SAND2022-16492C

Uncertainty Propagation of the Negative Spalart–Allmaras Turbulence Model Coefficients using Projection-based Reduced-Order Models

Elizabeth H. Krath[*] and Patrick J. Blonigan[†] and Eric J. Parish[‡]
Sandia National Laboratories, Livermore, CA, 94550, USA

This paper presents the uncertainty propagation of turbulent coefficients for the Spalart–Allmaras (SA) turbulence model using projection-based reduced-order models (ROMs). ROMs are used instead of Reynolds-averaged Navier–Stokes (RANS) solvers and stochastic collocation/Galerkin and Monte Carlo methods because they are computationally inexpensive and tend to offer more accuracy than a polynomial surrogate. The uncertainty propagation is performed on two benchmark RANS cases documented on NASA's turbulence modeling resource. Uncertainty propagation of the SA turbulent coefficients using a ROM is shown to compare well against uncertainty propagation performed using only RANS and using a Gaussian process regression (GP) model. The ROM is shown to be more robust to the size and spread of the training data compared to a GP model.

I. Nomenclature

Ω	=	Spatial domain
$\mathcal{D}$	=	Parameter domain
N	=	Number of volumes in spatial domain, *i.e.*, number of cells
p	=	Number of modes
C_f	=	wall coefficient of friction
$\boldsymbol{\mu}$	=	vector of system parameters
$\boldsymbol{U}$	=	Favre-averaged state vector
$\boldsymbol{q}$	=	Discretized state vector
$\boldsymbol{r}$	=	Discretized residual
$\boldsymbol{\Phi}$	=	Reduced basis
$\hat{\mathbf{q}}$	=	Reduced coordinates
$\mathbf{q}_{\text{ref}}$	=	Reference state
SA	=	Spalart-Allmaras
FOM	=	Full-order model
ROM	=	Projection-based reduced-order model
GP	=	Gaussian process regression
LHS	=	Latin hypercube sampling

II. Introduction

As computing power has increased, so too has the complexity of computational fluid dynamics (CFD) applications. However, while methods such as direct numerical simulation (DNS) and large eddy simulation (LES) have allowed for highly accurate and resolved simulations, the applications for these methods are constrained by their high computational expense [1]. For this reason, Reynolds-averaged Navier–Stokes (RANS) solvers are used along with turbulence closures that model the Reynolds stress. These closure models are typically constructed via a combination of physical expertise and empirical arguments, and are calibrated to provide accurate predictions for a class of canonical

[*] Postdoctoral Appointee, Thermal/Fluid Science and Engineering, MS 9042, AIAA Member.

[†] R&D S&E, Extreme-Scale Data Science and Analytics, MS 9159, AIAA Member.

[‡] R&D S&E, Extreme-Scale Data Science and Analytics, MS 9159

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

flows. As an example, one widely used turbulence closure model is the Spalart–Allmaras (SA) model. The SA model evolves a transport equation for the turbulent eddy viscosity and contains numerous model coefficients. The standard values for these coefficients are obtained by calibrating the model to canonical flows, but it is well appreciated that the coefficients may have varying ranges of optimal values depending on the problem [2].

Regardless of what closure is employed, quantifying the impact of parametric uncertainties in RANS models is imperative for simulation-based design and engineering. Various approaches have been employed for quantifying parametric uncertainties in RANS models. Borggard and collaborators, for instance, performed sensitivity analysis of the k–ϵ model via simulation of the forward sensitivity equations [3, 4]. Platteeuw et al. model uncertainty in solutions to the k–ϵ turbulence model due to uncertainties in its coefficients with a Probabilistic Collocation Method [5] and Margheri et al. propagate similar uncertainties in k–ϵ and k–ω models via generalized Polynomial Chaos (gPC); these processes involve modeling the solution as a random variable via a polynomial chaos expansion and employing projection to form a well-posed system. More directly, Dunn et al. [6] investigate parametric sensitivities for the k–ϵ model via Latin Hypercube Sampling (LHS). In addition to forward propagation, numerous bodies of work have performed “backward" approaches that attempt to infer an optimal parameter distribution [7]. In these cases, it is common to employ low-cost surrogate models to accelerate the simulation of the forward problem. For instance, Ray et al. employ support vector machines to learn an optimal eddy viscosity model for a jet-in-crossflow [8]. Ray et al. employed similar techniques to calibrate k–ϵ model constants for a jet-in-crossflow [9, 10] and SST (Shear Stress Transport) model constants and inflow conditions for HIFiRE-1 simulations. Various other turbulence model calibration efforts efforts exist in the literature [11, 12] and we refer to the review by Xiao and Cinnella [13] for a more comprehensive overview.

In this work we investigate the use of projection-based reduced-order models (ROMs) for quantifying parametric uncertainties in RANS models. Projection-based ROMs are weighted residual methods that consist of a computationally intensive offline training phase and a computationally cheap online evaluation phase. In the offline phase a reduced trial space is generated by executing the full-order model for various parameter instances and collecting solution data comprising “state snapshots". The reduced trial space is then identified, e.g., via the reduced-basis method [14–17] or the proper orthogonal decomposition method [18]. In the online phase, an approximate solution to the governing equations for a novel parameter instances is then sought within this reduced trial space via a (Petrov-)Galerkin projection or residual minimization. The result of this process is a ROM that can be executed much faster than its corresponding full-order model (FOM).

In this contribution we investigate accelerating UQ analyses of uncertain coefficients in the SA RANS model by (1) building a ROM from a small subset of FOM simulations obtained from LHS of a parameter space and (2) employing the ROM to generate approximate solutions for a much larger set of Latin hypercube samples. We investigate ROMs constructed via Galerkin projection with entropy variables and least-squares residual minimization, and results are presented for two benchmark problems from the NASA turbulence modeling resource. While the ROM methods employed in this work are only investigated within the context of accelerating LHS for forward propagation of uncertainties they could be used in numerous other ways to support the above literature. For instance, they could be employed to reduce the high computational burden associated with gPC methods by reducing the spatial dimension of the FOM [19], or they could be employed as a high-quality surrogate to accelerate parameter calibration similar to Ref [8].

This manuscript proceeds as follows. Section III describes the Favre–Averaged Navier–Stokes equations with the Spalart–Allmaras turbulence model along with their discretized form employed by the full-order model. Section IV formulates the reduced-order models employed in this work, while Section V provides numerical results. Conclusions are given in Section VI.

III. Full-Order Model

We consider solutions to the Favre-averaged Navier–Stokes equations closed with the Spalart–Allmaras–neg (SA-neg) turbulence model [20] defined on the spatial domain $\Omega \subset \mathbb{R}^3$. The system of equations can be written in conservative form as

$$\nabla \cdot \boldsymbol{F}(\boldsymbol{U}) - \nabla \cdot \boldsymbol{F}^{v}(\boldsymbol{U}, \nabla \boldsymbol{U}; \boldsymbol{\mu}) = \boldsymbol{S}(\boldsymbol{U}, \nabla \boldsymbol{U}; \boldsymbol{\mu}) \tag{1}$$

where $\boldsymbol{U} = \left[\overline{\rho},\ \overline{\rho}\tilde{\boldsymbol{u}},\ \overline{\rho}\tilde{E},\ \overline{\rho}\tilde{\nu}\right]^T : \Omega \to \mathbb{R}^6$ is the state comprising the Favre-averaged density, momentum, total energy, and SA-neg viscosity, $\boldsymbol{F}(\cdot) \in \mathbb{R}^6$ is the inviscid flux, $\boldsymbol{F}^{v}(\cdot,\cdot;\cdot) \in \mathbb{R}^6$ is the viscous flux, $\boldsymbol{S}(\cdot,\cdot;\cdot)$ is a source term, and for $\mathcal{D}$ representing the parameter domain then $\boldsymbol{\mu} \in \mathcal{D} \subset \mathbb{R}^{10}$ are parameters defining the SA-neg turbulence model. In particular there are ten model parameters: $\boldsymbol{\mu} = \left[\sigma, c_{w_1}, c_{w_2}, c_{w_3}, c_{b_1}, c_{b_2}, c_{v_1}, \kappa, c_{t_3}, c_{t_4}\right]^T$. The full definition of the inviscid fluxes, viscous fluxes, and source terms is given in Appendix A.

The governing equations are discretized with a cell-centered finite volume method. This discretization can be described by partitioning the domain into N non-overlapping volumes Ω_j such that $\cup\{\Omega_j\}_{j=1}^N = \Omega$. The finite volume method then integrates over each cell volume and, integrating the viscous and inviscid fluxes by parts, results in the system in each cell

$$\oint_{\Gamma_j} \left(\boldsymbol{F}\left(\boldsymbol{U}\right) - \boldsymbol{F}^v\left(\boldsymbol{U}, \nabla \boldsymbol{U}; \boldsymbol{\mu}\right)\right) \cdot \boldsymbol{n}_j dS = \int_{\Omega_j} \boldsymbol{S}\left(\boldsymbol{U}, \nabla \boldsymbol{U}; \boldsymbol{\mu}\right) dV, \tag{2}$$

where Γ_j is the boundary of the jth cell and $\boldsymbol{n}_j$ is the cell normal vector. A discrete system of equations is then obtained by discretizing the state $\boldsymbol{U}$ at each cell center, replacing the inviscid and viscous fluxes with their numerical counterparts (e.g., the Roe flux), and devising a scheme to compute the gradient of the state. All in all, the finite-volume method results in a $6N$-dimensional system of nonlinear algebraic equations

$$\boldsymbol{r}(\mathbf{q}(\boldsymbol{\mu}); \boldsymbol{\mu}) = \mathbf{0} \tag{3}$$

where $\boldsymbol{r} : \mathbb{R}^{6N} \times \mathcal{D} \to \mathbb{R}^{6N}$ is the discretized residual, and $\mathbf{q} : \mathcal{D} \to \mathbb{R}^{6N}$ is the discretized state at each cell center, i.e.,

$$\mathbf{q} = \begin{bmatrix} \overline{\rho}_1 & \overline{\rho}\widetilde{\boldsymbol{u}}_1 & \overline{\rho}\tilde{E}_1 & \overline{\rho}\tilde{\nu}_1 & \cdots & \overline{\rho}_N & \cdots & \overline{\rho}\tilde{\nu}_N \end{bmatrix}^T$$

with $\overline{\rho}_i$, $i = 1, \ldots, N$ referring to density at the ith cell center and the same for the other quantities. We note that the discrete system (3) has the same units as (2), i.e., at no point do we divide through by the cell volumes. We additionally note that the discrete system (3) can be interpreted as a weighted residual solution to the system (1) with a piecewise constant trial space described by the basis $\mathbf{\Phi}_{\text{fv}} = \mathbf{I} \in \mathbb{R}^{6N \times 6N}$.

