IOP Conf. Series: Journal of Physics: Conf. Series **1037** (2018) 072004 doi:10.1088/1742-6596/1037/7/072004

Data-Driven Machine Learning for Wind Plant Flow Modeling

R. N. King[1], C. Adcock[2], J. Annoni[1] and K. Dykes[1]

[1] National Renewable Energy Laboratory, Golden, CO, USA
[2] Massachusetts Institute of Technology, Cambridge, MA, USA

E-mail: ryan.king@nrel.gov

Abstract. In this paper, we introduce a data-driven machine learning framework for improving the accuracy of wind plant flow models by learning turbulence model corrections based on data from higher-fidelity simulations. First, a high-dimensional PDE-constrained optimization problem is solved using gradient-based optimization with adjoints to determine optimal eddy viscosity fields that improve the agreement of a medium-fidelity Reynolds-Averaged Navier Stokes (RANS) model with large eddy simulations (LES). A supervised learning problem is then constructed to find general, predictive representations of the optimal turbulence closure. A machine learning technique using Gaussian process regression is trained to predict the eddy viscosity field based on local RANS flow field information like velocities, pressures, and their gradients. The Gaussian process is trained on LES simulations of a single turbine and implemented in a wind plant simulation with 36 turbines. We show improvement over the baseline RANS model with the machine learning correction, and demonstrate the ability to provide accurate confidence levels for the corrections that enable future uncertainty quantification studies.

1. Introduction

Effective wind plant flow modeling requires balancing competing objectives in terms of physical fidelity and computational cost. While high-fidelity approaches, such as large eddy simulations (LES), are becoming increasingly common in wind energy research, they are computationally intractable for use in wind plant design and optimization. Similarly, remote sensing devices can now provide detailed experimental data about actual wind plant flow fields, but a rigorous framework for incorporating these observations into flow models is lacking. In this paper we demonstrate the use of data-driven corrections that can improve the accuracy of a less expensive models, and propose a machine learning framework for making the corrections predictive in new wind plant configurations or in new atmospheric conditions.

Our approach solves a high-dimensional optimization problem to find optimal corrections to turbulence closure models that produce better agreement with data from high-fidelity simulations. The same approach could be equally applied to experimental field data. A nonparametric machine learning model, using Gaussian process regression, is then trained to predict the optimal corrections. This Gaussian process (GP) can then be used to predict corrections to the low-fidelity model given new configurations outside of the training set, such as a different turbine rotor diameter, turbulence intensity, or inflow velocity. The resulting

workflow provides a rigorous, repeatable, and portable way to leverage high-fidelity wind plant data in improving the performance of reduced order models.

2. Approach to Learning Optimal Model Corrections

Our proposed approach builds on recent work in the area of using machine learning to find optimal turbulence closure models or build corrections to existing models in Reynolds-averaged Navier Stokes (RANS) and LES [1, 2, 3, 4, 5]. We develop a data-driven machine learning framework to improve the performance of WindSE [6], a medium-fidelity steady-state RANS models with 2D and 3D capabilities and nonrotating actuator disk turbine representations. WindSE is based on the Python FEniCS package [7], a general purpose PDE solver, along with dolfin-adjoint [8], an automatic differentiation package. The dolfin-adjoint library automates the process of deriving adjoint equations to PDE-constrained problems implemented in FEniCS [9]. By solving the adjoint equations to the RANS wind plant flow solver, we efficiently obtain high dimensional gradients for use in optimizing field variables in the turbulence closure model.

