Multiphase turbulence modeling using sparse regression and gene expression programming

Sarah Beetham[a], Jesse Capecelatro[a,b]

[a] *University of Michigan, Department of Mechanical Engineering*
[b] *University of Michigan, Department of Aerospace Engineering*

Abstract

In recent years, there has been an explosion of machine learning techniques for turbulence closure modeling, though many rely on augmenting existing models. While this has proven successful in single-phase flows, it breaks down for multiphase flows, particularly when the system dynamics are controlled by two-way coupling between the phases. In this work, we propose an approach that blends sparse regression and gene expression programming (GEP) to generate closed-form algebraic models from simulation data. Sparse regression is used to determine a minimum set of functional groups required to capture the physics and GEP is used to automate the formulation of the coefficients and dependencies on operating conditions. The framework is demonstrated on a canonical gas–solid flow in which two-way coupling generates and sustains fluid-phase turbulence.

Keywords: Turbulence modeling, particle-laden flow, sparse regression, gene expression programming

1. Introduction

In the last decade, data-driven approaches have become the predominant tool for developing turbulence models (Duraisamy et al., 2019). Of these approaches, Neural Networks (NNs) have gained a considerable amount of traction (Tracey et al., 2015; Milano and Koumoutsakos, 2002; Lu, 2010; Rajabi and Kavianpour, 2012; Duraisamy and Durbin, 2014; Duraisamy et al., 2015; Ma et al., 2016; Ling et al., 2016; Bode et al., 2019; Liu and Fang, 2019). In contrast, a relative minority of approaches have elected to pursue strategies that enable a compact, algebraic closure. Formulating models in this way has several important properties including increased interpretability, ease of dissemination and straightforward integration into existing solvers. These techniques generally fall into two categories, (1) symbolic regression and (2) gene expression programming.

In the case of sparse regression, Brunton et al. (2016) developed a strategy based on sparse regression that identifies the underlying functional form of the nonlinear physics by optimizing a coefficient matrix that acts upon a matrix of trial functions. This construct has the important benefit of including the user in the modeling process through selection of the trial functions. Schmelzer et al. (2020); Beetham and Capecelatro (2020) recently extended the sparse identification framework of Brunton et al. (2016) to infer algebraic stress models for the closure of the Reynolds-averaged Navier–Stokes (RANS) equations. In Schmelzer et al. (2020), the models are written as tensor polynomials and built from a library of candidate functions. In Beetham and Capecelatro (2020), Galilean invariance of the resulting models are guaranteed through thoughtful tailoring of the feature space.

Gene expression programming (GEP), a data-driven technique inspired by the Darwinian concept of survival-of-the-fittest, heuristically evolves symbolic models until error is reduced beyond a threshold. In recent years, this strategy has gained attention in the turbulence modeling. For example, GEP has demonstrated success in the contexts of modeling large eddy simulation (LES) subgrid scale closures (Reissmann et al., 2021), boundary layer theory (Dominique et al., 2021), turbulent pipe flow (Samadianfard, 2012) and informing RANS closures (Weatheritt and Sandberg, 2016; Zhao et al., 2020).

While these data-driven techniques have been increasingly utilized for modeling single-phase turbulence, their application to multiphase turbulence modeling is still relatively uncharted. Despite this, multiphase flows present a rich and diverse class of problems for which machine learning can prove useful. Due to the large parameter space frequently attributed to such flows, traditional modeling techniques have historically failed, especially beyond dilute regimes, where models extended from single-phase turbulence break down (see, e.g., Fox, 2014, and discussion therein). This divergence from single-phase turbulence theory can be attributed to the fact that at moderate volume fractions, particles generate turbulent kinetic energy (TKE) at the smallest scales. This is the direct antithesis to the classical notion of the turbulent energy cascade. Additionally, numerous practical applications span regimes from dilute to dense particle loadings, motivating the need for models that are accurate across regimes. This motivates the need for methodologies capable of formulating closures 'from scratch,' rather than augmenting existing models. These challenges, along with the societal importance and pervasiveness of these flows, make them excellent candidates for improvements in data-driven modeling techniques.