In the present work the Sandia Parallel Aerodynamics and Reentry Code (SPARC) [21] is employed to discretize and solve the system (1). A second-order Steger–Warming flux scheme is used for the inviscid fluxes and a least-squares scheme is used for the viscous fluxes. We employ min-mod as a limiter. All cases are run in a non-dimensional configuration such that the free-stream speed of sound is unity. In all cases, the heat capacity ratio is $\gamma = 1.4$, the Prandtl number is $\text{Pr} = 0.72$, and the turbulent Prandtl number is $\text{Pr}_{\text{t}} = 0.9$.

IV. Reduced-Order Models

In the present work, we focus on propagating uncertainties in the parameter vector $\boldsymbol{\mu}$ to solutions $\boldsymbol{q}(\boldsymbol{\mu})$. The most straightforward approach for quantifying this uncertainty is Monte Carlo sampling where the FOM (3) is repeatedly solved for different parameter samples. Repeatedly solving the FOM, unfortunately, is a computationally intensive process. Instead, we investigate the utility of projection-based reduced-order models (ROM) as a low-cost but high-fidelity approximate model. The ROMs in this work employ the reduced basis method for the state approximation and Galerkin and least-squares residual minimization for projection.

A. State approximation via the reduced basis method

Projection-based reduced-order models restrict the state to belong within a low-dimensional subspace. Mathematically, this is described by writing an approximation to the state as

$$\boldsymbol{q}(\boldsymbol{\mu}) \approx \mathbf{\Phi}\hat{\mathbf{q}}(\boldsymbol{\mu}) + \mathbf{q}_{\text{ref}} \tag{4}$$

where for p being the number of modes then $\mathbf{\Phi} \in \mathbb{R}^{6N \times p}$ with $p \ll 6N$ is a reduced basis, $\hat{\mathbf{q}} : \mathcal{D} \to p$ are reduced coordinates, and $\mathbf{q}_{\text{ref}} \in \mathbb{R}^{6N}$ is a prescribed reference state. Numerous approaches exist for computing the basis $\mathbf{\Phi}$ including proper orthogonal decomposition and the reduced basis method. Here, the latter approach is employed. Given a set of training parameter samples $\{\boldsymbol{\mu}_i\}_{i=1}^{n_{\text{samples}}}$, we define the basis matrix $\mathbf{\Phi}$ from a snapshot matrix $\mathbf{S} = \left[\tilde{\mathbf{q}}(\boldsymbol{\mu}_1) \ \cdots \ \tilde{\mathbf{q}}(\boldsymbol{\mu}_{n_{\text{samples}}})\right] \in \mathbb{R}^{6N \times n_{\text{samples}}}$. Specifically, in the present work we set $\mathbf{\Phi}$ to be $\mathbf{S}$ orthogonalized with respect to the cell volumes such that $\mathbf{\Phi}^T \mathbf{W} \mathbf{\Phi} = \mathbf{I}$, where $\mathbf{W} = \text{diag}(V_1, \ \cdots, \ V_1, \ V_2, \ \cdots, \ V_N)$ and V_i, $i = 1, \ldots, N$ is the volume of the ith cell. This definition associates with a discrete inner product that mimics its continuous counterpart, i.e., for a state $q(x) \in \mathbb{R}^6$ on the domain Ω discretized into the vector $\mathbf{q} \in \mathbb{R}^{6N}$ comprising the value of $q(x)$ at N finite volume cell centers we have

$$\int_\Omega q(x)^T q(x) dV \approx \mathbf{q}^T \mathbf{W} \mathbf{q}.$$

To ensure that the governing equations associated with the SA turbulence model are being satisfied, we additionally investigate a split-basis formulation where independent basis vectors are employed for the SA viscosity, $\tilde{\nu}$. Defining a "masking" operator as

$$\boldsymbol{\mathcal{M}}_{ij} = \begin{cases} 1 & i = j = 6, 12, \dots, 6N \\ 0 & \text{else} \end{cases} \in \mathbb{R}^{6N \times 6N}$$

we define a split basis as

$$\mathbf{\Phi} = \text{orthogonalize}\left(\left[(\mathbf{I} - \boldsymbol{\mathcal{M}})\, \mathbf{S},\ \boldsymbol{\mathcal{M}}\mathbf{S} \right] \right) \in \mathbb{R}^{6N \times 2 n_{\text{samples}}}. \tag{5}$$

Employing the split basis (5) results in ROMs that satisfy an independent governing equation for $\tilde{\nu}$ (as opposed to lumping the governing equation for $\tilde{\nu}$ into the governing equations for mass, momentum, and energy).

B. Galerkin reduced-order model discretized in entropy variables

Having defined the state approximation (4), we now define the projection scheme used to generate the ROM. Galerkin projection is one of the most common model reduction approaches. In Galerkin projection we restrict the state to live within the low-dimensional subspace defined by the basis $\mathbf{\Phi}$ and compute solutions within this low-dimensional subspace that satisfy the weak form of the governing equations. Mathematically this process results in the reduced-order model

$$\mathbf{\Phi}^T \boldsymbol{r}(\mathbf{\Phi}\hat{\mathbf{q}}(\boldsymbol{\mu}) + \mathbf{q}_{\text{ref}}; \boldsymbol{\mu}) = \mathbf{0}. \tag{6}$$

We emphasize that the Galerkin ROM (6) can be interpreted as a weighted residual solution to the weak form of (1) with a piecewise constant trial space described by the basis $\mathbf{\Phi}$. For non-symmetric and nonlinear systems, unfortunately, Galerkin projection is known to suffer from stability challenges [22]. Further, for vector-valued systems such as the compressible Navier–Stokes equations, Galerkin projection can result in ROMs that are dimensionally inconsistent if special care is not taken [23, 24].

In the present work we employ a variable transform to entropy variables to stabilize Galerkin projection by symmetrizing the governing equations. To this end we introduce a set of entropy variables as [25–28]

$$\mathbf{V} = \begin{Bmatrix} \frac{-s}{\gamma-1} + \frac{\gamma+1}{\gamma-1} - \frac{\overline{\rho}\tilde{E}}{\tilde{p}} - \frac{1}{2}\zeta\tilde{\nu}^2 \\ \frac{\overline{\rho}\tilde{u}_1}{\tilde{p}} \\ \frac{\overline{\rho}\tilde{u}_2}{\tilde{p}} \\ \frac{\overline{\rho}\tilde{u}_3}{\tilde{p}} \\ -\frac{\overline{\rho}}{\tilde{p}} \\ \zeta\tilde{\nu} \end{Bmatrix}$$

where $\zeta \in \mathbb{R}^+$ is a scalar weighting and $s = \ln(\tilde{p}) - \gamma \log(\overline{\rho})$. In the Galerkin ROM with entropy variables, we generate a basis for the entropy variables and compute solutions in entropy variables that satisfy the weak form of the governing equations. Mathematically this process results in the reduced-order model

$$\mathbf{\Phi}^T \boldsymbol{r}(\boldsymbol{\mathcal{V}}(\mathbf{\Phi}\hat{\mathbf{q}}(\boldsymbol{\mu}) + \mathbf{q}_{\text{ref}}); \boldsymbol{\mu}) = \mathbf{0} \tag{7}$$

where $\boldsymbol{\mathcal{V}} : \mathbb{R}^{6N} \to \mathbb{R}^{6N}$ is the mapping from entropy variables to conserved variables at each cell center. The Galerkin ROM (7) is solved with Newton's method.

Discretizing in entropy variables (and/or maintaining entropy stability) can result in numerous numerical advantages [25–27, 29]. Here we emphasize that, while we discretize our ROM in entropy variables, the full-order model SPARC still discretizes in conserved variables. Our past work has suggested that, even if the FOM is not discretized in entropy variables (or is entropy stable), discretizing in entropy variables for the ROM still results in improved performance.

C. Least-squares residual minimization reduced-order model

In addition to the Galerkin ROM in entropy variables we also consider ROMs based on least-squares residual minimization. Least-squares ROMs can be used over the more traditional Galerkin projection and have been shown to result in enhanced stability and accuracy [30–32]. In a Least-Squares Petrov Galerkin (LSPG) ROM, the residual minimization is performed by substituting the approximation of the state vector (4) into the FOM (3) and minimizing the residual in a weighted ℓ^2 norm

$$\hat{\mathbf{q}} = \underset{\hat{\mathbf{q}}^* \in \mathbb{R}^p}{\arg\min} \left\| \boldsymbol{r}\left(\mathbf{q}_{\mathrm{ref}} + \boldsymbol{\Phi}\hat{\mathbf{q}}^*; \boldsymbol{\mu}\right) \right\|_{\mathbf{M}} \tag{8}$$

where $\mathbf{M} \in \mathbb{R}^{6N \times 6N}$ is a symmetric positive definite weighting matrix defining the norm $\|\mathbf{v}\|_{\mathbf{M}}^2 \equiv \mathbf{v}^T \mathbf{M} \mathbf{v}$. The residual minimization problem is solved via the Gauss-Newton method with a QR decomposition.

In the present work we investigate weighted norms based on dimensional and physics-based arguments. In particular, due to the large difference in magnitudes between the state variables and the turbulence field variable, defining a physically-relevant residual minimization problem requires a norm that balances the contributions from the various state variables. Three norms are investigated: one based on the mean absolute value of the variables, one based on the mean residual norm of the variable, and a scaling based on a k-nearest-neighbors prediction of the residual norm for the variable. For simplicity, the weighting matrix $\mathbf{M}$ is defined as

$$\mathbf{M}_{ij} = \begin{cases} \mathcal{A}_\ell & i = j = \ell, \ell + 6, \ldots, 6N - 6 + \ell \\ 0 & \text{else} \end{cases}, \ \ell = 1, \ldots, 6 \tag{9}$$

where $\mathcal{A}_\ell \in \mathbb{R}^N$ represents the weighting for each state equation at each cell center. Let $\boldsymbol{q}_\ell = [\boldsymbol{q}_\ell, \boldsymbol{q}_{6+\ell}, \cdots, \boldsymbol{q}_{N-6+\ell}]^T \in \mathbb{R}^N$ and $\boldsymbol{r}_\ell = [\boldsymbol{r}_\ell, \boldsymbol{r}_{6+\ell}, \cdots, \boldsymbol{r}_{N-6+\ell}]^T \in \mathbb{R}^N$, $\ell = 1, ..., 6$ represent the state vector and residual for each respective state equation at all cell centers. Let $\boldsymbol{q}_\ell^{\mathrm{train}} \in \mathbb{R}^{N N_{\mathrm{train}}}$ and $\boldsymbol{r}_\ell^{\mathrm{train}} \in \mathbb{R}^{N N_{\mathrm{train}}}$ represent the state and residual solutions on a training set with size N_{train} and parameters $\boldsymbol{\mu}^{\mathrm{train}} \in \mathbb{R}^{\mathcal{D} \times N_{\mathrm{train}}}$. Then, the scaling based on the mean absolute value of the variable defines a norm that scales each residual component based on the mean absolute value of the state variable over the training set, *i.e.*,

$$\mathcal{A}_\ell = \frac{1}{\underset{\boldsymbol{q} \in \mathbb{R}^N}{\mathrm{avg}} \left(\left| \boldsymbol{q}_\ell^{\mathrm{train}} \right| \right)} \tag{10}$$

where $|\cdot|$ represents the absolute value. Analogously, the scaling based on the mean residual defines a norm that scales each component of the residual based on the mean value of the residual norm defined over the training set.

$$\mathcal{A}_\ell = \frac{1}{\|\bar{\boldsymbol{r}}_\ell^{\mathrm{train}}\|} \tag{11}$$

where here $(\bar{\cdot})$ represents the spatial average and $\|\bar{\boldsymbol{r}}_\ell^{\mathrm{train}}\|$ is an approximation for the mean residual norm. Finally, the scaling using k-NN trains a model to predict the mean residual for each state variable based on the values of the turbulent coefficients and then uses the inverse of that predicted mean residual to scale the residual for each state component

$$\mathcal{A}_\ell = \frac{1}{\mathcal{J}_\ell(\boldsymbol{\mu})} \tag{12}$$

where $\mathcal{J}_\ell : \mathcal{D} \to \mathbb{R}$ represents the k-NN functional with $\mathcal{J}_\ell : \boldsymbol{\mu} \mapsto \|\bar{\boldsymbol{r}}_\ell^*\|$. The k-NN model is trained from the the mean residual norm $\|\bar{\boldsymbol{r}}_\ell^{\mathrm{train}}\| \in \mathbb{R}^{N_{\mathrm{train}}}$ taken from the training data $\boldsymbol{r}_\ell^{\mathrm{train}}$ and from the parameters defined in the training set $\boldsymbol{\mu}^{\mathrm{train}}$.

V. Results

This section presents results for the propagation of uncertainties of the SA turbulence model coefficients for two benchmark RANS cases from the NASA Turbulence Modeling Resource: a turbulent flat plate with a zero pressure-gradient [33] and a wall-mounted hump with a separated flow [34]. The reduced-order models are implemented in Sandia's compressible CFD solver SPARC [21] with the open-source ROM package Pressio*. The ROM results are compared to results from a Gaussian Process Regression (GP) model.

A. Prior Uncertainty

The prior uncertainty was selected to be the same as Kwanten [2]. It's important to note that Kwanten used the standard version of the SA model while the negative SA model is used here. Table 1 summarizes the approximate

Coefficient	Lower	Upper
σ	0.6	1.0
c_{w_2}	0.2	0.5
c_{w_3}	1.0	3.0
c_{v_1}	4.0	10.0
c_{b_1}	0.13	0.14
c_{b_2}	0.61	0.69
κ	0.3	0.5

Table 1 Prior uncertainty of the SA turbulent coefficients.

upper and lower bounds for each turbulent coefficient that appears in the SA turbulence model equations. The ranges for c_{b_1} and c_{b_2} were found using a relation to σ derived by Kwanten. The relation was derived by using a quadratic polynomial fit to plots of c_{b_1} and c_{b_2} that appear in the original Spalart-Allmaras paper [35, Figure 1].

$$\begin{aligned} 10c_{b_1} &= 0.187\sigma^2 - 0.4973\sigma + 1.6046 \\ 1 + c_{b_2} &= 0.0751\sigma^2 + 0.0834\sigma + 1.5321 \end{aligned} \tag{13}$$

The coefficients were distributed uniformly between their lower and upper bounds for all training and test sets.

B. Zero pressure-gradient turbulent flatplate boundary layer

A zero-pressure-gradient turbulent flat plate flow was selected as the first case to propagate the uncertainty of the SA turbulence model coefficients. This case was based on the 2D turbulent flat plate verification case ran by NASA Langley [33]. In this case, the inlet flow comes in at Mach 0.2 at a Reynolds number of 5 million based on a unit length of the grid and a reference temperature of 300 K. The total pressure over the reference pressure at the inlet is 1.02828 and at the outlet is 1.0. The total temperature over the reference temperature is 1.008 at the inlet. Figure 1 shows the computational grid. The FOM solved for compressible flow over the flat plate.

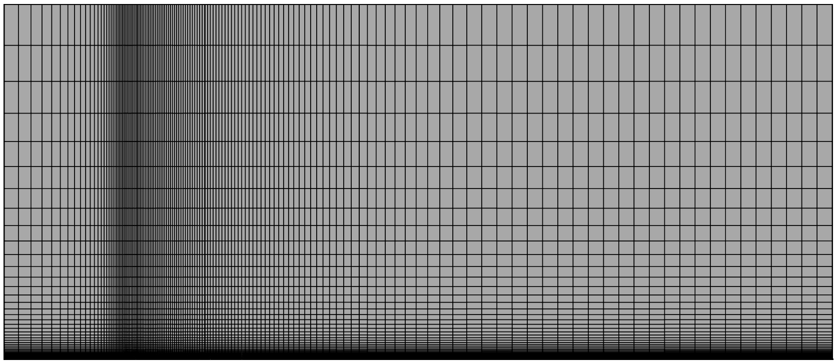


Fig. 1 Computational Grid of the 2D Flatplate

For this benchmark case, four SA turbulent coefficients were varied: σ, c_{w_2}, c_{w_3}, and c_{v_1}. A latin hypercube sampling (LHS) method was used to create the training points for the ROM. Sixteen training points were used. The ROM predicted solutions at a 50 additional points selected using LHS. These fifty additional points were not used to train the ROM.

The three weighted norms described in Sec. IV.C and the Galerkin entropy formulation ROM in Sec. IV.B are tested. Figure 2 demonstrates predictions for wall coefficient of friction made by the residual-minimization ROM with no scaling (top), the residual minimization ROM with k-NN scaling (middle) (12), and the Galerkin Entropy ROM

*https://github.com/Pressio

(bottom). The results for least-squares ROMs with the mean absolute value scaling (10) are not shown here because they are similar to the results obtained with no scaling, and results for the least-squares ROMs using a mean residual scaling (11) are also not shown here as they are similar to the results from k-NN scaling. The left figure shows the 95% confidence interval (CI) of the wall coefficient of friction overlayed across the "true" 95% CI that is the FOM solution at the test points and the right figure compares the prediction of the wall coefficient of friction to the "true" value, which is obtained from the FOM solution.

With no scaling used, the ROM predicts a small 95% CI and predicts an incorrect trend in the wall coefficient of friction. Using a k-NN scaling allows the ROM to predict a larger 95% CI, but the ROM prediction is not accurate compared to the "true" distribution, which is the distribution produced by the FOM solutions at the test points. The Galerkin entropy ROM performs the best of the methods considered: the method predicts an accurate distribution for skin friction and faithfully reproduces the FOM. We compare the ROM to another surrogate, a Gaussian process regression (GP) model trained on the same training points as the ROM. The GP model accepts a parameter instance (or a set of parameter instances) and outputs a scalar prediction; the GP model output is defined to be $\mathbf{f}_{\text{GP}} : \mathbb{R}^{\mathcal{D}} \to \mathbb{R}$ with $\mathbf{f}_{\text{GP}} : \boldsymbol{\mu} \mapsto \overline{C}_f\ (\boldsymbol{\mu})$ where $\overline{C}_f\ (\boldsymbol{\mu})$ denote the predicted average coefficient of friction across the bottom wall. The GP model used a constant kernel with an initial guess of 1 and bounds of [1e-5, 1e3] multiplied to a radial basis function kernel with an initial guess of 10 and bounds of [1e-5, 1e3]. The GP model is able to predict the average wall coefficient of friction at an R^2 of 0.99 and is competitive with the best ROM method.

We solve the FOM at two sets of points: the training set and the test set. These sets are used to both train the ROM and evaluate how well it performs in the parameter space. Because the FOM is so computationally expensive to run, it is necessary to evaluate whether the ROM adds any useful information or whether there was enough information to perform UQ with just the FOM solution at the training points. Figure 3 shows the 95% confidence interval (CI) for the wall coefficient of friction using three sets of data: (1) the FOM solution on the training set, (2) the FOM solution on the training set and the ROM prediction on the test set, and (3) the FOM solution on the training and test set. The third dataset is used as the "true" data set. With only the information that the FOM solution on the training points provides, the 95% CI for the wall coefficient of friction has a much larger range than the true 95% CI range. When the ROM prediction on the test set is included into the dataset with the FOM training set, the 95% CI range matches the range of the "true" dataset, *i.e.*, the ROM provides valuable information that helps capture the true range.

For this flatplate case, we showed that a Galerkin entropy ROM was able to match the "true" distribution of the wall coefficient of friction and predict the average wall coefficient of friction with an R^2 value of 0.99. A GP model performed similarly to the Galerkin entropy ROM for the same training set size of 16. In this next section, we will examine the effect of the size of the training set on the accuracy of the ROM and the GP model.

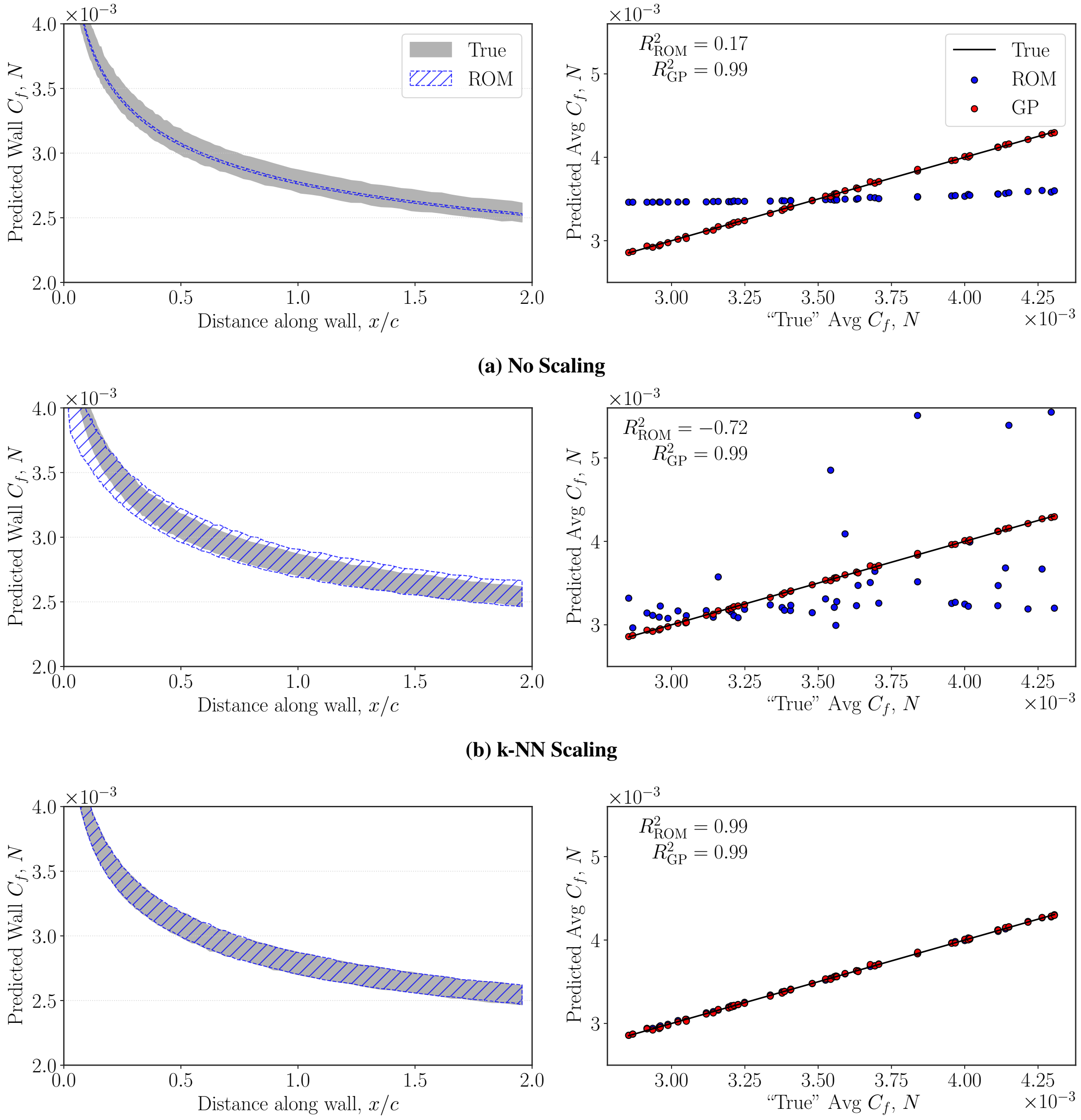
C. Wall-mounted hump with separated flow

A wall-mounted hump with a separated flow was selected as the second test case [34]. For this case, the reference freestream velocity is 34.6 m/s and reference temperature is 537 R. The Reynolds number based on chord for this flow is 936,000. The chord of the bump is 420 mm and the incoming fully turbulent boundary layer thickness is approximately 8% of the chord. The upper boundary is a slip boundary and the bottom boundary is a no-slip wall boundary. At the inlet, total pressure over the reference pressure is 1.007 and the total temperature over the reference temperature is 1.002. The static pressure over the reference pressure at the outlet is 0.99962. Figure 4 shows the computational grid for the wall-mounted hump. A no-slip boundary condition is used for the bottom wall and a slip boundary condition is used at the top wall.

Given its superior accuracy, only the Galerkin formulation using entropy variables is presented for this problem. Because the residual minimization ROMs were found to be less accurate than the entropy Galerkin formulation, they are omitted from this section. To further alleviate the difference in scaling and reduce numerical stiffness, the basis was again split between the turbulent viscosity and the other conserved quantities as described in Eq. (5).

For comparison, a Gaussian Process regression (GP) model was created to compare the performance of the ROM to a standard surrogate. A varying number of training points were used to train both the GP model and the ROM where both the GP model and the ROM used the same training points each time. Both models were tested on the same set of test points used before. This GP model used the same kernels as the GP model used for the flatplate.

For this benchmark case, seven SA turbulent coefficients were varied: σ, c_{w_2}, c_{w_3}, c_{b_1}, c_{b_2}, c_{v_1}, and κ. The ROM was trained using 16 training points sampled using LHS and then predicted solutions at 200 test points also sampled using LHS. Figure 5 shows the ROM prediction of the posterior distribution, the 95% CI and mean, of the wall coefficient of friction against the "true" distribution, which is the FOM solutions on the test set. At sixteen training



(a) No Scaling

(b) k-NN Scaling

(c) Galerkin Entropy ROM with Split Basis

Fig. 2 Prediction of wall coefficient of friction over a zero pressure-gradient turbulent flatplate using three methods: (a) an LSPG ROM with no residual scaling, (b) an LSPG ROM with a k-NN mean residual norm scaling, and (c) a Galerkin entropy ROM with a split basis.

points again, the ROM prediction of the distribution matches that of the "true" distribution and predicts an average wall coefficient of friction with an R^2 value of 0.997 when compared to the FOM solution. At sixteen training points, the ROM and the GP model have so far predicted at R^2 values greater than 0.9 for both the flatplate with four SA turbulent coefficients varied and for the wall-mounted hump with seven SA turbulent coefficients varied. For the next analysis, we reduced the training set size to analyze the accuracy of the ROM and to understand the dependence of the ROM's and the GP model's accuracy on the size and spread of the training points.

Table 2 summarizes the relative error, ϵ, and R^2 values across all training samples within a set of the ROM and

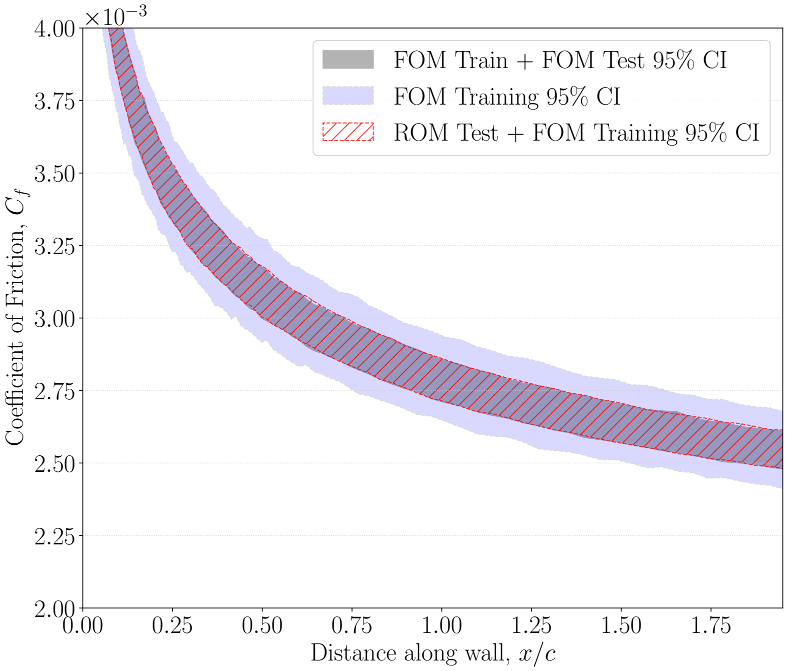


Fig. 3 Posterior distribution of the wall coefficient of friction for different data sets.

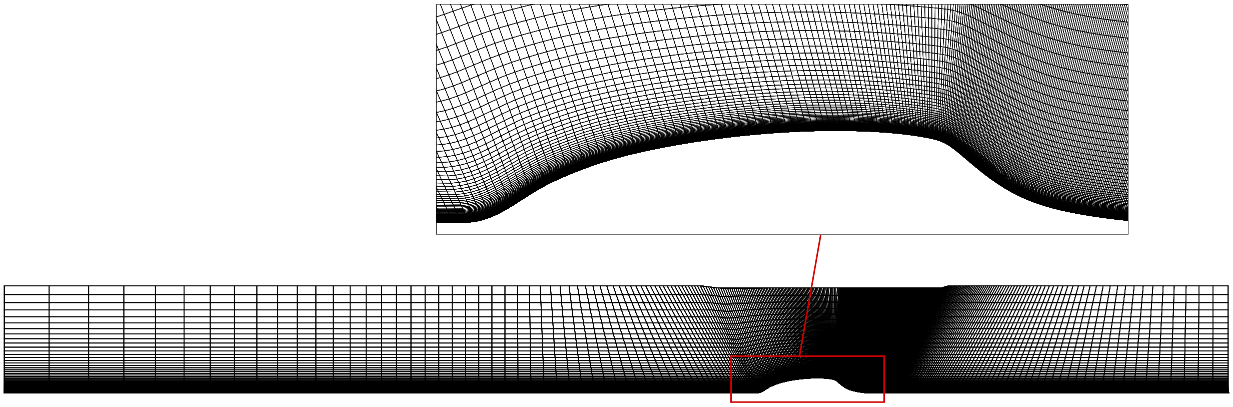


Fig. 4 Wall-mounted hump computational grid

GP model predictions of drag and average wall coefficient of friction over the hump to the FOM results. The training points were generated using LHS. Because of the randomness of LHS, two distinct sets were created for each number of training points. This reduces the likelihood that the randomly selected set of training points are located on spots in the parameter space that provide no useful information to the models and should also provide a better picture of how the models perform. The ROM result in Figure 5 is trained using the first row in the 16 training set size. As shown in Table 2, the ROM is more accurate than the GP model for a fewer number of training points. There are a few observations to point out here: (1) the ROM is more robust to the spread of the training points than the GP model and (2) the ROM is able to achieve a fixed accuracy level with fewer training samples than the GP.

For point 1, the GP model has varying performance within a set for a given number of training points. For example, for seven training points, the GP model is able to predict drag and the average wall coefficient of friction above an R^2 value of 0.5. With that same number of training points but a different selection, the GP model is unable to predict

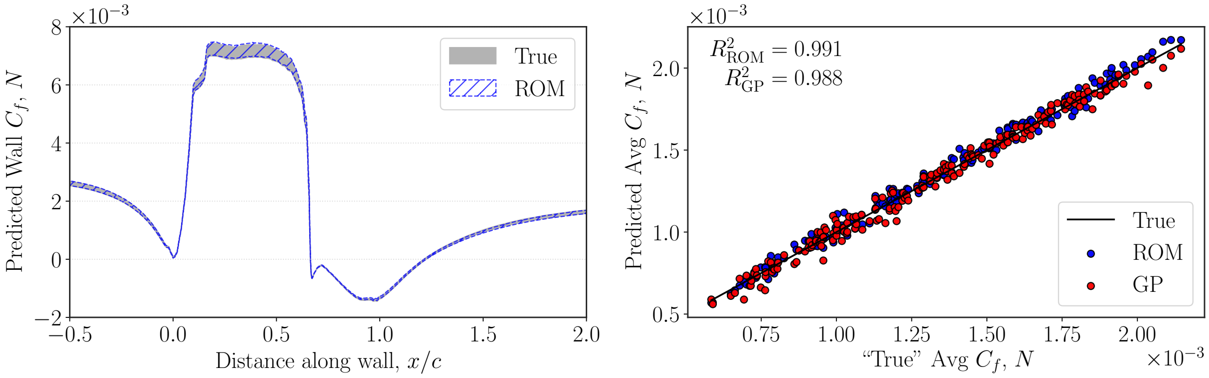


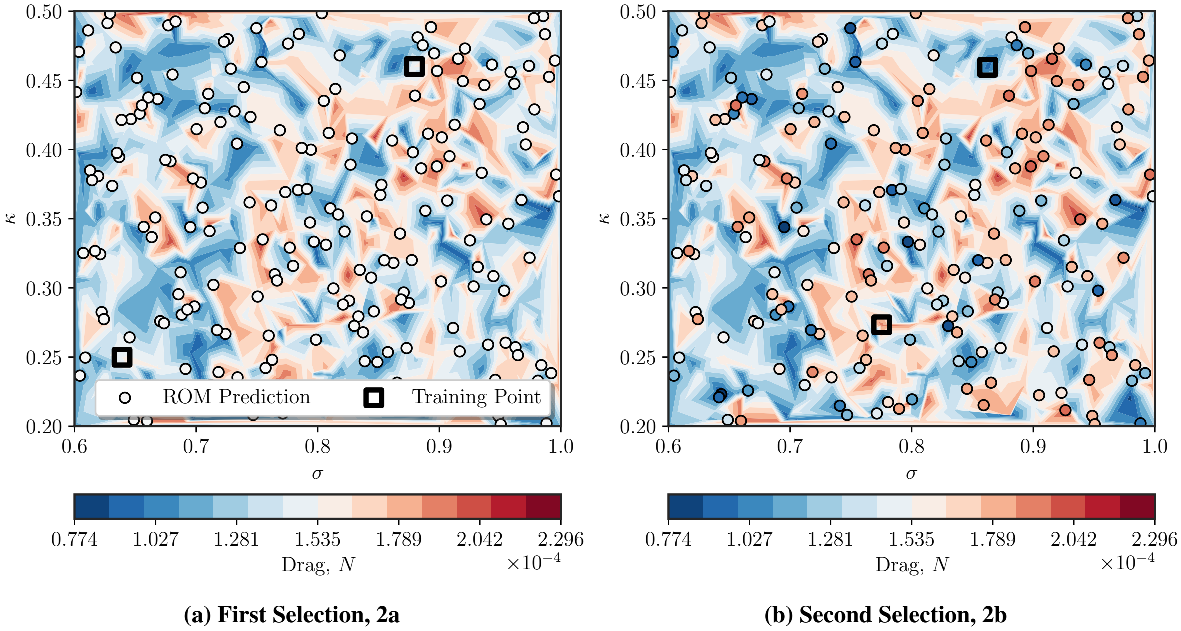
Fig. 5 Posterior distribution of the wall coefficient of friction for a wall-mounted hump.

No. of Training Points	Drag				Avg Wall Coefficient of Friction			
	ϵ_{ROM}	ϵ_{GP}	R^2_{ROM}	R^2_{GP}	ϵ_{ROM}	ϵ_{GP}	R^2_{ROM}	R^2_{GP}
2	21.98	21.99	−0.101	−0.101	31.78	32.67	−0.099	−0.097
	8.40	19.64	0.820	−0.004	12.40	28.83	0.786	0.000
4	7.33	20.35	0.801	−0.007	13.45	29.48	0.663	−0.002
	7.87	17.80	0.818	0.177	14.91	26.54	0.708	0.153
6	5.25	18.21	0.861	0.122	10.06	26.14	0.846	0.150
	2.69	8.97	0.977	0.776	5.16	14.14	0.956	0.742
7	4.19	11.30	0.938	0.632	6.94	18.00	0.926	0.526
	2.72	19.36	0.969	−0.019	12.54	27.93	0.787	−0.022
8	2.26	7.41	0.985	0.848	3.93	11.04	0.979	0.836
	1.44	10.14	0.992	0.680	3.46	15.67	0.981	0.645
10	1.57	1.51	0.992	0.994	2.65	2.36	0.989	0.991
	2.71	1.10	0.973	0.996	6.61	2.43	0.940	0.991
16	0.88	1.35	0.997	0.994	2.27	2.81	0.991	0.988
	1.81	1.44	0.997	0.993	1.81	2.30	0.991	0.991

Table 2 R^2 value and relative error comparison of the ROM and GP model to the FOM for drag and average wall coefficient of friction over the hump.

better than the mean of the test data. The ROM similarly struggles for this same set of training points, but predicts consistently above an R^2 value of 0.7 within the seven sample training set size and consistently above an R^2 value of 0.6 when using three or more training points. It is clear that the GP model has more sensitivity to the placement of the training points than the ROM, which is likely because it is a data-driven surrogate model whereas the ROM incorporates knowledge of the underlying physics based on its formulation.

With just two training points, the ROM is able to predict solutions for drag and average wall coefficient of friction with an R^2 value around 0.8. To better understand why the ROM is able to perform so well with so few training points, the ROM prediction for drag was plotted over the contour of the "true" values of drag (true taken to be the FOM solution) for the set of two training points in Figure 6. This contour represents a slice of a seven-dimensional parameter space highlighting the interaction between κ and σ on drag. We will refer to these sets as 2a and 2b where 2a refers to the first row and 2b refers to the second row of the "2" row in Table 2. Here, the square markers represent the location of the training points. For the 2a training set, the training points are located at points that share similar drag values. This training information does not convey any information on the shape of the drag surface. Therefore, the ROM predicts mostly constant solutions around the parameter space. For the 2b training set, the training points happen to be located at local maxima and minima points on the drag surface, which conveys some information on the shape of the drag surface. One would expect the ROM to follow a linear surface prediction based on the size of the training set, but the ROM is able to predict solutions outside of the parameter space it was trained and even predict other local optima.



(a) First Selection, 2a

(b) Second Selection, 2b

Fig. 6 Comparison of the ROM solution to the FOM solution of drag for a set of two training points.

Figure 7 compares the distribution and the probability density function (PDF) of the ROM and GP model results of drag along the wall-mounted hump to the "true" solution, *i.e.*, the FOM results, for training sets 7a, 8a, and 10a. These sets were chosen because they had the highest R^2 values for drag for both the ROM and the GP model. As the number of training points increase, the ROM and the GP model converge to the FOM solution. The ROM starts out with a distribution that is already close to the FOM solution at seven training points, but the mode for its drag prediction is shifted from the FOM's mode. This is corrected at 8 training points. The GP model predicts a mode for its drag distribution that is always aligned with the FOM's mode, but the GP model over predicts the frequency and under predicts the standard deviation of the drag distribution. At ten training points, both the ROM and the GP model closely match the FOM.

Figure 8 shows a correlation matrix of the SA turbulent coefficients to the average wall coefficient of friction from the ROM results. The ROM here was trained using the 10a set of training points and predicted solutions at 200 test points sampled using LHS. The correlation matrix was created from the test data. The bottom and right columns show the main effects of each turbulent coefficient on the average wall coefficient of friction and the interior rows and columns show the interactions between coefficients. Overall, most of the coefficients have small main and interaction effects on the average wall coefficient of friction. Of note are the coefficients κ and c_{ν_1}. Physically, the main effects of both coefficients make sense. The coefficient κ is tied to the diffusion and destruction terms in the SA equations. An increase in κ leads to smaller diffusion and destruction terms, which leads to a larger SA viscosity. The coefficient c_{ν_1} has a negative effect on the f_{ν_1} term, which appears in the calculation for the turbulent eddy viscosity. As c_{ν_1} increases, f_{ν_1} decreases and thus will decrease the SA viscosity.

In this wall-mounted hump case, we showed that a Galerkin entropy ROM was more robust to the size and spread of the training points than a GP model; the ROM was able to consistently predict drag and average wall coefficient of friction with an R^2 value above 0.8 when using a minimum of six training points while the GP model required 10 training points. We showed that the ROM could predict drag and average wall coefficient of friction above an R^2 value of 0.7 when using two training points if the training points are placed in unequal regions. Finally, we showed that the ROM predictions made physical sense when examining the effects of each SA turbulent coefficient on the SA viscosity. In this next section, we will examine the computational expense of running both the ROM and the GP model.

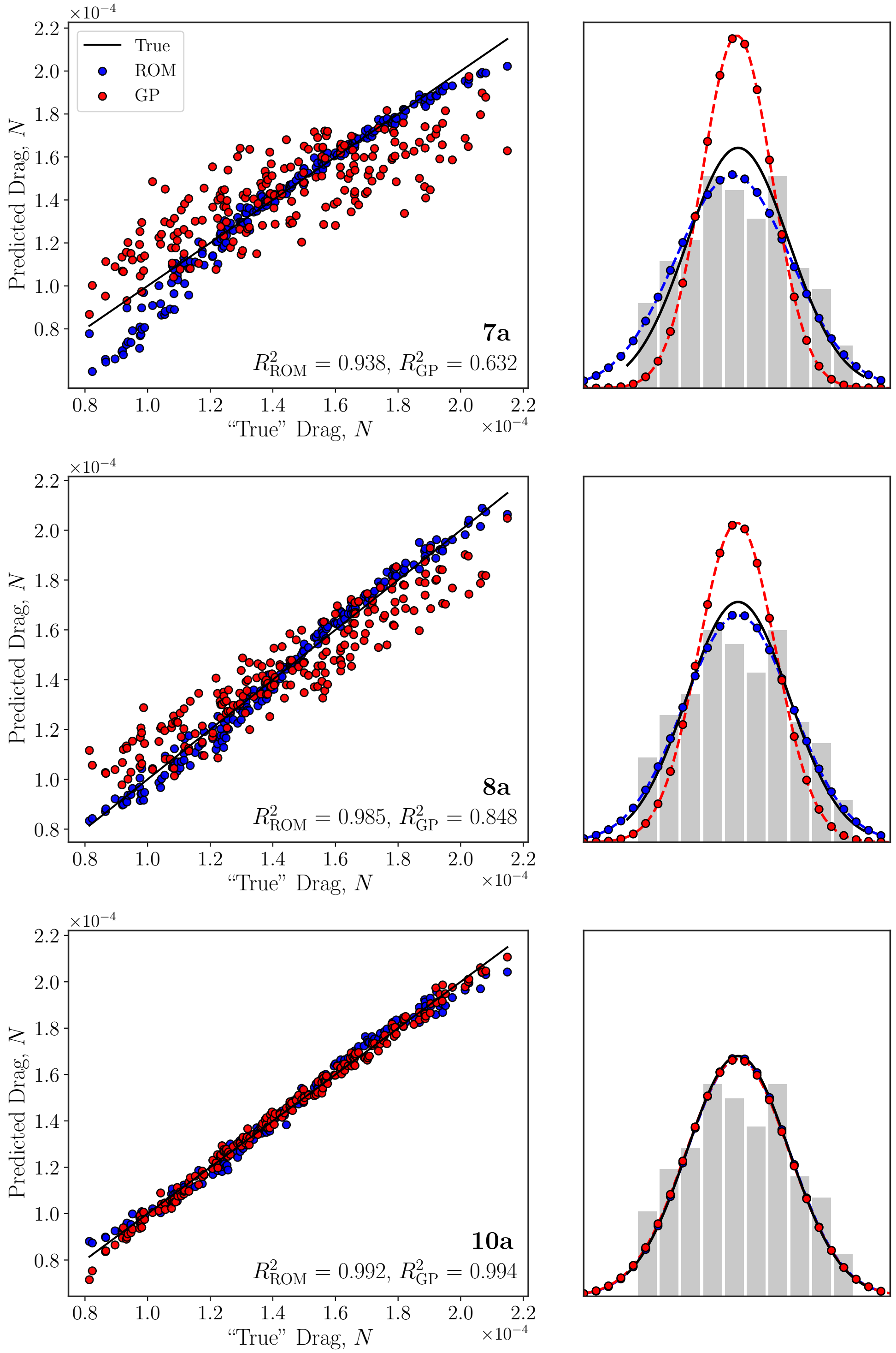


Fig. 7 Comparison of the posterior distribution and PDF of the ROM and GP model results of drag to the FOM solution using different training set sizes.

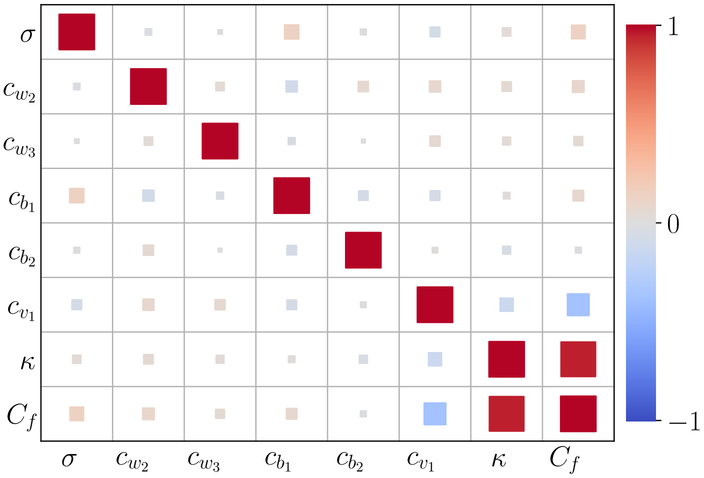


Fig. 8 Correlation matrix of the SA turbulent coefficients to the average wall coefficient of friction for 10 training points.

D. Computational Expense

The benefits of using a ROM or any surrogate model is that they are computationally inexpensive to run compared to the FOM. Depending on how the ROM is formulated or developed, it may run slower than most data-based surrogate models. To assess the performance of the ROM in this respect, the computational runtime of the ROM is compared to a Gaussian process regression model. The runtime is broken down into the offline phase and the online phase. For reference, the FOM runtime is included. It is important to note that the ROM solves for $\hat{\mathbf{q}} : \mathcal{D} \to 6p$ while the GP model solves for $\mathbf{f}_{\text{GP}} : \mathbb{R}^{\mathcal{D}} \to \mathbb{R}$ (i.e., the GP only predicts the quantity-of-interest while the ROM predicts the entire solution field within a low-dimensional subspace), so the CPU run times of the GP model can be a lot smaller than that of the ROM. The ROM can predict multiple QOIs at each cell center in the computational domain in a single run while a separate GP model has to be created for each QOI at each cell-center. Table 3 summarizes the CPU runtimes of all models for both the turbulent flatplate and the wall-mounted hump cases. For both cases, the ROM and the GP model were trained using 16 FOM snapshots.

	Flatplate			Wall-Mounted Hump		
	Offline [s]	Online [s]	Total [s]	Offline [s]	Online [s]	Total [s]
ROM	13.56	67.79	81.35	12.24	430.16	442.40
GP	0.15	0.01	0.16	0.11	0.01	0.12
FOM	-		956.08	-		93,704.33
Ratio of ROM to FOM	-		11.75	-		211.81
Ratio of GP to FOM	-		5832.19	-		792,561.09

Table 3 Computational runtime of all models.

The ROM is an order of magnitude faster than the FOM and the GP model is three orders of magnitude faster than the FOM. Looking at Table 2, we can see that the ROM was able to predict both drag and the average wall coefficient of friction with a consistent R^2 value greater than 0.8 starting at six training points while the GP models required greater than 10 training points to predict a consistent R^2 value greater than 0.8. At six training points, the total cost of training and using the ROM to predict drag and average wall coefficient of friction for 200 test points is simply the cost of running the FOM for six training points, performing the offline step once, and performing the online step 200 times: 367,158 seconds. At ten training points, the total cost of training the GP models and predicting drag and average wall coefficient of friction over 200 test points is the cost of running the FOM for ten training points, training two separate GP models for drag and average wall coefficient of friction (two offline costs), and using both models to make predictions (400 online costs): 937,048 seconds. The larger offline cost of training the GP model relative to the ROM exceeds its advantage over the ROM in its online cost.

It is important to note that the ROM used here is not necessarily optimized for runtime. There are two main areas in which the ROM's computational runtime can be improved: the computation of the action of the Jacobian on a vector in the Newton solve and the use of hyper-reduction. In the present work we employ Newton's method with a direct solve for the Galerkin ROM. We compute the reduced Jacobian employed by Newton's method in a two-step manner: first we compute the action of the Jacobian $\mathbf{J}$ on the basis $\mathbf{\Phi}$ via a matrix-free finite difference scheme and second we compute the reduced Jacobian $\mathbf{\Phi}^T\mathbf{J}\mathbf{\Phi}$. This approach is inefficient as computing the product $\mathbf{J}\mathbf{\Phi}$ via a matrix-free finite difference approach scales with the number of reduced states. For the Galerkin ROM we could alternatively employ a matrix-free Newton–Krylov method which only requires the action of the reduced Jacobian on a vector, i.e., we only require $\mathbf{\Phi}^T\mathbf{J}\mathbf{\Phi}\mathbf{a}$ for some vector $\mathbf{a}$. This term can be computed in a matrix-free manner with only one or two residual calls depending on the scheme; see the discussion in Ref. [36] for more details.

It is also well known that the calculation of the residual (6) scales with N, which inhibits the speedup of the ROM. Hyper-reduction can be utilized to reduce this cost by approximating the calculation of the residual or of $\mathbf{\Phi}^T\boldsymbol{r}$ when solving the ROM [37–40]. There is also a cost associated with using a FOM that does not use the same variable formulation as the ROM. Because of this difference in formulation, the ROM had to add a variable transformation in its call for the residual at each iteration step, which increased its computational expense.

VI. Conclusions

In this work, we used a projection-based reduced-order model to propagate the uncertainty of the SA turbulent coefficients on two quantities of interest, drag and the wall coefficient of friction. We applied the ROM for two benchmark RANS cases: a zero pressure-gradient turbulent flatplate and a wall-mounted hump with separated flow. We considered two ROMs: a Galerkin ROM discretized in entropy variables and a least-squares residual minimization ROM. The least-squares residual minimization ROM employed three weighted norms: (1) a norm based on the mean absolute value of the variables, (2) a norm based on the mean residual norm of the variable, and (3) a norm based on a k-nearest-neighbors prediction of the residual norm for the variable.

The turbulent flatplate case compared both ROMs and their respective norms by varying four SA turbulent coefficients: σ, c_{w_2}, c_{w_3}, and c_{v_1}. The Galerkin ROM discretized in entropy variables performed the best with an R^2 value (comparing to a respective FOM solution) of 0.991 for the average wall coefficient of friction. A GP model was created from the same training set and performed similarly to the Galerkin entropy ROM with an R^2 value of 0.988 for the average wall coefficient of friction. The residual minimization ROM performed poorly: the norm based on the mean absolute value of the variable predicted a posterior distribution similar to when no scaling was used and the norms based on a mean residual and k-NN scaling, while predicting a wider 95% CI than using no scaling, predicted an average wall coefficient of friction with a negative R^2 value. The Galerkin ROM discretized in entropy variables was used for all further results.

We analyzed whether the ROM provided any useful information to the posterior distribution of the wall coefficient of friction that the FOM training points did not already have. It was found that the information found in the FOM training points, which consisted of sixteen FOM snapshots, was not sufficient to match a "true" posterior distribution, true taken to be 216 FOM snapshots, of the wall coefficient of friction. When using the information from the FOM training points and the ROM predictions at two-hundred test points, the predicted posterior distribution matched the "true" posterior distribution of the coefficient of friction.

In the wall-mounted hump case, we examined the effect of training set size and spread on the accuracy of the ROM and GP model predictions by varying seven SA turbulent coefficients: σ, c_{w_2}, c_{w_3}, c_{b_1}, c_{b_2}, c_{v_1}, and κ. We found that the ROM was more robust to the size and spread of the training points compared to the GP model: the ROM was able to predict a consistent R^2 value above 0.8 using a minimum of six training points while the GP model required ten training points. The ROM was also able to predict both drag and average wall coefficient of friction with an R^2 value of above 0.7 when using two training points. We found that the placement of the training points at such a small training set size had a larger impact on the accuracy of the ROM: when the training points were placed in regions that had similar values of drag then the ROM predicted poorly with an R^2 value of -0.101 and when the training points were placed at a local maxima and minima then the ROM predicted drag with an R^2 value of 0.82. We also showed that the ROM predictions made physical sense: the effect of the SA coefficients on the average wall coefficient of friction matched what would be expected based on the physical meaning of each term in the SA equations.

Finally, we examined the computational expense of the ROM, the GP model, and the FOM. The ROM was an order of magnitude faster than the FOM and the GP model was three orders of magnitude faster than the FOM. Thanks to the robustness of the ROM to the size and spread of the training points, the ROM is more computationally efficient to use than a GP model. For example, in the wall-mounted hump case the ROM required six training points to predict above an R^2 value of 0.8 for drag and the average wall coefficient of friction while the GP model required ten training points. The ROM at six training points is around 2.5 times faster than a GP model at ten training points when considering the FOM cost for each training point and the offline and online costs for the ROM and GP model. A cost analysis of using the ROM compared to using a GP model (or any other surrogate model) may be beneficial if shown for additional benchmark cases. It was clear from the wall-mounted hump case that the sensitivity of the drag and the average wall coefficient of friction to the SA coefficients were not high. The ROM may show better cost benefits for a case that has a larger uncertainty or sensitivity of the QOIs to the SA coefficients.

Future work will focus on improving the computational efficiency of the ROM. Incorporating matrix-free and hyper-reduction approaches into the ROM will reduce its computational expense. However, we will investigate whether these methods have a significant impact on the ROM's accuracy. A cost-benefit analysis will be beneficial here to determine whether a loss in accuracy is made up for by the decrease in runtime of the ROM. Furthermore, the uncertainty introduced by using a ROM instead of the FOM was never accounted for in this paper. Accounting for this model uncertainty could further decrease the amount of FOM snapshots required to train the ROM and improve the overall computational expense.

A. Definition of governing equations

This work considered solutions to the Favre-averaged Navier–Stokes equations closed with the Spalart–Allmaras–neg (SA-neg) turbulence model defined on the spatial domain $\Omega \subset \mathbb{R}^3$. The system of equations can be written in conservative form as

$$\nabla \cdot \boldsymbol{F}\left(\boldsymbol{U}\right) - \nabla \cdot \boldsymbol{F}^{v}\left(\boldsymbol{U}, \nabla \boldsymbol{U}; \boldsymbol{\mu}\right) = \boldsymbol{S}\left(\boldsymbol{U}, \nabla \boldsymbol{U}; \boldsymbol{\mu}\right) \tag{14}$$

where $\boldsymbol{U} = \left[\overline{\rho},\ \overline{\rho}\tilde{\boldsymbol{u}},\ \overline{\rho}\tilde{E},\ \overline{\rho}\tilde{\nu}\right]^T : \Omega \longrightarrow \mathbb{R}^6$ is the state comprising the Reynolds-averaged density, and Favre-averaged momentum, total energy, and $\tilde{\nu}$ is the SA viscosity. The inviscid and viscous fluxes are given by

$$\boldsymbol{F}_i(\boldsymbol{U}) = \begin{Bmatrix} \overline{\rho}\tilde{u}_i \\ \overline{\rho}\tilde{u}_i\tilde{u}_1 + \tilde{p}\delta_{i1} \\ \overline{\rho}\tilde{u}_i\tilde{u}_2 + \tilde{p}\delta_{i2} \\ \overline{\rho}\tilde{u}_i\tilde{u}_3 + \tilde{p}\delta_{i3} \\ \tilde{u}_i\left(\tilde{E} + \tilde{p}\right) \\ \overline{\rho}\tilde{u}_i\tilde{\nu} \end{Bmatrix}, \qquad \boldsymbol{F}_j^{v} = \begin{Bmatrix} 0 \\ \sigma_{1j} + \tau_{1j} \\ \sigma_{2j} + \tau_{2j} \\ \sigma_{3j} + \tau_{3j} \\ \left(\sigma_{ij} + \tau_{ij}\right)\tilde{u}_i + \left(\frac{c_p\mu}{\mathrm{Pr}} + \frac{c_p\mu_t}{\mathrm{Pr_t}}\right)\frac{\partial \tilde{T}}{\partial x_j} \\ \frac{1}{\sigma}\left(\mu + \overline{\rho}\tilde{\nu}f_n\right)\frac{\partial \tilde{\nu}}{\partial x_j} \end{Bmatrix}$$

The source term is given as

$$\boldsymbol{S}\left(\boldsymbol{U}, \nabla \boldsymbol{U}; \boldsymbol{\mu}\right) = \begin{Bmatrix} 0 \\ \boldsymbol{0} \\ 0 \\ \begin{cases} \overline{\rho}c_{b_1}\left(1 - f_{t_2}\right)\hat{S}\tilde{\nu} - \overline{\rho}\left(c_{w_1}f_w - \frac{c_{b_1}}{\kappa^2}f_{t_2}\right)\left(\frac{\tilde{\nu}}{d}\right)^2 + \frac{\overline{\rho}}{\sigma}\left(c_{b_2}\frac{\partial \tilde{\nu}}{\partial x_i}\frac{\partial \tilde{\nu}}{\partial x_i}\right) - \frac{1}{\sigma}\left(\nu + \tilde{\nu}\right)\frac{\partial \overline{\rho}}{\partial x_j}\frac{\partial \tilde{\nu}}{\partial x_j} & \tilde{\nu} \geq 0 \\ \overline{\rho}c_{b_1}\left(1 - c_{t_3}\right)\Omega\tilde{\nu} + \overline{\rho}c_{w_1}\left(\frac{\tilde{\nu}}{d}\right)^2 + \frac{\overline{\rho}}{\sigma}\left[c_{b_2}\frac{\partial \tilde{\nu}}{\partial xi}\frac{\partial \tilde{\nu}}{\partial x_i}\right] - \frac{1}{\sigma}\left(\nu + \tilde{\nu}\right)\frac{\partial \overline{\rho}}{\partial x_j}\frac{\partial \tilde{\nu}}{\partial x_j} & \tilde{\nu} < 0 \end{cases} \end{Bmatrix}.$$

In the above, $\tau_{ij} = 2\mu_t\left(S_{ij} - \frac{1}{3}\frac{\partial \tilde{u}_k}{\partial x_k}\delta_{ij}\right) - \frac{2}{3}\overline{\rho}\tilde{k}\delta_{ij}$ the Boussinesq approximation to the Reynolds stress, $\sigma_{ij} = 2\tilde{\mu}\left(S_{ij} - \frac{1}{3}\frac{\partial \tilde{u}_k}{\partial x_k}\delta_{ij}\right)$ is the viscous stress tensor, $S_{ij} = \frac{1}{2}\left(\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_i}{\partial x_i}\right)$ is the mean rate-of-strain tensor, $\tilde{T}$ is the Favre-averaged temperature, μ is the dynamic viscosity, and μ_t is the turbulent viscosity. In this work we assume a calorically perfect gas such that the equation of state is given by

$$\tilde{p} = (\gamma - 1)\left[\overline{\rho}\tilde{E} - \frac{1}{2}\overline{\rho}\left(\tilde{u}_1^2 + \tilde{\mu}_2^2 + \tilde{\mu}_3^2\right)\right].$$

We note that turbulent kinetic energy is *not* included in the definition of total energy $\tilde{E}$. The turbulent eddy viscosity is found from

$$\mu_t = \overline{\rho}\tilde{\nu} f_{\nu_1}.$$

The remaining terms in Eq. (14) are given as

$$\hat{S} = \Omega + \frac{\tilde{\nu}}{\kappa^2 d^2} f_{\nu_2} \qquad f_{\nu_2} = 1 - \frac{\chi}{1 + \chi f_{\nu_1}} \qquad \Omega = \sqrt{2 W_{ij} W_{ij}} \qquad W_{ij} = \frac{1}{2}\left(\frac{\partial \tilde{u}_i}{\partial x_j} - \frac{\partial \tilde{u}_j}{\partial x_i}\right)$$

$$f_w = g\left(\frac{1 + c_{w_3}^6}{g^6 + c_{w_3}^6}\right)^{1/6} \qquad g = r + c_{w_2}\left(r^6 - r\right) \qquad r = \min\left(\frac{\tilde{\nu}}{\tilde{S}\kappa^2 d^2}, 10\right) \qquad f_{t_2} = c_{t_3} \exp\left(-c_{t_4}\chi^2\right)$$

$$f_n = \frac{c_{n_1} + \chi^3}{c_{n_1} - \chi^3} \qquad c_{w_1} = \frac{c_{b_1}}{\kappa^2} + \frac{1 + c_{b_2}}{\sigma} \qquad f_{\nu_1} = \frac{\chi^3}{\chi^3 + c_{\nu_1}^3} \qquad \chi = \frac{\tilde{\nu}}{\nu}.$$

where $c_{n_1} = 16$ and d is the distance from the field point to the nearest wall. There are a total of ten coefficients in the model: σ, c_{w_1}, c_{w_2}, c_{w_3}, c_{b_1}, c_{b_2}, c_{ν_1}, κ, c_{t_3}, c_{t_4}.

Acknowledgments

This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energys National Nuclear Security Administration under contract DE-NA0003525.

Bibliography

[1] Slotnick, J., Khodadoust, A., Alonso, J., Darmofal, D., Gropp, W., Lurie, E., and Mavriplis, D., "CFD vision 2030 study: a path to revolutionary computational aerosciences," Tech. Rep. NASA/CR2014-218178, NASA, March 2014.

[2] Kwanten, B., "Bayesian Inference Uncertainty Quantification of RANS Turbulence Models," Master's thesis, Delft University of Technology, 2014.

[3] Turgeon, E., Pelletier, D., and Borggaard, J., "Application of a sensitivity equation method to the K-epsilon model of turbulence," *15th AIAA Computational Fluid Dynamics Conference*, 2001. https://doi.org/10.2514/6.2001-2534.

[4] Colin, E., Etienne, S., Pelletier, D., and Borggaard, J., "Application of a sensitivity equation method to turbulent flows with heat transfer," *International Journal of Thermal Sciences*, Vol. 44, No. 11, 2005, pp. 1024–1038. https://doi.org/10.1016/j.ijthermalsci.2005.04.002.

[5] Platteeuw, P., Loeven, G., and Bijl, H., "Uncertainty Quantification Applied to the k-epsilon Model of Turbulence Using the Probabilistic Collocation Method," *49th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 16th AIAA/ASME/AHS Adaptive Structures Conference, 10th AIAA Non-Deterministic Approaches Conference, 9th AIAA Gossamer Spacecraft Forum, 4th AIAA Multidisciplinary Design Optimization Specialists Conference*, 2008. https://doi.org/10.2514/6.2008-2150.

[6] Dunn, M. C., Shotorban, B., and Frendi, A., "Uncertainty Quantification of Turbulence Model Coefficients via Latin Hypercube Sampling Method," *Journal of Fluids Engineering*, Vol. 133, No. 4, 2011. https://doi.org/10.1115/1.4003762, 041402.

[7] 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 & System Safety*, Vol. 96, No. 9, 2011, pp. 1137–1149. https://doi.org/10.1016/j.ress.2010.09.013.

[8] Ray, J., Lefantzi, S., Arunajatesan, S., and Dechant, L., "Learning an Eddy Viscosity Model Using Shrinkage and Bayesian Calibration: A Jet-in-Crossflow Case Study," *ASCE-ASME J Risk and Uncert in Engrg Sys Part B Mech Engrg*, Vol. 4, No. 1, 2017. https://doi.org/10.1115/1.4037557, 011001.

[9] Ray, J., Lefantzi, S., Arunajatesan, S., and Dechant, L., "Bayesian Parameter Estimation of a k-e Model for Accurate Jet-in-Crossflow Simulations," *AIAA Journal*, Vol. 54, No. 8, 2016, pp. 2432–2448. https://doi.org/10.2514/1.J054758.

[10] Ray, J., Dechant, L., Lefantzi, S., Ling, J., and Arunajatesan, S., "Robust Bayesian Calibration of a k-e Model for Compressible Jet-in-Crossflow Simulations," *AIAA Journal*, Vol. 56, No. 12, 2018, pp. 4893–4909. https://doi.org/10.2514/1.J057204.

[11] Kato, H., and Obayashi, S., "Approach for uncertainty of turbulence modeling based on data assimilation technique," *Computers & Fluids*, Vol. 85, 2013, pp. 2–7. https://doi.org/10.1016/j.compfluid.2012.09.002.

[12] Edeling, W., Cinnella, P., Dwight, R., and Bijl, H., "Bayesian estimates of parameter variability in the k–e turbulence model," *Journal of Computational Physics*, Vol. 258, 2014, pp. 73–94. https://doi.org/10.1016/j.jcp.2013.10.027.

[13] Xiao, H., and Cinnella, P., "Quantification of model uncertainty in RANS simulations: A review," *Progress in Aerospace Sciences*, Vol. 108, 2019, pp. 1–31. https://doi.org/10.1016/j.paerosci.2018.10.001.

[14] Prudhomme, C., Rovas, D. V., Veroy, K., Machiels, L., Maday, Y., Patera, A. T., and Turinici, G., "Reliable Real-Time Solution of Parametrized Partial Differential Equations: Reduced-Basis Output Bound Methods," *Journal of Fluids Engineering*, Vol. 124, No. 1, 2001, pp. 70–80. https://doi.org/10.1115/1.1448332.

[15] Rozza, G., Huynh, D. B. P., and Patera, A. T., "Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations," *Archives of Computational Methods in Engineering*, Vol. 15, No. 3, 2008, pp. 229–275. https://doi.org/10.1007/BF03024948.

[16] Veroy, K., and Patera, A. T., "Certified real-time solution of the parametrized steady incompressible Navier–Stokes equations: rigorous reduced-basis a posteriori error bounds," *International Journal for Numerical Methods in Fluids*, Vol. 47, No. 8-9, 2005, pp. 773–788. https://doi.org/10.1002/fld.867.

[17] Veroy, K., Prud'homme, C., Rovas, D., and Patera, A., "A Posteriori Error Bounds for Reduced-Basis Approximation of Parametrized Noncoercive and Nonlinear Elliptic Partial Differential Equations," *16th AIAA Computational Fluid Dynamics Conference*, 2003. https://doi.org/10.2514/6.2003-3847.

[18] Holmes, P., Lumley, J., and Berkooz, G., *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*, Cambridge University Press, 1996. https://doi.org/10.1017/CBO9780511622700.

[19] Jin, R., Rizzi, F., and Parish, E., "Space-time reduced-order modeling for uncertainty quantification," 2021. https://doi.org/10.48550/ARXIV.2111.06435.

[20] Allmaras, S., Johnson, F., and Spalart, P., "Modifications and clarifications for the implementation of the Spalart-Allmaras turbulence model," *Seventh International Conference on Computational Fluid Dynamics (ICCFD7)*, 2012, pp. 1–11.

[21] Howard, M., Bradley, A., Bova, S. W., Overfelt, J., Wagnild, R., Dinzl, D., Hoemmen, M., and Klinvex, A., "Towards Performance Portability in a Compressible CFD Code," *23rd AIAA Computational Fluid Dynamics Conference*, Vol. 1, AIAA, Denver, CO, 2017. https://doi.org/10.2514/6.2017-4407.

[22] Carlberg, K., Barone, M., and Antil, H., "Galerkin v. least-squares PetrovGalerkin projection in nonlinear model reduction," *Journal of Computational Physics*, Vol. 330, 2017, pp. 693 – 734. https://doi.org/10.1016/j.jcp.2016.10.033.

[23] Rowley, C. W., Colonius, T., and Murray, R. M., "Model reduction for compressible flows using POD and Galerkin projection," *Physica D: Nonlinear Phenomena*, Vol. 189, No. 1, 2004, pp. 115–129. https://doi.org/10.1016/j.physd.2003.03.001.

[24] Parish, E. J., and Rizzi, F., "On the impact of dimensionally-consistent and physics-based inner products for POD-Galerkin and least-squares model reduction of compressible flows," 2022.

[25] Harten, A., "On the symmetric form of systems of conservation laws with entropy," *Journal of Computational Physics*, Vol. 49, No. 1, 1983, pp. 151–164. https://doi.org/10.1016/0021-9991(83)90118-3.

[26] Tadmor, E., "Skew-selfadjoint form for systems of conservation laws," *Journal of Mathematical Analysis and Applications*, Vol. 103, No. 2, 1984, pp. 428–442. https://doi.org/10.1016/0022-247X(84)90139-2.

[27] Hughes, T., Franca, L., and Mallet, M., "A new finite element formulation for computational fluid dynamics: I. Symmetric forms of the compressible Euler and Navier-Stokes equations and the second law of thermodynamics," *Computer Methods in Applied Mechanics and Engineering*, Vol. 54, No. 2, 1986, pp. 223–234. https://doi.org/10.1016/0045-7825(86)90127-1.

[28] Lodares, D., Manzanero, J., Ferrer, E., and Valero, E., "An entropy–stable discontinuous Galerkin approximation of the Spalart–Allmaras turbulence model for the compressible Reynolds Averaged Navier–Stokes equations," *Journal of Computational Physics*, Vol. 455, 2022, p. 110998. https://doi.org/10.1016/j.jcp.2022.110998.

[29] Chan, J., "Entropy stable reduced order modeling of nonlinear conservation laws," *Journal of Computational Physics*, Vol. 423, 2020, p. 109789. https://doi.org/10.1016/j.jcp.2020.109789.

[30] LeGresley, P., and Alonso, J., *Airfoil design optimization using reduced order models based on proper orthogonal decomposition*, 2000. https://doi.org/10.2514/6.2000-2545.

[31] Bui-Thanh, T., Willcox, K., and Ghattas, O., "Model Reduction for Large-Scale Systems with High-Dimensional Parametric Input Space," *SIAM Journal on Scientific Computing*, Vol. 30, No. 6, 2008, pp. 3270–3288. https://doi.org/10.1137/070694855.

[32] Carlberg, K., Bou-Mosleh, C., and Farhat, C., "Efficient non-linear model reduction via a least-squares Petrov Galerkin projection and compressive tensor approximations," *International Journal for Numerical Methods in Engineering*, Vol. 86, No. 2, 2011, pp. 155–181. https://doi.org/10.1002/nme.3050.

[33] "SA Expected Results - 2D Zero Pressure Gradient Flat Plate," https://turbmodels.larc.nasa.gov/flatplate_sa.html, 2021. Accessed: 2021-05-20.

[34] "2D NASA Wall-Mounted Hump Separated Flow Validation Case," https://turbmodels.larc.nasa.gov/nasahump_val.html, 2021. Accessed: 2021-05-20.

[35] Spalart, P., and Allmaras, S., *A one-equation turbulence model for aerodynamic flows*, 1992. https://doi.org/10.2514/6.1992-439.

[36] Parish, E. J., Wentland, C. R., and Duraisamy, K., "The Adjoint Petrov–Galerkin method for non-linear model reduction," *Computer Methods in Applied Mechanics and Engineering*, Vol. 365, 2020, p. 112991. https://doi.org/10.1016/j.cma.2020.112991.

[37] Barrault, M., Maday, Y., Nguyen, N. C., and Patera, A. T., "An empirical interpolation method: application to efficient reduced-basis discretization of partial differential equations," *Comptes Rendus Mathematique*, Vol. 339, No. 9, 2004, pp. 667–672. https://doi.org/10.1016/j.crma.2004.08.006.

[38] Ryckelynck, D., "A priori hyperreduction method: an adaptive approach," *Journal of Computational Physics*, Vol. 202, No. 1, 2005, pp. 346–366. https://doi.org/10.1016/j.jcp.2004.07.015.

[39] Chaturantabut, S., and Sorensen, D. C., "Nonlinear Model Reduction via Discrete Empirical Interpolation," *SIAM J. Sci. Comput.*, Vol. 32, No. 5, 2010, pp. 2737–2764. https://doi.org/10.1137/090766498.

[40] Farhat, C., Chapman, T., and Avery, P., "Structure-preserving, stability, and accuracy properties of the energy-conserving sampling and weighting method for the hyper reduction of nonlinear finite element dynamic models," *International Journal for Numerical Methods in Engineering*, Vol. 102, No. 5, 2015, pp. 1077–1110. https://doi.org/10.1002/nme.4820.