2.1. Optimization Problem Formulation

WindSE is often run in 2D to reduce the computational cost during layout optimizations. The physics of 2D and 3D turbulent flows are fundamentally different and cause different wake recovery rates. Our data-driven corrections attempt to improve the agreement between 2D WindSE RANS and hub-height LES to retain the computational advantages of a 2D simulation, while emulating the physical behavior of a 3D simulation. The model corrections are determined from the following PDE-constrained optimization problem for the best eddy viscosity field $\widetilde{\nu}_T(\mathbf{x})$:

$$\widetilde{\nu}_T(\mathbf{x}) = \arg\min_{\nu_T(\mathbf{x})} \quad \int_\Omega \|\overline{u}_i(\mathbf{x}) - u_i^{\mathrm{LES,HH}}(\mathbf{x})\|_2^2 \,\mathrm{d}\mathbf{x} + \lambda \int_\Omega \|\nabla \nu_T(\mathbf{x})\|_2^2 \,\mathrm{d}\mathbf{x} \tag{1}$$

$$\text{s.t.} \quad \overline{u}_j \frac{\partial \overline{u}_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial \overline{p}}{\partial x_i} + \nu \frac{\partial^2 \overline{u}_i}{\partial x_j^2} - \frac{\partial \overline{u_i' u_j'}}{\partial x_j} + \frac{1}{\rho}\sum_{n=1}^{N} f_{AD,n}\hat{\mathbf{n}}_1 \tag{2}$$

$$\frac{\partial \overline{u}_i}{\partial x_i} = 0 \tag{3}$$

$$\overline{u_i' u_j'} = -2\nu_T \overline{S}_{ij} \tag{4}$$

$$f_{AD,n} = \frac{1}{2}\rho A_n c_{t,n}' \overline{u}_i^2 \beta_n^{-1} \varphi_n(\mathbf{x}) \tag{5}$$

where Ω is the whole computational domain, $\overline{u}_i(\mathbf{x})$ is the RANS velocity, $\overline{p}(\mathbf{x})$ is the RANS pressure field $u_i^{\mathrm{LES,HH}}(\mathbf{x})$ is the LES hub height eddy viscosity field, $\overline{u_i' u_j'}$ is the Reynolds stress tensor which we model using the gradient transport hypothesis, $f_{AD,n}$ is the body force from the nth actuator disk turbine, $\varphi_n(\mathbf{x})$ is a geometric smoothing kernel for distributing the turbine body force, ρ is the density of air, and λ is a regularization constant.

The objective function has two terms in it, one is an error term that penalizes deviations in the WindSE results from the time-averaged LES hub-height velocities, and the other is a regularization term that penalizes gradients in the eddy viscosity field. This regularization term enforces our prior belief that the eddy viscosity field should be reasonably smooth. The optimized field, $\widetilde{\nu}_T(\mathbf{x})$, will produce the best 2D WindSE RANS results relative to the LES hub height velocities, and will be the target of our machine learning.

The decision variable, $\nu_T(\mathbf{x})$, is a spatially varying field and as such has the same dimensions as the number of mesh vertices ($\sim 50,000$). Normally, such high dimensional optimization problems would be intractable; however we use gradient-based optimization techniques that efficiently find local optima. Obtaining the necessary gradients is also normally challenging. The adjoint capabilities in FEniCS automatically calculate high dimensional gradients at a cost

that is independent of the dimension of the control variable, enabling gradient based optimization of high dimensional problems.

A similar field optimization problem was studied by [12] to learn the best mixing length field in a mixing length turbulence closure model: $\nu_T = \ell_{mix}^2 \left(2\overline{S}_{ij}\overline{S}_{ij}\right)$. The resulting correction field was represented as the summation of three Gaussian distributions. In this study, we do not make a mixing length closure model assumption and instead focus on optimizing the eddy viscosity field itself. We then learn a general, nonparameteric function to predict the eddy viscosity field directly, without any assumption of a mixing length model or summation of Gaussian distributions representation. In our results section, we compare the GP improvements to this previous mixing length model.

2.2. High-Fidelity Training Data

Data from a high-fidelity computational fluid dynamics model known as the Simulator for Wind Farm Applications (SOWFA) was used to train the machine learning correction to WindSE. SOWFA is a large eddy simulation code that was developed at the National Renewable Energy Laboratory (NREL) to study atmospheric boundary layer turbulent flow within a wind plant [10]. In the simulations used in this paper, SOWFA uses an actuator disk and solves the unsteady, three-dimensional, filtered, incompressible Navier-Stokes equations and transport of potential temperature equations, which take into account the thermal buoyancy and earth rotation (Coriolis) effects in the atmosphere. SOWFA takes on the order of a day to compute 30 minutes of simulation time using NREL's supercomputer, which is prohibitively expensive for wind plant optimization studies or real-time control applications.

