# Probabilistic Machine Learning based Turbulence Model Learning with a Differentiable Solver

**Atul Agrawal, Phaedon-Stelios Koutsourelakis**
Professorship of Data-Driven Materials Modelling
Technical University of Munich
Munich, Germany
{atul.agrawal,p.s.koutsourelakis}@tum.de

## Abstract

We present a novel, data-driven closure model for Reynolds-averaged Navier-Stokes (RANS) equations which consists of two-parts. A parametric one, which a tensor basis neural-network and a non-parametric one which makes use of latent, random variables in order to capture aleatoric model uncertainty. Our fully Bayesian formulation, incorporating sparsity-inducing priors, identifies areas of the problem domain where the parametric closure falls short, requiring stochastic corrections to the Reynolds stress tensor. Training employs sparse, indirect data such as mean velocities and pressures, in contrast to the majority of alternatives which require direct, Reynolds stress data. For inference and learning, we employ Stochastic Variational Inference, facilitated by an adjoint-based differentiable solver. This end-to-end differentiable framework can ultimately yield accurate, probabilistic predictions for flow quantities, even in regions with model errors, as exemplified by the backward-facing step benchmark problem.

## 1 Introduction

The simulation of turbulence in fluid dynamics necessitates very fine spatio-temporal resolution of the Navier-Stokes equations, known as Direct Numerical Simulation (DNS), which is prohibitively expensive for real-world applications. Reynolds-averaged Navier-Stokes (RANS) models offer a more efficient way to predict mean flow properties and are the industry standard [1]. The accuracy of RANS predictions hinges upon the closure model adopted for the Reynolds stress (RS) tensor which formally depends on the unresolved velocity fluctuations [2, 3]. Multiple closure models have been proposed [4, 2] and in recent years, data-driven strategies have risen in prominence [5, 6, 7, 8]. We propose an intrusive, probabilistic framework which makes use of the RANS solver in the learning process and can make use of sparse, *indirect* mean velocity/pressure observations. To enable the learning, we developed an adjoint-based, differentiable RANS solver. Availability of the physics-based simulator gradients is a common problem across fields whenever it is involved in learning/inference tasks [9]. We utilize a parametric closure model of the RS tensor [10], to which a stochastic discrepancy tensor field is added in order to account for the insufficiency of the parametric part. As we demonstrate on a separated flow benchmark case, the framework proposed outperforms the baselines in terms of predictive accuracy while having reduced data requirements and simultaneously proving probabilistic predictive bounds.

This work shares similarities with data-driven models as in [10, 7, 11, 12, 13], which, nevertheless, do not involve the RANS solver in the training process (hence full field RS data is used for training). [14, 15, 16] reported inconsistency issues which might arise when the solver is not involved in the training process, leading to less satisfactory velocity/pressure predictions. Additionally, full-field RS

Machine Learning and the Physical Sciences Workshop, NeurIPS 2023.

data is required as opposed to our sparse data requirements. Recently, [17, 18, 19] incorporated the RANS solver in the training, however they do not account for aleatoric, model uncertainty.

## 2 Methods

**Problem Statement** The RANS equations arise by time-averaging from the NS equations and involve time-averaged pressure and velocity fields. They are however *unclosed* due to the so-called Reynolds-Stress (RS) tensor term $\boldsymbol{\tau}_{RS} = -\left\langle \tilde{u}\tilde{u}^T \right\rangle$[1], which depends on the cross-correlation of the unresolved, fluctuating part of the velocity $\tilde{u}$. [2] The most commonly employed strategy, i.e. the Linear Eddy Viscosity model (LEVM), assumes that $\boldsymbol{\tau}_{RS}$ can be expressed as a linear function of mean velocity and some other turbulent flow quantities, such as the turbulent kinetic energy $k$ and the turbulent energy dissipation $\epsilon$ (e.g. the $k-\epsilon$ model [20]), or the specific dissipation $\omega$ (e.g. the $k-\omega$ [21]). Although LEVMs provide accurate results for a range of flows, they can give rise to predictive inaccuracies which are particularly prominent when trying to capture flows with significant curvatures, recirculation zones, separation, reattachment, anisotropy etc [2, 22].

### 2.1 Probabilistic, data-driven differential framework

Upon discretisation using e.g. a finite element scheme, one can express the RANS equations in residual form as:

$$\mathcal{G}(\boldsymbol{z}) = \boldsymbol{B}\boldsymbol{\tau}; \qquad \text{or,} \quad \mathcal{R}(\boldsymbol{z};\boldsymbol{\tau}) := \mathcal{G}(\boldsymbol{z}) - \boldsymbol{B}\boldsymbol{\tau} = 0 \tag{1}$$

where $\boldsymbol{z} = [\boldsymbol{u}, \boldsymbol{p}]^T$ summarily denotes the discretized mean velocity $\boldsymbol{u}$ and pressure $p$ fields and $\boldsymbol{\tau}$ the discretized RS field. E.g. for a two-dimensional flow domain $\boldsymbol{z} \in \mathbb{R}^{N\times 3}$, $\boldsymbol{\tau} \in \mathbb{R}^{N\times 3}$ where $N$ is the number of grid points. We denote with $\mathcal{G}$ the discretized, non-linear operator accounting for the advective and diffusive terms as well as the conservation of mass, and with $\boldsymbol{B}$ the matrix (i.e. linear operator) arising from the divergence term on the RS tensor. Traditional, data-driven strategies postulate a closure e.g. $\boldsymbol{\tau}_{\boldsymbol{\theta}}(\boldsymbol{z})$ dependent on some tunable parameters $\boldsymbol{\theta}$, which they determine either by assuming that reference Reynolds-stress data is available from DNS simulations [13, 10, 23, 11]). Apart from the heavier data requirements, it does not guarantee that the trained model would yield accurate predictions of $\boldsymbol{z}$ [16] as even small errors in $\boldsymbol{\tau}$ might get amplified when solving Eq. (1). In contrast, we propose to learn the $\boldsymbol{\tau}_{\boldsymbol{\theta}}(\boldsymbol{z})$ using *indirect* and noisy mean field observations which necessitates parametric sensitivities, i.e. a differentiable solver, for training. Another critical aspect pertains to uncertainty quantification. Parametric uncertainty which is of epistemic origin has been extensively studied (e.g.,[24, 25, 26]). In the present work, we address the issue of aleatoric, model uncertainty in the closure equations. In particular, we augment the parametric closure model $\boldsymbol{\tau}_{\boldsymbol{\theta}}(\boldsymbol{u})$ with a set of latent (i.e. unobserved) random variables $\boldsymbol{\epsilon}_\tau$ which are embedded in the model equations and which quantify model discrepancies at each grid point. In reference to the discretized RS vector $\boldsymbol{\tau}$ in Eq. (1), we propose to use $\boldsymbol{\tau} = \boldsymbol{\tau}_{\boldsymbol{\theta}}(\boldsymbol{u}) + \boldsymbol{\epsilon}_\tau$.

### 2.2 Models, data, likelihood and posterior

For the parametric part of the closure, i.e. $\boldsymbol{\tau}_{\boldsymbol{\theta}}(\boldsymbol{u})$, we make use of the invariant neural network architecture proposed by [10] which relates the anisotropic part of the RS tensor with the symmetric and antisymmetric components of the velocity gradient tensor. By using tensor invariants, the neural network is able to achieve both Galilean invariance as well as rotational invariance. For the NN parameters $\boldsymbol{\theta}$, as a prior we employ a Student's $\mathcal{T}$-distribution centered at zero. For the latent variables $\boldsymbol{\epsilon}_\tau$, we propose a dimension-reduced representation that facilitates inference tasks given the high values that $N$ takes in most simulations. In particular, we represent $\boldsymbol{\epsilon}_\tau = \boldsymbol{W}\boldsymbol{E}_{\boldsymbol{\tau}}$, where the entries of $\boldsymbol{W}$ are 1 if a corresponding grid point (row of $\boldsymbol{W}$) belongs in a certain subdomain (column of $\boldsymbol{W}$) and 0 otherwise. The vector $\boldsymbol{E}_\tau = \{\boldsymbol{E}_{\tau,J}\}_{J=1}^{N_d}$ contains therefore the RS discrepancy terms for each of the $N_d$ subdomains. For a two-dimensional flow, $dim(\boldsymbol{E}_{\tau,J}) = 3$. As there would be an infinity of combinations of $\boldsymbol{\theta}$ and $\boldsymbol{\epsilon}_\tau/\boldsymbol{E}_\tau$ that could fit the data equally well, we make use of a sparsity-inducing Bayesian prior [27] based on the Automatic Relevance Determination (ARD) [28] , in combination

[1] $\langle\cdot\rangle$ denotes the time average of the arguments.

with the Gamma hyperprior. In particular:

$$p(\boldsymbol{E}_\tau|\boldsymbol{\Lambda}) = \prod_{J=1}^{N_d} \mathcal{N}\left(\boldsymbol{E}_{\tau,J}\,|\mathbf{0},\ diag(\boldsymbol{\Lambda}_J)^{-1}\right);\qquad p(\boldsymbol{\Lambda}) = \prod_{J=1}^{N_d}\prod_{\ell=1}^{L} Gamma(\Lambda_{J,\ell}\,|\alpha_0,\beta_0) \tag{2}$$

where $\Lambda_{J,\ell}$ denotes the $\ell^{th}$ entry (e.g. $L=3$ for two-dimensional flows) of the vector of precision hyperparameters in subdomain $J$. We note that when $\Lambda_{J,\ell}\to\infty$, then the corresponding model discrepancy term $E_{\tau,J,\ell}\to 0$. The resulting prior for $\boldsymbol{E}_\tau$ arising by marginalizing the hyperparameters $\boldsymbol{\Lambda}$ is a light-tailed, Student's t-distribution [29] that promotes solutions in the vicinity of 0 unless strong evidence in the data suggests otherwise. For training, we consider $M\geq 1$ flow settings and denote the observations collected as $\mathcal{D}=\{\hat{\boldsymbol{z}}^{(m)}\}_{m=1}^{M}$. These consist of time-averaged velocity/pressure values where $dim(\hat{\boldsymbol{z}}^{(m)})=N_{obs}$. We utilize a Gaussian likelihood given by:

$$p(\mathcal{D}\mid\boldsymbol{\theta},\boldsymbol{\epsilon}_\tau^{(1:M)}) = \prod_{m=1}^{M} p(\hat{\boldsymbol{z}}^{(m)}\mid\boldsymbol{\theta},\boldsymbol{\epsilon}_\tau^{(m)}) = \prod_{m=1}^{M}\mathcal{N}(\hat{\boldsymbol{z}}^{(m)}\mid\boldsymbol{z}(\boldsymbol{\theta},\boldsymbol{\epsilon}_\tau^{(m)}),\boldsymbol{\Sigma}) \tag{3}$$

where $\boldsymbol{z}(\boldsymbol{\theta},\boldsymbol{\epsilon}_\tau^{(m)})$ denotes the solution vector of the discretized RANS equations and the covariance is expressed as $\boldsymbol{\Sigma}=diag(\sigma_1^2,\cdots,\sigma_{3N_{obs}}^2)$. By combining the priors with the likelihood model, we arrive at the following posterior:

$$p(\boldsymbol{\theta},\boldsymbol{E}_\tau^{(1:M)},\boldsymbol{\Lambda}|\ \mathcal{D}) \propto \left(\prod_{m=1}^{M} p(\hat{\boldsymbol{z}}^{(m)}\mid\boldsymbol{\theta},\boldsymbol{E}_\tau^{(m)})\ p(\boldsymbol{E}_\tau^{(m)}|\boldsymbol{\Lambda})\right)\ p(\boldsymbol{\theta})\ p(\boldsymbol{\Lambda}) \tag{4}$$

### 2.3 Learning and predictions

To identify the latent variables and unknown variables in the probabilistic model proposed, we advocate the use of Stochastic Variational Inference (SVI) [30] which results in a closed-form approximation of the posterior $p(\boldsymbol{\theta},\boldsymbol{E}_\tau^{(1:M)},\boldsymbol{\Lambda}|\mathcal{D})$. Given a family of probability densities $q_{\boldsymbol{\xi}}\left(\boldsymbol{\theta},\boldsymbol{\Lambda},\boldsymbol{E}_\tau^{(1:M)}\right)$ parametrized by $\boldsymbol{\xi}$, we find the optimal, i.e. the one that is closest to the exact posterior in terms of their Kullback-Leibler divergence, by maximizing the Evidence Lower Bound (ELBO) $\mathcal{F}(\boldsymbol{\xi})$ [31]. We employ a mean-field assumption [32] for the approximate posterior $q_{\boldsymbol{\xi}}$. The updates of the parameters $\boldsymbol{\xi}$ are carried out using derivatives of the ELBO. These entail expectations with respect to $q_{\boldsymbol{\xi}}$ which are estimated (with noise) by Monte Carlo in conjunction with the ADAM stochastic optimization scheme [33] and the reparametrization trick [34]. If we summarily denote with $\boldsymbol{\eta}=\{\boldsymbol{\theta},\boldsymbol{\Lambda},\boldsymbol{E}_\tau^{(1:M)}\}$ and given that the approximate posterior $q_{\boldsymbol{\xi}}(\boldsymbol{\eta})$ can be represented by deterministic transform $\boldsymbol{\eta}=g_\xi(\boldsymbol{\phi})$, where $\boldsymbol{\phi}$ follows a known density $q(\boldsymbol{\phi})$, the expectations involved in the ELBO and, more importantly, in its gradient can be rewritten as:

$$\nabla_{\boldsymbol{\xi}}\mathcal{F}(\boldsymbol{\xi}) = \mathbb{E}_{q(\boldsymbol{\phi})}\left[\nabla_{\boldsymbol{\xi}} g_\xi(\boldsymbol{\phi})\nabla_{\boldsymbol{\eta}}\left(\log p(\mathcal{D},\boldsymbol{\eta}) - \log q_\xi(\boldsymbol{\eta})\right)\right] \tag{5}$$

One observes that derivatives of the log-likelihood with respect to $\boldsymbol{\eta}$ are needed. This in turn would imply derivatives of the RANS-model outputs with respect to $\{\boldsymbol{\theta},\boldsymbol{\Lambda},\boldsymbol{E}_\tau^{(1:M)}\}$ which appear indirectly through $\boldsymbol{\tau}$. Such derivatives are rendered possible by using an adjoint formulation of the discretized RANS model that yields in effect a *differentiable* solver (details in Appendix B).

*Probabilistic*, predictive estimates of any quantity of interest related to the RANS-simulated flow can be produced using the trained model. In particular, one can obtain a *predictive, posterior density* $p(\mathbf{z}|\mathcal{D})$ on the whole solution vector $\mathbf{z}$ of the RANS equations as follows:

$$p(\mathbf{z}|\mathcal{D}) = \int p(\mathbf{z}|\boldsymbol{E}_\tau,\boldsymbol{\theta})\ p(\boldsymbol{E}_\tau|\boldsymbol{\Lambda})p(\boldsymbol{\theta},\boldsymbol{\Lambda}|\mathcal{D})\ d\boldsymbol{E}_\tau\ d\boldsymbol{\theta}\ d\boldsymbol{\Lambda} \tag{6}$$

The third of the densities in the intergrand is the posterior which is substituted by its variational approximation i.e. $q_{\boldsymbol{\xi}}$ with its optimal parameter values $\boldsymbol{\xi}$. The second represents the prior model prescribed in Eq. (2). Finally the first is simply a Dirac-delta that corresponds to the solution of the RANS equations obtained when using a closure model of the form $\boldsymbol{\tau}=\tau_{\boldsymbol{\theta}}(\boldsymbol{u})+\boldsymbol{W}\boldsymbol{E}_\tau$. Owing to the intractability of the integral, a Monte Carlo scheme is employed.

## 3 Numerical Illustrations

We select the backward facing step configuration (Fig. (1)) in order to assess the proposed modeling framework. This is a classic benchmark problem that has been widely used for studying the performance of turbulence models as it poses significant challenges due to the complex flow features such as flow separation, reattachment and recirculation [35]. Following boundary conditions/inputs are prescribed: constant inlet bulk velocity $u_b = 1$ in the $x$-direction on the left boundary, no-slip condition on top/bottom boundary and zero traction at the outflow boundary. The implementation of the differentiable FE solver is detailed in Appendix. A. The $k - \epsilon$ model is choosen as the *baseline RANS model* as its the most widely used RANS model in industrial applications. For training data generation, LES is performed for various $Re$ number ($Re = [300, 700, 900, 1100]$). In the subsequent results, mean velocity/pressure observations at approximately $8\%$ of the grid points of the RANS-FE mesh were used. For the learning, the popular machine learning library `PyTorch` [36] was used. The ELBO maximization was performed using the ADAM scheme [33] with learning rate being $10^{-6}$. The neural network architecture employed for the parametric RS model was identical to the one suggested by [10].

We assessed the trained model for the test-case with $Re = 500$ which was not contained in the training data. We observe from Fig. (2) that even though no RS training data was provided, the $\boldsymbol{\tau_\theta}$ model is able to capture the features, hence the high precision in some areas (e.g., downstream). In the regions the model is under-performing, the precision attains small values, thus allowing the $\boldsymbol{\epsilon}_\tau$ to provide a correction to the RS tensor. The impact of this correction is evident in Fig. (1), wherein the mean field predictions agree well with the reference LES and the predictive bounds envelop the LES values in most area, as opposed to the baseline $k - \epsilon$. Furthermore, Table 1 compares our model against others for the reattachment length, a key parameter in the study of separated flows [37], which gives the distance from the step where the flow separates to the point at which it reattaches. We observe while previous works deviated significantly from the reference LES value, our probabilistic prediction is able to envelop it.

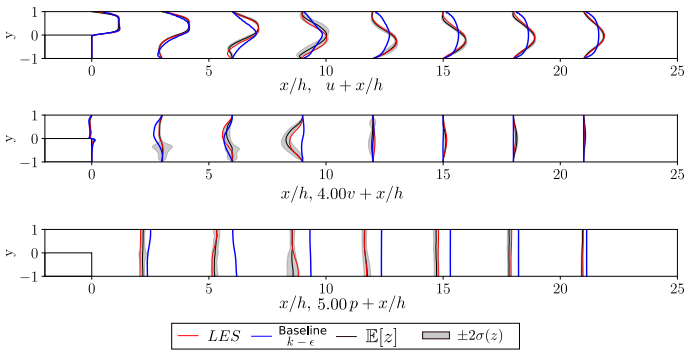


Figure 1: Section plots at different locations $x/h$ comparing the LES and *Baseline* RANS mean fields with the posterior predictive mean ( solid black line) and $\pm 2\times$ standard deviation (shaded area). **top** - velocity in x-direction ($\boldsymbol{u}$), **middle** - velocity in the y-direction ($\boldsymbol{v}$), **bottom** - pressure ($\boldsymbol{p}$).

| Model | $x_{reattach}$[x/h] |
|---|---|
| LES (reference) | 9.10 |
| Biswas et al. [37] | 8.9 |
| Baseline RANS | 5.61 |
| Geneva et al.[13] | 5.52 |
| **proposed model** | **10.06 ± 1.21** |

Table 1: Predictions of reattachment length ($x/h$) of the primary recirculation region behind the backward-facing step (expansion ratio $H/h = 2$). $\pm$ : $3\times$ s.d

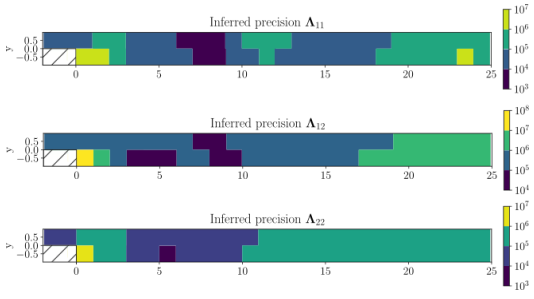


Figure 2: The inferred hyper-parameter $\boldsymbol{\Lambda}$ of the (reduced) discrepancy tensor $\boldsymbol{E_\tau}$ corresponding to the three components $\boldsymbol{E_{\tau,11}}$, $\boldsymbol{E_{\tau,12}}$ and $\boldsymbol{E_{\tau,22}}$

## 4 Conclusions

The model proposed, which incorporates the RANS solver in the training process, uses sparse velocity/pressure observations instead of the full field Reynolds stress observations and employs a stochastic, model discrepancy term, leads to better predictions while quantifying the predictive uncertainties. This is demonstrated in the backward-facing step benchmark problem while comparing our approach to baseline RANS, high-fidelity LES and other data-driven model(s).

## 5 Broader Impact Statement

Turbulence is a crucial physical characteristic of a broad range of fluid flows. Grasping this occurrence is vital for intricate designs, environmental simulations, and a myriad of engineering uses. Over the last few decades, computational capabilities have surged significantly, allowing for high-resolution simulations such as direct numerical simulations (DNS) for various turbulent flows. Nevertheless, approximations like Reynolds-averaged Navier–Stokes (RANS) persist as indispensable for industrial purposes, where precision heavily relies on turbulence closure models.

The present research demonstrated that including the RANS solver in the training process can improve the prediction quality even with limited training data. This underscored the impact augmentation of training data with information that can be extracted from the physical simulator can have. This deep integration of differentiable and probabilistic programming frameworks involving physical simulators is the need of the hour for scientific machine learning. This can revolutionize many fields in physics and engineering like fluid mechanics, molecular dynamics, particle physics, cosmology, material science, drug discovery, to name a few. Furthermore, we demonstrated a method to account for the involved uncertainties, that are unavoidable when any sort of modeling or learning with finite data is involved.

We do not see any direct ethical concerns associated with this research. The impact on society is primarily through the over-arching context of research using machine learning to improve our general understanding of fluid turbulence.

## References


[1] J. Slotnick et al. "CFD Vision 2030 Study: A Path to Revolutionary Computational Aerosciences". In: *NNASA/CR-2014-218178* (2014).

[2] Stephen B Pope. *Turbulent flows*. Cambridge university press, 2000.

[3] Philippe R Spalart. "Philosophies and fallacies in turbulence modeling". In: *Progress in Aerospace Sciences* 74 (2015), pp. 1–15.

[4] Giancarlo Alfonsi. "Reynolds-Averaged Navier-Stokes Equations for Turbulence Modeling". In: *Applied Mechanics Reviews - APPL MECH REV* 62 (July 2009). DOI: 10.1115/1.3124648.

[5] Karthik Duraisamy, Gianluca Iaccarino, and Heng Xiao. "Turbulence modeling in the age of data". In: *Annual Review of Fluid Mechanics* 51 (2019), pp. 357–377.

[6] Steven L Brunton, Bernd R Noack, and Petros Koumoutsakos. "Machine learning for fluid mechanics". In: *Annual Review of Fluid Mechanics* 52 (2020), pp. 477–508.

[7] Julia Ling, Reese Jones, and Jeremy Templeton. "Machine learning strategies for systems with invariance properties". In: *Journal of Computational Physics* 318 (2016). Publisher: Elsevier Inc., pp. 22–35. ISSN: 10902716. DOI: 10.1016/j.jcp.2016.05.003. URL: http://dx.doi.org/10.1016/j.jcp.2016.05.003.

[8] Atul Agrawal and Phaedon-Stelios Koutsourelakis. "A probabilistic, data-driven closure model for RANS simulations with aleatoric, model uncertainty". In: *arXiv preprint arXiv:2307.02432* (2023).

[9] Kyle Cranmer, Johann Brehmer, and Gilles Louppe. "The frontier of simulation-based inference". In: *Proceedings of the National Academy of Sciences* 117.48 (2020), pp. 30055–30062.

[10] Julia Ling, Andrew Kurzawski, and Jeremy Templeton. "Reynolds averaged turbulence modelling using deep neural networks with embedded invariance". In: *Journal of Fluid Mechanics* 807 (2016), pp. 155–166.

[11] Mikael LA Kaandorp and Richard P Dwight. "Data-driven modelling of the Reynolds stress tensor using random forests with invariance". In: *Computers & Fluids* 202 (2020), p. 104497.

[12] Jian-Xun Wang et al. "A comprehensive physics-informed machine learning framework for predictive turbulence modeling". In: *arXiv preprint arXiv:1701.07102* (2017).

[13] Nicholas Geneva and Nicholas Zabaras. "Quantifying model form uncertainty in Reynolds-averaged turbulence models with Bayesian deep neural networks". In: *Journal of Computational Physics* 383 (2019), pp. 125–147.

[14] Salar Taghizadeh, Freddie D Witherden, and Sharath S Girimaji. "Turbulence closure modeling with data-driven techniques: physical compatibility and consistency considerations". In: *New Journal of Physics* 22.9 (2020), p. 093023.

[15] Karthik Duraisamy. "Perspectives on Machine Learning-augmented Reynolds-averaged and Large Eddy Simulation Models of Turbulence". In: *Physical Review Fluids* 6.5 (May 2021). arXiv:2009.10675 [physics], p. 050504. ISSN: 2469-990X. DOI: 10.1103/PhysRevFluids.6.050504. URL: http://arxiv.org/abs/2009.10675 (visited on 03/10/2023).

[16] Roney L Thompson et al. "A methodology to evaluate statistical errors in DNS data of plane channel flows". In: *Computers & Fluids* 130 (2016), pp. 1–7.

[17] Oliver Brenner, Pasha Piroozmand, and Patrick Jenny. "Efficient Assimilation of Sparse Data into RANS-Based Turbulent Flow Simulations Using a Discrete Adjoint Method". en. In: *Journal of Computational Physics* (Sept. 2022), p. 111667. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2022.111667. URL: https://www.sciencedirect.com/science/article/pii/S0021999122007306 (visited on 09/29/2022).

[18] Carlos A Michelén Ströfer and Heng Xiao. "End-to-end differentiable learning of turbulence models from indirect observations". In: *arXiv preprint arXiv:2104.04821* (2021).

[19] Xin-Lei Zhang et al. "Ensemble Kalman method for learning turbulence models from indirect observation data". en. In: *Journal of Fluid Mechanics* 949 (Oct. 2022). Publisher: Cambridge University Press, A26. ISSN: 0022-1120, 1469-7645. DOI: 10.1017/jfm.2022.744. (Visited on 10/04/2022).

[20] B. E. Launder and B. I. Sharma. "Application of the energy-dissipation model of turbulence to the calculation of flow near a spinning disc". In: *Letters in Heat and Mass Transfer* 1.2 (Nov. 1974), pp. 131–137. ISSN: 00944548. DOI: 10.1016/0094-4548(74)90150-7.

[21] David C. Wilcox. "Formulation of the k-$\omega$ turbulence model revisited". In: *AIAA Journal*. Vol. 46. 11. Nov. 2008, pp. 2823–2838. DOI: 10.2514/1.36541. URL: https://arc.aiaa.org/doi/abs/10.2514/1.36541.

[22] David C Wilcox et al. *Turbulence modeling for CFD*. Vol. 2. DCW industries La Canada, CA, 1998.

[23] Jian-Xun Wang, Jin-Long Wu, and Heng Xiao. "Physics-informed machine learning approach for reconstructing Reynolds stress modeling discrepancies based on DNS data". In: *Physical Review Fluids* 2.3 (2017), p. 034603.

[24] Todd Oliver and Robert Moser. "Uncertainty quantification for RANS turbulence model predictions". In: *APS division of fluid dynamics meeting abstracts*. Vol. 62. 2009, pp. LC–004.

[25] Todd A Oliver and Robert D Moser. "Bayesian uncertainty quantification applied to RANS turbulence models". In: *Journal of Physics: Conference Series*. Vol. 318. SECTION 4. Institute of Physics Publishing, Dec. 2011, p. 042032. DOI: 10.1088/1742-6596/318/4/042032. URL: https://iopscience.iop.org/article/10.1088/1742-6596/318/4/042032.

[26] Wouter Nico Edeling et al. "Bayesian estimates of parameter variability in the k–$\varepsilon$ turbulence model". In: *Journal of Computational Physics* 258 (2014), pp. 73–94.

[27] L Felsberger and PS Koutsourelakis. "Physics-constrained, data-driven discovery of coarse-grained dynamics". In: *Communications in Computational Physics* 25.5 (2019), pp. 1259–1301. DOI: 10.4208/cicp.OA-2018-0174.

[28] Radford M Neal. *Bayesian learning for neural networks*. Vol. 118. Springer Science & Business Media, 2012.

[29] M. E. Tipping. "The Relevance Vector Machine". In: *Advances in Neural Information Processing Systems 12*. Ed. by S. A. Solla, T. K. Leen, and K.-R. Müller. MIT Press, 2000, pp. 652–658.

[30] Matthew D Hoffman et al. "Stochastic variational inference". In: *Journal of Machine Learning Research* (2013).

[31] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*. Vol. 4. 4. Springer, 2006.

[32] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. "Variational inference: A review for statisticians". In: *Journal of the American statistical Association* 112.518 (2017), pp. 859–877.

[33] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[34] Diederik P Kingma and Max Welling. "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114* (2013).

[35] Pankaj M Nadge and RN Govardhan. "High Reynolds number flow over a backward-facing step: structure of the mean separation bubble". In: *Experiments in fluids* 55.1 (2014), pp. 1–22.

[36] Adam Paszke et al. "Pytorch: An imperative style, high-performance deep learning library". In: *Advances in neural information processing systems* 32 (2019).

[37] Gautam Biswas, Michael Breuer, and Franz Durst. "Backward-facing step flows for various expansion ratios at low and moderate Reynolds numbers". In: *J. Fluids Eng.* 126.3 (2004), pp. 362–374.

[38] Martin Alnæs et al. "The FEniCS project version 1.5". In: *Archive of Numerical Software* 3.100 (2015).

[39] Sebastian K Mitusch, Simon W Funke, and Jørgen S Dokken. "dolfin-adjoint 2018.1: automated adjoints for FEniCS and Firedrake". In: *Journal of Open Source Software* 4.38 (2019), p. 1292.

[40] Peter Deuflhard. *Newton methods for nonlinear problems: affine invariance and adaptive algorithms*. Vol. 35. Springer Science & Business Media, 2005.

[41] Jean Donea and Antonio Huerta. *Finite element methods for flow problems*. John Wiley & Sons, 2003.

[42] Leopoldo Penna Franca and Eduardo Gomes Dutra Do Carmo. "The Galerkin gradient least-squares method". In: *Computer Methods in Applied Mechanics and Engineering* 74.1 (1989), pp. 41–54.

# Appendix

## A Differentiable RANS solver

In the present study, the RANS equations (Eq. (1)) are numerically solved using the finite element discretization, implemented in the open source package FEniCS [38], due to its innate adjoint solver [39]. The discrete equations are obtained by representing the solution and test functions in appropriate finite dimensional function spaces. In particular, we employed the standard Taylor-Hood pair of basis functions with polynomial degree one for the pressure interpolants and two for the velocities. This choice is made to avoid stability issues potentially arising from the interaction between the momentum and continuity equations.

The turbulence scaling terms, $k$ and $\epsilon$, are obtained by solving the respective standard transport equations [2, 4]. Symmetry is enforced in the RS tensor, i.e. $\tau_{xy}$ and $\tau_{yx}$ are identical without any redundancy in the representation. The discretized system is solved with damped Newton's method. For robustness and global convergence, pseudo-time stepping is used with the backward Euler discretization [40]. As the Reynolds number is increased, the convection term dominates, leading to stability [41].This elicits a need to add stabilization terms to the weak form, such as the least-square stabilization, according to which the weighted square of the strong form is added to the weak form residual. However, these extra terms have to be chosen carefully in order not to compromise the correctness of the approximate solution. Classically, researchers added artificial diffusion terms or a numerical diffusion by using upwind scheme for the convection term instead of central diffusion. The extra infused term corrupted the solution quality. To avoid this, in practice, it is common to use schemes like Streamline-Upwind Petrov-Gelarkin method (SUPG) and Galerkin Least Squares (GLS). In the present study, we have utilized a self-adjoint numerical stabilisation scheme which is an extension of Gelarkin Least Squares (GLS) Stabilisation called Galerkin gradient least square method [42]. This amounts to adding a stabilization term to the residual weak form. For additional details, interested readers are referred to [42, 41].

## B Adjoint Formulation and Estimation of the Gradient of the ELBO

As discussed in Section 2.3, the SVI framework advocated, in combination with the reparametrization trick, to arrive at the following ELBO $\mathcal{F}(\boldsymbol{\xi})$:

$$\mathcal{F}(\boldsymbol{\xi}) = \mathbb{E}_{q_\xi\left(\boldsymbol{\theta},\boldsymbol{\Lambda},\boldsymbol{E}_\tau^{(1:M)}\right)}\left[\log\left(\frac{p(\mathcal{D}\mid\boldsymbol{\theta},\boldsymbol{E}_\tau^{(1:M)})\;p(\boldsymbol{E}_\tau^{(1:M)}|\boldsymbol{\Lambda})\;p(\boldsymbol{\theta})\;p(\boldsymbol{\Lambda})}{q_\xi\left(\boldsymbol{\theta},\boldsymbol{\Lambda},\boldsymbol{E}_\tau^{(1:M)}\right)}\right)\right] \tag{7}$$

The derivatives of the ELBO with respect to the variables which we summarily denoted by $\boldsymbol{\eta} = \{\boldsymbol{\theta}, \boldsymbol{\Lambda}, \boldsymbol{E}_\tau^{(1:M)}\}$, i.e. (as in Eq. (5)):

$$\nabla_{\boldsymbol{\xi}}\mathcal{F}(\boldsymbol{\xi}) = \mathbb{E}_{q(\boldsymbol{\phi})}\left[\nabla_{\boldsymbol{\xi}} g_\xi(\boldsymbol{\phi})\nabla_{\boldsymbol{\eta}}\left(\log p(\mathcal{D},\boldsymbol{\eta}) - \log q_\xi(\boldsymbol{\eta})\right)\right] \tag{8}$$

where from Eq. (4):

$$\begin{aligned}\log p(\mathcal{D},\boldsymbol{\eta}) &= \log\left(p(\mathcal{D}\mid\boldsymbol{\theta},\boldsymbol{E}_\tau^{(1:M)})\;p(\boldsymbol{E}_\tau^{(1:M)}|\boldsymbol{\Lambda})\;p(\boldsymbol{\theta})\;p(\boldsymbol{\Lambda})\right)\\ &= \left(\sum_{m=1}^{M}\log p(\hat{\boldsymbol{z}}^{(m)}\mid\boldsymbol{\theta},\boldsymbol{E}_\tau^{(m)}) + \log p(\boldsymbol{E}_\tau^{(m)}|\boldsymbol{\Lambda})\right) \;+\log p(\boldsymbol{\theta}) + \log\; p(\boldsymbol{\Lambda})\end{aligned} \tag{9}$$

The form of the (log-)priors $p(\boldsymbol{E}_\tau^{(m)}|\boldsymbol{\Lambda})$ , $p(\boldsymbol{\theta})$, $p(\boldsymbol{\Lambda})$ (Eq. (2)) as well as of the approximate posterior $q_\xi(\boldsymbol{\eta})$ suggest that most of these derivatives can be analytically computed with the exception of the ones involving the log-likelihoods, i.e.:

$$\ell^{(m)}(\boldsymbol{\theta},\boldsymbol{E}_\tau^{(m)}) = \log p(\hat{\boldsymbol{z}}^{(m)}\mid\boldsymbol{\theta},\boldsymbol{E}_\tau^{(m)}). \tag{10}$$

This is because each of these terms depends implicitly on $\boldsymbol{\theta}$, $\boldsymbol{E}_\tau^{(m)}$ through the output of the RANS solver $\boldsymbol{z}(\boldsymbol{\theta}, \boldsymbol{\epsilon}_\tau^{(m)} = \boldsymbol{W}\boldsymbol{E}_\tau^{(m)})$ with the closure model for the discretized RS tensor field suggested by $\boldsymbol{\tau} = \boldsymbol{\tau}_{\boldsymbol{\theta}}(\boldsymbol{u}) + \boldsymbol{W}\boldsymbol{E}_\tau^{(m)}$. In view of the governing equations (Eq. (1)), we explain below how adjoint equations can be formulated that enable efficient computation of the aforementioned derivatives of the log-likelihoods.

In particular, and if we drop the superscript $m$ for each term in the log-likelihood in order to simplify the notation, we formulate a Lagrangian with the help of a vector $\lambda$ of Lagrangian multipliers, i.e.:

$$\mathcal{L} = \ell + \boldsymbol{\lambda}^T(\mathcal{G}(\boldsymbol{z}) - \boldsymbol{B}\boldsymbol{\tau}) \tag{11}$$

where $\mathcal{G}$, $\boldsymbol{B}$, $\boldsymbol{\tau}$ and $\boldsymbol{z}$ are as defined in Section (2.1).Differentiating with respect to $\boldsymbol{\tau}$ yields:

$$\begin{aligned}\frac{d\mathcal{L}}{d\boldsymbol{\tau}} &= \frac{\partial \ell}{\partial \boldsymbol{z}}\frac{d\boldsymbol{z}}{d\boldsymbol{\tau}} + \frac{d\boldsymbol{\lambda}^T}{d\boldsymbol{\tau}}(\mathcal{G}(\boldsymbol{z}) - \boldsymbol{B}\boldsymbol{\tau}) + \boldsymbol{\lambda}^T\left(\frac{\partial \mathcal{G}}{\partial \boldsymbol{z}}\frac{d\boldsymbol{z}}{d\boldsymbol{\tau}} - \boldsymbol{B}\right) \\ &= \left(\frac{\partial \ell}{\partial \boldsymbol{z}} + \boldsymbol{\lambda}^T\frac{\partial \mathcal{G}}{\partial \boldsymbol{z}}\right)\frac{d\boldsymbol{z}}{d\boldsymbol{\tau}} - \boldsymbol{\lambda}^T\boldsymbol{B}\end{aligned} \tag{12}$$

We select $\boldsymbol{\lambda}^T$ so that the first term in parentheses vanishes, i.e. :

$$\frac{\partial \ell}{\partial \boldsymbol{z}} + \boldsymbol{\lambda}^T\frac{\partial \mathcal{G}}{\partial \boldsymbol{z}} = 0 \quad \text{or,} \quad \left(\frac{\partial \mathcal{G}}{\partial \boldsymbol{z}}\right)^T\boldsymbol{\lambda} = -\left(\frac{\partial \ell}{\partial \boldsymbol{z}}\right)^T \tag{13}$$

The linear system of equations was solved using a direct LU solver. The vector $\boldsymbol{\lambda}$ found was substituted in Eq. (12) in order to obtain the desired gradient which is given by:

$$\frac{d\mathcal{L}}{d\boldsymbol{\tau}} = \frac{d\ell}{d\boldsymbol{\tau}} = -\boldsymbol{\lambda}^T\boldsymbol{B} \tag{14}$$

Subsequently, and by application of the chain rule we can obtain derivatives with respect to $\boldsymbol{\theta}$ as:

$$\frac{d\ell}{d\boldsymbol{\theta}} = \underbrace{\frac{\partial \ell}{\partial \boldsymbol{\tau}}}_{\substack{\text{Adjoint} \\ \text{model}}}\underbrace{\frac{\partial \boldsymbol{\tau}}{\partial \boldsymbol{\theta}}}_{\substack{\text{NN} \\ \text{auto-diff}}} \tag{15}$$

where $\partial\boldsymbol{\tau}/\partial\boldsymbol{\theta}$ was efficiently computed by back-propagation, which is a reverse accumulation automatic differentiation algorithm for deep neural networks that applies the chain rule on a per-layer basis. We note that since the parameters $\boldsymbol{\theta}$ are common for each likelihood $\ell^{(m)}$ the aforementioned terms would need to be added as per Eq. (9).

Similarly by chain rule, the gradient with respect to the vector $\boldsymbol{E}_\tau$ is given by:

$$\frac{d\ell}{d\boldsymbol{E}_\tau} = \boldsymbol{W}^T\frac{d\ell}{d\boldsymbol{\tau}} \tag{16}$$

We note finally that the expectations involved in the ELBO and its gradient (Eq. (5)) are approximated by Monte Carlo i.e.:

$$\mathcal{F}(\boldsymbol{\xi}) \approx \frac{1}{K}\left(\sum_{k=1}^{K}\left(\sum_{m=1}^{M}\ell^{(m)}(\boldsymbol{\theta}, \boldsymbol{E}_\tau^{(m,k)}) + \log p(\boldsymbol{E}_\tau^{(m,k)}|\boldsymbol{\Lambda})\right) + \log p(\boldsymbol{\theta}) + \log\ p(\boldsymbol{\Lambda}) - \log q_\xi(\boldsymbol{\theta}, \boldsymbol{\Lambda}, \boldsymbol{E}_\tau^{(m,k)})\right) \tag{17}$$

where:

$$\boldsymbol{\phi}^{(m,k)} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}), \ \ \boldsymbol{E}_\tau^{(m,k)} = g_{\boldsymbol{\xi}}(\boldsymbol{\phi}^{(m,k)}), \tag{18}$$

and:

$$\nabla_{\boldsymbol{\xi}}\mathcal{F}(\boldsymbol{\xi}) \approx \frac{1}{K}\sum_{k=1}^{K}\nabla_{\boldsymbol{\xi}}g_\xi(\boldsymbol{\phi}^{(k)})\nabla_{\boldsymbol{\eta}}\left(\log p(\mathcal{D}, \boldsymbol{\eta}^{(k)}) - \log q_\xi(\boldsymbol{\eta}^{(k)})\right) \tag{19}$$

where $\boldsymbol{\eta}^{(k)} = g_\xi(\boldsymbol{\phi}^{(k)})$.