An Efficient Surrogate Model for Emulation and Physics Extraction of Large Eddy Simulations

Simon Mak, Chih-Li Sung, Xingjian Wang, Shiang-Ting Yeh, Yu-Hung Chang, V. Roshan Joseph, Vigor Yang & C. F. Jeff Wu

Taylor & Francis
Taylor & Francis Group

Check for updates

An Efficient Surrogate Model for Emulation and Physics Extraction of Large Eddy Simulations

Simon Mak[a,*], Chih-Li Sung[a,*], Xingjian Wang[b], Shiang-Ting Yeh[b], Yu-Hung Chang[b], V. Roshan Joseph[a], Vigor Yang[b], and C. F. Jeff Wu[a]

[a]School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA; [b]School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA

ABSTRACT

In the quest for advanced propulsion and power-generation systems, high-fidelity simulations are too computationally expensive to survey the desired design space, and a new design methodology is needed that combines engineering physics, computer simulations, and statistical modeling. In this article, we propose a new surrogate model that provides efficient prediction and uncertainty quantification of turbulent flows in swirl injectors with varying geometries, devices commonly used in many engineering applications. The novelty of the proposed method lies in the incorporation of known physical properties of the fluid flow as simplifying assumptions for the statistical model. In view of the massive simulation data at hand, which is on the order of hundreds of gigabytes, these assumptions allow for accurate flow predictions in around an hour of computation time. To contrast, existing flow emulators which forgo such simplifications may require more computation time for training and prediction than is needed for conducting the simulation itself. Moreover, by accounting for coupling mechanisms between flow variables, the proposed model can jointly reduce prediction uncertainty and extract useful flow physics, which can then be used to guide further investigations. Supplementary materials for this article, including a standardized description of the materials available for reproducing the work, are available as an online supplement.

1. Introduction

In the quest for designing advanced propulsion and power-generation systems, there is an increasing need for an effective methodology that combines engineering physics, computer simulations, and statistical modeling. A key point of interest in this design process is the treatment of turbulence flows, a subject that has far-reaching scientific and technological importance (McComb 1990). Turbulence refers to the irregular and chaotic behavior resulting from motion of a fluid flow (Pope 2001), and is characterized by the formation of eddies and vortices, which transfer flow kinetic energy due to rotational dynamics. Such a phenomenon is an unavoidable aspect of everyday life, present in the earth's atmosphere and ocean waves, and also in chemically reacting flows in propulsion and power-generation devices. In this article, we develop a surrogate model, or emulator, for predicting turbulent flows in a swirl injector, a mechanical component with a wide variety of engineering applications.

There are two reasons why a statistical model is required for this important task. First, the time and resources required to develop an effective engineering device with desired functions may be formidable, even at a *single* design setting. Second, even with the availability of high-fidelity simulation tools, the computational resources needed can be quite costly, and only a handful of design settings can be treated in practical times. For example, the flow simulation of a single injector design takes over 6

days of computation time, parallelized using 200 CPU cores. For practical problems with large design ranges and/or many design inputs, the use of only high-fidelity simulations is insufficient for surveying the full design space. In this setting, emulation provides a powerful tool for efficiently predicting flows at any design geometry, using a small number of flow simulations as training data. A central theme of this article is that, by properly *eliciting* and *applying* physical properties of the fluid flow, simplifying assumptions can be made on the emulator which greatly reduce computation and improve prediction accuracy. In view of the massive simulation datasets, which can exceed many gigabytes or even terabytes in storage, such efficiency is paramount for the usefulness of emulation in practice.

The proposed emulator uses a popular technique called *kriging* (Mathéron 1963), which employs a Gaussian process (GP) for modeling computer simulation output over a desired input domain. The main appeal of kriging lies in the fact that both the emulation predictor and its associated uncertainty can be evaluated in closed-form. For our application, a kriging model is required that can predict flows at any injector geometry setting; we refer to this as *flow kriging* for the rest of the article. In recent years, there have been important developments in flow kriging, including the works of Williams et al. (2006) and Rougier (2008) on regular spatial grids (i.e., outputs are observed at the same spatial locations over all simulations), and Hung, Joseph, and

Melkote (2015) on irregular grids. Unfortunately, it is difficult to apply these models to the more general setting in which the *dimensions* of spatial grids vary greatly for different input variables. In the present work, for instance, the desired design range for injector length varies from 20 mm to 100 mm. Combined with the high spatial and temporal resolutions required in simulation, the resulting flow data is much too large to process using existing models, and data-reduction methods are needed.

There has been some work on using reduced-basis models to compact data for emulation, including the functional linear models by Fang, Li, and Sudjianto (2006), wavelet models by Bayarri et al. (2007) and principal component models by Ramsay and Silverman (2002), and Higdon et al. (2008). Here, we employ a generalization of the latter method called *proper orthogonal decomposition (POD)* (Lumley 1967), which is better known in statistical literature as the Karhunen–Loève decomposition (Karhunen 1947, Loève 1955). From a flow physics perspective, POD separates a simulated flow into key instability structures, each with its corresponding spatial and dynamic features. Such a decomposition is, however, inappropriate for emulation, because there is no way to connect the extracted instabilities of one input setting to the instabilities of another setting. To this end, we propose a new method called the *common POD* (CPOD) to extract *common* instabilities over the design space. This technique exploits a simple and physically justifiable linearity assumption on the spatial distribution of instability structures.

In addition to efficient flow emulation, our model also provides two important features. First, the same domain-specific model simplifications (e.g., on the spatio-temporal correlation structure) that enable efficient prediction also allow for an efficient uncertainty quantification (UQ) for such a prediction. This UQ is highly valuable in practice, since the associated uncertainties for variable disturbance propagations can then be used for mitigating flow instabilities (You, Ku, and Yang 2013). Second, by incorporating known properties of the fluid flow into the model, the proposed emulator can in turn provide valuable insights on the dominant physics present in the system, which can then be used to guide further scientific investigations. One key example of this is the learning of dominant flow coupling mechanisms using a large cokriging model (Stein and Corsten 1991, Banerjee, Carlin, and Gelfand 2014) under sparsity constraints.

The article is structured as follows. Section 2 provides a brief overview of the physical model of concern, including injector design, governing equations, and experimental design. Section 3 introduces the proposed emulator model, and proposes a parallelized algorithm for efficient parameter estimation. Section 4 presents the emulation prediction and UQ for a new injector geometry, and interprets important physical correlations extracted by the emulator. Section 5 concludes with directions for future work.

2. Injector Schematic and Large Eddy Simulations

We first describe the design schematic for the swirl injector of concern, then briefly outline the governing partial differential equations and simulation tools. A discussion on experimental design is provided at the end of this section.
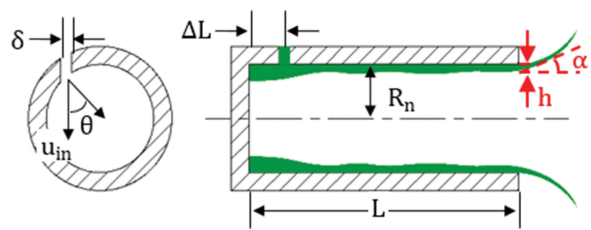

Figure 1. Schematic of injector configuration.

2.1. Injector Design

Figure 1 shows a schematic of the swirl injector under consideration. It consists of an open-ended cylinder and a row of tangential entries for liquid fluid injection. The configuration is typical of many propulsion and power-generation applications (Zong and Yang 2008, Wang et al. 2017a, Wang, Wang, and Yang 2017b). Liquid propellant is tangentially introduced into the injector and forms a thin film attached to the wall due to the swirl-induced centrifugal force. A low-density gaseous core exists in the center region in accordance with conservation of mass and angular momentum. The liquid film exits the injector as a thin sheet and mixes with the ambient gas. The swirl injection and atomization process involves two primary mechanisms: disintegration of the liquid sheet as it swirls and stretches, and sheet breakup due to the interaction with the surroundings. The design of the injector significantly affects the atomization characteristics and stability behaviors.

Figure 1 shows the five design variables considered for injector geometry: the injector length L, the nozzle radius R_n, the inlet diameter δ, the injection angle θ, and the distance between inlet and head-end ΔL. From flow physics, these five variables are influential for liquid film thickness h and spreading angle α (see Figure 1), which are key measures of injector performance of a swirl injector. For example, a larger injection angle θ induces greater swirl momentum in the liquid oxygen (LOX) flow, which in turn causes thinner film thickness and smaller spreading angle. Table 1 summarizes the design ranges for these five variables. To ensure the applicability of our work, broad geometric ranges are considered, covering design settings for several existing rocket injectors. Specifically, the range for injector length L covers the length of RD-0110 and RD-170 liquid-fuel rocket engines.

2.2. Flow Simulation

The numerical simulations here are performed with a pressure of 100 atm, which is typical of contemporary liquid rocket engines with LOX as the propellant. The physical processes modeled here are turbulent flows, in which various sizes of turbulent eddies are involved. A direct numerical simulation to resolve all eddy length-scales is computationally prohibitive. To

Table 1. Range of geometric parameters.

Parameter	Range
L	20 mm–100 mm
R_n	2.0 mm–5.0 mm
δ	0.5 mm–2.0 mm
θ	45°–75°
ΔL	1.0 mm–4.0 mm

this end, we employ the large eddy simulation (LES) technique, which directly simulates large turbulent eddies and employs a model-based approach for small eddies. To provide initial turbulence, broadband Gaussian noise is superimposed onto the inlet velocity components. Thermodynamic and transport properties are simulated using the techniques in Huo, Wang, and Yang (2014) and Wang, Huo, and Yang (2015); the theoretical and numerical framework can be found in Oefelein and Yang (1998) and Zong et al. (2004). To optimize computational speed, a multiblock domain decomposition technique combined with the message-passing interface for parallel computing is applied. Each LES simulation takes 6 days of computation time, parallelized over 200 CPU cores, to obtain $T = 1000$ snapshots with a time-step of 0.03 ms after the flow reaches statistically stationary state. From this, six flow variables of interest can be extracted: axial (u), radial (v), and circumferential (w) components of velocity, temperature (T), pressure (P), and density (ρ).

Numerical simulations are conducted for $n = 30$ injector geometries in the timeframe set for this project. These simulation runs are allocated over the design space in Table 1 using the maximum projection (MaxPro) design proposed by Joseph, Gul, and Ba (2015). Compared to Latin-hypercube-based designs (e.g., McKay, Beckman, and Conover 1979; Morris and Mitchell 1995), MaxPro designs enjoy better space-filling properties in all possible projections of the design space, and also provide better predictions for GP modeling. While $n = 30$ simulation runs may appear to be too small of a dataset for training the proposed flow emulator, we show this sample size can provide accurate flow predictions for the application at hand, through an elicitation of flow physics and the incorporation of such physics into the model. For these 30 runs, one issue which arises is that the simulation data are massive, requiring nearly a 100 GB in computer storage. For such large data, a blind application of existing flow kriging methods may require weeks for flow prediction, which entirely defeats the purpose of emulation, because simulated flows can generated in 6 days. Again, by properly eliciting and incorporating physics as simplifying assumptions for the emulator model, accurate flow predictions can be achieved in hours despite a limited run size. We elaborate on this elicitation procedure in the following section.

3. Emulator Model

We first introduce the new idea of CPOD, then present the proposed emulator model and a parallelized algorithm for parameter estimation. A key theme in this section (and indeed, for this article) is the elicitation and incorporation of flow physics within the emulator model. This not only allows for efficient and accurate flow predictions through simplifying model assumptions, but also provides a data-driven method for extracting useful flow physics, which can then guide future experiments. As demonstrated in Section 4, both objectives can be achieved despite limited runs and complexities inherent in flow data. Table 2 summarizes the elicited flow physics and the corresponding emulator assumptions; we discuss each point in greater detail below.

Table 2. Elicited flow physics and corresponding assumptions for the emulator model.

Flow physics	Model assumption
Coherent structures in turbulent flow (Lumley 1967)	POD-based kriging
Similar Reynolds numbers for cold-flows (Stokes 1851)	Linear-scaling modes in CPOD
Dense simulation time-steps	Time-independent emulator
Couplings between flow variables (Pope 2001)	Cokriging framework with covariance matrix **T**
Few-but-significant couplings (Pope 2001)	Sparsity on **T**$^{-1}$

3.1. Common POD

A brief overview of POD is first provided, following Lumley (1967). For a *fixed* injector geometry, let $Y(\mathbf{x}, t)$ denote a flow variable (e.g., pressure) at spatial coordinate $\mathbf{x} \in \mathbb{R}^2$ and flow time t. POD provides the following decomposition of $Y(\mathbf{x}, t)$ into separable spatial and temporal components:

$$Y(\mathbf{x}, t) = \sum_{k=1}^{\infty} \beta_k(t) \phi_k(\mathbf{x}), \tag{1}$$

with the spatial eigenfunctions $\{\phi_k(\mathbf{x})\}_{k=1}^{\infty}$ and temporal coefficients $\{\beta_k(t)\}_{k=1}^{\infty}$ given by

$$\phi_k(\mathbf{x}) = \underset{\|\psi\|_2 = 1, \langle \psi, \phi_l \rangle = 0, \forall l < k}{\operatorname{argmax}} \int \left\{ \int Y(\mathbf{x}, t) \psi(\mathbf{x}) \, d\mathbf{x} \right\}^2 \, dt,$$
$$\beta_k(t) = \int Y(\mathbf{x}, t) \phi_k(\mathbf{x}) \, d\mathbf{x}. \tag{2}$$

Following Berkooz, Holmes, and Lumley (1993), we refer to $\{\phi_k(\mathbf{x})\}_{k=1}^{\infty}$ as the *spatial POD modes* for $Y(\mathbf{x}, t)$, and its corresponding coefficients $\{\beta_k(t)\}_{k=1}^{\infty}$ as *time-varying coefficients*.

There are two key reasons for choosing POD over other reduced-basis models. First, one can show (Loève 1955) that any truncated representation in (1) gives the best flow reconstruction of $Y(\mathbf{x}, t)$ in L_2-norm, compared to any other linear expansion of space/time products with the same number of terms. This property is crucial for our application, since it allows the massive simulation data to be optimally reduced to a smaller training dataset for the proposed emulator. Second, the POD has a special interpretation in terms of turbulent flow. In the seminal article by Lumley (1967), it is shown that, under certain conditions, the expansion in (1) can extract *physically meaningful* coherent structures that govern turbulence instabilities. For this reason, physicists use POD as an experimental tool to pinpoint key flow instabilities, simply through an inspection of $\phi_k(\mathbf{x})$ and the dominant frequencies in $\beta_k(t)$. For example, using POD analysis, Zong and Yang (2008) showed that the two flow phenomena, hydrodynamic wave propagation on LOX film and vortex core excitation near the injector exit, are the key mechanisms driving flow instability. This is akin to the use of principal components in regression, which can yield meaningful results in applications where such components have innate interpretability.

Unfortunately, POD is only suitable for extracting instability structures at a *single* geometry, whereas for emulation, a
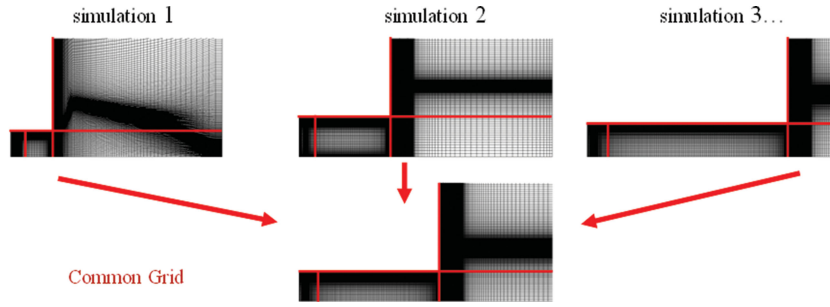
Figure 2. Common grid using linearity assumption for CPOD.

method is needed that can extract common structures over *varying* geometries. With this in mind, we propose a new decomposition called common POD (CPOD). The key assumption of CPOD is that, under a *physics-guided partition* of the computational domain, the spatial distribution of coherent structures *scales linearly* over varying injector geometries. For cold flows, this can be justified by similar Reynolds numbers (which characterize flow dynamics) over different geometries (Stokes 1851). This is one instance of model simplification through elicitation, because such a property likely does not hold for general flows. This linearity assumption is highly valuable for computational efficiency, because flows from different geometries can then be rescaled onto a common spatial grid for instability extraction. Figure 2 visualizes this procedure. The grids for each simulation are first split into four parts: from injector head-end to the inlet, from the inlet to the nozzle exit, and the top and bottom portions of the downstream region. Each part is then proportionally rescaled to a common, reference grid according to changes in the geometric variables L, R_n, and ΔL (see Figure 1). From a physics perspective, such a partition is necessary for the linearity assumption to hold.

Stating this mathematically, let $\mathbf{c}_1, \ldots, \mathbf{c}_n \in \mathbb{R}^p$ be the n simulated geometries, let $Y(\mathbf{x}, t; \mathbf{c}_i)$ be the simulated flow at setting $\mathbf{c}_i$, and fix some setting $\mathbf{c} \in \{\mathbf{c}_i\}_{i=1}^n$ as the geometry for the common grid. Next, define $\mathcal{M}_i : \mathbb{R}^2 \to \mathbb{R}^2$ as the linear map, which rescales spatial modes on the common geometry $\mathbf{c}$ back to the ith simulated geometry $\mathbf{c}_i$ according to geometric changes in L, R_n, and ΔL. $\mathcal{M}_i$ can be viewed as the inverse map of the procedure described in the previous paragraph and visualized in Figure 2, which rescales modes from $\mathbf{c}_i$ to the common geometry $\mathbf{c}$ (see Appendix A.1 for details). CPOD provides the following decomposition of $Y(\mathbf{x}, t; \mathbf{c}_i)$:

$$Y(\mathbf{x}, t; \mathbf{c}_i) = \sum_{k=1}^{\infty} \beta_k(t; \mathbf{c}_i) \mathcal{M}_i\{\phi_k(\mathbf{x})\}, \qquad (3)$$

with the spatial CPOD modes $\{\phi_k(\mathbf{x})\}$ and time-varying coefficients $\{\beta_k(t; \mathbf{c}_i)\}$ defined as

$$\phi_k(\mathbf{x}) = \underset{\substack{\|\psi\|_2=1, \\ \langle\psi,\phi_l\rangle=0,\forall l<k}}{\operatorname{argmax}} \sum_{i=1}^{n} \int \left\{\int Y(\mathbf{x}, t; \mathbf{c}_i)\mathcal{M}_i\{\psi(\mathbf{x})\} \, d\mathbf{x}\right\}^2 \, dt,$$

$$\beta_k(t; \mathbf{c}_i) = \int Y(\mathbf{x}, t; \mathbf{c}_i)\mathcal{M}_i\{\phi_k(\mathbf{x})\} \, d\mathbf{x}. \qquad (4)$$

Here, $\phi_k(\mathbf{x})$ is the spatial distribution for the kth common flow structure, with $\beta_k(t; \mathbf{c}_i)$ its time-varying coefficient for geometry $\mathbf{c}_i$. As in POD, leading terms in CPOD can also be

interpreted in terms of flow physics, a property we demonstrate later in Section 4. CPOD, therefore, not only provides optimal data-reduction for the simulation data, but also extracts physically meaningful structures which can then be incorporated for emulation.

Algorithmically, the CPOD expansion can be computed by rescaling and interpolating all flow simulations to the common grid, computing the POD expansion, and then rescaling the resulting modes back to their original grids. Interpolation is performed using the inverse distance weighting method in Shepard (1968), and can be justified by dense spatial resolution of the data (with around 100,000 grid points for each simulation). Letting T be the total number of time-steps, a naive implementation of this decomposition requires $O(n^3 T^3)$ work, due to a singular-value-decomposition step. Such a decomposition therefore becomes computationally intractable when the number of runs grows large or when simulations have dense time-steps (as is the case here). To avoid this computational issue, we use an iterative technique from Lehoucq, Sorensen, and Yang (1998) called the implicitly restarted Arnoldi method, which approximates leading terms in (3) using periodically restarted Arnoldi decompositions. The full algorithm for CPOD is outlined in Appendix A.

3.2. Model Specification

After the CPOD extraction, the extracted time-varying coefficients $\{\beta_k(t; \mathbf{c}_i)\}_{i,k}$ are then used as data for fitting the proposed emulator. There has been some existing work on dynamic emulator models, such as Conti et al. (2009), Liu and West (2009), and Conti and O'Hagan (2010), but the sheer number of simulation time-steps here can impose high computation times and numerical instabilities for these existing methods (Hung, Joseph, and Melkote 2015). As mentioned previously, computational efficiency is paramount for our problem, since simulation runs can be performed within a week. Moreover, existing emulators cannot account for cross-correlations between different dynamic systems, while the flow physics represented by different CPOD modes are known to be highly coupled from governing equations. Here, we exploit the dense temporal resolution of the flow by using a *time-independent*(TI) emulator that employs independent kriging models at each slice of time. The rationale is that, because time-scales are so fine, there is no practical need to estimate temporal correlations (even when they exist), since prediction is not required between time-steps. This TI simplification is key for emulator efficiency, since it allows us to fully exploit the power of parallel computing for model fitting and flow prediction.

The model is as follows. Suppose R flow variables are considered (with $R = 6$ in the present case), and the CPOD expansion in (3) is truncated at K_r terms for flow $r = 1, \ldots, R$. Let $\boldsymbol{\beta}^{(r)}(t; \mathbf{c}) = (\beta_1^{(r)}(t; \mathbf{c}), \ldots, \beta_{K_r}^{(r)}(t; \mathbf{c}))^T$ be the vector of K_r time-varying coefficients for flow variable r at design setting $\mathbf{c}$, with $\boldsymbol{\beta}(t; \mathbf{c}) = (\boldsymbol{\beta}^{(1)}(t; \mathbf{c})^T, \ldots, \boldsymbol{\beta}^{(R)}(t; \mathbf{c})^T)^T$ the coefficient vector for all flows at $\mathbf{c}$. We assume the following TI GP model on $\boldsymbol{\beta}(t; \mathbf{c})$:

$$\boldsymbol{\beta}(t; \mathbf{c}) \sim GP\{\boldsymbol{\mu}(t), \boldsymbol{\Sigma}(\cdot, \cdot; t)\}, \quad \boldsymbol{\beta}(t; \mathbf{c}) \perp \boldsymbol{\beta}(t'; \mathbf{c}) \text{ for } t \neq t'. \tag{5}$$

Here, $K = \sum_{r=1}^R K_r$ is the number of extracted modes over all R flow variables, $\boldsymbol{\mu} \in \mathbb{R}^K$ is the process mean vector, and $\boldsymbol{\Sigma}(\cdot, \cdot) : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}^{K \times K}$ its corresponding covariance matrix function defined below. Since the GPs are now TI, we present the specification for *fixed* time t, and refer to $\boldsymbol{\beta}(t; \mathbf{c})$, $\boldsymbol{\mu}(t)$ and $\boldsymbol{\Sigma}(\cdot, \cdot; t)$ as $\boldsymbol{\beta}(\mathbf{c})$, $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}(\cdot, \cdot)$ for brevity.

For computational efficiency, the following separable form is assumed for $\boldsymbol{\Sigma}(\cdot, \cdot)$:

$$\boldsymbol{\Sigma}(\mathbf{c}_1, \mathbf{c}_2) = r_\tau(\mathbf{c}_1, \mathbf{c}_2)\mathbf{T}, \quad r_\tau(\mathbf{c}_1, \mathbf{c}_2) = \prod_{j=1}^p \tau_j^{4(c_{1j} - c_{2j})^2},$$
$$\mathbf{c}_1, \mathbf{c}_2 \in \mathbb{R}^p, \quad \tau_j \in (0, 1), \tag{6}$$

where $\mathbf{T} \in \mathbb{R}^{K \times K}$ is a symmetric, positive definite matrix called the *CPOD covariance matrix*, and $r_\tau(\cdot, \cdot)$ is the correlation function over the design space, parameterized by $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_p)^T \in (0, 1)^p$. This can be viewed as a large cokriging model (Stein and Corsten 1991) over the design space, with the multivariate observations being the extracted CPOD coefficients for all flow variables. Note that r_τ is a reparameterization of the squared-exponential (or Gaussian) correlation function $\exp\{-\sum_{j=1}^p \theta_j(c_{1j} - c_{2j})^2\}$, with $\theta_j = -4\log\tau_j$. In our experience, such a reparameterization allows for a more numerically stable optimization of maximum likelihood estimations (MLEs), because the optimization domain $\tau_j \in (0, 1)$ is now bounded. Our choice of the Gaussian correlation is also well-justified for the application at hand, since fully developed turbulence dynamics are known to be relatively smooth.

Suppose simulations are run at settings $\mathbf{c}_1, \ldots, \mathbf{c}_n$, and assume for now that model parameters are known. Invoking the conditional distribution of the multivariate normal distribution, the time-varying coefficients at a new setting $\mathbf{c}_{\text{new}}$ follow the distribution:

$$\boldsymbol{\beta}(\mathbf{c}_{\text{new}})|\{\boldsymbol{\beta}(\mathbf{c}_i)\}_{i=1}^n$$
$$\sim \mathcal{N}\Big(\boldsymbol{\mu} + (\mathbf{T} \otimes \mathbf{r}_{\tau,\text{new}})^T (\mathbf{T}^{-1} \otimes \mathbf{R}_\tau^{-1})(\boldsymbol{\beta} - \mathbf{1}_n \otimes \boldsymbol{\mu}),$$
$$\mathbf{T} - (\mathbf{T} \otimes \mathbf{r}_{\tau,\text{new}})^T (\mathbf{T}^{-1} \otimes \mathbf{R}_\tau^{-1})(\mathbf{T} \otimes \mathbf{r}_{\tau,\text{new}})\Big), \tag{7}$$

where $\mathbf{r}_{\tau,\text{new}} = (r_\tau(\mathbf{c}_{\text{new}}, \mathbf{c}_1), \ldots, r_\tau(\mathbf{c}_{\text{new}}, \mathbf{c}_n))^T$ and $\mathbf{R}_\tau = [r_\tau(\mathbf{c}_i, \mathbf{c}_j)]_{i=1}^n {}_{j=1}^n$. Using algebraic manipulations, the minimum-MSE (MMSE) predictor for $\boldsymbol{\beta}(\mathbf{c}_{\text{new}})|\{\boldsymbol{\beta}(\mathbf{c}_i)\}_{i=1}^n$ and its corresponding variance is given by

$$\hat{\boldsymbol{\beta}}(\mathbf{c}_{\text{new}}) = \boldsymbol{\mu} + ((\mathbf{r}_{\tau,\text{new}}^T \mathbf{R}_\tau^{-1}) \otimes \mathbf{I}_K)(\boldsymbol{\beta} - \mathbf{1}_n \otimes \boldsymbol{\mu}),$$
$$\mathbb{V}\{\boldsymbol{\beta}(\mathbf{c}_{\text{new}})|\{\boldsymbol{\beta}(\mathbf{c}_i)\}_{i=1}^n\} = (1 - \mathbf{r}_{\tau,\text{new}}^T \mathbf{R}_\tau^{-1} \mathbf{r}_{\tau,\text{new}})\mathbf{T},$$
$$\tag{8}$$

where $\mathbf{I}_K$ and $\mathbf{1}_n$ denote a $K \times K$ identity matrix and a 1-vector of n elements, respectively. Substituting this into the CPOD expansion (3), the predicted rth flow variable becomes

$$\hat{Y}^{(r)}(\mathbf{x}, t; \mathbf{c}_{\text{new}}) = \sum_{k=1}^{K_r} \hat{\beta}_k^{(r)}(\mathbf{c}_{\text{new}})\mathcal{M}_{\text{new}}\{\phi_k^{(r)}(\mathbf{x})\}, \tag{9}$$

with the associated spatio-temporal variance

$$\mathbb{V}\{Y^{(r)}(\mathbf{x}, t; \mathbf{c}_{\text{new}})|\{Y^{(r)}(\mathbf{x}, t; \mathbf{c}_i)\}_{i=1}^n\}$$
$$= \sum_{k=1}^{K_r} \mathbb{V}\{\beta_k^{(r)}(\mathbf{c}_{\text{new}})|\{\boldsymbol{\beta}(\mathbf{c}_i)\}_{i=1}^n\}\Big[\mathcal{M}_{\text{new}}\{\phi_k^{(r)}(\mathbf{x})\}\Big]^2, \tag{10}$$

where $\phi_k^{(r)}(\mathbf{x})$ is the kth CPOD mode for flow variable r. This holds because the CPOD modes for a fixed flow variable are orthogonal (see Section 3.1).

It is worth noting that, when model parameters are known, the MMSE predictor in (8) from the proposed cokriging model (which we call M_A) is the same as the MMSE predictor from the simpler *independent* GP model with $\mathbf{T}$ diagonal (which we call M_0). One advantage of the cokriging model M_A, however, is that it provides improved UQ compared to the independent model M_0, as we show below. Moreover, the MMSE predictor for a derived function g of the flow can be quite different between M_A and M_0. This is demonstrated in the study of turbulent kinetic energy (TKE) in Section 4.3.

3.2.1. CPOD Covariance Matrix

We briefly describe why the CPOD covariance matrix $\mathbf{T}$ is appealing from both a physical and a statistical perspective. From the underlying governing equations, it is well known that certain dynamic behaviors are strongly *coupled* for different flow variables (Pope 2001). For example, pressure oscillation in the form of acoustic waves within an injector can induce velocity and density fluctuations. In this sense, $\mathbf{T}$ incorporates knowledge of these physical couplings within the emulator itself, with $\mathbf{T}_{ij} \gg 0$ indicating the presence of a significant coupling between modes i and j, and vice versa. The covariance selection and estimation of $\mathbf{T}$, therefore, provide a data-driven way to *extract* and *rank* significant flow couplings, which is of interest in itself and can be used to guide further experiments. Note that the block submatrices of $\mathbf{T}$ corresponding to the same flow variable (marked in red in Figure 3) should be diagonal, by the orthogonality of CPOD modes.
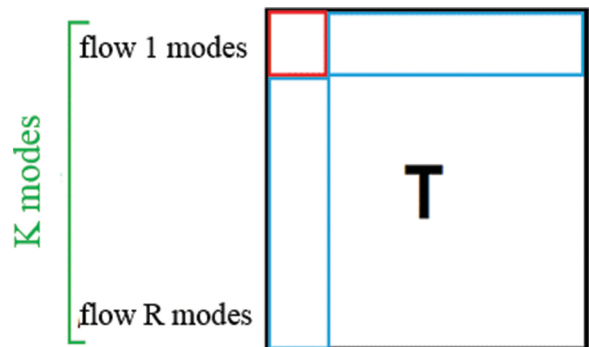

Figure 3. Illustration of the CPOD correlation matrix $\mathbf{T}$. Red indicates a diagonal matrix, while blue indicates nondiagonal entries.

The CPOD covariance matrix $\mathbf{T}$ also plays an important statistical role in emulation. Specifically, when significant cross-correlations exist between modes (which we know to be true from the flow couplings imposed by governing equations), the incorporation of this correlation structure within our model ought to provide a more accurate quantification of uncertainty. This is indeed true, and is made precise by the following theorem.

Theorem 3.1. Consider the two models $M_0 : \boldsymbol{\beta}(\mathbf{c}) \in \mathbb{R}^K \sim GP\{\boldsymbol{\mu}, \boldsymbol{\Sigma}^{(0)}\}$ and $M_A : \boldsymbol{\beta}(\mathbf{c}) \sim GP\{\boldsymbol{\mu}, \boldsymbol{\Sigma}^{(A)}\}$, where $\boldsymbol{\Sigma}^{(0)}(\mathbf{c}_1, \mathbf{c}_2) = r_\tau(\mathbf{c}_1, \mathbf{c}_2)\mathbf{D}$ and $\boldsymbol{\Sigma}^{(A)}(\mathbf{c}_1, \mathbf{c}_2) = r_\tau(\mathbf{c}_1, \mathbf{c}_2)\mathbf{T}$ with $\mathbf{T} \succeq 0$ and $\mathbf{D} = \text{diag}\{\mathbf{T}\}$. Let C_0 be the $100(1-\alpha)\%$ highest-density confidence region (HDCR, see Hyndman 1996) of $\boldsymbol{\beta}(\mathbf{c}_{\text{new}})|\{\boldsymbol{\beta}(\mathbf{c}_i)\}_{i=1}^n$ under M_0. Suppose $\lambda_{\min}(\mathbf{T}^{1/2}\mathbf{D}^{-1}\mathbf{T}^{1/2}) > 1$. Then

$$\mathbb{P}\left\{\boldsymbol{\beta}(\mathbf{c}_{\text{new}}) \in C_0 | M_A, \{\boldsymbol{\beta}(\mathbf{c}_i)\}_{i=1}^n\right\} < 1 - \alpha.$$

Proof. For brevity, let $\boldsymbol{\beta} \equiv \boldsymbol{\beta}(\mathbf{c}_{\text{new}})|\{\boldsymbol{\beta}(\mathbf{c}_i)\}_{i=1}^n$, and let $\hat{\boldsymbol{\beta}} \equiv \mathbb{E}[\boldsymbol{\beta}(\mathbf{c}_{\text{new}})|\{\boldsymbol{\beta}(\mathbf{c}_i)\}_{i=1}^n]$. Letting $\mathbf{Z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_K)$, it is easy to show that

$$\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}|M_0 \sim \mathcal{N}\left\{\mathbf{0}, \left(1 - \mathbf{r}_{\tau,\text{new}}^T \mathbf{R}_\tau^{-1} \mathbf{r}_{\tau,\text{new}}\right)\mathbf{D}\right\}$$
$$\stackrel{d}{=} \sqrt{1 - \mathbf{r}_{\tau,\text{new}}^T \mathbf{R}_\tau^{-1} \mathbf{r}_{\tau,\text{new}}}\mathbf{D}^{1/2}\mathbf{Z}, \quad \text{and}$$

$$\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}|M_A \sim \mathcal{N}\left\{\mathbf{0}, \left(1 - \mathbf{r}_{\tau,\text{new}}^T \mathbf{R}_\tau^{-1} \mathbf{r}_{\tau,\text{new}}\right)\mathbf{T}\right\}$$
$$\stackrel{d}{=} \sqrt{1 - \mathbf{r}_{\tau,\text{new}}^T \mathbf{R}_\tau^{-1} \mathbf{r}_{\tau,\text{new}}}\mathbf{T}^{1/2}\mathbf{Z}.$$

Under the independent model M_0, the $100(1-\alpha)\%$ HDCR becomes

$$C_0 = \{\boldsymbol{\xi} : \left(1 - \mathbf{r}_{\tau,\text{new}}^T \mathbf{R}_\tau^{-1} \mathbf{r}_{\tau,\text{new}}\right)^{-1}(\boldsymbol{\xi} - \hat{\boldsymbol{\beta}})^T \mathbf{D}^{-1}(\boldsymbol{\xi} - \hat{\boldsymbol{\beta}})$$
$$\leq \chi_K^2(1 - \alpha)\},$$

where $\chi_K^2(1 - \alpha)$ be the $(1-\alpha)$-quantile of a χ^2-distribution with K degrees of freedom. Now, let $\lambda_{\min}$ denote the minimum eigenvalue of $\mathbf{T}^{1/2}\mathbf{D}^{-1}\mathbf{T}^{1/2}$. It follows that

$$\mathbb{P}\left(\boldsymbol{\beta} \in C_0 | M_A\right) = \mathbb{P}\left\{(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})^T \mathbf{D}^{-1}(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})\right.$$
$$\left. \leq \left(1 - \mathbf{r}_{\tau,\text{new}}^T \mathbf{R}_\tau^{-1} \mathbf{r}_{\tau,\text{new}}\right)\chi_K^2(1 - \alpha)\Big| M_A\right\}$$
$$= \mathbb{P}\left\{\mathbf{Z}^T(\mathbf{T}^{1/2}\mathbf{D}^{-1}\mathbf{T}^{1/2})\mathbf{Z} \leq \chi_K^2(1 - \alpha)\right\}$$
$$\leq \mathbb{P}\left\{\mathbf{Z}^T\mathbf{Z} \leq \lambda_{\min}^{-1}\chi_K^2(1 - \alpha)\right\},$$

since $\mathbf{Z}^T(\mathbf{T}^{1/2}\mathbf{D}^{-1}\mathbf{T}^{1/2})\mathbf{Z} \geq \lambda_{\min}\mathbf{Z}^T\mathbf{Z}$ almost surely. The asserted result follows because $\mathbb{P}\{\mathbf{Z}^T\mathbf{Z} \leq \lambda_{\min}^{-1}\chi_K^2(1 - \alpha)\}$ is strictly less than $1 - \alpha$ when $\lambda_{\min} > 1$. $\square$

In words, this theorem quantifies the effect on coverage probability when the true cokriging model M_A, which accounts for cross-correlations between modes, is misspecified as M_0, the independent model ignoring such cross-correlations. Note that an increase in the number of significant nonzero cross-correlations in $\mathbf{T}$ causes $\mathbf{T}^{1/2}\mathbf{D}^{-1}\mathbf{T}^{1/2}$ to deviate further from unity, which in turn may increase $\lambda_{\min}$. Given enough such correlations, Theorem 3.1 shows that the coverage probability from the misspecified model M_0 is less than the desired $100(1 - \alpha)\%$ rate. In the present case, this suggests that when there are enough significant flow couplings, the cokriging model M_A provides

more accurate UQ for the *joint* prediction of flow variables when compared to the misspecified, independent model M_0. This improvement also holds for functions of flow variables (as we demonstrate later in Section 4), although a formal argument is not presented here.

It is important to mention here an important trade-off for cokriging models in general, and why the proposed model is appropriate for the application at hand in view of such a trade-off. It is known from spatial statistics literature (see, e.g., Banerjee, Carlin, and Gelfand 2014, Mak, Bingham, and Lu 2016) that when the matrix $\mathbf{T}$ exhibits strong correlations and can be estimated well, one enjoys improved predictive performance through a cokriging model (this is formally shown for the current model in Theorem 3.1). However, when such correlations are absent or cannot be estimated well, a cokriging model can yield poorer performance to an independent model! We claim that the former is true for the current application at hand. First, the differential equations governing the simulation procedure explicitly impose strong dependencies between flow variables, so we know a priori the existence of strong correlations in $\mathbf{T}$. Second, we will show later in Section 4.4 that the dominant correlations selected in $\mathbf{T}$ are physically interpretable in terms of fluid mechanic principles and conservation laws, which provides strong evidence for the correct estimation of $\mathbf{T}$.

One issue with fitting M_A is that there are many more parameters to estimate. Specifically, since the CPOD covariance matrix $\mathbf{T}$ is $K \times K$ dimensional, there is insufficient data for estimating all entries in $\mathbf{T}$ using the extracted coefficients from the CPOD expansion. One solution is to impose the sparsity constraint $\|\mathbf{T}^{-1}\|_1 \leq \gamma$, where $\|\mathbf{A}\|_1 = \sum_{k=1}^K \sum_{l=1}^K |A_{kl}|$ is the element-wise L_1 norm. For a small choice of γ, this forces nearly all entries in $\mathbf{T}^{-1}$ to be zero, thus permitting consistent estimation of the few significant correlations. Sparsity can also be justified from an engineering perspective, because the number of significant couplings is known to be small from flow physics. γ can also be adjusted to extract a prespecified number of flow couplings, which is appealing from an engineering point-of-view. The justification for sparsifying $\mathbf{T}^{-1}$ instead of $\mathbf{T}$ is largely computational, because, algorithmically, the former problem can be handled much more efficiently than the latter using the graphical least absolute shrinkage and selection operator (LASSO) (Friedman, Hastie, and Tibshirani 2008; see also Bien and Tibshirani 2011). Such efficiency is crucial here, since GP parameters need to be jointly estimated as well. Although the proposed model is similar to the one developed in Qian, Wu, and Wu (2008) for emulating qualitative factors, there are two key distinctions. First, our model allows for different process variances for each coefficient, whereas their approach restricts all coefficients to have equal variances. Second, our model incorporates sparsity on the CPOD covariance matrix, an assumption necessary from a statistical point-of-view and appealing from a physics extraction perspective. Last, the algorithm proposed below can estimate $\mathbf{T}$ more efficiently than the semidefinite programming approach in Qian, Wu, and Wu (2008).

3.3. Parameter Estimation

To estimate the model parameters $\boldsymbol{\mu}$, $\mathbf{T}$, and $\boldsymbol{\tau}$, MLE is used in favor of a Bayesian implementation. The primary reason for this

choice is computational efficiency: for the proposed emulator to be used as a fast investigative tool for surveying the design space, it should generate flow predictions much quicker than a direct LES simulation, which requires several days of parallelized computation.

Algorithm 1 BCD algorithm for maximum likelihood estimation

1: **for** each time-step $t = 1, \cdots, T$ **do parallel**
2:    • Set initial values $\mu \leftarrow \mathbf{0}_K$, $T \leftarrow I_K$ and $\tau \leftarrow \mathbf{1}_p$, and set $B \leftarrow (\beta(c_1), \cdots, \beta(c_n))^T$
3:    **repeat**
4:       Optimizing T:
5:       • Set $W \leftarrow \frac{1}{n}(B - \mathbf{1}_n \otimes \mu^T)^T R_\tau^{-1}(B - \mathbf{1}_n \otimes \mu^T) + \lambda \cdot I_K$
6:       **repeat**
7:          **for** $j = 1, \cdots, K$ **do**
8:             • Solve $\tilde{\delta} = \text{argmin}_\delta \left\{ \frac{1}{2} \|W_{-j,-j}^{1/2} \delta\|_2^2 + \lambda \|\delta\|_1 \right\}$ using LASSO
9:             • Update $W_{-j,j} \leftarrow W_{-j,-j}\tilde{\delta}$ and $W_{j,-j}^T \leftarrow W_{-j,-j}\tilde{\delta}$
10:       **until** W converges
11:       • Update $T \leftarrow W^{-1}$
12:       Optimizing μ and τ:
13:       • Update $\tau \leftarrow \text{argmin}_\tau l_\lambda(\mu_\tau, T, \tau)$ with L-BFGS, with $\mu_\tau = (\mathbf{1}_n^T R_\tau^{-1} \mathbf{1}_n)^{-1}(\mathbf{1}_n^T R_\tau^{-1} B)$
14:       • Update $\mu \leftarrow \mu_\tau$
15:    **until** μ, T and τ converge
16: **end parallel for**
17: • **return** $\mu(t)$, $T(t)$ and $\tau(t)$

From (5) and (6), the maximum likelihood formulation can be written as $\text{argmin}_{\mu, T, \tau}\, l_\lambda(\mu, T, \tau)$, where $l_\lambda(\mu, T, \tau)$ is the *penalized* negative log-likelihood

$$l_\lambda(\mu, T, \tau) = n \log \det T + K \log \det R_\tau$$
$$+ (B - \mathbf{1}_n \otimes \mu)^T [R_\tau^{-1} \otimes T^{-1}]$$
$$\times (B - \mathbf{1}_n \otimes \mu) + \lambda \|T^{-1}\|_1. \tag{11}$$

Note that, because the formulation is convex in T^{-1}, the sparsity constraint $\|T^{-1}\|_1 \leq \gamma$ has been incorporated into the likelihood through the penalty $\lambda \|T^{-1}\|_1$ using strong duality. Similar to γ, a larger λ results in a smaller number of selected correlations, and vice versa. The tuning method for λ should depend on the desired end-goal. For example, if predictive accuracy is the primary goal, then λ should be tuned using cross-validation techniques (Hastie, Tibshirani, and Friedman 2009). However, if correlation extraction is desired or prior information is available on flow couplings, then λ should be set so that a fixed (preset) number of correlations is extracted. We discuss this further in Section 4.

Assume for now a fixed penalty $\lambda > 0$. To compute the MLEs in (11), we propose the following *blockwise coordinate descent* (BCD) algorithm. First, assign initial values for μ, T, and τ. Next, iterate the following two updates until parameters converge: (a) for fixed GP parameters μ and τ, optimize for T in (11); and (b) for fixed covariance matrix T, optimize for μ and τ in (11). With the use of the graphical LASSO algorithm from Friedman, Hastie, and Tibshirani (2008), the first update can be computed efficiently. The second update can be computed using

nonlinear optimization techniques on τ by means of a closed-form expression for μ. In our implementation, this is performed using the L-BFGS algorithm (Liu and Nocedal 1989), which offers a super-linear convergence rate without the cumbersome evaluation and manipulation of the Hessian matrix (Nocedal and Wright 2006). The following theorem guarantees that the proposed algorithm converges to a stationary point of (11) (see Appendix B for proof).

Theorem 3.2. The BCD scheme in Algorithm 1 converges to some solution $(\hat{\mu}, \hat{T}, \hat{\tau})$, which is stationary for the penalized log-likelihood $l_\lambda(\mu, T, \tau)$.

It is worth noting that the proposed algorithm does not provide global optimization. This is not surprising, because the log-likelihood l_λ is nonconvex in τ. To this end, we run multiple threads of Algorithm 1 in parallel, each with a different initial point τ_0 from a large space-filling design on $[10^{-3}, 1 - 10^{-3}]^p$, then choose the converged parameter setting, which yields the largest likelihood value from (11). In our experience, this heuristic performs quite well in practice.

4. Emulation Results

In this section, we present in four parts the emulation performance of the proposed model, when trained using the database of $n = 30$ flow simulations described in Section 2. First, we briefly introduce key flow characteristics for a swirl injector, and physically interpret the flow structures extracted from CPOD. Second, we compare the numerical accuracy of our flow prediction with a validation simulation at a new injector geometry. Third, we provide a spatio-temporal quantification of uncertainty for our prediction, and discuss its physical interpretability. Last, we summarize the extracted flow couplings from T, and explain why these are both intuitive and intriguing from a flow physics perspective.

4.1. Visualization and CPOD Modes

We employ three flow snapshots of circumferential velocity (shown in Figure 4) to introduce key flow characteristics for a swirl injector: the fluid transition region, spreading angle, surface wave propagation, and center recirculation. These characteristics will be used for assessing emulator accuracy, UQ, and extracted flow physics.

- *Fluid transition region:* The fluid transition region is the region that connects compressed-liquid near the wall (colored blue in Figure 4) to light-gas (colored red) near the centerline at supercritical pressure (Wang et al. 2017a). This region is crucial for analyzing injector flow characteristics, as it provides the instability propagation and feedback mechanisms between the injector inlet and exit. An important emulation goal is to accurately predict both the spatial location of this region and its dynamics, because such information can be used to assess feedback behavior at new geometries.

- *Spreading angle:* The spreading angle α (along with the LOX film thickness h) is an important physical metric for
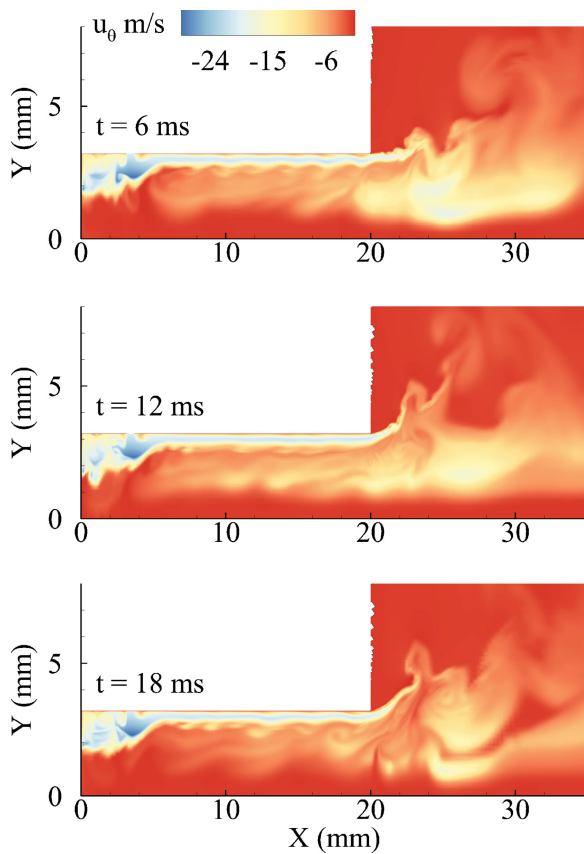
Figure 4. Flow snapshots of circumferential velocity at $t = 6$, 12, and 18 ms.

measuring the performance of a swirl injector. A larger α and smaller h indicate better performance of injector atomization and breakup processes. The spreading angle can be seen in Figure 4 from the blue LOX flow at injector exit (see Figure 1 for details).

- *Surface wave propagation:* Surface waves, which transfer energy through the fluid medium, manifest themselves as wavy structures in the flowfield. These waves allow for propagation of flow instabilities between upstream and downstream regions of the injector, and can be seen in the first snapshot of Figure 4 along the LOX film boundary.
- *Center recirculation:* Center recirculation, another key instability structure, is the circular flow of a fluid around a rotational axis (this circular region is known as the vortex core). From the third snapshot in Figure 4, a large vortex core (in white) can be seen at the injector exit, which is expected because of sudden expansion of the LOX stream and subsequent generation of adverse pressure gradient.

Regarding the CPOD expansion, Figure 5 shows the energy ratio captured using the leading M terms in (3) for circumferential velocity, with this ratio defined as follows:

$$\xi(M) = \frac{\sum_{k=1}^{M} \sum_{i=1}^{n} \int \left[\int \beta_k(t; \mathbf{c}_i) \mathcal{M}_i\{\phi_k(\mathbf{x})\} \, d\mathbf{x} \right]^2 \, dt}{\sum_{k=1}^{\infty} \sum_{i=1}^{n} \int \left[\int \beta_k(t; \mathbf{c}_i) \mathcal{M}_i\{\phi_k(\mathbf{x})\} \, d\mathbf{x} \right]^2 \, dt}.$$

Only $M = 10$ and $M = 45$ modes are needed to capture 90% and 99% of the total flow energy over *all* $n = 30$ simulation cases, respectively. Compared to a similar experiment in Zong and Yang (2008), which required around $M = 20$ modes to capture 99% flow energy for a *single* geometry, the current results
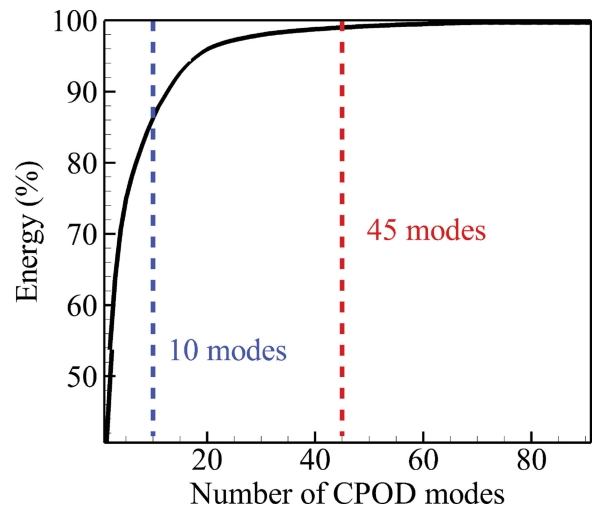

Figure 5. Energy distribution of CPOD modes for circumferential velocity flow.

are very promising, and show that the CPOD gives a reasonably compact representation. This also gives empirical evidence for the linearity assumption used for computation efficiency. Similar results also hold for other flow variables as well, and are not reported for brevity. Additionally, the empirical study in Zong and Yang (2008) showed that the POD modes capturing the top 95% energy have direct physical interpretability in terms of known flow instabilities. To account for these (and perhaps other) instability structures in the model, we set the truncation limit K_r as the smallest value of M satisfying $\xi(M) \geq 99\%$, which appears to provide a good balance between predictive accuracy and computational efficiency.

The extracted CPOD terms can also be interpreted in terms of flow physics. We illustrate this using the leading two CPOD terms for circumferential velocity, whose spatial distributions are shown in Figure 6. Upon an inspection of these spatial plots and their corresponding spectral frequencies, both modes can be identified as hydrodynamic instabilities in the form of longitudinal waves propagating along the LOX film boundary. Specifically, the first mode corresponds to the first harmonic mode for this wave, and the second mode represents the second harmonic
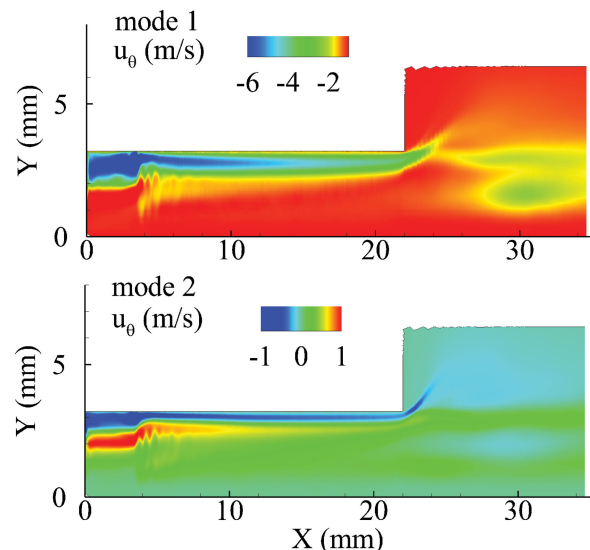

Figure 6. The leading two spatial CPOD modes for circumferential velocity flow.

and shows the existence of an antinode in wave propagation. As we show in Section 4.4, the interpretability of CPOD modes allows the proposed model to extract physically meaningful couplings for further analysis.

4.2. Emulation Accuracy

To ensure that our emulator model provides accurate flow predictions, we perform a validation simulation at the new geometric setting: $L = 22$ mm, $R_n = 3.215$ mm, $\Delta L = 3.417$ mm, $\theta = 58.217°$, and $\delta = 0.576$ mm. This new geometry provides a 10% variation on an existing injector used in the RD-0110 liquid-fuel engine (Yang and Anderson 1995). Since the goal is predictive accuracy, the sparsity penalty λ in (11) is tuned using 5-fold cross-validation (Hastie, Tibshirani, and Friedman 2009). We provide below a qualitative comparison of the predicted and simulated flows, and then discuss several metrics for quantifying emulation accuracy.

Figure 7 shows three snapshots of the simulated and predicted fully developed flows for temperature, in intervals of 1.5 ms starting at 21.75 ms. From visual inspection, the predicted flow closely mimics the simulated flow on several performance metrics, including the fluid transition region, film thickness, and spreading angle. The propagation of surface waves is also captured quite well within the injector, with key downstream recirculation zones correctly identified in the prediction as well. This comparison illustrates the effectiveness of the proposed emulator in capturing key flow physics, and demonstrates the importance of incorporating known flow properties of the fluid as assumptions in the statistical model.

Next, three metrics are used to quantify emulation accuracy. The first metric, which reports the mean relative error (MRE) in important subregions of the injector, measures the *spatial* aspect of prediction accuracy. The second metric, which inspects spectral similarities between the simulated and predicted flows, measures *temporal* accuracy. The last metric investigates how well the predicted flow captures the underlying flow physics of an injector.

For spatial accuracy, the following MRE metric is used:

$$\text{MRE}(t; \mathcal{S}) = \frac{\int_{\mathcal{S}} |Y(\mathbf{x}, t; \mathbf{c}_{\text{new}}) - \hat{Y}(\mathbf{x}, t; \mathbf{c}_{\text{new}})| \, d\mathbf{x}}{\int_{\mathcal{S}} |Y(\mathbf{x}, t; \mathbf{c}_{\text{new}})| \, d\mathbf{x}} \times 100\%,$$

where $Y(\mathbf{x}, t; \mathbf{c}_{\text{new}})$ is the simulated flow at setting $\mathbf{c}_{\text{new}}$, and $\hat{Y}(\mathbf{x}, t; \mathbf{c}_{\text{new}})$ is the flow predictor in (9) (for brevity, the superscript for flow variable r is omitted here). In words, MRE$(t; \mathcal{S})$ provides a measure of emulation accuracy within a desired subregion $\mathcal{S}$ at time t, relative to the overall flow energy in $\mathcal{S}$. Since flow behaviors within the injector inlet, fluid transition region, and injector exit (outlined in Figure 9) are crucial for characterizing injector instability, we investigate the MRE specifically for these three sub-regions. Figure 8 plots MRE$(t, \mathcal{S})$ for $t = 15$–30 ms, when the flow has fully developed. For all three sub-regions, the relative error is within a tolerance level of 10% for nearly all time-steps, which is very good from an engineering perspective.

To assess temporal accuracy, we conduct a power spectral density (PSD) analysis of predicted and simulated pressure flows
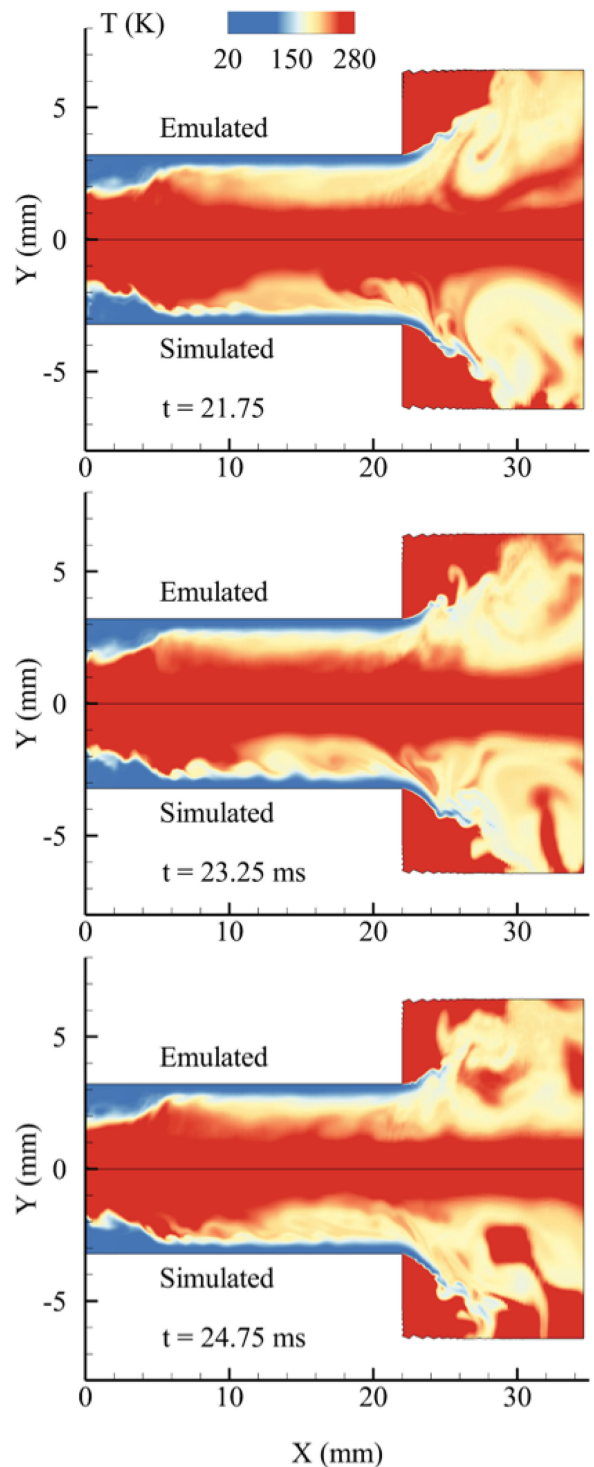

Figure 7. Simulated and emulated temperature flow at $t = 21.75$ ms, 23.25 ms, and 24.75 ms.

at eight specific probes along the region of surface wave propagation (see Figure 9). This analysis is often performed as an empirical tool for assessing injector stability (see Zong and Yang 2008), because surface waves allow for feedback loops between upstream and downstream oscillations (Bazarov and Yang 1998). Figure 10 shows the PSD spectra for the predicted and simulated flow at four of these probes. Visually, the spectra look very similar, both at low and high frequencies, with peaks nearly identical for the predicted and simulated flow. Such peaks
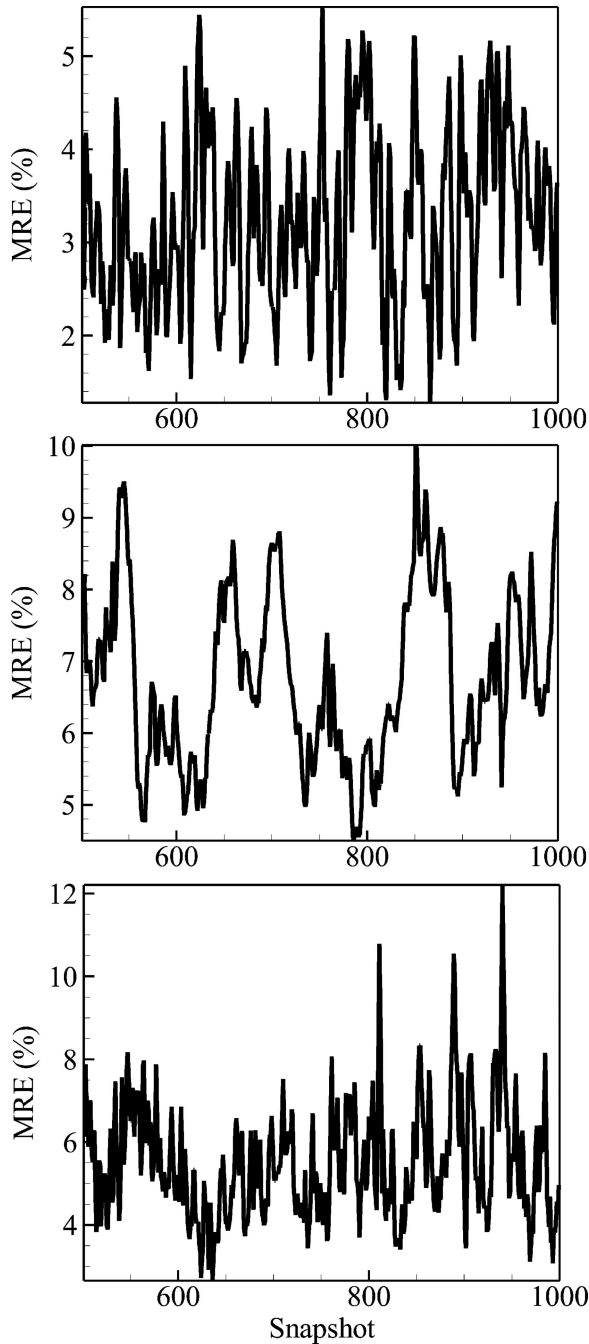
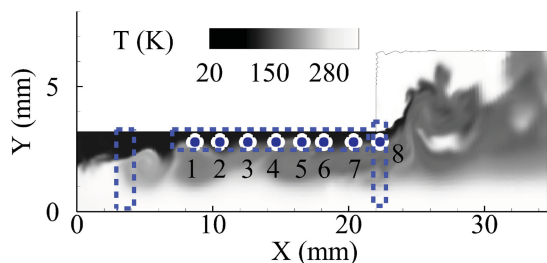**Figure 8.** MRE at injector inlet (top), fluid transition region (middle), and injector exit (bottom).

Figure 9. Injector subregions (dotted in blue) and probe locations (circled in white).

are highly useful for analyzing flow physics, because they can be used to identify physical properties (e.g., hydrodynamic, acoustic) of dominant instability structures. In this sense, the proposed emulator does an excellent job in mimicking important physics of the simulated flow.

Finally, we investigate the film thickness h and spreading angle α, which are key performance metrics for injector performance. Since both of these metrics are computed using spatial gradients of flow variables, an accurate emulation of these measures suggests accurate flow emulation as well. For the validation setting, the simulated (predicted) flow has a film thickness of 0.47 mm (0.42 mm) and a spreading angle of 103.63° (107.36°), averaged over the fully developed timeframe from $t = 15$–30 ms. This corresponds to relative errors of 10.6% and 3.60%, respectively, and is within the desired error tolerance from an engineering perspective.

4.3. Uncertainty Quantification

For computer experiments, the quantification of predictive uncertainty can be as important as the prediction itself. To this end, we provide a spatio-temporal representation of this UQ, and show that it has a useful and appealing physical interpretation. For spatial UQ, the top plot of Figure 11 shows the one-sided width of the 99% pointwise confidence interval (CI) from (10) for x-velocity at $t = 15$ ms. It can be seen that the emulator is most certain in predicting near the inlet and centerline of the injector, but shows high predictive uncertainty at the three gaseous cores downstream (in green). This makes physical sense, because these cores correspond to flow recirculation vortices, and therefore exhibit highly unstable flow behavior. From the bottom plot of Figure 11, which shows the absolute emulation error of the same flow, the pointwise confidence band not only covers the realized prediction error, but roughly mimics its spatial distribution as well.

For temporal UQ, Figure 12 shows the same one-sided CI width at probe 1 (see Figure 9). We see that this temporal uncertainty is relatively steady over t, except for two abrupt spikes at time-steps around 300 and 800. These two spikes have an appealing physical interpretation: the first indicates a flow displayment effect of the central vortex core, whereas the second can be attributed to the boundary development of the same core. This again demonstrates the usefulness of UQ not only as a measure of predictive uncertainty, but also as a means for extracting useful flow physics without the need for expensive simulations.

To illustrate the improved UQ of the proposed model (see Theorem 3.1), we use a derived quantity called TKE. TKE is typically defined as follows:

$$\kappa(\mathbf{x}, t) = \frac{1}{2} \sum_{r \in \{u,v,w\}} \left\{ Y^{(r)}(\mathbf{x}, t) - \bar{Y}^{(r)}(\mathbf{x}) \right\}^2, \qquad (12)$$

where $Y^{(u)}(\mathbf{x}, t)$, $Y^{(v)}(\mathbf{x}, t)$, and $Y^{(w)}(\mathbf{x}, t)$ are flows for x-, y-, and circumferential velocities, respectively, with $\bar{Y}^{(u)}(\mathbf{x})$, $\bar{Y}^{(v)}(\mathbf{x})$, and $\bar{Y}^{(w)}(\mathbf{x})$ its corresponding time-averages. Such a quantity is particularly important for studying turbulent instabilities, because it measures fluid rotation energy within eddies and vortices.
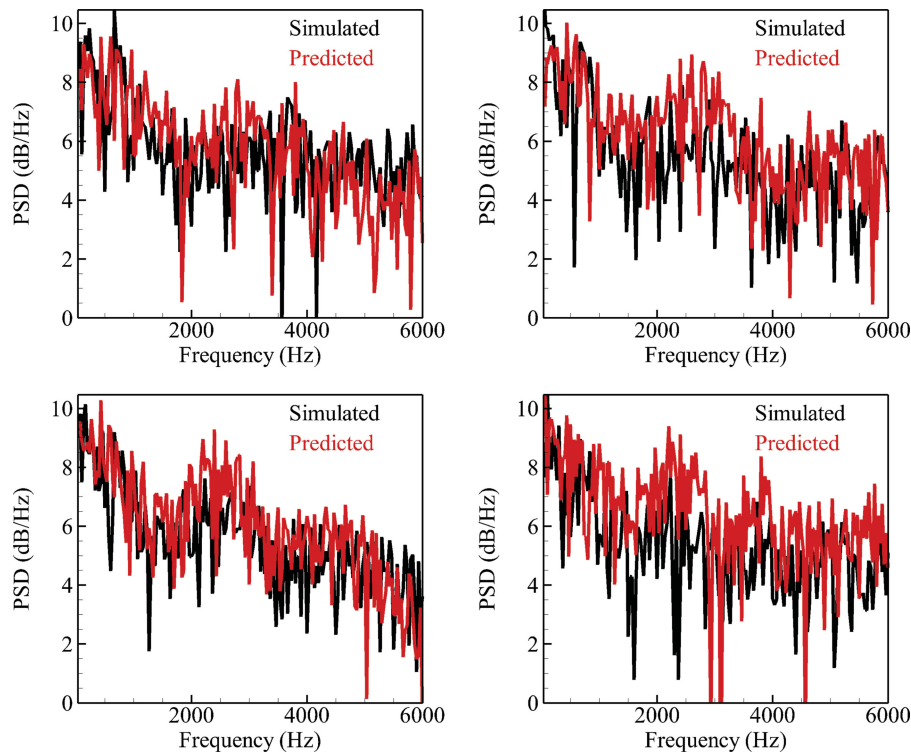
Figure 10. PSD spectra for pressure flow at probes 1, 3, 5, and 7 (see Figure 9).

For the sake of simplicity, assume that (a) the time-averages $\bar{Y}^{(u)}(\mathbf{x})$, $\bar{Y}^{(v)}(\mathbf{x})$, and $\bar{Y}^{(w)}(\mathbf{x})$ are fixed, and (b) the parameters $(\boldsymbol{\mu}, \mathbf{T}, \boldsymbol{\tau})$ are known. The following theorem provides the MMSE predictor and pointwise CI for $\kappa(\mathbf{x}, t)$ (proof in Appendix C).

Theorem 4.1. For fixed $\mathbf{x}$ and t, the MMSE predictor of $\kappa(\mathbf{x}, t)$ at a new setting $\mathbf{c}_{\text{new}}$ is

$$\hat{\kappa}(\mathbf{x}, t) = \frac{1}{2} \sum_{r \in \{u,v,w\}} \left\{ \hat{Y}^{(r)}(\mathbf{x}, t) - \bar{Y}^{(r)}(\mathbf{x}) \right\}^2 + \text{tr}\{\Phi(\mathbf{x}, t)\},$$

(13)

where $\hat{Y}^{(u)}(\mathbf{x}, t)$, $\hat{Y}^{(v)}(\mathbf{x}, t)$, and $\hat{Y}^{(w)}(\mathbf{x}, t)$ are predicted flows for x-, y-, and circumferential velocities from (9), and $\Phi(\mathbf{x}, t)$ is defined in (C.1) of Appendix C. Moreover, $\hat{\kappa}(\mathbf{x}, t)$ is distributed as a weighted sum of noncentral χ^2 random variables, with an explicit expression given in (C.3) of Appendix C.

In practice, plug-in estimates are used for both time-averaged flows and model parameters.

With this in hand, we compare the prediction and UQ of TKE from the proposed model M_A and the independent model M_0 (see Theorem 3.1) with the simulated TKE at the validation setting. Figure 13 shows the predicted TKE $\hat{\kappa}(\mathbf{x}, t)$ at probe 8 over the fully developed time-frame of $t = 15$–30 ms, along with the 90% lower pointwise confidence band constructed using Theorem 4.1. Visually, the proposed model M_A provides an
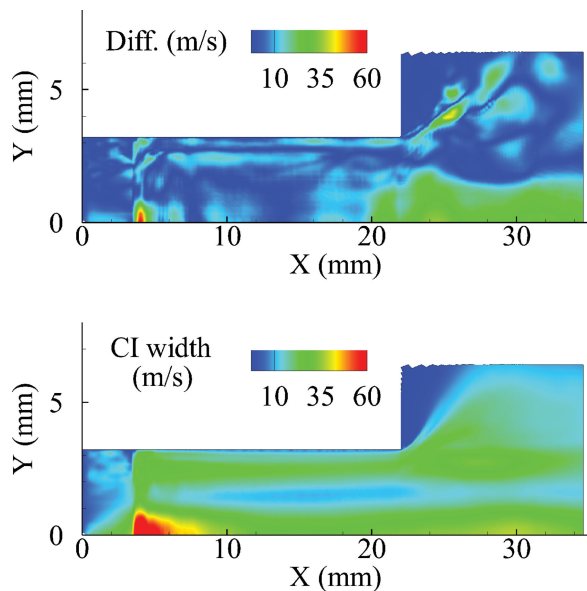

Figure 11. Absolute prediction error (top) and pointwise CI width (bottom) for x-velocity at $t = 15$ ms.
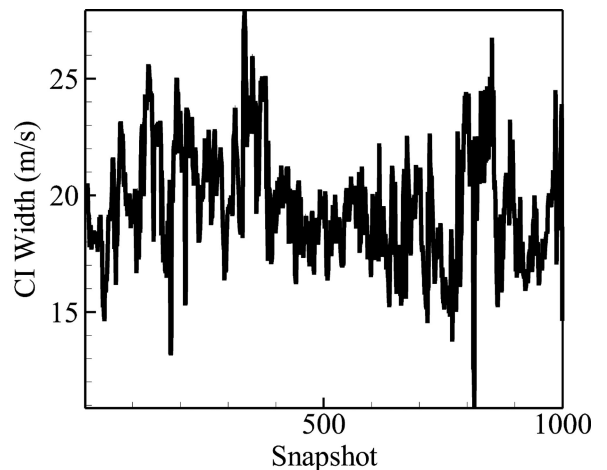

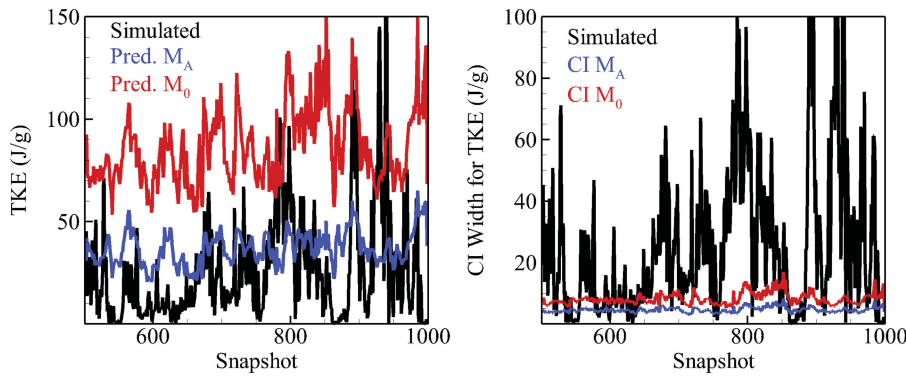**Figure 12.** CI width of x-velocity at probe 1.

Figure 13. Predicted TKE and lower 90% confidence band for M_A and M_0 at probe 8.

improved prediction of the simulated TKE than the independent model M_0. As for the confidence bands, the average coverage rate for M_A over the fully developed time-frame (85.0%) is much closer to the desired nominal rate of 90% compared to that for M_0 (73.8%). The proposed model therefore provides a coverage rate closer to the desired nominal rate of 90%. The poor coverage rate for the independent model is shown in the right plot of Figure 13, where the simulated TKE often dips below the lower confidence band. By incorporating prior knowledge of flow couplings, the proposed model can provide improved predictive performance and UQ.

4.4. Correlation Extraction

Finally, we demonstrate the use of the proposed model as a tool for extracting common flow couplings on the design space. Setting the sparsity penalty λ so that only the top nine correlations are chosen, Figure 14 shows the corresponding graph of the extracted couplings of CPOD modes. Nodes on this graph represent CPOD modes for each flow variable, with edges indicating the presence of a nonzero correlation between two modes. Each connected subgraph in Figure 14 is interpretable in terms of flow physics. For example, the subgraph connecting u_1, w_1, and P_1 (first modes for x-velocity, circumferential velocity, and pressure) makes physical sense, because u_1 and w_1 are inherently coupled by Bernoulli's equation for fluid flow (Shames and

Shames 1982), while w_1 and P_1 are connected by the centrifugal acceleration induced by circular momentum of LOX flow. Likewise, the subgraph connecting T_1, ρ_1, and w_2 also provides physical insight: T_1 and ρ_1 are coupled by the equation of state and conservation of energy, while ρ_1 and w_2 are connected by conservation of momentum.

The interpretability of these extracted flow couplings in terms of fundamental conservation laws from fluid mechanics is not only appealing from a flow physics perspective, but also provides a reassuring check on the estimation of the cokriging matrix **T**. Recall from the discussion in Section 3.2.1 that an accurate estimate of **T** is needed for the improved predictive guarantees of Theorem 3.1 to hold. The consistency of the selected flow couplings (and the ranking of such couplings) with established physical principles provides confidence that the proposed estimation algorithm indeed returns an accurate estimate of **T**. These results nicely illustrate the dual purpose of the CPOD matrix **T** in our cokriging model: not only does it allow for more accurate UQ, it also extracts interesting flow couplings that can guide further experiments.

4.5. Computation Time

In addition to accurate flow emulation and physics extraction, the primary appeal of the proposed emulator is its efficiency. Table 3 summarizes the computation time required for each step of the emulation process, with timing performed on a parallelized system of 200 Intel Xeon E5-2603 1.80 GHz processing cores. Despite the massive training dataset, which requires nearly 100 GB of storage space, we see that the proposed model can provide accurate prediction, UQ and coupling extraction in slightly over an hour of computation time. Moreover, because both CPOD extraction and parameter estimation need to be performed only once, the surrogate model can generate flow predictions for hundreds of new settings within a day's time, thereby allowing for the exploration of the full design space in practical
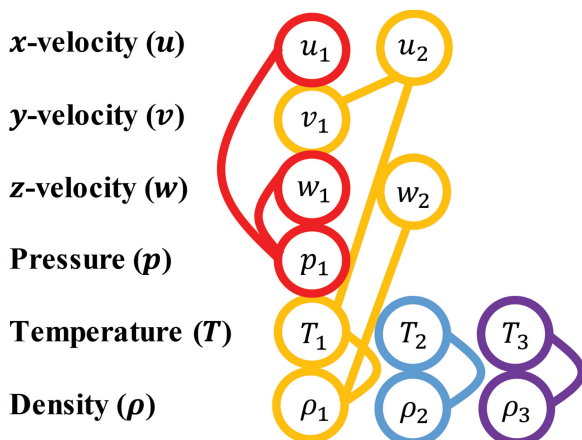

Figure 14. Graph of selected flow couplings from **T**. Nodes represent CPOD modes, and edges represent nonzero correlations.

Table 3. Computation time for each step of the proposed emulator, parallelized over 200 processing cores.

Step	Comp. time (mins)
CPOD extraction	33.91
Parameter estimation	11.31
Flow prediction	20.19
Total	65.41

turn-around times. Through a careful elicitation and incorporation of flow physics into the surrogate model, we show that an efficient and accurate flow prediction is possible despite a limited number of simulation runs, with the trained model extracting valuable physical insights that can be used to guide further investigations.

5. Conclusions and Future Work

In this article, a new emulator model is proposed which efficiently predicts turbulent cold-flows for rocket injectors with varying geometries. An important innovation of our work lies in its *elicitation* and *incorporation* of flow properties as model assumptions. First, exploiting the deep connection between POD and turbulent flows (Lumley 1967), a novel CPOD decomposition is used for extracting common instabilities over the design space. Next, taking advantage of dense temporal resolutions, a tTI emulator is proposed that considers independent emulators at each simulation time-step. Last, a sparse covariance matrix **T** is employed within the emulator model to account for the few significant couplings among flow variables. Given the complexities inherent in spatio-temporal flows and the massive datasets at hand, such simplifications are paramount for accurate flow predictions in practical turn-around times. This highlights the need for careful elicitation in flow emulation, particularly for engineering applications, where the time-consuming nature of simulations limits the number of available runs.

Applying the model to simulation data, the proposed emulator provides accurate flow predictions and captures several key metrics for injector performance. In addition, the proposed model offers two appealing features: (a) it provides a physically meaningful quantification of spatio-temporal uncertainty, and (b) it extracts significant couplings between flow instabilities. A key advantage of our emulator over existing flow kriging methods is that it provides accurate predictions using only a fraction of the time required by simulation. This efficiency is very appealing for engineers, because it allows them to fully explore the desired design space and make timely decisions.

Looking ahead, we are pursuing several directions for future research. First, while the CPOD expansion appears to work well for cold-flows, the justifying assumption of similar Reynolds numbers does not hold for more complicated (e.g., reacting) turbulent flows. To this end, we are working on ways to incorporate pattern recognition techniques (Fukunaga 2013) into the GP kriging framework to jointly (a) identify common instability structures that scale nonlinearly over varying geometries, then (b) predict such structures at new geometric settings. The key hurdle is again computational efficiency, and the treed GP models in Taddy, Gramacy, and Polson (2011) or the local GP models in Gramacy and Apley (2015), and Sung, Gramacy, and Haaland (2017) appear to be attractive options. Next, a new design is proposed recently in Mak and Joseph (2017), which combines the MaxPro methodology with minimax coverage, and it will be interesting to see whether such designs can provide improved performance. Last, to evaluate the stability of new injector geometries, the UQ for the emulated flow needs to be fed forward through an acoustics solver. Since each evaluation of the solver can be time-intensive, this forward uncertainty propagation can be performed more quickly by reducing this UQ to

a set of representative points, and the support points in Mak and Joseph (2016) can prove to be useful for conducting such a task.

Supplementary Materials

This article is accompanied by two supplementary files: the first ("appendix.pdf") provides further details on algorithmic implementation and technical proofs, and the second ("code.zip") contains relevant code files.

References

Banerjee, S., Carlin, B. P., and Gelfand, A. E. (2014), *Hierarchical Modeling and Analysis for Spatial Data*, Boca Raton, FL: CRC Press. [1444,1448]

Bayarri, M., Berger, J., Cafeo, J., Garcia-Donato, G., Liu, F., Palomo, J., Parthasarathy, R., Paulo, R., Sacks, J., and Walsh, D. (2007), "Computer Model Validation With Functional Output," *The Annals of Statistics*, 35, 1874–1906. [1444]

Bazarov, V. G., and Yang, V. (1998), "Liquid-Propellant Rocket Engine Injector Dynamics," *Journal of Propulsion and Power*, 14, 797–806. [1451]

Berkooz, G., Holmes, P., and Lumley, J. L. (1993), "The Proper Orthogonal Decomposition in the Analysis of Turbulent Flows," *Annual Review of Fluid Mechanics*, 25, 539–575. [1445]

Bien, J., and Tibshirani, R. J. (2011), "Sparse Estimation of a Covariance Matrix," *Biometrika*, 98, 807–820. [1448]

Conti, S., Gosling, J. P., Oakley, J. E., and O'Hagan, A. (2009), "Gaussian Process Emulation of Dynamic Computer Codes," *Biometrika*, 96, 663–676. [1446]

Conti, S., and O'Hagan, A. (2010), "Bayesian Emulation of Complex Multi-Output and Dynamic Computer Models," *Journal of Statistical Planning and Inference*, 140, 640–651. [1446]

Fang, K.-T., Li, R., and Sudjianto, A. (2006), *Design and Modeling for Computer Experiments*, Boca Raton, FL: CRC Press. [1444]

Friedman, J., Hastie, T., and Tibshirani, R. (2008), "Sparse Inverse Covariance Estimation with the Graphical Lasso," *Biostatistics*, 9, 432–441. [1448,1449]

Fukunaga, K. (2013), *Introduction to Statistical Pattern Recognition*, San Diego, CA: Academic Press. [1455]

Gramacy, R. B., and Apley, D. W. (2015), "Local Gaussian Process Approximation for Large Computer Experiments," *Journal of Computational and Graphical Statistics*, 24, 561–578. [1455]

Hastie, T., Tibshirani, R., and Friedman, J. (2009), *The Elements of Statistical Learning*, New York: Springer. [1449,1451]

Higdon, D., Gattiker, J., Williams, B., and Rightley, M. (2008), "Computer Model Calibration using High-Dimensional Outputs," *Journal of the American Statistical Association*, 103, 570–583. [1444]

Hung, Y., Joseph, V. R., and Melkote, S. N. (2015), "Analysis of Computer Experiments with Functional Response," *Technometrics*, 57, 35–44. [1444,1446]

Huo, H., Wang, X., and Yang, V. (2014), "A General Study of Counterflow Diffusion Flames at Subcritical and Supercritical Conditions: Oxygen/Hydrogen Mixtures," *Combustion and Flame*, 161, 3040–3050. [1445]

Hyndman, R. J. (1996), "Computing and Graphing Highest Density Regions," *The American Statistician*, 50, 120–126. [1448]

Joseph, V. R., Gul, E., and Ba, S. (2015), "Maximum Projection Designs for Computer Experiments," *Biometrika*, 102, 371–380. [1445]

Karhunen, K. (1947), *Über lineare Methoden in der Wahrscheinlichkeitsrechnung*, (Vol. 37), Universitat Helsinki. [1444]

Lehoucq, R. B., Sorensen, D. C., and Yang, C. (1998), *ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems With Implicitly Restarted Arnoldi Methods* (Vol. 6), Philadelphia: SIAM. [1446]

Liu, D. C., and Nocedal, J. (1989), "On the Limited Memory BFGS Method for Large Scale Optimization," *Mathematical Programming*, 45, 503–528. [1449]

Liu, F., and West, M. (2009), "A Dynamic Modelling Strategy for Bayesian Computer Model Emulation," *Bayesian Analysis*, 4, 393–411. [1446]

Loève, M. (1955), *Probability Theory; Foundations, Random Sequences*, New York: D. Van Nostrand Company. [1444,1445]

Lumley, J. L. (1967), "The Structure of Inhomogeneous Turbulent Flows," in *Atmospheric Turbulence and Radio Wave Propagation*, eds. A. M. Yaglom and V. I. Tatarsky, pp. 166–178, Moscow, Russia: Nauka [1444,1445,1455]

Mak, S., Bingham, D., and Lu, Y. (2016), "A Regional Compound Poisson Process for Hurricane and Tropical Storm Damage," *Journal of the Royal Statistical Society*, Series C, 65, 677–703. [1448]

Mak, S., and Joseph, V. R. (2017), "Support Points," *Annals of Statistics*. [1455]

—— (2017), "Minimax and Minimax Projection Designs using Clustering," *Journal of Computational and Graphical Statistics*, 27, 166–178. [1455]

Mathéron, G. (1963), "Principles of Geostatistics," *Economic Geology*, 58, 1246–1266. [1443]

McComb, W. D. (1990), *The Physics of Fluid Turbulence*, Oxford, UK: Clarendon Press. [1443]

McKay, M. D., Beckman, R. J., and Conover, W. J. (1979), "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code," *Technometrics*, 42, 55–61. [1445]

Morris, M. D., and Mitchell, T. J. (1995), "Exploratory Designs for Computational Experiments," *Journal of Statistical Planning and Inference*, 43, 381–402. [1445]

Nocedal, J., and Wright, S. (2006), *Numerical Optimization*, New York: Springer-Verlag. [1449]

Oefelein, J. C., and Yang, V. (1998), "Modeling High-Pressure Mixing and Combustion Processes in Liquid Rocket Engines," *Journal of Propulsion and Power*, 14, 843–857. [1445]

Pope, S. B. (2001), *Turbulent Flows*, Cambridge: Cambridge University Press. [1443,1445,1447]

Qian, P. Z. G., Wu, H., and Wu, C. F. J. (2008), "Gaussian Process Models for Computer Experiments With Qualitative and Quantitative Factors," *Technometrics*, 50, 383–396. [1448]

Ramsay, J. O., and Silverman, B. W. (2002), *Applied Functional Data Analysis: Methods and Case Studies*, New York: Springer-Verlag. [1444]

Rougier, J. (2008), "Efficient Emulators for Multivariate Deterministic Functions," *Journal of Computational and Graphical Statistics*, 17, 827–843. [1443]

Shames, I. H., and Shames, I. H. (1982), *Mechanics of Fluids*, New York: McGraw-Hill. [1454]

Shepard, D. (1968), "A Two-Dimensional Interpolation Function for Irregularly-Spaced Data," in *Proceedings of the 23rd ACM National Conference*, pp. 517–524. [1446]

Stein, A., and Corsten, L. (1991), "Universal Kriging and Cokriging as a Regression Procedure," *Biometrics*, 47, 575–587. [1444,1447]

Stokes, G. G. (1851), *On the Effect of the Internal Friction of Fluids on the Motion of Pendulums* (Vol. 9), Cambridge, UK: Pitt Press. [1445]

Sung, C.-L., Gramacy, R. B., and Haaland, B. (2018), "Exploiting Variance Reduction Potential in Local Gaussian Process Search," *Statistica Sinica*, 28, 577–600 [1455]

Taddy, M. A., Gramacy, R. B., and Polson, N. G. (2011), "Dynamic Trees for Learning and Design," *Journal of the American Statistical Association*, 106, 109–123. [1455]

Wang, X., Huo, H., Wang, Y., and Yang, V. (2017a), "Comprehensive Study of Cryogenic Fluid Dynamics of Swirl Injectors at Supercritical Conditions," *AIAA Journal*, 55, 3109–3119. [1444]

Wang, X., Huo, H., and Yang, V. (2015), "Counterflow Diffusion Flames of Oxygen and N-Alkane Hydrocarbons ($CH_4 - C_{16}H_{34}$) at Subcritical and Supercritical Conditions," *Combustion Science and Technology*, 187, 60–82. [1445]

Wang, X., Wang, Y., and Yang, V. (2017b), "Geometric Effects on Liquid Oxygen/Kerosene Bi-Swirl Injector Flow Dynamics at Supercritical Conditions," *AIAA Journal*, 55, 3467–3475. [1444]

Williams, B., Higdon, D., Gattiker, J., Moore, L., McKay, and M., Keller-McNulty, S. (2006), "Combining Experimental Data and Computer Simulations, with an Application to Flyer Plate Experiments," *Bayesian Analysis*, 1, 765–792. [1443]

Yang, V., and Anderson, W. (1995), *Liquid Rocket Engine Combustion Instability* (Vol. 169), New York: AIAA. [1451]

You, D., Ku, D. D., and Yang, V. (2013), "Acoustic Waves in Baffled Combustion Chamber with Radial and Circumferential Blades," *Journal of Propulsion and Power*, 29, 1453–1467. [1444]

Zong, N., Meng, H., Hsieh, S.-Y., and Yang, V. (2004), "A Numerical Study of Cryogenic Fluid Injection and Mixing Under Supercritical Conditions," *Physics of Fluids*, 16, 4248–4261. [1445]

Zong, N., and Yang, V. (2008), "Cryogenic Fluid Dynamics of Pressure Swirl Injectors at Supercritical Conditions," *Physics of Fluids*, 20, Art. no. 056103. [1444,1445,1450,1451]