In this paper, initial RANS model improvements were learned from an LES study of a single turbine. The NREL 5MW turbine [11], which has a rotor diameter of 126 m and a hub height of 90 m, was simulated in a 5 km $\times$ 2 km $\times$ 1 km domain with grid spacing of 10 m. The simulations were run on 200 cores and the surface roughness was adjusted to produce a low turbulence intensity of 5.6%. The resulting LES fields were time averaged to enable comparison with RANS fields.

After training our machine learning model on a single turbine, we used it to predict the turbulence closure on a simulation of a much larger wind plant with 36 turbines. This larger plant had turbines arranged in a 6$\times$6 regular grid with 6 rotor diameter spacing in the streamwise and spanwise directions. The domain size was 5 km $\times$ 5 km $\times$ 1 km with grid spacing of 10 m. The simulations were run on 500 cores. Surface roughness was also adjusted to produce similar atmospheric conditions, with a resulting turbulence intensity of 6%. The RANS results with our machine learning correction were then compared against these LES results, as discussed in Section 4.

We note that although we demonstrate our framework on LES data, the optimization and learning procedures are not restricted to only LES. Any data source could be used for learning model improvements, such as wind tunnel data, lidar data, or any simulation data that comes from a higher-fidelity model. Our framework is a general approach to deriving spatially-distributed corrections and then developing a machine learning prediction of those corrections in new simulations.

2.3. Supervised Learning of Predictive Correction Function

After determining the optimal eddy viscosity field, we solve a machine learning problem to predict future values of the eddy viscosity in other simulations. We learn a nonparameteric predictive function f, drawn from some hypothesis space $\mathcal{F}$, that best maps from a set of inputs $\boldsymbol{\theta}(\mathbf{x})$ to outputs $\widetilde{\nu}_T(\mathbf{x})$. To do so, we construct a training set consisting of N input–output pairs $\mathcal{D} = \left\{\boldsymbol{\theta}\left(\mathbf{x}^{(i)}\right), \widetilde{\nu}_T\left(\mathbf{x}^{(i)}\right)\right\}_{i=1}^{N}$ sampled from N mesh points. This is considered a supervised learning problem because we have training data labeled as inputs and outputs. Furthermore,

because we seek outputs in a continuous space rather than a discrete set, this is a function regression problem instead of a classification problem.

It is important to construct the input vectors $\boldsymbol{\theta}$ in a way that makes the learned function f applicable in different simulations. Therefore, we use only local RANS flowfield information and avoid introducing any dependence on location or rotor diameter. The flowfield variables that are available to be used as inputs are

$$\boldsymbol{\theta} = \left[\overline{u}_1, u_2, \frac{\partial \overline{u}_1}{\partial x_1}, \frac{\partial \overline{u}_1}{\partial x_2}, \frac{\partial \overline{u}_2}{\partial x_1}, \frac{\partial \overline{u}_2}{\partial x_2}, \frac{\partial^2 \overline{u}_1}{\partial x_1^2} + \frac{\partial^2 \overline{u}_1}{\partial x_2^2}, \frac{\partial^2 \overline{u}_2}{\partial x_1^2} + \frac{\partial^2 \overline{u}_2}{\partial x_2^2}, \overline{p}, \frac{\partial \overline{p}}{\partial x_1}, \frac{\partial \overline{p}}{\partial x_2}\right]. \tag{6}$$

Our desired function approximates the optimal eddy viscosity with some, possibly nonlinear, function of these variables $\widetilde{\nu}_T \approx f(\boldsymbol{\theta})$ that performs well on our training data $\mathcal{D}$ and also generalizes to unseen data.

We choose our function based on the principle of empirical risk minimization [13], by solving the following optimization problem:

$$\min_f \quad \frac{1}{N}\sum_{i=1}^{N} V\left[\widetilde{\nu}_T^{(i)}, f\left(\boldsymbol{\theta}^{(i)}\right)\right] + \gamma R(f) \tag{7}$$

where $\widetilde{\nu}_T^{(i)} = \widetilde{\nu}_T\left(\mathbf{x}^{(i)}\right)$ and $\boldsymbol{\theta}^{(i)} = \boldsymbol{\theta}\left(\mathbf{x}^{(i)}\right)$, $V[\cdot]$ is a loss function corresponding to errors in predicting the optimal eddy viscosity with our function f such as an ℓ_2 norm, and $R(f)$ is a regularization function that penalizes complex functions f. This empirical risk minimization is used as a proxy for minimizing the true expected risk, which would depend on the unknown joint pdf of the input and output data. By applying regularization to the learned function, we express a preference for simpler or smoother functions that are likely to generalize well and not overfit to the training data. A natural measure of f's complexity for $R(\cdot)$ is the norm $\|f\|_{\mathcal{F}}$ in the Hilbert space $\mathcal{F}$ where f is chosen from. This is readily accomplished using Gaussian process regression, as discussed in the following section.

3. Machine Learning with Gaussian Process Regression

To solve the supervised learning problem we model the function f as a Gaussian process (GP), i.e. we assume a Gaussian prior probability distribution of possible functions, and condition on observations from our training data using a kernel covariance function. In doing so, a GP performs Bayesian inference to learn a posterior distribution of possible functions. When evaluating a Gaussian process for a given input, it returns a normal random variable, and each finite combination of these random variables is a multivariate normal distribution.

The GP approach to supervised learning has three remarkable features. First, the kernel choice implicitly defines a reproducing kernel Hilbert space (RKHS), $\mathcal{F}$, which is our function hypothesis space. Based on the Representer Theorem [14], it can be shown a priori that the solution to empirical risk minimization problems can be efficiently expressed as linear combinations of kernel evaluations on training points. This allows for learning very nonlinear or high dimensional functions f because the training costs scale with the number of training points and not the function complexity. Second, it is nonparametric, meaning that we do not impose any particular mathematical or operator form on our closure model a priori. As the GP is provided more training data, its expressiveness or ability to capture complex functions grows. Third, because the GP is Bayesian it provides more complete statistical information about the prediction, including an analytical formulation of the uncertainty in the mean prediction. A comprehensive review of Gaussian process techniques is provided in [15].

The GP representation of f is fully described by a mean function $\mu(\cdot)$ and kernel covariance function $k(\cdot,\cdot)$:

$$f(\boldsymbol{\theta}) \sim \mathcal{GP}\left(\mu(\boldsymbol{\theta}), k\left(\boldsymbol{\theta}, \boldsymbol{\theta}'\right)\right) \tag{8}$$

where without a loss of generality we assume $\mu(\boldsymbol{\theta}) = 0$ and $k(\boldsymbol{\theta}, \boldsymbol{\theta}')$ is the Matern kernel. We use a Matern kernel of degre 3/2, which is a product of a linear polynomial and an exponential function, given as

$$k_{\nu=3/2}(r) = \left(1 + \frac{\sqrt{3}r}{l}\right) \exp\left(-\frac{\sqrt{3}r}{l}\right) \tag{9}$$

where $r = |\boldsymbol{\theta} - \boldsymbol{\theta}'|$ is the distance between inputs and l is a lengthscale hyperparamter that is tuned by optimizing the log-likelihood of the hyperparameter.

After conditioning on the training data $\mathcal{D}$, we arrive at the posterior distribution used to predict the eddy viscosity at a new input vector $\boldsymbol{\theta}^*$:

$$f(\boldsymbol{\theta}^* \mid \mathcal{D}) \sim \mathcal{GP}\left(\mu_{f|\mathcal{D}}, k_{f|\mathcal{D}}\right) \tag{10}$$

$$\mu_{f|\mathcal{D}}(\boldsymbol{\theta}^*) = \mathbf{k}_*^\mathsf{T}\mathbf{K}^{-1}\mathbf{y} \tag{11}$$

$$k_{f|\mathcal{D}}(\boldsymbol{\theta}^*, \boldsymbol{\theta}) = k(\boldsymbol{\theta}^*, \boldsymbol{\theta}^*) - \mathbf{k}_*^\mathsf{T}\mathbf{K}^{-1}\mathbf{k}_* \tag{12}$$

$$\mathbf{k}_* = [k(\boldsymbol{\theta}^*, \boldsymbol{\theta}_1), \ldots, k(\boldsymbol{\theta}^*, \theta_N)] \tag{13}$$

$$\mathbf{K} = K_{ij} = k(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j) \quad \text{for } i, j = 1, \ldots, N \tag{14}$$

Here the new vector $\mathbf{k}_*$ is the vector of kernel function outputs when evaluated with the desired input point $\boldsymbol{\theta}^*$ and each of the training points. Similarly, the $N \times N$ matrix $\mathbf{K}$ contains the kernel covariance function evaluated for each combination of the training points. The vector $\mathbf{y} = y_i = \widetilde{\nu}_T^{(i)}$ contains the training data outputs.

While the GP model is insensitive to the complexity of the function f, its costs do scale with the number of training points N. This arises in an offline step where the matrix $\mathbf{K}$ must be inverted to solve $\mathbf{K}^{-1}\mathbf{y}$, and in several online prediction steps such as calculating $\mathbf{k}_*$ and the matrix-vector multiplication of $\mathbf{K}^{-1}$ and $\mathbf{k}_*$. The offline training is not an issue for implementation in a computational fluid dynamics code, however the online components must be recomputed at each location in the flow. To reduce the online computational cost, we randomly select $N = 100$ points from the complete set of mesh vertices, of which there are around 50,000, and constrain our training data to this subset of the complete data. Future work on a more principled selection of these points could be fruitful.

After conditioning on the training data, the GP makes both a mean prediction of the eddy viscosity, and provides an estimate of the uncertainty in the prediction. As seen in Eq. 10, the posterior uncertainty is determined by the prior covariance predicted by the kernel, reduced by the information gained by the nearby training points. This allows for further model development and a sampling strategy for selecting high fidelity training points to minimize the uncertainty in the machine learning prediction.

4. Optimization Results and Prediction of Wind Plant Flows

The optimized eddy viscosity field is shown in Figure 1. This eddy viscosity should produce the best agreement with LES regardless of which RANS model produces it, i.e. a mixing length model, a $k-\epsilon$ model, or a Gaussian process machine learning model. The desired eddy viscosity features strong mixing in the wake shear layer that peaks approximately 10 rotor diameters downstream of the turbine. It also features a localized increase in mixing immediately upstream of the rotor. The Gaussian process model was trained on $N = 100$ locations and then used to predict the eddy viscosity at every point in the domain. The GP is able to capture the strong mixing in the shear layer and the correct location of the peak, but slightly over predicts the mixing in a broad area upwind and transverse to the rotor diameter. We also compare to the eddy viscosity produced by the original mixing length model in WindSE. The mixing length model generally underpredicts the eddy viscosity and lacks the spike in mixing downstream of

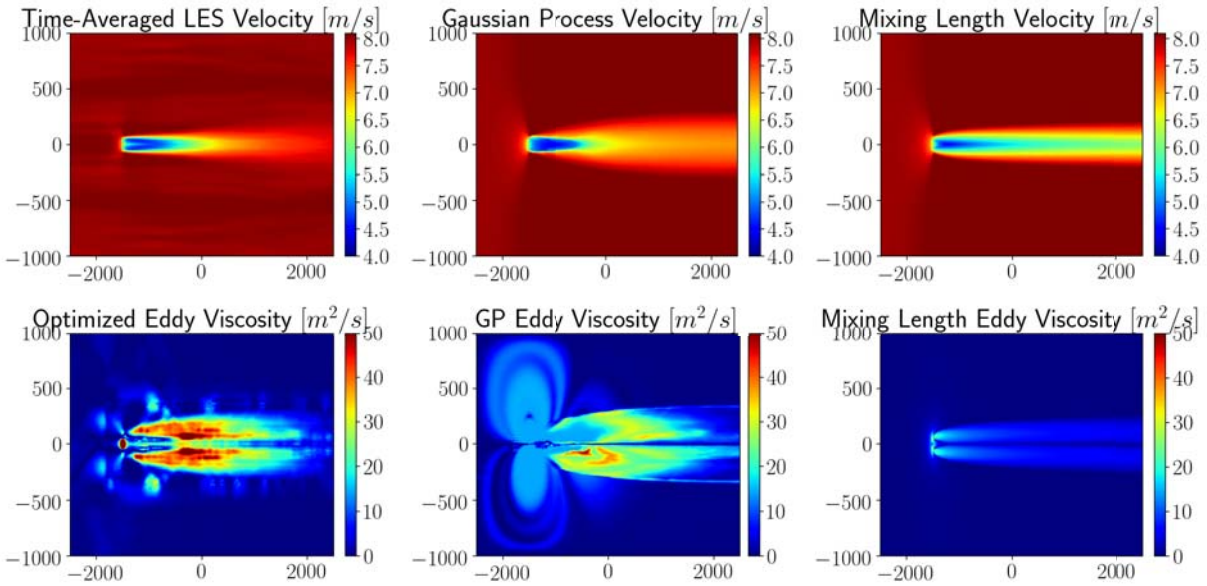

Figure 1: The top row contains a comparison of LES and RANS velocity fields computed from the Gaussian process prediction and the original mixing length RANS model. In the bottom row, the left panel shows the eddy viscosity field that produces optimal agreement with LES, the middle panel shows the eddy viscosity prediction from the Gaussian process trained on 100 mesh points, and the right panel shows the eddy viscosity field produced by the mixing length model. The Gaussian process is alble to capture the increase shear layer mixing, particularly in the far wake, while the mixing length model overpredicts the eddy viscosity in the near wake and underpredicts in the far wake.

the rotor. It also overpredicts the eddy viscosity just at the rotor and in the near wake shear layer.

One valuable feature of Gaussian processes is that they provide analytical estimates of the uncertainty in predictions because each input evaluation actually returns a Gaussian distribution. This provides a confidence level in the machine learning prediction that can be useful for the flow modeling practitioner, as well as a useful tool in experimental design for selecting new training points. Figure 2 shows the GP eddy viscosity mean prediction, the standard deviation for that prediction, and the error in the mean viscosity prediction relative to the true optimal. The strong correlation between the regions with high uncertainty and high errors shows that the GP model fails gracefully, i.e. it can correctly anticipate and inform users about regions where the model is likely to fail.

These regions of high uncertainty typically occur where the GP is extrapolating outside of training points, which is a usually problematic region for surrogate models. Here we see that when asked to extrapolate outside of training points, the GP mean prediction decays back to the mean value, which is usually assumed to be 0 or the mean of the training points, at a rate determined by the covariance kernel length scales. At the same time, the GP reports a growing standard deviation at those extrapolated points, resulting in the broad region of average viscosity and elevated uncertainty upwind of the rotor.

To demonstrate the predictive capability of the machine learning model, we implement the GP model on a wind plant simulation with 36 turbines. As shown in Figure 3, the GP model is able to predict eddy viscosity corrections for unseen data arising in simulations unrelated to the training set. This predictive ability is a crucial characteristic of machine learning models that

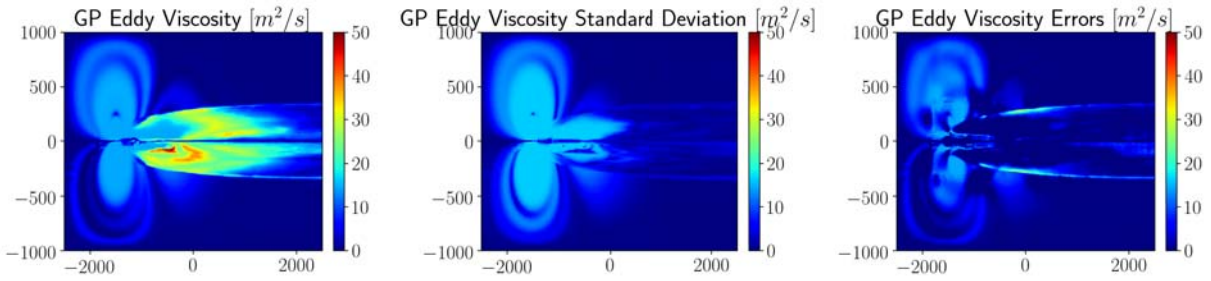

Figure 2: A valuable feature of Gaussian processes is that they provide analytical estimates of the prediction uncertainty. The left panel shows the mean GP prediction, the center panel shows the estimated standard deviation, and the right panel shows the error of the GP prediction relative to the true value. The strong correlation between the standard deviation prediction and the error demonstrates that the GP provides an accurate assessment of its own uncertainty.

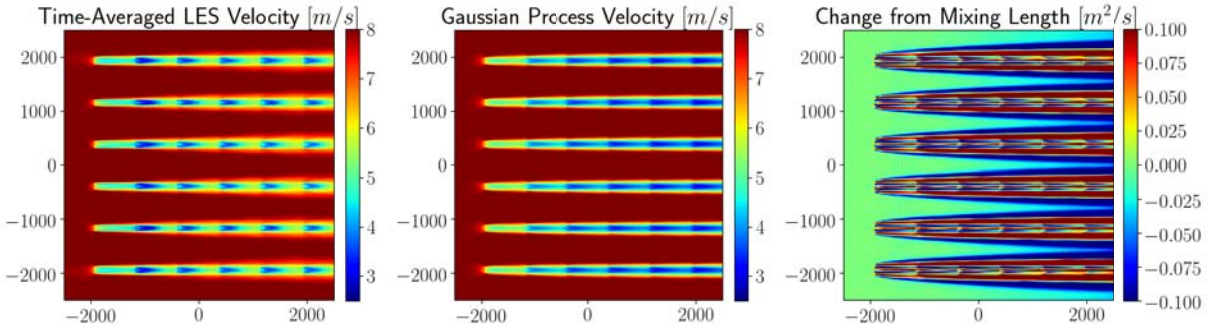

Figure 3: The GP model is truly predictive, and can be used in wind plant simulations that are unrelated to the training set, such as the 36 turbine wind plant shown above. This results in plant-wide improvements to the predicted velocity field. As shown in the right panel, this is difference is particularly apparent in the wake structure and shear layer behavior relative to the original mixing length model.

distinguishes them from ordinary model tuning. The resulting plant flowfield is improved from the original mixing length model, and produces structured changes in the wake shear layers. Because the trained GP only uses local flowfield information as inputs, it can easily adapt to new turbine layouts or different rotor diameters without requiring retraining.

5. Conclusion

This paper has demonstrated a data-driven framework for learning corrections to lower fidelity models from higher fidelity data sources such as LES or experimental data. The desired corrections were identified by solving a high-dimensional optimization problem for a turbulence closure field variable. While such high-dimensional optimization problems are typically challenging, the adjoint capabilities in WindSE enabled an efficient gradient-based optimization of a high dimensional (~ 50000) variable.

Instead of assuming a fixed turbulence closure model as commonly done, we learn a general, nonparametric function to predict the eddy viscosity with no a priori assumptions about its mathematical form. The closure function was found by solving a supervised learning problem using training points sampled from the optimized correction field. A Gaussian process representation of the closure function provided a Bayesian inference pathway to learning the nonparametric function and enabled efficient learning of highly nonlinear representations.

The GP prediction of the eddy viscosity field captures many important features of the desired correction with just 100 training points, such the elevated mixing in the far wake shear layer. Additionally, the GP showed a remarkable ability to predict its own errors thanks to the analytical expression for the variance provided by the GP formulation. This information is invaluable in uncertainty quantification studies and can also enabled informed selection training points in experimental design studies.

Finally, the GP model is truly predictive. It showed improvements to the simulation of a 36 turbine wind plant relative to the original mixing length model when only trained on a single turbine. This predictive ability is enabled by the choice of local field variables, like velocities, pressures, and their gradients, as inputs to the GP model. Because none of the inputs are domain or turbine specific, the GP model can be applied to any layout or turbine configuration, yielding a flexible tool for improving wind plant modeling.

References

[1] Parish E J and Duraisamy K 2016 *Journal of Computational Physics* **305** 758–774 ISSN 0021-9991 URL http://www.sciencedirect.com/science/article/pii/S0021999115007524

[2] Ling J and Templeton J 2015 *Physics of Fluids (1994-present)* **27** 085103 ISSN 1070-6631, 1089-7666 URL http://scitation.aip.org/content/aip/journal/pof2/27/8/10.1063/1.4927765

[3] Singh A P and Duraisamy K 2016 *Physics of Fluids* **28** 045110 ISSN 1070-6631 URL http://aip.scitation.org/doi/abs/10.1063/1.4947045

[4] Wang J X, Wu J L and Xiao H 2017 *Physical Review Fluids* **2** ISSN 2469-990X URL https://link.aps.org/doi/10.1103/PhysRevFluids.2.034603

[5] King R N, Hamlington P E and Dahm W J A 2016 *Physical Review E* **93** 031301 URL http://link.aps.org/doi/10.1103/PhysRevE.93.031301

[6] King R N, Dykes K, Graf P and Hamlington P E 2017 *Wind Energy Science* **2** 115–131 ISSN 2366-7451 URL http://www.wind-energ-sci.net/2/115/2017/

[7] Logg A, Mardal K A and Wells G (eds) 2012 *Automated Solution of Differential Equations by the Finite Element Method* (*Lecture Notes in Computational Science and Engineering* vol 84) (Berlin, Heidelberg: Springer Berlin Heidelberg) ISBN 978-3-642-23098-1 978-3-642-23099-8 fEniCS URL http://link.springer.com/10.1007/978-3-642-23099-8

[8] Farrell P E, Ham D A, Funke S W and Rognes M E 2013 *SIAM Journal on Scientific Computing* **35** C369–C393 ISSN 1064-8275, 1095-7197 URL http://epubs.siam.org/doi/abs/10.1137/120873558

[9] Funke S W and Farrell P E 2013 *arXiv:1302.3894 [cs]* ArXiv: 1302.3894 URL http://arxiv.org/abs/1302.3894

[10] Churchfield M J, Lee S, Michalakes J and Moriarty P J 2012 *Journal of Turbulence* N14 URL http://www.tandfonline.com/doi/abs/10.1080/14685248.2012.668191

[11] Jonkman J, Butterfield S, Musial W and Scott G 2009 Definition of a 5-MW Reference Wind Turbine for Offshore System Development Technical Report NREL/TP-500-38060 National Renewable Energy Laboratory URL http://www.osti.gov/bridge/servlets/purl/947422-nhrlni/

[12] Adcock C, King R N and Annoni J 2018 Data-driven wind farm optimization under varying atmospheric stabilities (To appear in 2018 American Controls Conference)

[13] Hastie T, Tibshirani R and Friedman J H 2009 *The elements of statistical learning: data mining, inference, and prediction* 2nd ed Springer series in statistics (New York, NY: Springer) ISBN 978-0-387-84857-0 978-0-387-84858-7

[14] Schlkopf B and Smola A J 2002 *Learning with kernels: support vector machines, regularization, optimization, and beyond* Adaptive computation and machine learning (Cambridge, Mass: MIT Press) ISBN 978-0-262-19475-4

[15] Rasmussen C E and Williams C K I 2006 *Gaussian processes for machine learning* Adaptive computation and machine learning (Cambridge, Mass: MIT Press) ISBN 978-0-262-18253-9