Dimensionality reduction for regularization of sparse data-driven RANS simulations

Journal of Computational Physics 492 (2023) 112404

Contents lists available at ScienceDirect

Journal of Computational Physics

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

Dimensionality reduction for regularization of sparse data-driven RANS simulations

Pasha Piroozmand *, Oliver Brenner, Patrick Jenny *

Institute of Fluid Dynamics, Sonneggstrasse 3, ETH Zurich, Zurich CH-8092, Switzerland

A R T I C L E I N F O

Article history:
Received 9 September 2022
Received in revised form 20 July 2023
Accepted 28 July 2023
Available online 14 August 2023

Keywords:
Data assimilation
Data-driven simulation
Regularization
Dimensionality reduction
RANS

A B S T R A C T

Data assimilation can reduce the model-form errors of RANS simulations. A spatially distributed corrective parameter field can be introduced into the closure model, whose optimal values can be efficiently found by an adjoint method and gradient-based optimization. When assimilating experimental data, which in most cases are sparsely distributed or based on a low-resolution grid, the inverse problem is highly underdetermined and thus ambiguous. Therefore, to reduce the ambiguity, regularization is required. Established regularization approaches such as total variation and Sobolev gradient methods can produce smooth and physically meaningful velocity fields in many cases. However, if the measurements are located close to walls, spiky and unphysical wall shear stress profiles can occur. A new regularization strategy based on a piecewise linear approximation of the corrective field is proposed. This method is shown to lead to a very accurate free stream velocity field and smooth wall shear stress profiles. The resulting skin friction drag error for the case of flow over periodic hills was around 1.38% which is seven times lower than the error obtained with the Sobolev gradient and two orders of magnitude lower than that obtained with the other two methods.

1. Introduction

Experimental methods can be used to study turbulent flow by measuring the velocity and pressure distributions. However, most of these methods only provide sparse measurements (e.g. hot-wire anemometry, laser Doppler, etc.) or low-resolution fields (e.g. PIV, PTV, etc.) [1]. Computational Fluid Dynamics (CFD), on the other hand, provides high-resolution fields that are subject to some degree of uncertainty, depending on the level of description. Despite the surge of computational power in recent decades, it still is computationally demanding, or even prohibitive, to compute high-Reynolds number turbulent flows with Direct Numerical Simulation (DNS) or even Large Eddy Simulation (LES). For example, the computational cost of wall-bounded turbulent LES roughly scales with $Re^{1.8}$ [2]. Reynolds-averaged Navier-Stokes (RANS) models, mostly based on the linear eddy viscosity (LEV) assumption, are often the only computationally affordable option. However, their accuracy is limited for complex flows, e.g., flows with large pressure gradients, unsteady separation zones, strong streamline curvatures, abrupt changes in mean flow, etc. [3].

* Corresponding authors.
E-mail addresses: ppasha@ethz.ch (P. Piroozmand), brennero@ethz.ch (O. Brenner), jenny@ifd.mavt.ethz.ch (P. Jenny).

https://doi.org/10.1016/j.jcp.2023.112404

Combining low-resolution or sparse point-wise measurement data with RANS simulations has the potential to provide computationally affordable, highly-resolved, and accurate results [4–7]. Such an approach can be called *data-driven* simulation and the mathematical framework for its implementation is data assimilation (DA) [8].

Another approach to improve RANS models is *physics-driven* [8]. In this approach, more involved Reynolds stress models or more accurate eddy viscosity models within the LEV framework are proposed. Nonlinear eddy viscosity models [9], explicit algebraic stress models [10], and differential Reynolds stress models [11] have all been proposed to directly improve the computation of Reynolds stresses and can achieve higher accuracy for some applications. However, because of their lack of robustness and numerical stability, such models have not been widely used in engineering applications [12]. On the other hand, more sophisticated models for the computation of the eddy viscosity in the context of LEV models, such as adding more turbulence transport equations with more terms, lead to more free parameters that need to be calibrated and also to a higher computational cost. In fact, no new, widely-accepted general-purpose RANS turbulence model has been developed in recent years, and the currently used models, despite their unsatisfactory performance in many cases, (e.g. the k-ϵ [13], k-ω [14], Spalart–Allmaras [15] and SST k-ω models [16]) were all developed decades ago [17]. Therefore, data-driven RANS simulations are an alternative to obtain accurate, high-resolution results at a reasonable computational cost, at least when measurements exist within the domain.

It should be noted that such data-driven RANS simulations cannot replace general-purpose physics-driven RANS models. The goal of DA here is to obtain accurate results only for the specific problem of interest. DA can indeed also be used as a step towards data-driven modeling, for example using machine learning techniques, where the trained model should give results even when data are not available in other configurations. However, this is outside the scope of this paper.

There are numerous DA techniques to incorporate data into the model. The best choice is highly case dependent. Depending on whether the problem is steady-state or transient, the available data is sparse or abundant, time-dependent or averaged, and if it involves uncertainties, one can choose the best relevant approach.

Three main DA approaches are variational, statistical, and nudging [18]. Variational methods involve formulating an optimization problem that minimizes the difference between model simulations and observations, while allowing for uncertainties in both. Statistical approaches, on the other hand, employ probabilistic frameworks such as the Kalman filter to estimate the state of a system by updating model forecasts using measurement data information [19,20]. Lastly, nudging techniques adjust model variables towards observed values through a relaxation process, effectively blending model dynamics with observed data while preserving key model features [21].

In the current work, the goal is to reconstruct mean velocities and wall shear stresses from sparse time-averaged measurement data. The data is assumed to be deterministic and has no uncertainty. To reach accurate results, such a reconstruction problem demands a high dimensional parameter space. Having these conditions, the variational data assimilation can be the best choice due to its computational advantage for high dimensional problems. In variational data assimilation for RANS simulations, the data is incorporated by introducing parameters to the RANS model equations. Discrepancies between sparse measurement data and RANS results at those locations are minimized by tuning the parameter values.

By minimizing the discrepancy, the model should match the data at the measurement points, and the overall accuracy of the RANS results is expected to improve. To solve the minimization problem, a gradient-based optimization approach can be used, where the sensitivity of the discrepancy function with respect to the parameters is used to find the minimum of the cost function. Various other optimization methods exist, e.g., Newton methods, that rely on second-order derivatives albeit with a substantial increase in computational and memory costs. Quasi-Newton methods such as BFGS and Gauss-Newton methods can circumvent this issue by approximating the Hessian matrix, while considering only the first-order derivative information [22,23]. Depending on the problem, the sensitivities can be computed by the adjoint method, automatic differentiation, analytically, or numerically using finite difference methods [24].

The other two alternatives to variational DA, namely statistical methods and nudging, may not be suitable for the problem at hand. The former, e.g., the ensemble Kalman filter method, is computationally expensive, if the number of tuning parameters is very large [19]. The latter tends to result in spiky solutions, if sparse point-wise measurement data are used [21].

The level at which the parameters are introduced into the RANS equations is another crucial task. There exists a spectrum of possibilities from assuming the Reynolds stresses in each cell to be the parameters, to only correcting a global coefficient of a specific RANS model (e.g. correcting constant C_μ in the standard k-ϵ model). In the former approach, the parameters are a symmetric tensor field, i.e., $6 \times N$ parameters have to be considered (N is the mesh size), while in the latter approach, the parameter is only one scalar value.

It is tempting to try to derive from the data the Reynolds stress tensor that provides the greatest *flexibility* [5]. However, it is necessary to address some important issues

The first issue arises from the non-uniqueness of the solution in such inverse problems. In the order of N parameters have to be adjusted, but typically much fewer measurements exist. Therefore, there may exist many different Reynolds stress tensor fields that lead to a match of the numerical solution with the reference velocities [12,4,5].

The second issue is the ability of the optimizer to find proper minima. The optimizer relies solely on the local gradient information pointing in the direction of a local minimum.

The last issue is the ill-conditioned nature of the data-driven Reynolds stress model, which affects the optimization process and makes the search for the minimum very sensitive to tiny changes in the Reynolds stress tensor field. Wu et al. [12] showed that even in an ideal scenario where the DNS Reynolds stress data are directly inserted as explicit source term into a RANS solver, tiny deviations in the inserted Reynolds stresses can lead to large errors in the resulting mean velocity fields. This shows that the forward problem itself is ill-conditioned. In the case of the inverse problem, where the Reynolds stresses are to be estimated by the optimizer, inaccuracies in the gradient calculation or the optimization process make it exceedingly difficult to obtain accurate results.

In summary, inferring the Reynolds stress tensor field from only a few point velocity measurements would be an extremely difficult task. Two attempts in this regard are the works of [25] and [26], and more recently, [7], in which the Reynolds forcing vector field is inferred from partial data while ensuring the divergence-free constraint. The task becomes significantly easier and more robust if the degrees of freedom of the problem are reduced. Further, to reduce ambiguity, some sort of regularization is required, e.g.:

- By introducing further constraints for the Reynolds stresses, e.g., that they must be realizable or that the eddy viscosity Ansatz must be used [27,5,28].
- By adding penalties to the cost function which that penalize, e.g., spiky solutions and large deviations from a baseline state [5],
- By applying filtering to remove unphysical noise [29],
- Or by reducing the degrees of freedom (number of parameters) by spatial coarsening of the parameter field, e.g., by piecewise decomposition.

One can start by modeling the Reynolds stress field using a turbulence model and then try to correct the modeled stresses with additive data-driven corrective stresses. In these cases, realizability can be enforced by perturbing only the eigenvalues or eigenvectors of the Reynolds stress tensors within realizable limits [30,31]. So far such methods were mostly applied when full DNS data was available.

To further reduce the degrees of freedom, instead of a tensor field, a corrective vector field resembling the divergence of a corrective tensor field can be introduced into the momentum equations [28].

In their study, Brenner et al. [4] maintained the Boussinesq eddy viscosity assumption and adjusted the eddy viscosity to match sparse measurement data. They demonstrated that while the eddy viscosity assumption may be somewhat unphysical when combined with an effective regularization method such as the total variation (TV) penalty that prevents abrupt changes in the eddy viscosity field, it can produce physically reasonable and accurate results for complex problems such as the periodic hill case involving separation and reattachment. However, it can be shown that this method leads to jagged wall shear stress profiles when the measurement data is situated near the walls, and that traditional regularization approaches like the Sobolev gradient [29] and TV methods cannot address this issue.

In this study, the concept of piecewise linear dimension reduction (PLDR) is presented as a regularization strategy that not only results in physically acceptable and accurate internal velocity profiles, but also provides accurate wall shear stresses. More specifically, piecewise linear Lagrange elements are used to further constrain the parameter field, resulting in a reduced number of parameters to be tuned. Note that the approach is similar in some ways to the P1 Finite element method (FEM). Both P1 FEM and PLDR decompose the domain into linear Lagrange elements. However, the goal of the two is different. The FEM subdivides a domain into smaller elements by constructing local element equations that approximate the PDEs. These equations eliminate spatial derivatives and are solved as algebraic equations. The P1 element, with linear interpolation between nodes within an element, enables the local approximation of the solution. In the case of piecewise linear dimensionality reduction (PLDR), the use of triangular elements aims to remove the nonphysical variations in the eddy viscosity caused by the gradient-based optimizer. Triangular elements serve as a regularization technique to mitigate these variations.

To assess the accuracy of this approach, it is compared with classical regularization techniques such as the total variation and Sobolev gradient methods. It is argued that this approach provides a good trade-off between flexibility and accuracy.

The rest of the paper is structured as follows. In section 2, the problem statement of both forward and inverse problems is introduced. In addition, the PLDR method and some conventional regularization schemes are presented. In section 3 results using the PLDR method are presented and compared with those obtained with other methods, and in section 4 conclusions are provided.

2. Methods

In this section, the mathematical framework of the DA approach and the test case setups are explained.

2.1. Problem statement

Two problems are inverse with respect to each other, if the formulation of one involves the solution of the other [18]. The problem that is easy to solve and generally well-posed is called the forward problem. The problem that is more difficult to solve and generally ambiguous is called the inverse problem. Regarding turbulence modeling, the forward problem consists of solving the RANS or LES equations based on initial and boundary conditions. In the present case, the inverse problem is to infer spatially varying model parameters from sparse measurement data [32].

2.1.1. Formulation of the forward problem

The steady-state RANS equations for an incompressible Newtonian fluid are introduced as

$$\frac{\partial \bar{u}_i \bar{u}_j}{\partial x_j} + \frac{\partial \tau_{ij}}{\partial x_j} = -\frac{\partial \bar{p}}{\partial x_i} + \frac{\partial}{\partial x_j}\left(\nu \frac{\partial \bar{u}_i}{\partial x_j}\right) , \tag{1}$$

where $\bar{u}_i$ are the mean velocity components, $\bar{p}$ is the mean kinematic pressure assuming constant density ρ, and ν the constant kinematic viscosity. The Reynolds-stress tensor $\tau_{ij} = \overline{u_i' u_j'}$ is responsible for the momentum transfer by velocity fluctuations. This unclosed term needs to be modeled due to the loss of information through the averaging process. The above equations are solved together with the continuity equation

$$\frac{\partial \bar{u}_i}{\partial x_i} = 0 \tag{2}$$

to obtain the mean velocity and pressure fields. For the reasons explained in the introduction, the Boussinesq hypothesis is used, which assumes a linear constitutive relationship between the deviatoric part of the Reynolds stress tensor and the mean flow strain rate, i.e., that

$$\tau_{ij} = \frac{2}{3}\bar{k}\delta_{ij} - 2\nu_t \bar{S}_{ij} , \tag{3}$$

where δ_{ij} is the Kronecker delta, $\bar{S}_{ij} = \frac{1}{2}\left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right)$ is the mean flow strain rate, and $\bar{k} = \frac{1}{2}\overline{u_i' u_i'}$ is the turbulent kinetic energy. In practice, the hydrostatic part (the first part of the right-hand side term of equation (3)) is absorbed into the pressure gradient term in equation (1), and the kinematic pressure $\bar{p}$ is replaced by the modified kinematic pressure $\bar{p}^* = \bar{p} + \frac{2}{3}\bar{k}$. Therefore, $\bar{k}$ is not required even for a closure where only an accurate velocity field is sought. The proportionality factor ν_t is the eddy viscosity, which must be evaluated locally in each cell. One could try to obtain ν_t directly from sparse measurement data. However, the task seems easier if one starts with a good ν_t field obtained, e.g., with a conventional LEV model. Then only a moderate correction to ν_t by a correction factor α^m (also it is called the model parameter here and is initially one) may be needed [33]. The resulting governing PDE reads

$$\frac{\partial \bar{u}_i \bar{u}_j}{\partial x_j} = -\frac{\partial \bar{p}^*}{\partial x_i} + \frac{\partial}{\partial x_j}\left(2\left(\nu + \alpha^m \nu_t^b\right)\bar{S}_{ij}\right) , \tag{4}$$

where ν_t^b is the baseline eddy viscosity computed by an eddy viscosity RANS model. In the remainder of this manuscript, a simplified notation will be used. In particular, the modified kinematic pressure is denoted as pressure p, the mean velocity components as $\{u_1, u_2, u_3\}$, and the vector of state variables as $U = \{u_1, u_2, u_3, p\}$.

2.1.2. Formulation of the inverse problem

The data assimilation problem involves minimizing the discrepancy between the state variables computed by the RANS model and the reference data. This is formulated as an optimization problem where a cost function is minimized subject to the residual of the governing equation ($\boldsymbol{R}$) and the constraint that the optimized parameter must be greater than zero (details are given in section 2.3.1), which can be mathematically expressed as

$$\min_{\boldsymbol{\alpha}} \quad f(\boldsymbol{\alpha}, \boldsymbol{U}) \tag{5a}$$

$$\text{subject to} \quad \boldsymbol{R}(\boldsymbol{\alpha}, \boldsymbol{U}) = 0 \tag{5b}$$

$$\text{and} \quad \boldsymbol{\alpha} > 0 . \tag{5c}$$

The cost function consists of two parts: one that depends solely on the parameter $\boldsymbol{\alpha}$ and includes regularization functions, and another that directly measures the deviation of the forward solution $\boldsymbol{U}$ from the reference data, i.e.,

$$f(\boldsymbol{\alpha}, \boldsymbol{U}) = f_\alpha(\boldsymbol{\alpha}) + f_U(\boldsymbol{U}) . \tag{6}$$

For the sake of conciseness, the explicit dependency is removed ($f(\boldsymbol{\alpha}, \boldsymbol{U}) = f$). In addition, for now, the regularization part f_α is ignored. The cost function is thus defined as the squared Euclidean distance between the two data sets, i.e.,

$$f = \hat{\boldsymbol{V}} \cdot \sum_{k=1}^{4} \left(\mathbf{H}\boldsymbol{U}_k - \hat{\boldsymbol{U}}_k\right)^{\circ 2} , \tag{7}$$

where $\boldsymbol{U}_k$ is the vector of the k-th state variable computed in each cell, $\mathbf{H}$ is the linear observation operator [18] (an $R \times N$ matrix with R being the number of measurement locations), $(\cdot) \cdot (\cdot)$ is the inner product, $(\cdot)^{\circ 2}$ is the Hadamard power of two (element-wise square of the vector), $\hat{\boldsymbol{U}}_k$ is the vector of the k-th measured state variables, and $\hat{\boldsymbol{V}}$ is the corresponding vector of cell volumes. In practice, one may only measure a single component of the state variables, e.g., the horizontal velocity component u_1, resulting in

$$f = \hat{\boldsymbol{V}} \cdot \left(\mathbf{H}\boldsymbol{u}_1 - \hat{\boldsymbol{u}}_1\right)^{\circ 2} \,, \tag{8}$$

where the components of $\boldsymbol{u}_1$ and $\hat{\boldsymbol{u}}_1$ are the calculated and measured horizontal velocities, respectively.

Knowing the sensitivity of the cost function f with respect to the parameter vector $\boldsymbol{\alpha}$ would allow the parameter values to be optimized iteratively using a gradient-based algorithm as [4]

$$\boldsymbol{\alpha}^{(t+1)} = \boldsymbol{\alpha}^{(t)} - \Delta \left(\frac{\mathrm{d}f^{\,(t)}}{\mathrm{d}\boldsymbol{\alpha}}\right)^T \,, \tag{9}$$

where $t = 0, \ldots, T_{max}$ denotes the iteration number of the optimization, with T_{max} being the maximum number of iterations and Δ is the learning rate or optimization step size. Next, it is discussed how to obtain the cost function gradient $\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{\alpha}}$, while the choice of Δ, which can be either a scalar or a matrix, will be discussed later.

2.1.3. Sensitivity calculation within the discrete adjoint method

The most straightforward way to compute the sensitivity of the cost function f with respect to the parameter vector $\boldsymbol{\alpha}$ is the brute-force finite difference (FD) method, where each parameter is perturbed separately while the change in the cost function is measured [34]. However, this requires M evaluations of the forward problem, where M is the number of parameters (length of the vector $\boldsymbol{\alpha}$). Of course, this approach becomes extremely expensive for of large M (M is equal to the number of cells if each cell is allowed to attain an individual α value). In addition to the large computational cost of the method, the obtained sensitivities depend very much on the perturbation amplitude. The complex-step method [35] can circumvent this issue, but the computational cost still scales with the number of parameters.

Another option would be to apply automatic differentiation (AD) in a reverse mode to the forward problem solver code [36]. This way, the number of forward problem evaluations required becomes independent of the number of parameters. However, this approach requires an enormous amount of memory to store all the intermediate steps of the iterative forward solver.

Instead, the sensitivity can be computed at low computational cost and low memory requirements using the discrete adjoint method. Using the chain rule, the sensitivity can be written as

$$\underbrace{\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{\alpha}}}_{1\times M} = \underbrace{\frac{\partial f}{\partial \boldsymbol{\alpha}}}_{1\times M} + \underbrace{\frac{\partial f}{\partial \boldsymbol{U}}}_{1\times 4N} \underbrace{\frac{\partial \boldsymbol{U}}{\partial \boldsymbol{\alpha}}}_{4N\times M} \,, \tag{10}$$

where $\boldsymbol{U}$ is the vector of all four state variables in all N cells. The term $\frac{\partial \boldsymbol{U}}{\partial \boldsymbol{\alpha}}$ is a $4N \times M$ matrix that quantifies the sensitivities of the state variables in each cell (rows) with respect to all parameters (columns). While the vectors $\frac{\partial f}{\partial \boldsymbol{\alpha}}$ and $\frac{\partial f}{\partial \boldsymbol{U}}$ can be explicitly computed, there is no explicit relationship between the state variables and the parameters to obtain $\frac{\partial \boldsymbol{U}}{\partial \boldsymbol{\alpha}}$. However, $\boldsymbol{U}$ and $\boldsymbol{\alpha}$ are implicitly related by the residual equations $\boldsymbol{R}\left(\boldsymbol{U}, \boldsymbol{\alpha}\right) = \boldsymbol{0}$ of the forward problem. Since by definition $\boldsymbol{R}$ and its total derivative with respect to α are identical to zero, it is

$$\underbrace{\frac{\mathrm{d}\boldsymbol{R}}{\mathrm{d}\boldsymbol{\alpha}}}_{4N\times M} = \underbrace{\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{\alpha}}}_{4N\times M} + \underbrace{\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}}}_{4N\times 4N} \underbrace{\frac{\partial \boldsymbol{U}}{\partial \boldsymbol{\alpha}}}_{4N\times M} = 0 \,. \tag{11}$$

The above equations can be rearranged to obtain an expression for $\frac{\partial \boldsymbol{U}}{\partial \boldsymbol{\alpha}}$, which is then substituted into equation (10) to give

$$\underbrace{\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{\alpha}}}_{1\times M} = \underbrace{\frac{\partial f}{\partial \boldsymbol{\alpha}}}_{1\times M} - \overbrace{\underbrace{\frac{\partial f}{\partial \boldsymbol{U}}}_{1\times 4N} \underbrace{\left(\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}}\right)^{-1}}_{4N\times 4N}}^{\boldsymbol{\lambda}^T} \underbrace{\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{\alpha}}}_{4N\times M} = \underbrace{\frac{\partial f}{\partial \boldsymbol{\alpha}}}_{1\times M} - \underbrace{\boldsymbol{\lambda}^T}_{1\times 4N} \underbrace{\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{\alpha}}}_{4N\times M} \,, \tag{12}$$

where the components of $\boldsymbol{\lambda}$ are called adjoint variables. These are computed by evaluating the so-called *adjoint* equations

$$\underbrace{\left(\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}}\right)^T}_{4N\times 4N} \underbrace{\boldsymbol{\lambda}}_{4N\times 1} = \underbrace{\left(\frac{\partial f}{\partial \boldsymbol{U}}\right)^T}_{4N\times 1} \,. \tag{13}$$

This system of equations is solved iteratively, similar to the solution of the forward problem. Note that the system matrix and the right-hand side are both independent of $\boldsymbol{\alpha}$. Therefore, the computational cost of solving the adjoint equations is independent of the number of parameters and in the order the cost of a single evaluation of the linearized forward problem. This is a major advantage compared to, e.g., the finite difference method which requires M additional forward problem evaluations.

As stated, the vectors $\frac{\partial f}{\partial \boldsymbol{\alpha}}$ and $\frac{\partial f}{\partial \boldsymbol{U}}$ can be computed analytically, but the matrices $\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}}$ and $\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{\alpha}}$ need to be evaluated in a different way. For this purpose the reverse mode of AD could be used. However, the naive use of AD also leads to a large memory usage in this case. Numerous approaches to circumvent this issue have been proposed [37,38,24]. