Computers and Fluids 210 (2020) 104626

Contents lists available at ScienceDirect

Computers and Fluids

journal homepage: www.elsevier.com/locate/compfluid

Data-driven surrogate modeling of multiphase flows using machine learning techniques

Himakar Ganti, Prashant Khare*

Department of Aerospace Engineering and Engineering Mechanics, University of Cincinnati, Cincinnati, OH, 45221-0070, USA

ARTICLE INFO

Article history:
Received 30 December 2018
Revised 6 January 2020
Accepted 11 June 2020
Available online 13 June 2020

Keywords:
Liquid Jets
Atomization
Machine Learning
Multiphase Flows

ABSTRACT

This study focuses on the development of a theoretical framework and corresponding algorithms to establish spatio-temporal surrogate models for multiphase flow processes using Gaussian process (GP) based machine learning technique trained by direct numerical simulation (DNS) data. The training (and testing) datasets are obtained by solving the incompressible form of the Navier Stokes equations with surface tension in an Eulerian reference frame. The liquid-gas interfacial evolution is resolved using a volume-of-fluid (VOF) interface capturing method. The overall framework proceeds in four steps: 1) design of experiments study to identify the training and testing points and generation of corresponding datasets using DNS calculations; 2) dimensionality reduction using proper orthogonal decomposition; 3) Gaussian process regression (supervised training) over the reduced training dataset over the entire range of operating conditions under consideration; and 4) Galerkin reconstruction and error quantification by comparing the emulated flowfields (at test conditions) with the testing dataset. The machine learning framework predicts both the spatial basis-functions and the time-coefficients, thus, predicting the entire flowfield in time and space. The capabilities of the algorithm are demonstrated for two canonical flow configurations: 1) flow over a circular cylinder for a range of Reynolds numbers from 10 to 200; and 2) diesel jet injected into a quiescent nitrogen environment at chamber pressure of 30 atm and room temperature conditions, and injection velocities from 10 to 55 m/s, corresponding to a range of gas-based Weber numbers from 11.5 to 348. The emulations from the learned GP algorithm show excellent agreement with high-fidelity numerical data for test conditions; average error (in both space and time) at the testing point of $Re = 185$ for the flow over cylinder case is 4.4%, and for the diesel jet injection configuration at a testing point corresponding to velocity of 22.5 m/s is found to be 15.5%. The tip penetration location of the diesel jet is predicted within 2.5% of the DNS calculations. Corresponding to these two representative test points, speedup of 256 and 8000 is achieved for flow over cylinder and diesel jet atomization configurations, respectively. This paper represents the first effort of its kind on the development of a general machine learning framework to predict multiphase flows.

1. Introduction

Numerical simulations play a major role in the development of engineering devices and are increasingly becoming the primary design and analysis tool in liquid fueled propulsion devices, such as diesel, gas-turbine and rocket engines, medical and other infrastructure based industries [1,2]. To this end, on one hand, high-fidelity computer modeling and simulation has resulted in tremendous progress in enhancing the quantitative understanding of turbulent, multi-scale, multi-phase, and multi-physics processes underlying various engineering systems [3]. On the other hand, because of the wide range of length (sub-micron to meter scale) and time scales (100ns to 10s) involved in these processes [4,5], such high-fidelity simulations are cost-prohibitive and cannot be readily incorporated in the design and development of real-world devices [6].

Specifically, for the liquid jet atomization process, significant progress has been made to enhance the mechanistic understanding of flow dynamics, detailed spray physics and statistical behavior of resulting droplets and velocity distributions using interface tracking and capturing techniques, such as level-set and volume-of-fluid (VOF) methods [7–10]. However, even with the use of adaptive-mesh-refinement (AMR) techniques, such simulations are subject to severe grid resolution requirements at the liquid-gas interface

* Corresponding author.
E-mail addresses: gantihr@mail.uc.edu (H. Ganti), Prashant.Khare@uc.edu (P. Khare).

https://doi.org/10.1016/j.compfluid.2020.104626
0045-7930/

that makes high-fidelity numerical studies computationally expensive with large turn-around times, thus rendering them impractical and prohibitive for routine design calculations.

One way to incorporate high-fidelity physics from aforementioned simulations in design tools is by developing data driven surrogate models that are cost-effective, and at the same time capture the detailed flow behaviors. There remains a major gap between data-driven techniques and high-fidelity calculations, especially in the context of multiphase flows. Currently existing data-driven methods reported in fluid dynamics literature are limited to the determination of system dynamics [11–13]; the inclusion of data-driven techniques to predict the complete flowfield with high-fidelity is in its infancy. The current research effort is an attempt to bridge the gap between high-fidelity simulations of single and multiphase flows, and data-enabled surrogate modeling. This emerging research opportunity, which is ubiquitous to many fields that exhibit highly non-linear behaviors, is the focus of this paper.

Previous efforts on analyzing flowfield data from experimental and numerical simulations have been primarily focused on feature extraction using Eigen value decomposition techniques to identify underlying structure of nonlinear dynamics of single [14,15] and multiphase flows [16–18]. Most commonly used data analysis and identification techniques for non-reacting and reacting flows are based on Karhunan-Loeve procedures such as the proper orthogonal decomposition (POD) and dynamic mode decomposition techniques [19–23]. A few attempts have also been made in the past to develop machine learning based models to predict system dynamics by combining modal decomposition with Gaussian process regression [24,25], artificial neural networks (ANN) [26,27] and deep learning [28,29]. A notable application of such an approach for automotive design was recently presented by Moiz et al. [30] where they used a machine learning (ML) based stochastic global optimization genetic algorithm (GA) to optimize nine input parameters to minimize fuel consumption while satisfying CO and NO_x emission constraints. 75% reduction in turn-around time was reported for this optimization problem with the ML-GA approach, as compared to sequential CFD calculations with a GA optimizer.

Recently, Brunton at al. [31] leveraged advances in sparsity and machine learning techniques in combination with symbolic regression to predict flow over circular cylinders. They used the first two most energetic POD modes and a shift mode as coordinate axes to represent the steady state vortex shedding and transient dynamics, respectively. The system dynamics in POD coordinates, quadratic nonlinearity and time scales associated with the mean flow compared well with reference data. A similar study was conducted by Yeh et al. [32] using a kriging-based framework trained with high-fidelity data to predict spatio-temporal behaviors of swirling gas flow in an injector element and measured the accuracy of the mean flow. Uncertainty quantification capabilities were also demonstrated for different injector geometries.

Grigo and Koutsourelakis [33] recently developed a Bayesian formulation to simultaneously perform dimensionality reduction for processes governed by partial differential equations. A Stochastic Variational Inference (SVI) scheme was used to train the developed model. In addition to uncertainty propagation, the authors suggested that their formulation can be used for the solution of inverse problems. In a follow up study, using the SVI scheme, Grigo and Koutsourelakis [34] developed a probabilistic machine learning framework and demonstrated prediction capabilities for unknown extrapolative conditions.

No study in the past has discussed a generalized machine learning framework that can predict the spatio-temporal behaviors of multiphase flows. Therefore, in this research effort we develop such a framework, based on Gaussian process based supervised machine learning techniques.. One of the advantages of using the Gaussian process machine learning algorithm is that it is based on the Bayesian approach that maximizes the probability of predictions and can be optimized to remove noise from the underlying signals.

The overall framework proceeds in four steps: 1) design of experiment studies and generation of high-fidelity training and testing datasets; 2) dimensionality reduction using proper orthogonal decomposition; 3) Gaussian process regression (supervised training) over the reduced training dataset for the range of operating conditions under consideration; and 4) Galerkin reconstruction and error quantification by comparing the emulated flowfields (at test conditions) with the testing dataset (not used for training). To demonstrate the robustness and applicability to single and multiphase flows, the framework is applied to two canonical configurations: (1) flow over a circular cylinder; and (2) diesel jet injection in quiescent environment. This study is a step in the development of a comprehensive surrogate modeling framework that can be used as a routine design tool to investigate the entire process from liquid injection, atomization, vaporization, ignition and combustion with high-fidelity.

This paper is organized as follows – the emulation framework is described in detail first. This is followed by the governing equations that are solved to establish the training and testing databases using direct numerical simulations. Mathematical description of the POD technique and Gaussian process regression are discussed next. This is followed by the application of the developed framework to flow over a circular cylinder and diesel jet injection in quiescent environment.

2. Emulation framework

Fig. 1 shows the overall philosophy of the proposed framework. There are four major steps, further details of which are provided in subsequent sub-sections:

Step 1: generation of training database using the truth model

A design of experiment (DoE) study is conducted to identify the appropriate number and distribution of training points (in terms of non-dimensional numbers such as Re, We, Oh), within the operating range of interest. Direct numerical simulations are conducted at the identified training conditions to generate a spatially and temporally varying training dataset. The machine learning algorithm is trained using this database. For clarity, the predictions from the trained machine learning algorithm will be referred to as emulations and the results from direct numerical simulations will be referred to as truth model for the rest of this paper.

Step 2: reduction of dimensionality

Dimensionality reduction of the training data is achieved using proper orthogonal decomposition (POD) [35,36]. POD is a well-known principal component analysis technique that has been extensively used to extract system dynamics. The output of a POD analysis are Eigen modes (also called POD modes), spatial basis-functions and time-coefficients.

Step 3: Gaussian process regression

The third step is kernel selection and Gaussian process regression – the inputs are Eigen modes, basis-functions and corresponding time-coefficients generated in the previous step for the entire training database. After this step, for any given operating condition within the training bounds, the algorithm can predict the dynamics of the system, that is, the Eigen modes, basis-functions and time-coefficients. The spatio-temporal emulated flowfield is reconstructed by using Galerkin reconstruction [20].

Step 4: error quantification

Galerkin reconstruction is used to generate the spatio-temporal flowfield to emulate the flow behavior at all points within the training bounds. To establish the robustness and accuracy of the framework, L_1error norm is calculated to quantify the difference between the emulated results and the corresponding truth model

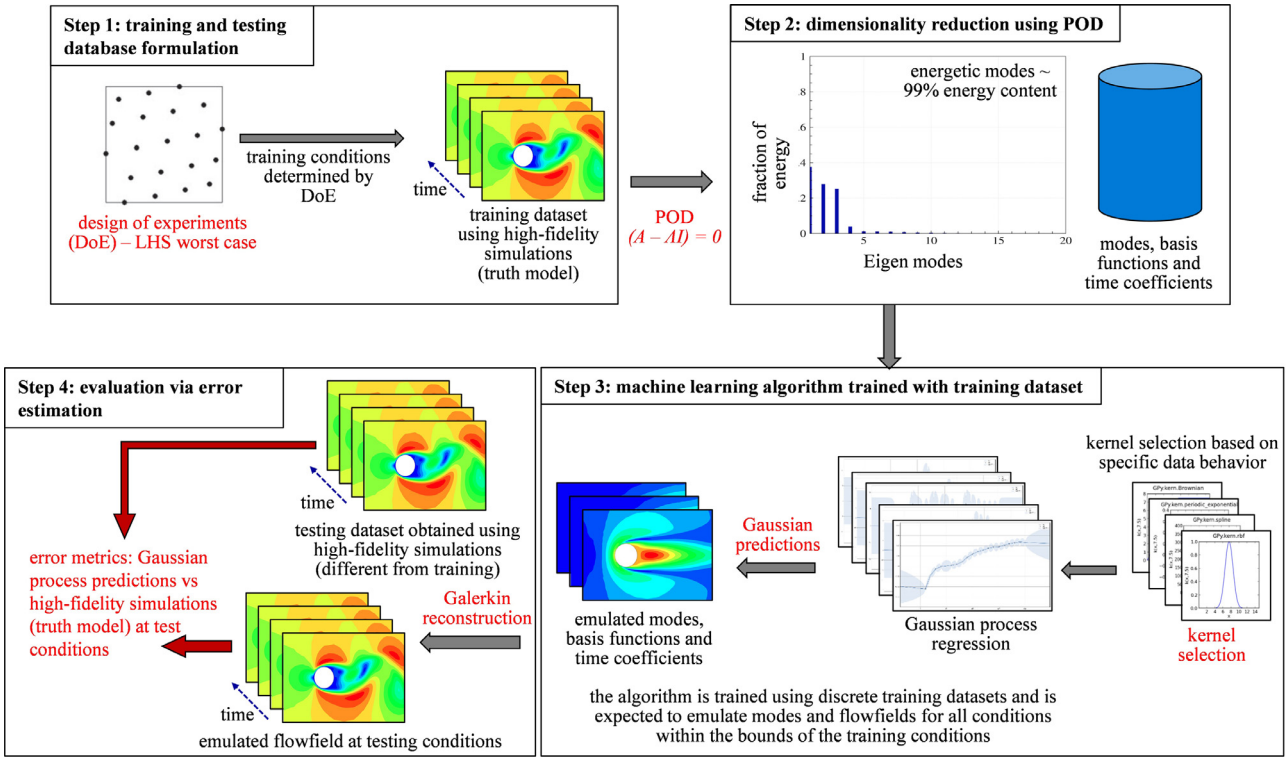

Fig. 1. Outline of Gaussian process-based machine learning algorithm as applied to a fluids problem; clockwise from above: Step 1 – design of experiments; Step 2 – reduction of dimensionality; Step 3 – Gaussian process training and machine emulation; Step 4 – flowfield reconstruction and evaluation via error quantification.

at test points. The theoretical formulations of each of these steps are discussed in the next five sub-sections.

2.1. Identification of training points

High fidelity numerical simulations are often computationally expensive and conducting a parameter sweep for even a single input variable can be prohibitive. Under such circumstances, selecting a set of equally spaced input variable points within the training bounds is sufficient for generating the training data as long as the set captures the salient features of the flow. In addition, prior knowledge of the physical processes based on the literature are also considered when choosing training points, such that points where significant phenomenological changes occur are included in the training dataset. In the current work, training points are identified by a single variable – Reynolds number for flow over a cylinder configuration and injection velocity (or Weber number) for diesel jet injection case. Each truth-model-based simulation can then be considered as independent and identically distributed (IID). This removes any bias and simplifies the Gaussian processes required for building the emulator. Similar to the study of McKay et al. [37], we use the worst-case scenario Latin Hypercube Sampling (LHS) methodology for our DoE study [38,39]. LHS sampling states that if $\boldsymbol{N}$ intervals are needed for each component $\boldsymbol{K}$, of an input variable $\boldsymbol{X}$, then the total number of points required to cover the sample space $\boldsymbol{X}$ is given as:

$$Samples = \mathrm{N}^{\mathrm{K}} \tag{1}$$

The above equation does not specify the spacing between samples; and hence as mentioned before, we sample points at regular intervals, corresponding to the worst case scenario.

2.2. Governing equations of the truth model

The truth model is built on high-fidelity numerical simulations based on three-dimensional, incompressible, variable-density form of the Navier-Stokes equations with surface tension [40]. These partial differential equations for mass and momentum conservation are:

$$\frac{\partial \rho}{\partial t} + \nabla.(\rho \vec{u}) = 0 \tag{2}$$

$$\rho\left(\frac{\partial \vec{u}}{\partial t} + \vec{u}.\nabla\vec{u}\right) = -\nabla p + \nabla.(2\mu\bar{\bar{D}}) + \sigma\kappa\delta_s\vec{n} \tag{3}$$

where $\vec{u} = u(\vec{x}, t)$is the fluid velocity, $\rho = \rho(\vec{x}, t)$ the density, $\mu = \mu(\vec{x}, t)$the dynamic viscosity, and $\bar{\bar{D}}$ is the deformation tensor defined as $D_{ij} = 1/2(\partial u_i/\partial x_j + \partial u_j/\partial x_i)$. σ is the surface tension coefficient, κ and $\vec{n}$are the radius of curvature and the unit vector normal to the interface, respectively. The Dirac delta distribution function, δ_s expresses the fact that the surface tension term is concentrated on the interface. This methodology is also called the one-fluid approach in the literature [41]. To capture the multi-fluid interface, a volume-of-fluid (VOF) variable $c = c(\vec{x}, t)$is introduced. It is defined as the volume fraction of the fluid in a given computational cell. The density and viscosity in each finite volume are then defined as linear functions of c:

$$\rho(c) = c\rho_1 + (1-c)\rho_2 \tag{4}$$

$$\mu(c) = c\mu_1 + (1-c)\mu_2 \tag{5}$$

Subscripts 1 and 2 denote the first and second fluid, respectively. Substituting Eqs. (4) and (5) in (2) transforms the advection equation for the density to an equivalent equation for the volume

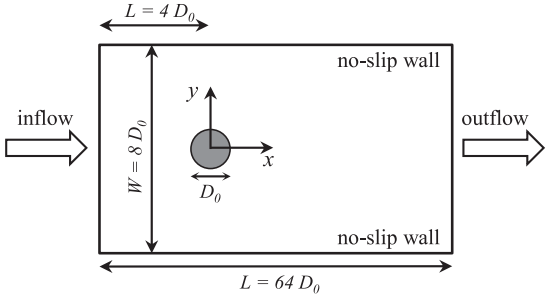

Fig. 2. Schematic of flow over cylinder.

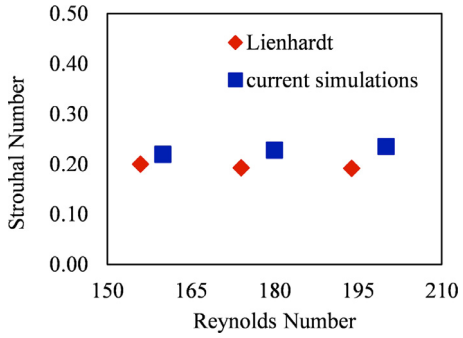

Fig. 3. Comparison of Strouhal number from simulations with experimental measurements of [57].

fraction, given by:

$$\frac{\partial c}{\partial t} + \nabla.(c\vec{u}) = 0 \tag{6}$$

The theoretical formulation outlined above is solved numerically using a finite volume method augmented by an adaptive mesh refinement (AMR) technique to improve the solution accuracy and efficiency. The framework is furnished with a multi-block domain decomposition feature to facilitate parallel processing on MPI-based distributed HPC environments. The details of the numerical methodology can be referred to in the literature [42–46,59,60].

2.3. Properorthogonal decomposition (POD)

Proper orthogonal decomposition (POD) is a principle component analysis technique and is widely used to identify system dynamics [14,17,18,26,47]. POD yields an optimal set of basis-functions from an input of ensemble data; the data source can be numerical simulations or experiments. POD can be applied to scalars as temperature, pressure, volume fractions, species concentrations, or to vectors such as velocity and vorticity. For flowfields, velocity vector field, $\vec{u} = u(\vec{x}, t)$ is often used for POD analysis because its correlation matrix directly corresponds to the kinetic energy of system. The basic concept is to represent the velocity vector as a finite sum in a variable-separable form to approximate the entire field, given by:

$$u(\vec{x}, t) = \sum_{i=1}^{M} a_i(t)\phi_i(\vec{x}) \tag{7}$$

where, u is the velocity vector field, $\vec{x}$ is spatial coordinate and t is temporal coordinate. M is the number of modes selected for summation. The vector field can be further modified by subtracting the mean, such that only the unsteady components are retained in the basis-functions. The mean value can be either spatial or temporal. Eq. (7) thus transforms to:

$$u(\vec{x}, t) - \bar{u}(\vec{x}, t) = \sum_{i=1}^{M} a_i(t)\phi_i(\vec{x}) \tag{8}$$

A covariance matrix is obtained by multiplying this vector with its transpose, resulting in a square matrix as follows:

$$\vec{u}_{corr} = \vec{u} * \vec{u}^T \tag{9}$$

Eigenvalue decomposition based on the method proposed by Sirovich [36], also called the method of snapshots, is then performed on this correlation matrix. The eigenvalues and eigenvectors obtained by this analysis and are orthonormal to each other. To determine the number of modes, r needed to reconstruct flow data, eigenvalues are identified such that they capture 99% of the total energy. The flowfield can then be reconstructed using r modes using corresponding basis-functions and time-coefficients, as follows:

$$u(\vec{x}, t) - \bar{u}(\vec{x}, t) = \sum_{i=1}^{r} a_i(t)\phi_i(\vec{x}) \tag{10}$$

2.4. Gaussian process regression for fluid flows

2.4.1. Gaussian process regression

Gaussian process regression (GPR) is a non-parameterized Bayesian analysis method that can be used as a supervised machine learning tool to build surrogate models from experimental, simulation or probabilistic-distribution datasets. The underlying assumption and idea behind GPR is that the mean and covariance of a dataset, which can be represented as a sample from a Gaussian distribution, if modelled by assuming a mean (assumed to be 0 in this paper) and covariance function, $k(x, x')$, can be used to estimate outputs and their probabilities for desired inputs. As reported in subsequent sections, this hypothesis is validated via two case studies concerning gaseous and multiphase flows. It should be noted that the goal of GPR is not to find a functional form that maps inputs to output but to predict the output and associated uncertainties (at inputs where output is not known) based on n observations. The covariance function is then calculated over all possible combinations in the dataset, given by three matrices [48,49]:

$$K = \begin{bmatrix} k(x_1,x_1) & k(x_1,x_2) & \dots & k(x_1,x_n) \\ k(x_2,x_1) & k(x_2,x_2) & \dots & k(x_2,x_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_n,x_1) & k(x_n,x_2) & \dots & k(x_n,x_n) \end{bmatrix}$$
$$K_* = \begin{bmatrix} k(x_*,x_1) & k(x_*,x_2) & \dots & k(x_*,x_n) \end{bmatrix} \text{ and } K_{**} = k(x_*,x_*) \tag{11}$$

where x_* is the point where we are trying to estimate the likelihood of the solution, y_* given the training dataset, represented by a Gaussian distribution:

$$\begin{bmatrix} \mathbf{y} \\ y_* \end{bmatrix} \sim N\left(0, \begin{bmatrix} K & K_*^T \\ K_* & K_{**} \end{bmatrix}\right) \tag{12}$$

where T indicates transpose. We are interested in the conditional probability $p(y_*|\mathbf{y})$, that is, given the training data, what is the probability (and predicted value) of y_* at x_*.This probability turns out to be a Gaussian distribution given by [48]:

$$y_*|\mathbf{y} \sim N\left(K_*K^{-1}\mathbf{y}, K_{**} - K_*K^{-1}K_*^T\right) \tag{13}$$

The mean and variance of this distribution then represents y_* and the associated uncertainty, respectively, as shown below:

$$\begin{aligned} \bar{y}_* &= K_*K^{-1}\mathbf{y} \\ \text{var}(y_*) &= K_{**} - K_*K^{-1}K_*^T \end{aligned} \tag{14}$$

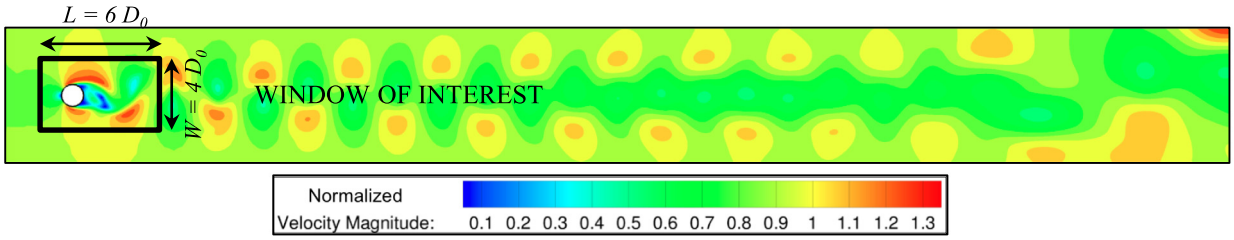

Fig. 4. Instantaneous snapshot of flow over a cylinder for Re =180. Window of interest for capturing flow features, as compared to the flow domain.

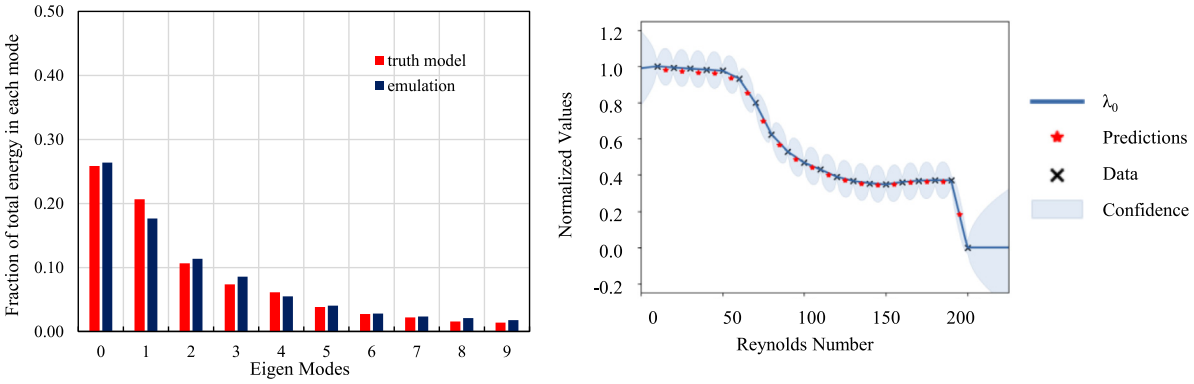

Fig. 5. Comparison of (a) Eigen values for Re = 185; (b) GP model for first Eigen mode, λ_0.

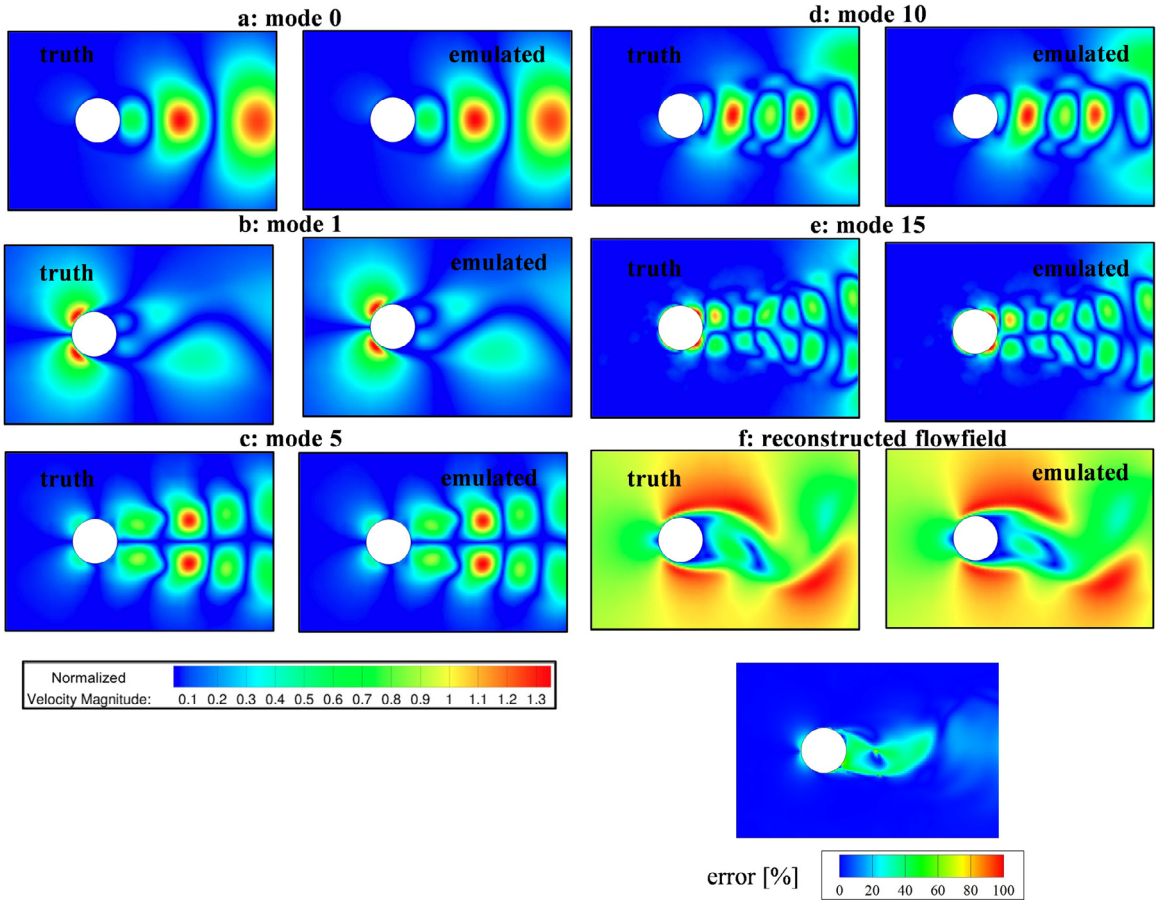

Fig. 6. Comparison of emulated and truth model for test point of Re = 185: POD modes and reconstructed flowfield. L_1 error contours (between the truth model and emulation) are shown in bottom right.

The preceding discussion relies on how the covariance matrices are formed. One way, is to calculate it pairwise for all points, which can easily become very expensive. Another way, which is followed in this paper, is to use a function, $k(x, x')$ also known as kernel to build the covariance matrices. The parameters associated with the kernel are often called hyperparameters, denoted by θ. The reliability of predictions through GPR is therefore, dependent on the choice of kernels and the corresponding prior estimates of the hyperparameters. In the current research effort, a uniform distribution of priors is used based on the conclusions of Chen and Wang [50], where they showed that for a given kernel, the choice of initial hyperparameters have no significant impact on GPR predictions. The optimum value of θ that fits the training dataset (assuming that little to no information is known about θ) then occurs when $p(\theta|\mathbf{x}, \mathbf{y})$ is largest, which from Bayes' theorem corresponds to maximizing $\log p(\mathbf{y}|\mathbf{x}, \theta)$ and is given by:

$$\log p(\mathbf{y}|\mathbf{x}, \theta) = -\frac{1}{2}\mathbf{y}^T K^{-1}\mathbf{y} - \frac{1}{2}\log|K| - \frac{n}{2}\log 2\pi \tag{15}$$

Once the hyperparameters are calculated, the predictions for y_* corresponding to x_* can be obtained from Eq. (14). In essence, GPR is non-parametric in the sense that a functional form is not assumed for the data, but for the covariance function. There are other methods that are completely non-parametric and can be referred to elsewhere [51]. The GPR training algorithm in the paper follows the implementation of Gaussian Process Summer School [52,53].

2.4.2. General considerations and kernel selection for fluid flows

For supervised learning of fluid flows, kernel selection depends on the behaviour of data for the model to perform reliably within the bounds of the input variables of the training dataset. The goal of the current research effort is to predict the spatio-temporal dynamics of single and multiphase flows for operating conditions that lie within the GPR training range. To achieve this objective, GPR is applied at each point (defined by the numerical grid) in the flowfield for the following data types obtained after the POD analysis: 1) Eigen values, 2) mean parameter values, 3) modal basis-function values and 4) time-coefficients. The challenge is to identify a kernel function that simultaneously satisfies the aforementioned four data types at each grid point. The following were considered prior to selecting a kernel for emulating fluid flows:

1 As the data used for training the machine learning algorithm is obtained from direct numerical simulations is deterministic the GP model has to pass through the training data points.
2 Based on DoE, an optimum set of training points is identified. The distribution of these points is determined by the physics under consideration - training points where significant phenomenological changes are observed are selected.
3 The model behaviour should not change when additional training data points are added within the training bounds. The model however, should change when data points are removed from this set. For both case studies described in this paper, no changes were observed in the predicted flowfields when additional datasets were added. This check ensures reliability of the model.
4 Kernel functions were selected based on characteristics of the four data types at a few locations in the flowfield, for the entire input parameter space. Since it is not possible to observe the behaviors at all locations, representative locations like the wake for the cylinder flow or the liquid-gas interface were chosen for selecting the kernel function. The kernel function for each data type is kept constant in the current effort over the entire flowfield.
5 The rationale underlying the selection of a kernel function is to capture as much variation in the signal as possible across all training datasets at each point in space and time. There are several strategies to accomplish this: (1) selecting spatial locations where the flow behaviors are most dynamic, such as recirculation zones immediately downstream of the cylinder wake in case of flow over cylinder case, and the recirculating zones appearing behind the ligaments formed due to primary atomization. Based on the specific flow behaviors, other dynamic features can be used to identify the appropriate spatial locations. To ensure that one kernel depicts the entire flowfield in space and time, we use the additive, multiplicative and convolution properties, which preserve the positive definiteness of the kernels; (2) dividing the domain in zones (based on a predefined criteria) and creating individual kernels for each of them. In this study we used the first method, where there is only one kernel for the entire domain.

Details pertaining to kernel functions specific to each case study is accompanied with their descriptions below.

2.5. Error quantification and speedup

All errors reported in this study are based on the L_1 norm, defined as follows:

$$error = \frac{1}{n}\sum_{i}^{n}\left(\frac{|truth - emulation|}{truth}\right)_i \tag{16}$$

The speedup factor is estimated as shown below:

$$\text{speedup} = \left(\frac{t_{truth}}{t_{emulation}}\right) \tag{17}$$

3. Case studies

To verify the numerical treatment and to establish confidence, the framework described in the previous sections is applied to two

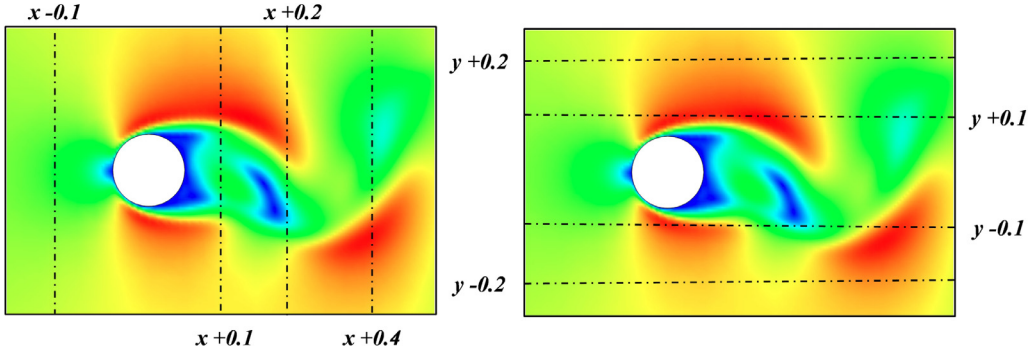

Fig. 7. Comparison of velocity profiles in vertical (x = -0.1, 0.1, 0.2, 0.4) and horizontal (y = -0.2, -0.1, 0.1, 0.2) directions between the truth model and emulated results for the test point of Re = 185.

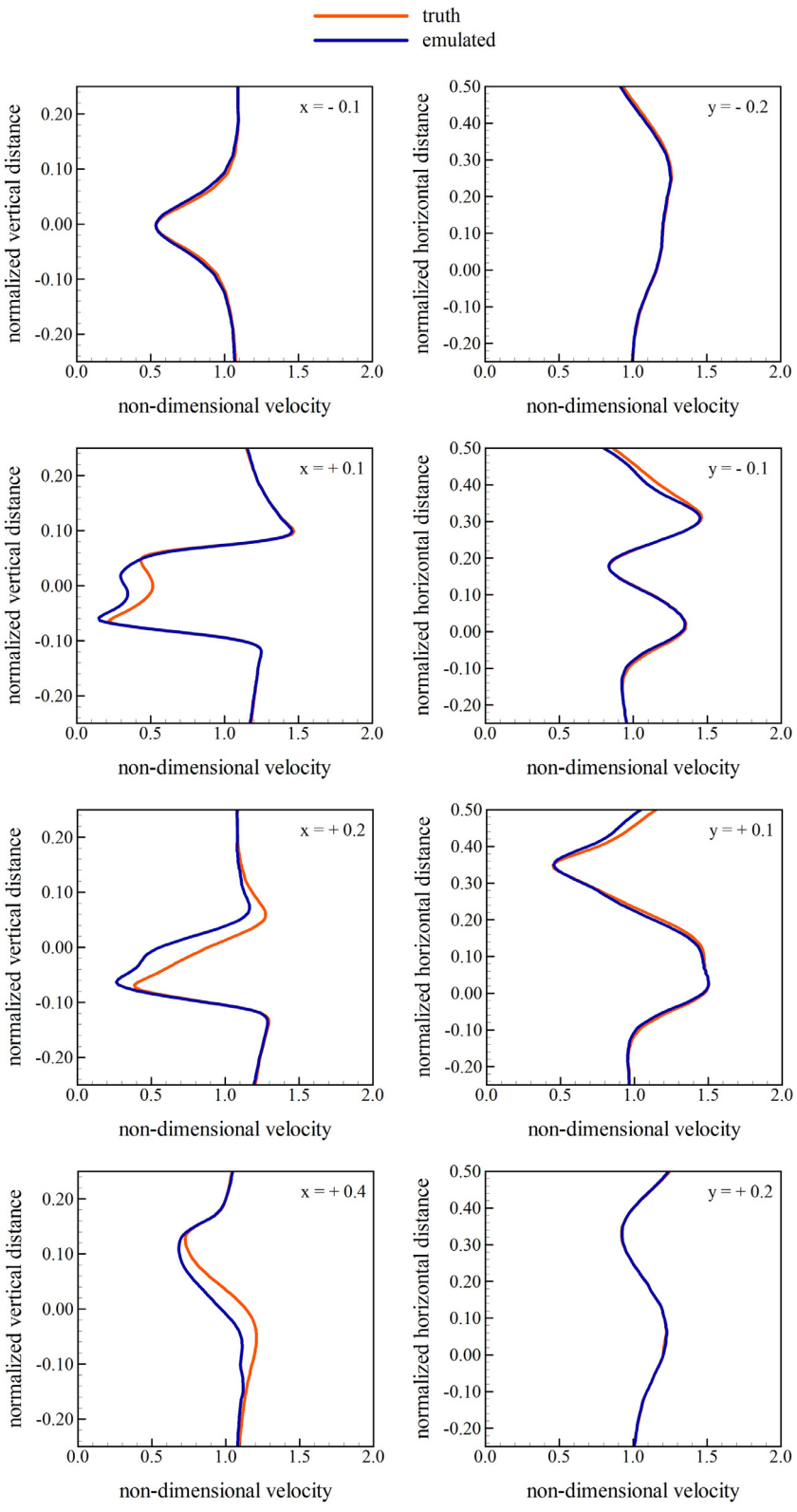

Fig. 7. Continued

different sample problems. These include 2 canonical laminar flow configurations: (1) air flow over a circular cylinder; and (2) liquid jet injection in quiescent environment. Four steps described in Fig. 1 are applied to both flow situations.

3.1. Flow over a circular cylinder

Laminar flow of air over a 2D circular cylinder is studied first as schematically shown in Fig. 2. The computational domain encom-

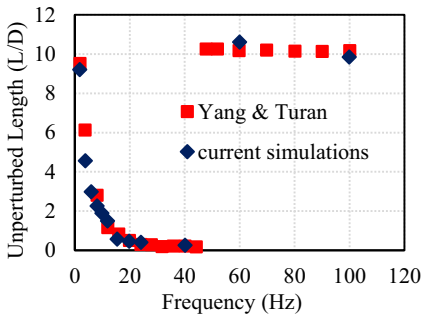

Fig. 8. Comparison of unperturbed normalized jet lengths with Yang and Turan [58]. We = 287.5 with 5% velocity amplitude and a frequency range of 0 to 100 Hz.

Table 1
Operating conditions and relevant non-dimensional numbers for flow over a cylinder.

Diameter	0.125 m
Viscosity	1.84×10^{-5}Pa-s
Density	1.225 kg/m^3
Operating pressure	1 atm
Operating temperature	298 K
Reynolds number	10 to 200

passes a region of 8.00 m × 1.00 m. Air flows through the channel that contains a cylinder of diameter 0.125 m, with a bulk speed, U ranging from 0.121 to 2.414 cm/s, at atmospheric conditions (p = 1 atm and T = 298 K). The corresponding Reynolds numbers based on the cylinder diameter varies from 10 to 200. The cylinder is located at a distance of 0.5 m, downstream of the entrance. The grid adapts based on the vorticity gradient after every timestep to capture the unsteady wake and the periodic vortex shedding [54–56]. Fig. 2 shows the schematic of the configuration and the corresponding operating conditions are listed in Table 1.

Step 1. training dataset

The flow conditions are consolidated in one input variable – the Reynolds number. Since the system behavior is not known apriori, uniform spacing is chosen based on LHS worst case scenario approach, such that the training points consist of results from high-fidelity spatio-temporally varying flowfields at 20 equally spaced intervals in the Reynolds number space, i.e., 10, 20, 30, ..., 200. In addition to the training database, a testing database is also established represented by Re = 15, 25, 35, ..., 195. To ensure accuracy of the high-fidelity calculations that will become the basis of Gaussian process training, Strouhal numbers (non-dimensional vortex shedding frequency) obtained from the current study at different Reynolds numbers are compared with measurements of Lienhard [57]. As shown in Fig. 3, our results corroborate very well with published literature.

Step 2. dimensionality reduction

Fig. 4 shows an instantaneous snapshot of flow over a cylinder for Re = 180. Each grid point in the flowfield is associated with discrete values of primitive and derived variables, including the velocity vector, pressure and vorticity. For demonstration purposes, GPR is performed in the near field of the cylinder, shown by the box with dimensions of 0.75 × 0.50 m^2 in Fig. 4 that consists of 15000uniformly spaced points that capture the critical flow features like the wake and vortex shedding immediately downstream of the cylinder. These points are kept consistent for the entire training dataset described in step 1.

To identify system dynamics and reduce the dimensionality, POD analysis is performed on the spatio-temporally varying high-fidelity results over the entire range of training Reynolds numbers. The result of this analysis are the Eigen values, basis-functions and time-coefficients. As shown in Fig. 5a, the first 16 modes (out of a total of 101) retain 99%, and the first mode captures 40%%of the energy content. These modes and corresponding basis-functions and time-coefficients are used for GPR.

Step 3. Gaussian process training

As described before, the GP training algorithm has to be applied for four different data types, and an appropriate kernel function has to be chosen based on their behaviours. Kernel function used for emulating Eigen values has the following mathematical form:

$$\text{kernel} = (\text{rational quadratic} + \text{spline} + \text{brownian}) \tag{18}$$

$$k(x,x') = \left(1 + \frac{(x-x')^2}{2\alpha l^2}\right)^{-\alpha} + \left(\xi + xx' + \Gamma\left(\frac{1}{2}|x-x'|\min(x,x') + \frac{1}{3}\min(x,x')^3\right)\right) + (\min(x,x')) \tag{19}$$

where α, ξ and l are kernel hyper-parameters. The training values are normalized before the emulation is conducted to reduce model complexity and stiffness. As shown in Fig. 5a, the emulated Eigen values for testing condition of Re = 185 agrees very well with results from the truth model. The error between the truth model and emulated Eigen modes that contain 99% of the kinetic energy content for this testing point of Re = 185 (not used for training) is 7.87%. Fig. 5b shows the mean and the corresponding confidence interval when using the above kernel for the prediction of the Eigen mode-0 (λ_0). This model has a 95% confidence within the bounds of the training range. However, outside that range, uncertainties are high, as a result we recommend the use of this model only within the bounds of the training range. Similar emulation models exist for all the other Eigen modes.

Similar to Eigen values, kernel functions are established for the spatial basis-functions and time-coefficients, such that they accurately predict the training data. The expectation is that if training data is predicted with sufficient confidence, any testing point within the range will be predicted with low uncertainty. Kernel functions developed for spatial basis-functions and time coefficients are given by:

$$\text{kernel} = (\text{Polynomial}(1 + \text{Brownian}) + \text{Spline}(1 + \text{Periodic exponential} + \text{Brownian})) \tag{20}$$

$$k(x,x') = (\alpha^2 x^{\mathrm{T}}x + \beta^2) * (1 + (\min(x,x'))) + \left(\xi + xx' + \Gamma\left(\frac{1}{2}|x-x'|\min(x,x') + \frac{1}{3}\min(x,x')^3\right)\right) * \left(1 + \left(\exp\left(-\frac{2\sin^2\left(\frac{x-x'}{2}\right)}{l^2}\right)\right) + (\min(x,x'))\right) \tag{21}$$

$$\text{kernel} = (\text{Polynomial} + \text{Spline}(1 + \text{Brownian})) \tag{22}$$

$$k(x,x') = (\alpha^2 x^{\mathrm{T}}x + \beta^2) + \left(\xi + xx' + \Gamma\left(\frac{1}{2}|x-x'|\min(x,x') + \frac{1}{3}\min(x,x')^3\right)\right) * (1 + (\min(x,x'))) \tag{23}$$

where α, β, and ξ are the kernel hyper-parameters. To ensure that the emulated results are not dependent on the number of training points, training points were sequentially added until the emulated predictions were invariant. The GP-based machine learning algorithm described above is then trained with the POD modes, basis-function values and time-coefficients of the training dataset.

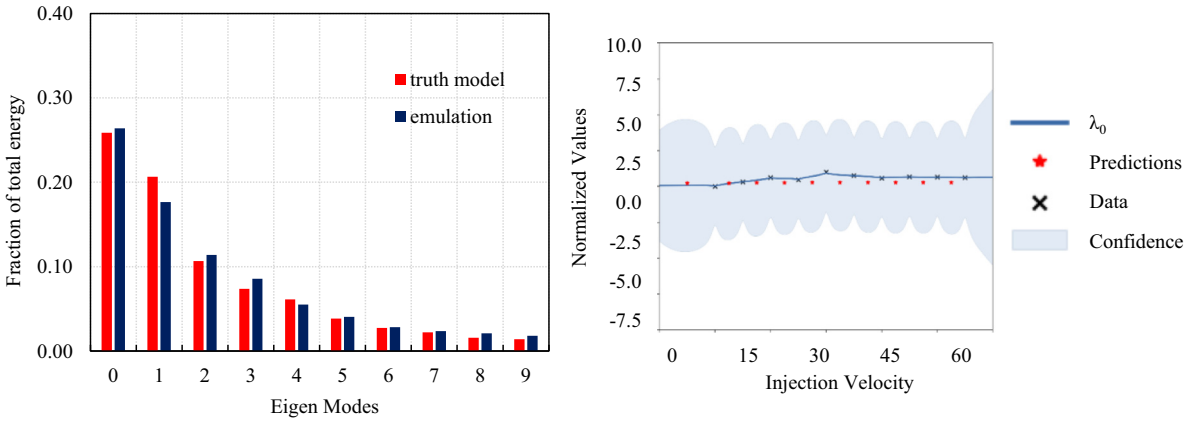

Fig. 9. Comparison of (a) Eigen values for U_j= 22.5 m/s; (b) GP model for first Eigen mode, λ_0

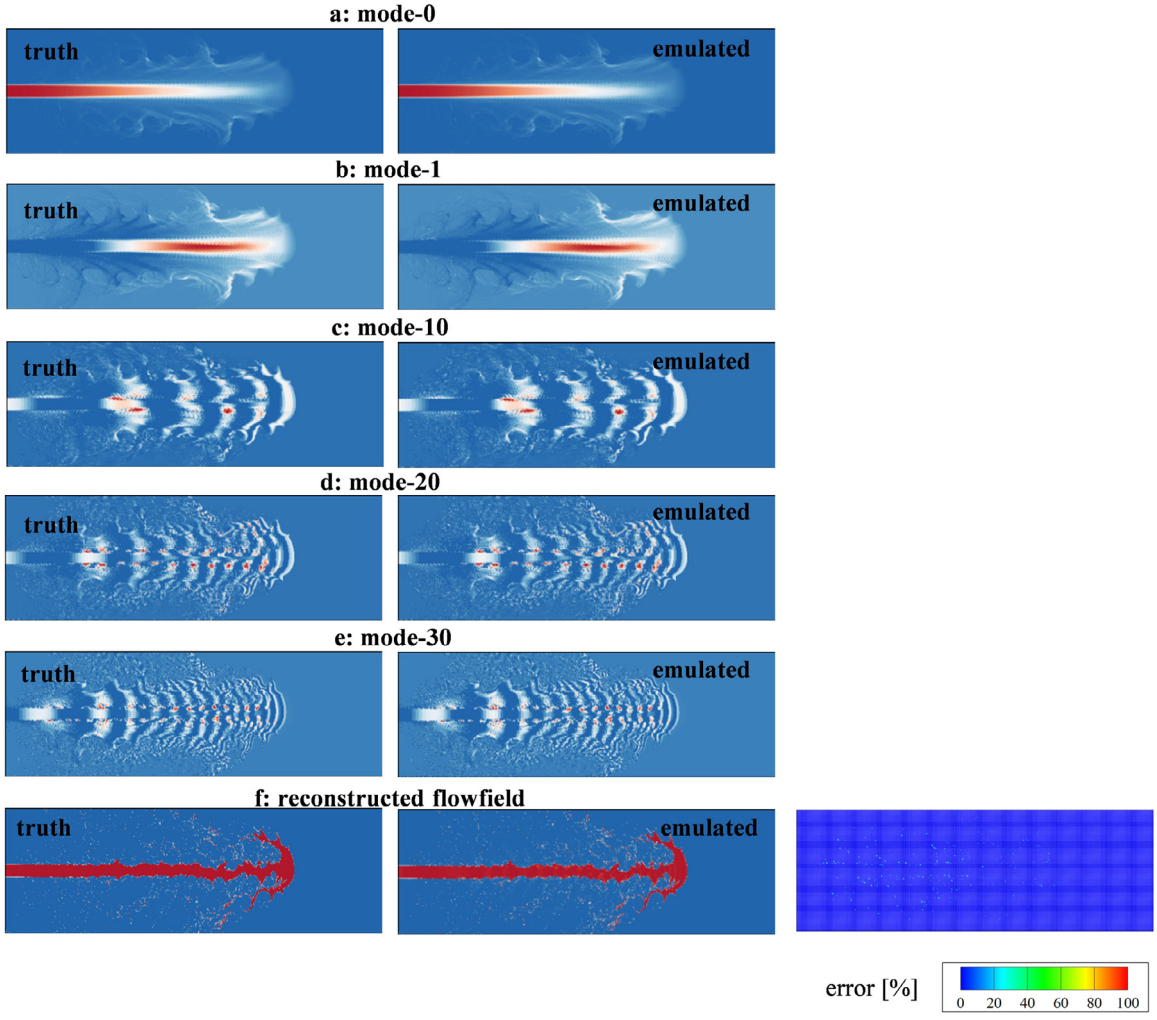

Fig. 10. Comparison of truth model vs emulation predictions for training point, U_j = 45 m/s. From top to bottom: Modes 0, 1, 10, 20, 30 and reconstructed flowfield. L_1 error contours (between the truth model and emulation) are shown in bottom right.

Once the training dataset is learned, measured by how well training data is predicted, the algorithm is used to predict the flowfield at testing conditions.

Step 4.error quantification and speedup

The spatio-temporal flowfield is reconstructed using Galerkin reconstruction of POD modes, basis-functions and time-coefficients proposed by Newman [20]. Fig. 6 shows a comparison between the emulated basis-functions corresponding to modes 0, 1, 5, 10 and 15 and the reconstructed flowfield with simulations conducted using the truth model. von Karman vortex street, Föppl vortices and associated flow features are retained in the emulation, validating our framework. As expected, minor loss of information is observed and is quantified by the L_1error between the truth model and the emulated flowfield for velocity magnitudes. For Re = 185, the er-

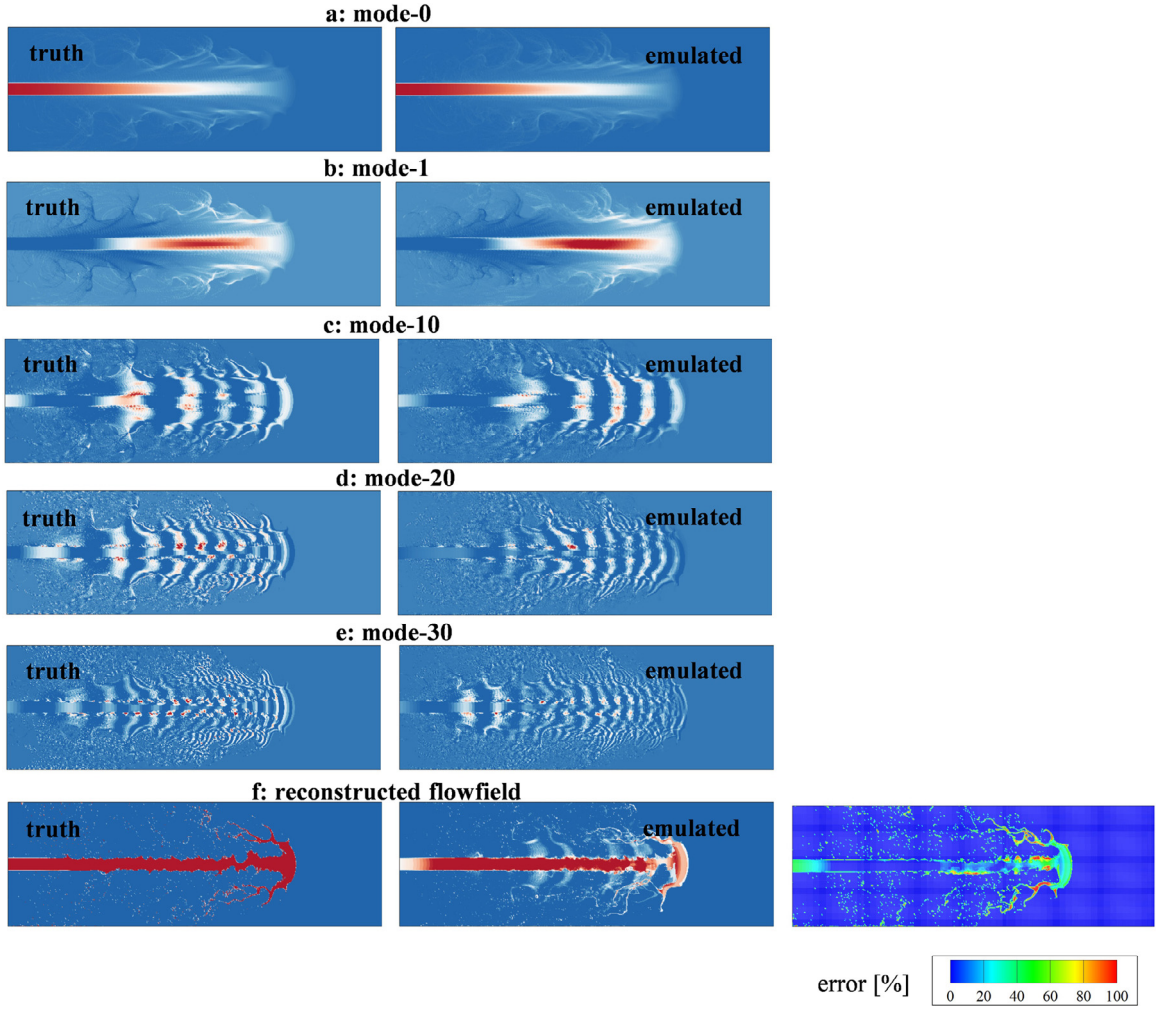

Fig. 11. Comparison of truth model vs emulation predictions for training point, U_j = 22.5 m/s. From top to bottom: Modes 0, 1, 10, 20, 30 and reconstructed flowfield. L_1 error contours (between the truth model and emulation) are shown in bottom right.

ror is 4.40% and the speedup achieved is 256. To further demonstrate the accuracy of our framework, Fig. 7 shows the comparison of velocity profiles in the horizontal and vertical directions at four different sections for the test point, Re = 185. As can be observed from the figure, there is an excellent agreement between the truth model and emulated results.

3.2. Liquid injection in quiescent environment

Next, the framework is applied to 2-D diesel jet injection in quiescent N_2environment. Table 2 lists the fuel properties, operating conditions and the corresponding range of the relevant non-dimensional numbers. All simulations are conducted at a chamber pressure of 30.1 atm and room temperature conditions. The injection velocities are varied from 10 to 55 m/s, corresponding to Weber numbers ranging from 11.5 to 348. The grid adapts based on the value and gradient of the VOF variable to refine the interior of the liquid field and the gas-liquid interface, respectively. Prior to applying the Gaussian process based framework, 3-D simulations of diesel injection in stagnant ambient at p = 30 atm and T = 300 K, and inlet fluctuations with amplitude of 5% of injection velocity in the frequency range of 0-100 Hz are conducted and compared with published literature [58] to establish confidence in the truth model.

Fig. 8 shows the comparison of unperturbed jet lengths as a function of fluctuation frequency from current simulations as compared to published data by Yang and Turan [58] for We = 287.5, showing excellent agreement, including the jump in the unperturbed length at 40 Hz.

Table 2
Diesel fuel properties, operating conditions and corresponding non-dimensional numbers.

(a) Liquid – Diesel		
Jet diameter	D	100 μm
Density	ρ_l	840 kg/m^3
Viscosity	μ_l	2.87e-3 Pa-s
Surface tension	σ_l	2.95e-2 N/m
Jet velocity	U_j	10 to 55 m/s
(b) Gas – Nitrogen		
Density	ρ_g	34.5 kg/m^3
Viscosity	μ_g	1.97e-5 Pa-s
Pressure	p	30.1 atm
Temperature	T	300 K
(c) Non-Dimensional Numbers		
Density ratio	ρ_l/ρ_g	24.58
Reynolds number	$\rho_l\ U_j\ D/\mu_l$	295 to 1675
Weber number	$\rho_g\ U_j\ ^2D_/\sigma_l$	11.5 to 348

Step 1. training dataset

The training points consist of results from 2D spatio-temporally varying spray fields at 10 equally spaced points in injection velocity parameter space, given by: 10, 15, 20, ..., 55. A testing database is also established represented by injection velocities of 12.5, 17.5,

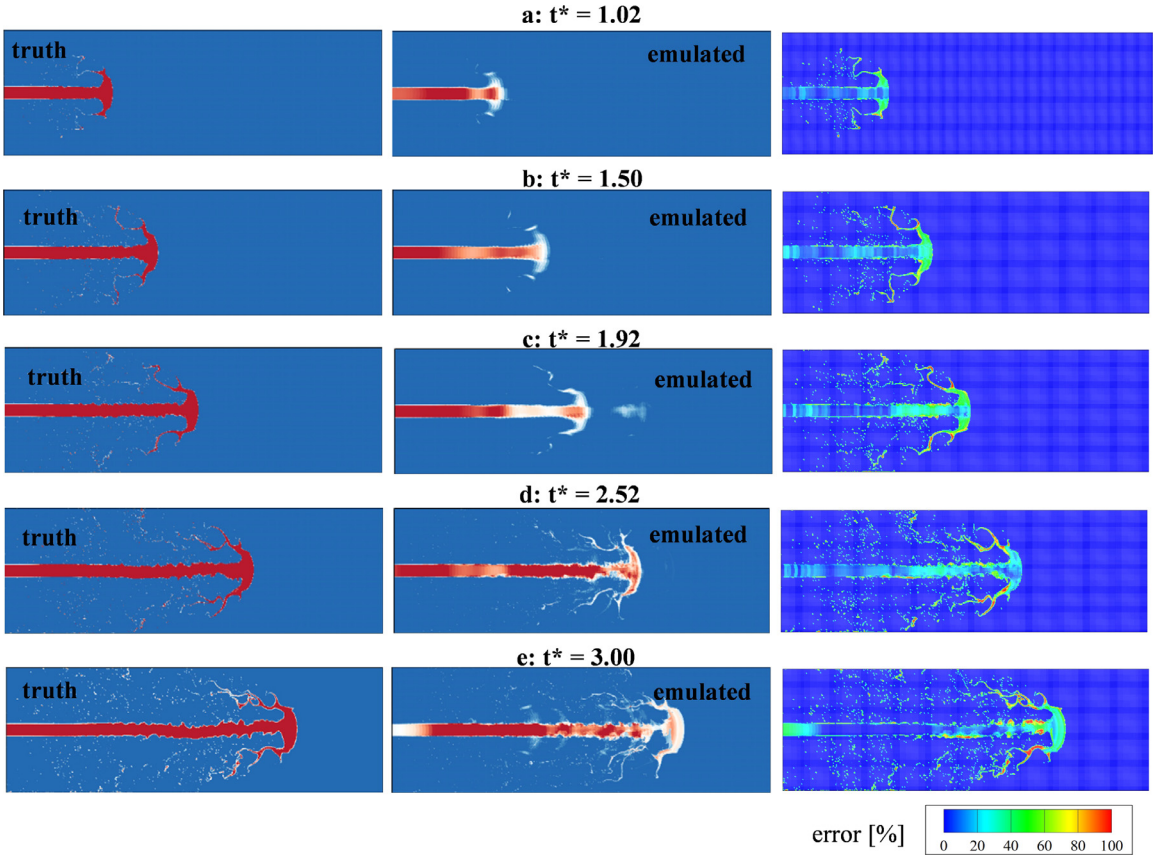

Fig. 12. Time evolution of liquid jet injection and atomization for $U_j = 22.5$; comparison between truth model and emulation. Non-dimensional time, $t^* = t/(U_j/D_0)$. L_1 error contours (between the truth model and emulation) are shown on the right at each time.

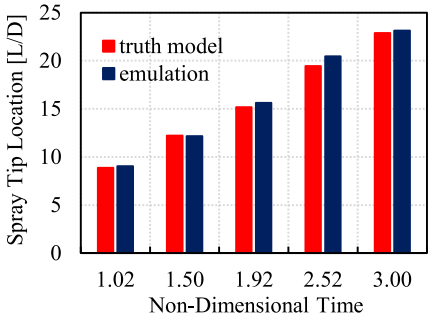

Fig. 13. Comparison of spray tip location between predicted by emulations and truth model at various non-dimensional times.

22.5, ..., 52.5. To ensure that results of emulation are independent of the number of training points, datasets are added progressively until the emulation results are invariant; this resulted in the chosen training dataset mentioned above.

Step 2. dimensionality reduction

Similar to the cylinder flow simulation, a windowing function with 120000 discrete and uniformly spaced grid points was used to capture data in a region of interest inside the flow domain. These set of points are kept consistent for the training dataset described in step 1. Dimensionality reduction is performed with POD on the simulation data to extract the dominant energetic modes containing 95% of flow energy. The training data was normalized to reduce stiffness before performing POD that resulted in a set of mean values. The GP algorithm is then run on four different data value types - POD modes, mean and basis-function values, and time-coefficients. As shown in Fig. 8a, for the testing point of $U_j = 22.5$ m/s, the first 37 (out of 101) modes retain the required 95% of energy content, with mode 0 capturing 26.4% of it. The other injection velocities show similar results.

Step 3. Gaussian process training

The kernel functions used for flow over a cylinder emulated the data for liquid jet atomization process well without modifications, therefore, only machine learning emulation results are discussed in the rest of this section. Fig. 9a show the comparison of the first 10 emulated POD modes with the truth model for the testing point identified by $U_j = 22.5$ m/s; excellent agreement is achieved. The error between the emulated Eigen modes (that contains 95% of the total energy) and the truth model for the test point, $U_j = 22.5$ m/s is 9.49%. Fig. 9b shows the prediction of the largest energy containing mode, $\lambda_0$for the entire range of injection velocities from 10 to 55 m/s using the machine learning algorithm. The prediction for the training as well as testing datasets is within 2σ confidence interval, and as expected, the uncertainty increases for conditions that are out of bounds of the training conditions.

Before predicting the spatio-temporally varying flowfield for the test conditions using the trained machine learning algorithm, its accuracy and consistency is confirmed by predicting the basis-functions and time-coefficients of the training dataset. As demonstrated in Fig. 10a-f, which shows excellent agreement is achieved between the basis-function values from POD (of the truth model, shown on the left) and GPML (on the right) for modes 0, 1, 10, 20, 30 for a training point corresponding to $U_j = 45$ m/s. The reconstructed spray field, shown in Fig. 10f also compares very well with the truth model. The algorithm is then used to predict the

modes and spray field for test condition of $U_j = 22.5$ m/s, shown in Fig. 11.

Fig. 12 shows the time evolution of spray field for the test point corresponding to $U_j = 22.5$. Emulation results are in the middle, the L_1 error contours on the right and the the truth model on the left. The reconstructed flowfields compare well with the truth model and capture ligamentation, droplet formation and shedding accurately. The average L_1 error between the truth model and emulation for the VOF variable at an injection velocity of 22.5 m/s is 15.76%. The corresponding speedup is estimated to be 8000, with the truth model taking 33 hours and 20 minutes on 32 compute cores to simulate the spatio-temporal evolution of diesel injection and atomization process and emulation taking 8 min on a single core.

The L_1 error for this configuration is considerably higher than the cylinder case. This is primarily due to the fact that during the POD analysis only 95% of the energy is captured for this case as opposed to 99% for the cylinder case. Capturing 99% of the energy for the liquid jet injection case, requires 89 of the 101 modes that would have led to significantly higher training costs thus reducing the speedup. That said, the framework can easily produce more accurate results at the expense of speedup.

Spray tip locations predicted by the GP based surrogate model are also extracted. Fig. 13 shows the comparison of the time evolution of the spray tip for 5 different non-dimensional times for the test point corresponding to diesel injection velocity of 22.5 m/s, showing excellent agreement with the truth model. The average error for the 5 times between the truth model and emulation results is 2.40%.

4. Conclusions

A data-driven surrogate modeling framework was developed to emulate spatio-temporal gaseous and spray fields using Gaussian process-based machine learning techniques. The developed framework is applied to two canonical configurations of flow over a circular cylinder and diesel jet injection in quiescent ambient. The GP algorithm was trained using data generated by solving the incompressible form of the Navier-Stokes equations with surface tension (for the diesel injection case). To identify system dynamics and reduce the dimensionality of the training dataset, POD was applied as the first step. Next, based on the flow behaviors, kernel functions that are positive, definite and stationary are selected. Once trained the predictions have a 2σ confidence interval and acceptable L_1 error norm, especially considering the enormous speedup (8000 for the diesel injection case) achieved as a result of using the GP based surrogate model. A mathematical formalism of selecting kernel functions is yet to be developed, however, we used only one set of kernel functions for both the case studies. It is hoped that this is a good first step in making it possible to conduct previously unaffordable and prohibitive sensitivity analysis and uncertainty quantification studies for routine design calculations of single and multiphase flow dynamics of engineering devices.

Acknowledgments

The support from University of Cincinnati's Office of Research, the College of Engineering and Applied Science, and the department of Aerospace Engineering and Engineering Mechanics is gratefully acknowledged. The authors also thank Dr. Stephane Popinet for the development of VOF method and the Gaussian process summer school team for developing and sharing the Gaussian process-based machine learning tools.

References

[1] Soller S, Behr R, Beyer S, Laithier F, Lehmann M, Preuss A, et al. Design and testing of liquid propellant injectors for additive manufacturing. In: Galfetti Luciano, editor. 7th European Conference for Aeronautics and Space Sciences (Eucass 2017). EUCASS 2017; 2017.
[2] Battistoni M, Grimaldi CN. Numerical analysis of injector flow and spray characteristics from diesel injectors using fossil and biodiesel fuels. Appl Energy 2012;97:656–66.
[3] Gorokhovski M, Herrmann M. Modeling Primary Atomization. Ann Rev Fluid Mech 2008;40:343–66.
[4] Yang V. Modeling of supercritical vaporization, mixing, and combustion processes in liquid-fueled propulsion systems. Proc Combust Inst 2000;28:925–42.
[5] Balachandar S, Eaton JK. Turbulent dispersed multiphase flow. Ann Rev Fluid Mech 2010;42:111–33.
[6] Queipo NV, Haftka RT, Shyy W, Goel T, Vaidyanathan R, Kevin Tucker P. Surrogate-based analysis and optimization. Prog Aerosp Sci 2005;41:1–28.
[7] Arienti M, Li X, Soteriou MC, Eckett CA, Sussman M, Jensen RJ. Coupled level-set/volume-of-fluid method for simulation of injector atomization. J Propul Power 2013;29:147–57.
[8] Brackbill JU, Kothe DB, Zemach C. A continuum method for modeling surface tension. J Comput Phys 1992;100:335–54.
[9] Reitz RD, Braco FV. Mechanism of atomization of a liquid jet. Phys Fluids 1982;25:1730.
[10] Herrmann M. On simulating primary atomization using the refined level set grid method. At Sprays 2011;21:283–301.
[11] Bai Z, Brunton SL, Brunton BW, Kutz JN, Kaiser E, Spohn A, et al. Data-driven methods in fluid dynamics: sparse classification from experimental data. In: Pollard A, Castillo L, Danaila L, Glauser M, editors. Whither turbulence and big data in the 21st century?. Cham: Springer International Publishing; 2017. p. 323–42.
[12] Matthew OW, Clarence WR, Igor M, Ioannis GK. Data fusion via intrinsic dynamic variables: an application of data-driven Koopman spectral analysis. EPL (Europhys Lett) 2015;109:40007.
[13] Hemati MS, Williams MO, Rowley CW. Dynamic mode decomposition for large and streaming datasets. Phys Fluids 2014;26:111701.
[14] Taira K, Brunton SL, Dawson STM, Rowley CW, Colonius T, McKeon BJ, et al. Modal analysis of fluid flows: an overview. AIAA J 2017;55:4013–41.
[15] Kitsios V, Cordier L, Bonnet JP, Ooi A, Soria J. On the coherent structures and stability properties of a leading-edge separated aerofoil with turbulent recirculation. J Fluid Mech 2011;683:395–416.
[16] Arndt REA, Long DF, Glauser NM. The proper orthogonal decomposition of pressure fluctuations surrounding a turbulent jet. J Fluid Mech 1997;340:1–33.
[17] Arienti M, Soteriou MC. Time-resolved proper orthogonal decomposition of liquid jet dynamics. Phys Fluids 2009;21:1–15.
[18] Arienti M, Soteriou MC, Hagen G, Corn ML. Analysis of liquid jet atomization dynamics using proper orthogonal decomposition. In: 47th AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition, January 5, 2009 - January 8; 2009. p. 1–17.
[19] Lumley JL. The strucure of inhomogeneous turbulent flows. Atmos Turbul Radio Wave Propagat 1967:166–78.
[20] Newman AJ. Model reduction via the karhunen-loeve expansion. Institute for Systems Research; 1996. p. 15.
[21] Sirovich L. Turbulence and the dynamics of coherent structures. I-III. Quart Appl Math 1987;45:561–90.
[22] Cordier L, Bergmann M. Two typical applications of POD: coherent structures reduction and reduced order modelling. von Karman Institute for Fluid Dynamics; 2003.
[23] Cordier L, Bergmann M. Post-processing of experimental and numerical data: POD an overview. von Karman Institute for Fluid Dynamics; 2003. p. 1–46.
[24] Moonen P, Allegrini J. Employing statistical model emulation as a surrogate for CFD. Environ Model Software 2015;72:77–91.
[25] Bright I, Lin G, Kutz JN. Compressive sensing based machine learning strategy for characterizing the flow around a cylinder with limited pressure measurements. Phys Fluids 2013:25.
[26] Siegel SG, Seidel J, Fagley C, Luchtenburg DM, Cohen K, McLaughlin T. Low-dimensional modelling of a transient cylinder wake using double proper orthogonal decomposition. J Fluid Mech 2008;610:1–42.
[27] Wang Q, Hesthaven JS, Ray D. Non-intrusive reduced order modeling of unsteady flows using artificial neural networks with application to a combustion problem. J Comput Phys 2019;384:289–307.
[28] Takbiri-Borujeni A, Kazemi H, Nasrabadi N. A data-driven proxy to Stoke's flow in porous media. arXiv preprint arXiv:190506327. 2019.
[29] Raissi M, Perdikaris P, Karniadakis GE. Physics informed deep learning (part ii): data-driven discovery of nonlinear partial differential equations. arXiv preprint arXiv:171110561. 2017.
[30] Moiz AA, Pal P, Probst D, Pei Y, Zhang Y, Som S, et al. A machine learning – genetic algorithm (ML-GA) approach for rapid optimization using high-performance computing. SAE Techn Papers 2018;01:15.
[31] Brunton S, Proctor JL, Kutz JN. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. Proc Natl Acad Sci 2016:3932–7.
[32] Yeh S-t, Wang X, Sung C-l, Mak S, Chang Y-h, Zhang L, et al. Common proper orthogonal decomposition-based spatiotemporal emulator for design exploration. AIAA J 2018:1–15.

[33] Grigo C, Koutsourelakis PS. Bayesian model and dimension reduction for uncertainty propagation: Applications in random media. SIAM-ASA J Uncertain Quant 2019;7:292–323.
[34] Grigo C, Koutsourelakis P-S. A physics-aware, probabilistic machine learning framework for coarse-graining high-dimensional systems in the Small Data regime. arXiv preprint arXiv:190203968. 2019.
[35] Lumley JL. The structure of inhomogeneous turbulent flows. Atmos Turbul Wave Propagat 1967:166–78.
[36] Sirovich L. Turbulence and the dynamics of coherent structures. I. Coherent structures. Quart Appl Math 1987;45:561–71.
[37] McKay MD, Beckman RJ, Conover WJ. A Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics 1979;21:239–45.
[38] Shields MD, Zhang J. The generalization of Latin hypercube sampling. Reliab Eng Syst Saf 2016;148:96–108.
[39] Viana FAC. A tutorial on latin hypercube design of experiments. Qual Reliab Eng Int 2016;32:1975–85.
[40] Pópinet S. An accurate adaptive solver for surface-tension-driven interfacial flows. J Comput Phys 2009;228:5838–66.
[41] Tryggvason G, Scardovelli R, Zaleski S. Direct numerical simulations of gas-liquid multiphase. Flows: Cambridge University Press; 2011.
[42] Ma D, Chen X, Khare P, Yang V. Atomization and breakup characteristics of liquid sheets formed by two impinging jets. 49th AIAA Aerospace Sciences Meeting Orlando, FL. AIAA; 2011.
[43] Chen X, Ma D, Khare P, Yang V. Energy and mass transfer during binary droplet collision. 49th AIAA aerospace sciences meeting Orlando, FL. AIAA; 2011.
[44] Fuster D, Agbaglah G, Josserand C, Popinet S, Zaleski S. Numerical simulation of droplets, bubbles and waves: state of the art. Fluid Dyn Res 2009;41:24.
[45] Khare P, Yang V. A study on the breakup of water droplets. submitted (under review). 2018.
[46] Notaro V, Khare P, Lee JG. Mixing characteristics of non-newtonian impinging jets at elevated pressures. Flow Turbul Combust 2018.
[47] Chen H, Reuss DL, Sick V. On the use and interpretation of proper orthogonal decomposition of in-cylinder engine flows. Meas Sci Technol 2012;23:085302.
[48] Ebden M. Gaussian processes: a quick introduction. arXiv preprint arXiv:150502965. 2015.
[49] Rasmussen CE. Gaussian processes in machine learning. In: Bousquet O, von Luxburg U, Rätsch G, editors. Advanced lectures on machine learning: ML summer schools 2003, Canberra, Australia, February 2 - 14, 2003, Tübingen, Germany, August 4 - 16, 2003, Revised Lectures. Berlin, Heidelberg: Springer Berlin Heidelberg; 2004. p. 63–71.
[50] Chen Z, Wang B. How priors of initial hyperparameters affect Gaussian process regression models. Neurocomputing 2018;275:1702–10.
[51] Sivia D, Skilling J. Data analysis: a Bayesian tutorial, Chapter 6. 2 ed. New York: OUP Oxford; 2006.
[52] Hadfield S, Lawrence ND, Wilkinson R, Durrande N. Gaussian process summer school. Gaussian Process Summer School; 2017.
[53] Gaussian Process Summer School. GPy: A Gaussian process framework in Python, 2014. https://github.com/SheffieldML/GPy.
[54] Williamson CHK. Defining a universal and continuous Strouhal-Reynolds number relationship for the laminar vortex shedding of a circular cylinder. Phys Fluids 1988;31:2742–4.
[55] Norberg C. LDV-measurements in the near wake of a circular cylinder. Advances in understanding of bluff body wakes and vortex-induced vibration 1998. p. 1-12.
[56] Inoue O, Sakuragi A. Vortex shedding from a circular cylinder of finite length at low Reynolds numbers. Phys Fluids 2008:20.
[57] Lienhard JH. Synopsis of lift, drag, and vortex frequency data for rigid circular cylinders. Washinton State Univ Bull 1966;300:32.
[58] Yang X, Turan A. Simulation of liquid jet atomization coupled with forced perturbation. Phys Fluids 2017;29:022103.
[59] Ganti Himakar, Khare Prashant. Binary collision of CMAS droplets—Part I: Equal-sized droplets. Journal of Materials Research 2020(Sandphobic Thermal/Environmental Barrier Coatings for Gas Turbine Engines):1–15. doi:10.1557/jmr.2020.138.
[60] Ganti Himakar, Khare Prashant. Binary collision of CMAS droplets—Part II: Unequal-sized droplets. Journal of Materials Research 2020(Sandphobic Thermal/Environmental Barrier Coatings for Gas Turbine Engines):1–13. doi:10.1557/jmr.2020.153.