In this work, we propose a blending modeling approach which combines the strengths of both sparse regression and GEP to inform multiphase turbulence closures in a way that leverages the physical knowledge of the modeler and automates the determination of model components for which physical insight is not obvious or does not exist. To demonstrate the utility of such an approach, we present a simple configuration in which strong two-way coupling between fluid and particles generates and sustains turbulence. This configuration has been discussed extensively in prior work (see, e.g., Capecelatro et al., 2014, 2015; Beetham and Capecelatro, 2020, for more exhaustive details) and serves as a case study here.

2. Methodology

It is well established (Pope, 2000) that any tensor quantity, $\mathcal{D}_{ij}$, can be exactly described by an infinite sum given as

$$\mathcal{D}_{ij} = \sum_{n=1}^{\infty} \beta^{(n)} \mathcal{T}_{ij}^{(n)}, \tag{1}$$

where $\beta^{(n)}$ represents the n-th coefficient associated with a corresponding basis tensor, $\mathcal{T}_{ij}^{(n)}$. The coefficients may range in complexity from constants to nonlinear functions of the principal invariants of the tensor bases. For many configurations, this infinite sum can reduced to a finite sum by leveraging the Cayley–Hamilton theorem. This results in a reduced set of tensors termed a minimal invariant basis (e.g., Spencer and Rivlin, 1958). Using knowledge of the system physics, a minimal invariant basis can be derived. Once this basis is established, modeling can be broken into two tasks: (1) *Which of the basis tensors are most important for capturing the physics at play?* and (2) *How do the coefficients depend on principal invariants or system parameters?*

Sparse regression has been shown to be successful at addressing the first task (Beetham and Capecelatro, 2020; Beetham et al., 2021) and works well for the second task when *constant* coefficients are sufficient. However, when the system has a complex and large parameter space, as is the case for multiphase turbulence, constant coefficients are no longer sufficient for capturing physics across scales. In this situation, sparse regression is not an efficient method for determining the form of the coefficients and requires the modeler to supply all potential test functions to the algorithm manually. While this has important benefits for embedding physics-based reasoning and properties into the resultant model (e.g., form invariance), it implies a tedious, 'guess-and-check' exercise if physics-based arguments can no longer be used to supply test functions. In the present method, we propose to offload this work to a gene expression algorithm when naivity in functional form is unavoidable. This effectively automates the process of evaluating trial functions for the coefficients, while preserving the benefits of using sparse regression to inform the tensorial building blocks of the model.

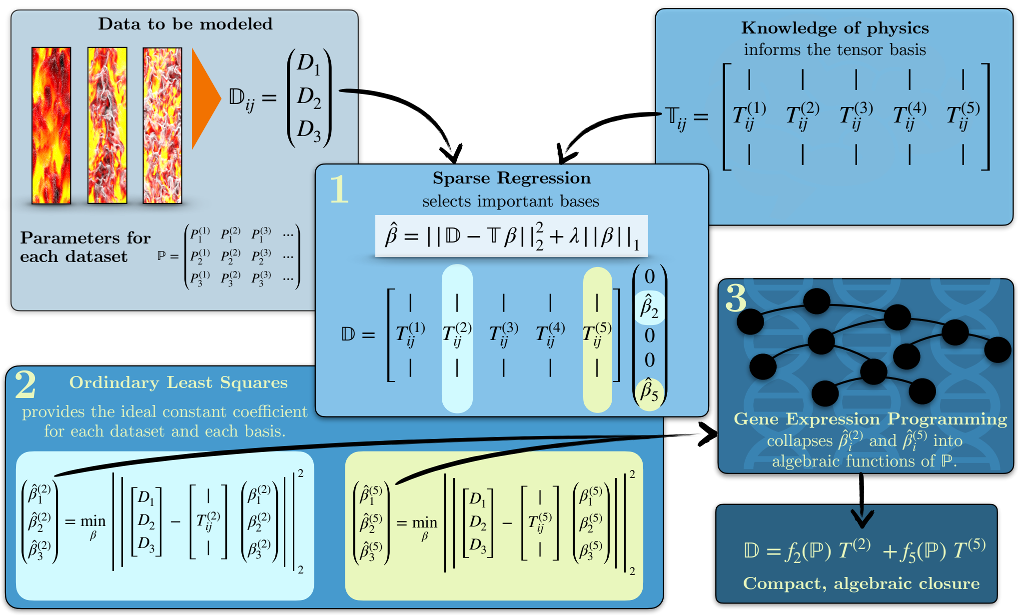

Figure 1: The modeling methodology has three steps: (1) Sparse regression identifies the important basis tensors, (2) OLS squares provides the ideal coefficients for each of the data sets for each of the identified bases and (3) GEP collapses the ideal coefficients for each case into a compact, algebraic closure.

The method can be summarized by three distinct modeling steps, as shown in Fig. 1, and outlined here for data spanning s unique conditions in the parameter space (in the context of multiphase, turbulent flows, these parameters might include solids volume fraction, Reynolds number, etc.):

1. Use sparse regression to identify the basis tensors required to describe physics by optimizing

$$\hat{\beta} = \min_{\beta} ||\mathbb{D} - \mathbb{T}\beta||_2^2 + \lambda||\beta||_1 \tag{2}$$

and assuming constant coefficients. Each base associated with a nonzero coefficient in $\hat{\beta}$ is deemed to be 'essential' and is retained in the final model. The surviving bases are then condensed into a subset of $\mathbb{T}$, denoted $T^{\subset}$.

2. For each of the s conditions, compute the ideal constant coefficients associated with the p essential bases, using Ordinary Least Squares (OLS):

$$\hat{\beta}_s^{\subset} = \min_{\hat{\beta}_s^{\subset}} ||D_s - T_s^{\subset}\beta_s^{\subset}||_2^2, \tag{3}$$

where $\hat{\beta}_s^{\subset}$ is of size $p \times 1$, D_s is size $q \times 1$, where q is the amount of data in the dataset (e.g. the number of time steps) and $T_s^{\subset}$ is size $q \times p$. Note that both $T^{\subset}$ and D require tensorial data to be reorganized as vertical vectors (see Beetham and Capecelatro, 2020, for details). After this process has been done for all s conditions, concatenate each of the $\hat{\beta}_s^{\subset}$ vectors into a matrix of size $p \times s$.

3. Finally, provide each p-th row of $\hat{\beta}^{\subset}$ and matrix of parameters, $\mathbb{P}$, associated with the s conditions as input to the GEP algorithm. The resulting functional model for $\hat{\beta}^{\subset}$ effectively collapses the vector of discrete values for $\hat{\beta}^{\subset}$ to a continuous, closed form with algebraic dependence on system parameters.

This modeling flow is illustrated in Fig. 1, where $s = 3$ and $p = 2$ for demonstration purposes.

3. Case study description

Multiphase flows span large parameter spaces, making modeling challenging. Thus, we use a simple gas–solid flow in which two-way coupling between the phases drives the underlying turbulence as a case study to evaluate the effectiveness of the proposed modeling framework. In this configurations, rigid spherical particles are suspended in an unbounded (triply periodic) domain containing an initially quiescent gas. As particles settle under the influence of gravity, they spontaneously form clusters. Due to momentum exchange between phases, particles entrain the fluid, generating turbulence therein. A frame of reference with the fluid phase is considered, such that the mean streamwise fluid velocity is null. Key non-dimensional numbers that characterize the system include the Reynolds number, the Archimedes number, defined as

$$\mathrm{Ar} = (\rho_p/\rho_f - 1)d_p^3 g/\nu_f^2. \tag{4}$$

Alternatively, a Froude number can be introduced to characterize the balance between gravitational and inertial forces, defined as $\mathrm{Fr} = \tau_p^2 g/d_p$, where $\tau_p = \rho_p d_p^2/(18\rho_f \nu_f)$ is the particle response time.

The mean particle-phase volume fraction is varied from $0.001 \leq \langle \varepsilon_p \rangle \leq 0.05$ and gravity is varied from $0.8 \leq g \leq 8.0$. Here, angled brackets denote an average in all three spatial dimensions and time. Due to the large density ratios under consideration, the mean mass loading, $\varphi = \langle \varepsilon_p \rangle \rho_p / \left(\langle \varepsilon_f \rangle \rho_f \right)$, ranges from $\mathcal{O}(10)$–$\mathcal{O}(10^2)$, and consequently two-way coupling between the phases is important. A large enough domain with a sufficiently large number of particles is needed to observe clustering. To enable simulations on this scale, we use an Eulerian–Lagrangian approach (Capecelatro and Desjardins, 2013). Fluid equations are solved on a staggered grid with second-order spatial accuracy and advanced in time with second-order accuracy using the semi-implicit Crank–Nicolson scheme. Particles are tracked individually in a Lagrangian frame of reference and integrated using a second-order Runge–Kutta method. Particle data is projected to the Eulerian mesh using a Gaussian filter described in Capecelatro and Desjardins (2013).

Derivation of the *single-phase* RANS equations is done by directly averaging the Navier–Stokes equations. Derivation of the *multiphase* RANS equations, however, will retain additional physics if averaging is performed on the volume-filtered Navier-Stokes equations Fox (2014). Volume fraction weighted averages, or phase averaging (PA), analogous to Favre averaging of variable density flows, has previously been derived (Capecelatro et al., 2015). For the relatively simple configuration used here, which is homogeneous in all spatial directions, statistically stationary in time and symmetric in the counter-gravity direction, the transport equations for the fluid-phase Reynolds stresses can be reduced to two unique, non-zero components. In the streamwise direction this equation is given as

$$
\begin{aligned}
\frac{1}{2}\frac{\partial \langle u_f'''^2\rangle_f}{\partial t} = {} & \underbrace{\frac{1}{\rho_f}\left\langle p_f \frac{\partial u_f'''}{\partial x}\right\rangle}_{\text{pressure strain (PS)}} - \underbrace{\frac{1}{\rho_f}\left\langle \sigma_{f,1i}\frac{\partial u_f'''}{\partial x_i}\right\rangle}_{\text{viscous dissipation (VD)}} + \underbrace{\frac{\varphi}{\tau_p^\star}\left(\langle u_f''' u_p''\rangle_p - \langle u_f'''^2\rangle_p\right)}_{\text{drag exchange (DE)}} + \\
& \underbrace{\frac{\varphi}{\tau_p^\star}\langle u_f'''\rangle_p \langle u_p\rangle_p}_{\text{drag production (DP)}} + \underbrace{\frac{\varphi}{\rho_p}\left\langle u_f''' \frac{\partial p_f'}{\partial x}\right\rangle_p}_{\text{pressure exchange (PE)}} - \underbrace{\frac{\varphi}{\rho_p}\left\langle u_f''' \frac{\partial \sigma_{f,1i}'}{\partial x_i}\right\rangle_p}_{\text{viscous exchange (VE)}}, \qquad (5)
\end{aligned}
$$

Where u_p is the particle-phase velocity in an Eulerian frame of reference. Here, $\langle(\cdot)\rangle_p = \langle\varepsilon_p(\cdot)\rangle/\langle\varepsilon_p\rangle$. Fluctuations about PA terms are denoted with a double prime. In a similar fashion, the PA operator in the fluid phase is defined as $\langle(\cdot)\rangle_f = \langle\varepsilon_f(\cdot)\rangle/\langle\varepsilon_f\rangle$. Fluctuations about the PA fluid velocity are given by $\boldsymbol{u}_f''' = \boldsymbol{u}_f(\boldsymbol{x},t) - \langle\boldsymbol{u}_f\rangle_f$. With this, the fluid-phase TKE is given by $k_f = \langle\boldsymbol{u}_f''' \cdot \boldsymbol{u}_f'''\rangle_f/2$.

It is notable that all the terms appearing on the right hand side of (5) are unclosed and require modeling. This work has already been carried out using sparse regression exclusively (Beetham et al., 2021). Here, we select the drag production term to demonstrate the present methodology. This term is chosen due to its importance in this class of flows. In the absence of mean shear, it is the only source of fluid-phase TKE. Additionally, all components of the drag production tensor are identically zero, except for the contribution in the gravity aligned direction. This condition often presents challenges for modeling.

4. Results and discussion

We now demonstrate the modeling methodology presented in Sec. 2 on the multiphase case study summarized in Sec. 3, focusing on the drag production term, $\mathcal{R}^{\mathrm{DP}}$, in particular. Here, we follow the three modeling steps as outlined previously.

In the first step, we conduct modeling of drag production using sparse regression with embedded invariance and the assumption of constant coefficients to inform the bases that comprise the reduced set, $T^{\subset}$. The model consisting of the reduced basis tensors is given as

$$\mathcal{R}^{\mathrm{DP}} = \beta_1 \mathbb{I} + \beta_2 \hat{\mathbb{U}}_r, \qquad (6)$$

where $\hat{\mathbb{U}}_r$ is a tensor formulated using the mean slip velocity between the phase, $\mathbb{I}$ is the identity tensor and the coefficients, β_1 and β_2, have functional dependency upon configuration parameters that are unknown and cannot be informed by physics-based reasoning.

Next, we evaluate the ideal *constant* coefficients for each unique configuration studied, by conducting OLS and allowing the coefficients, β_1 and β_2 to take on unique values for each configuration. In other words, the values of β_1 and β_2 associated with the case for $\langle \varepsilon_p \rangle = 0.001$ and $g = 0.8$ need not be the same as the values for $\langle \varepsilon_p \rangle = 0.05$ and $g = 2.4$.

As described in Sec. 2, the ideal coefficients are arranged into two vectors: one for each of the basis tensors, $\mathbb{I}$ and $\hat{\mathbb{U}}_r$. Each vector of ideal coefficients is used as input, along with the associated parameters and invariants, to the GEP algorithm (Searson, 2009). Here, the GEP algorithm selects models that reduce the R^2 between the ideal coefficient values and the candidate models, which are all functions of the parameters and invariants. This effectively collapses the vector of ideal coefficients to a single, compact algebraic expression.

The resultant model learned from this methodology is given as

$$\begin{aligned}\mathcal{R}^{\mathrm{DP}} = &\left(0.258\varphi + (0.03\varphi)^3 + 1.9\frac{\langle \varepsilon_p \rangle}{\mathcal{S}^{(2)}}\right)\mathbb{I} + \\ &\left(1.9\varphi - 5.8\varphi^{1/2}\right)\hat{\mathbb{U}}_r\end{aligned} \tag{7}$$

where $\mathcal{S}^{(2)}$ is a principal invariant, defined as $\mathrm{tr}\left(\hat{\mathbb{U}}_r\hat{\mathbb{R}}_f\hat{\mathbb{R}}_p^2\right)$, and the basis tensor, $\hat{\mathbb{U}}_r$ is defined by the normalized slip tensor. This slip tensor is given as the outer product of the mean slip velocity, $(\langle \boldsymbol{u}_p \rangle_p - \langle \boldsymbol{u}_f \rangle_f) \otimes (\langle \boldsymbol{u}_p \rangle_p - \langle \boldsymbol{u}_f \rangle_f)$. The two other basis tensors, $\hat{\mathbb{R}}_f$ and $\hat{\mathbb{R}}_p$, are the anisotropic stress tensors associated with the fluid and particle phase, respectively. In terms of solution variables, the mean phase velocities are solved by associated momentum equations and the Reynolds stresses are informed by transport equations in the multiphase RANS equations (see Beetham et al., 2021). This model has an error of 0.012, where the error is defined as

$$\epsilon = \frac{||\mathbb{D} - \mathbb{T}\hat{\beta}||_2^2}{||\mathbb{D}||_2^2}. \tag{8}$$

This is comparable performance to the model learned using sparse regression exclusively ($\epsilon = 0.013$), however, the proposed method does not require a manual selection of trial functions for the coefficients, thus making it far more efficient from a modeling perspective.

As a counter argument to the blended modeling approach, we also allowed the GEP algorithm to learn the full model (i.e., the mean values of drag production, all 24 basis tensors, the principal invariants and configuration parameters from the Euler–Lagrange simulations were provided to the GEP algorithm as input). The learned model is given as

$$\begin{aligned}\mathcal{R}^{\mathrm{DP}} =&24.4\hat{\mathbb{U}}_r + 30.4e^{-\hat{\mathbb{R}}_p} - 1.42\left(\hat{\mathbb{U}}_r\hat{\mathbb{R}}_p + (\hat{\mathbb{U}}_r\hat{\mathbb{R}}_p)^{\mathrm{T}}\right)^{1/2} + \\ &1.41\times10^5\left(\hat{\mathbb{U}}_r^2\hat{\mathbb{R}}_f + (\hat{\mathbb{U}}_r^2\hat{\mathbb{R}}_f)^{\mathrm{T}}\right)^2\langle\varepsilon_p\rangle^2 - 30.4,\end{aligned} \tag{9}$$

with associated error 0.13 (an order of magnitude higher than the blended or sparse regression only approach). This degradation in performance can be attributed to the fact that the model now depends upon $\hat{\mathbb{R}}_p$ and $\hat{\mathbb{R}}_f$, the particle and fluid-phase Reynolds stress tensors, in addition to $\hat{\mathbb{U}}_r$. On a fundamental level, since the drag production is a gravity-based phenomenon (i.e., TKE is generated *solely* due to the presence of gravity in this configuration), we can anticipate that $\hat{\mathbb{U}}_r$ would be the predominant tensor from the basis for describing the physics. Additionally, since the Reynolds stresses contain nonzero diagonal entries, including these terms makes it difficult to drive the cross stream directions of the drag production model to zero. Finally, and perhaps most importantly, GEP does not enforce the relation that resultant model be linear with respect to the basis tensors. The stipulation of linearity in the basis tensors is critical for ensuring form invariance in the resultant model and for ensuring a physics-based description of the data, as described by (1). These results suggest that sparse regression and GEP are both needed in order to select a minimal set of tensors to describe physics and automate the complex dependencies of the coefficients when physical intuition cannot guide this process.

5. Acknowledgements

This material is based upon work supported by the National Science Foundation (NSF CAREER, CBET-1846054). The computing resources and assistance provided by the staff of Advanced Research Computing at the University of Michigan, Ann Arbor is greatly appreciated. Additionally, this work used the Extreme Science and Engineering Discovery Environment

(XSEDE), which is supported by National Science Foundation grant number ACI-1548562 (Towns et al., 2014).

References

S. Beetham and J. Capecelatro. Formulating turbulence closures using sparse regression with embedded form invariance. *Physical Review Fluids*, 5: 084611, 2020.

S. Beetham, R. O. Fox, and J. Capecelatro. Sparse identification of multiphase turbulence closures for coupled fluid-particle flows. *Journal of Fluid Mechanics*, 914, A11, 2021.

M. Bode, M. Gauding, K. Kleinheinz, and H. Pitsch. Deep learning at scale for subgrid modeling in turbulent flows: regression and reconstruction. *arXiv:1910.00928v1*, 2019.

S. L. Brunton, J. L. Proctor, and J. N. Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016.

J. Capecelatro and O. Desjardins. An Euler–Lagrange strategy for simulating particle-laden flows. *Journal of Computational Physics*, 238:1–31, 2013.

J. Capecelatro, O. Desjardins, and R. O. Fox. Numerical study of collisional particle dynamics in cluster-induced turbulence. *Journal of Fluid Mechanics*, 747:R2 1–13, 2014.

J. Capecelatro, O. Desjardins, and R. O. Fox. On fluid-particle dynamics in fully-developed cluster-induced turbulence. *Journal of Fluid Mechanics*, 780:578–635, 2015.

Joachim Dominique, Julien Christophe, Christophe Schram, and Richard D Sandberg. Inferring empirical wall pressure spectral models with Gene Expression Programming. *Journal of Sound and Vibration*, 506:116162, 2021.

K. Duraisamy and P. A. Durbin. Transition modeling using data driven approaches. *Proceedings of the Summer Program*, page 427, 2014.

K. Duraisamy, Z. J. Zhang, and A. P. Singh. New approaches in turbulence and transition modeling using data-driven techniques. *53rd AIAA Aerospace Sciences Meeting*, page 1284, 2015.

Karthik Duraisamy, Gianluca Iaccarino, and Heng Xiao. Turbulence modeling in the age of data. *Annual Review of Fluid Mechanics*, 51:357–377, 2019.

R. O. Fox. On multiphase turbulence models for collisional fluid–particle flows. *Journal of Fluid Mechanics*, 742:368–424, 2014.

J. Ling, A. Kurzawski, and J. Templeton. Reynolds averaged turbulence modeling using deep neural networks with embedded invariance. *Journal of Fluid Mechanics*, 807:155–166, 2016.

W. Liu and J. Fang. Iterative framework of machine-learning based turbulence modeling for Reynolds-averaged Navier-Stokes simulations. *arXiv:1910.01232v1*, 2019.

C. Lu. Artificial neural network for behavior learning from meso-scale simulations, application to multi-scale multimaterial flows. *PhD thesis*, 2010.

M. Ma, J. Lu, and G. Tryggvason. Using statistical learning to close two-fluid multiphase flow equations for bubbly flows in vertical channels. *International Journal of Multiphase Flow*, 85:336–347, 2016.

M. Milano and P. Koumoutsakos. Neural network modeling for near wall turbulent flow. *Journal of Computational Physics*, 182:1–26, 2002.

S. B. Pope. Turbulent flows. *Cambridge University Press*, 2000.

E. Rajabi and M. R. Kavianpour. Intelligent prediction of turbulent flow over backward-facing step using direct numerical simulation data. *Engineering Applications of Computational Fluid Mechanics*, 6(4):490–503, 2012.

Maximilian Reissmann, Josef Hasslberger, Richard D Sandberg, and Markus Klein. Application of Gene Expression Programming to a-posteriori LES modeling of a Taylor Green vortex. *Journal of Computational Physics*, 424:109859, 2021.

Saeed Samadianfard. Gene expression programming analysis of implicit colebrook–white equation in turbulent flow friction factor calculation. *Journal of Petroleum Science and Engineering*, 92:48–55, 2012.

M. Schmelzer, R. P. Dwight, and P. Cinnella. Discovery of algebraic Reynolds-stress models using sparse symbolic regression. *Flow, Turbulence and Combustion*, 104(2):579–603, 2020.

D. Searson. Gptips: Genetic programming & symbolic regression for MATLAB. *http://gptips.sourceforge.net*, 2009.

A. J. M. Spencer and R. S. Rivlin. The theory of matrix polynomials and its application to the mechanics of isotropic continua. *Archive for Rational Mechanics and Analysis*, 2:309–336, 1958.

J. Towns, T. Cockerill, M. Dahan, I. Foster, K. Gaither, A. Grimshaw, V. Hazlewood, S. Lathrop, D. Lifka, G. D. Peterson, R. Roskies, J. R. Scott, and N. Wilkins-Diehr. Xsede: Accelerating scientific discovery. *Computing in Science & Engineering*, 16:62–74, 2014.

B. Tracey, K. Duraisamy, and J. J. Alonso. A machine learning strategy to assist turbulence model development. *AIAA Paper*, 1287, 2015.

Jack Weatheritt and Richard Sandberg. A novel evolutionary algorithm applied to algebraic modifications of the RANS stress–strain relationship. *Journal of Computational Physics*, 325:22–37, 2016.

Yaomin Zhao, Harshal D Akolekar, Jack Weatheritt, Vittorio Michelassi, and Richard D Sandberg. RANS turbulence model development using cfd-driven machine learning. *Journal of Computational Physics*, 411:109413, 2020.