theory and Modelling*, Vol. 8, No. 3, 2004, pp. 607–632.

[7]Rasmussen, C. E. and Williams, C. K. I., *Gaussian Processes for Machine Learning*, The MIT Press, 2006.

[8]Watson, G. S., "Smooth Regression Analysis," *The Indian Journal of Statistics*, 1964.

[9]Cleveland, W. S., Devlin, S. J., and Grosse, E., "Variance Reduction Methods I," *Monte Carlo Simulation: IEOR E4703*, 2004.

[10]Duda, R. O., Hart, P. E., and Stork, D. G., *Pattern Classification*, Wiley, New York, 2nd ed., 2001.

[11]Saghafian, A. and Pitsch, H., "Direct numerical simulation of supersonic combustion with finite-rate chemistry," *64th Annual Meeting of the APS Division of Fluid Dynamics*, Vol. 56, 2011.

[12]Marquillie, M., Laval, J.-P., and Dolganov, R., "Direct Numerical Simulation of a separated channel flow with a smooth profile," *Journal of Turbulence*, Vol. 9, 2008, pp. 1–23.

[13]Menter, F., "Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications," *AIAA Journal*, Vol. 32, No. 8, 1994.

[14]Banerjee, S., Ertunc, O., Durst, F., Kartalopoulos, S., Buikis, A., Mastorakis, N., and Vladareanu, L., "Anisotropy properties of turbulence," *WSEAS International Conference. Proceedings. Mathematics and Computers in Science and Engineering*, No. 13, WSEAS, 2008.

[15]Terrapon, V., Ham, F., Pecnik, R., and Pitsch, H., "A flamelet-based model for supersonic combustion," *Annual Research Briefs*, 2009, pp. 47–58.

[16]Webster, D., DeGraaff, D., and Eaton, J., "Turbulence characteristics of a boundary layer over a two-dimensional bump," *Journal of Fluid Mechanics*, Vol. 320, No. 1, 1996, pp. 53–69.

[17]Radhakrishnan, S., Keating, A., Piomelli, U., and Silva Lopes, A., "Reynolds-averaged and Large-eddy simulations of turbulent non-equilibrium flows," *Journal of Turbulence*, Vol. 7, 2006.

[18]Duraisamy, K. and Iaccarino, G., "Curvature Correction and Application of the $v^2 - f$ Turbulence Model to Tip Vortex Flows," *Annual Research Briefs of the Center for Turbulence Research, Stanford University*, 2005.

[19]Abdi, H. and Williams, L., "Principal component analysis," *Wiley Interdisciplinary Reviews: Computational Statistics*, Vol. 2, 2010, pp. 433–459.

[20]Emory, M., Larsson, J., and Iaccarino, G., "Manuscript under preparation," 2012.

Appendix A: Flamelet equations

The governing equations are the Favre-averaged equations for mass, momentum and energy along with a Reynolds analogy for turbulent heat flux and a Boussinesq approximation for the turbulent transport. These equations are supplemented by:

$$\begin{aligned}
\frac{\partial}{\partial t}(\bar{\rho}\tilde{Z}) + \frac{\partial}{\partial x_j}(\bar{\rho}\tilde{u}\tilde{Z}) &= \frac{\partial}{\partial x_j}\left[\left(\frac{\lambda}{c_p} + \frac{\mu_t}{Sc_z}\right)\frac{\partial \tilde{Z}}{\partial x_j}\right] \\
\frac{\partial}{\partial t}(\bar{\rho}\tilde{Z''^2}) + \frac{\partial}{\partial x_j}(\bar{\rho}\tilde{u}\tilde{Z''^2}) &= \frac{\partial}{\partial x_j}\left[\left(\frac{\lambda}{c_p} + \frac{\mu_t}{Sc_{z''^2}}\right)\frac{\partial \tilde{Z''^2}}{\partial x_j}\right] + 2\frac{\mu_t}{Sc_{z''^2}}\frac{\partial \tilde{Z}}{\partial x_j}\frac{\partial \tilde{Z}}{\partial x_j} - c_\chi \bar{\rho}\tilde{\chi}'' \\
\frac{\partial}{\partial t}(\bar{\rho}\tilde{C}) + \frac{\partial}{\partial x_j}(\bar{\rho}\tilde{u}\tilde{C}) &= \frac{\partial}{\partial x_j}\left[\left(\frac{\lambda}{c_p} + \frac{\mu_t}{Sc_c}\right)\frac{\partial \tilde{C}}{\partial x_j}\right] + \tilde{\dot{\omega}}_C
\end{aligned}$$

where μ_t and $Sc_{()}$ represent the turbulent eddy viscosity and Schmidt number, respectively. Note that a unity Lewis number assumption has been used. The turbulent kinetic energy k and eddy viscosity μ_t are calculated using Menter's Shear Stress Transport[13] model. For closure, the equations of state, material properties and the source term $\tilde{\dot{\omega}}_C$ are derived from quantities interpolated from pre-computed tables as a function of $\tilde{Z}, \tilde{Z''^2}, \tilde{C}$.

The equations of state are

$$\hat{T} = \hat{T}_0 + \frac{\tilde{\gamma}_0 - 1}{a_\gamma}\left(e^{a_\gamma(\tilde{e}-\tilde{e}_0)/\tilde{R}_0} - 1\right) \tag{15}$$

$$\tilde{e} = \tilde{e}_0 + \frac{\tilde{R}_0}{a_\gamma}\ln\left(1 + \frac{a_\gamma(\hat{T}-\hat{T}_0)}{\tilde{\gamma}_0 - 1}\right) \tag{16}$$

$$\tilde{h} = \tilde{e} + \tilde{R}_0\hat{T} \tag{17}$$

$$\bar{p} = \bar{\rho}\tilde{R}_0\hat{T} \tag{18}$$

$$\frac{\lambda}{c_p} = \left(\frac{\lambda}{c_p}\right)_0\left[\frac{\hat{T}}{\hat{T}_0}\right]^{0.62} \tag{19}$$

$$\mu = \mu_0\left[\frac{\hat{T}}{\hat{T}_0}\right]^{0.7} \tag{20}$$

$$\gamma = \tilde{\gamma}_0 + a_\gamma(\hat{T}-\hat{T}_0) \tag{21}$$

$$a = \sqrt{\gamma\tilde{R}_0\hat{T}} \tag{22}$$

In the above equations, the variables $\hat{T}_0, \tilde{\gamma}_0, \tilde{e}_0, \tilde{R}_0, a_\gamma, \tilde{\gamma}_0, \left(\frac{\lambda}{c_p}\right)_0, \mu_0$ are pre-computed using flamelet libraries and tabulated as functions of $\tilde{Z}, \tilde{Z''^2}, \tilde{C}$ and interpolated when required. Note that the flamelet library is pre-computed for a specific reference pressure p_{ref}. To sensitize the model to changes in pressure, the the source term in the progress variable reaction is scaled as $\tilde{\dot{\omega}}_C = \tilde{\dot{\omega}}_{0C}\frac{\bar{p}^2}{p_{ref}^2}$. Finally, the scalar dissipation rate term $\tilde{\chi}''$ is approximated as 0.18ω.

Appendix B: Definition of marker function m

The "marker function" m was introduced by Emory et al.[20] to describe regions of the flow which deviate from parallel shear flow. This can be used as a proxy to indicate regions in which RANS models (particularly eddy viscosity models) are likely to be inaccurate. Given the Reynolds averaged velocity vector field u_i, the gradient of the component of the velocity vector aligned with the local streamline is given by

$$g_j = \frac{u_i}{\sqrt{u_k u_k}}\frac{\partial u_i}{\partial x_j}. \tag{23}$$

The cosine of the angle between this gradient and the streamline direction is then

$$f = \frac{|g_j s_j|}{\sqrt{g_k g_k}}. \tag{24}$$

The marker is defined as a composite of f along with an indicator of the local importance of turbulence, such that far-field and potential flows may be ignored. The final expression for the marker is then

$$m = \frac{k}{\sqrt{u_k u_k}} f, \tag{25}$$

where k is the turbulent kinetic energy.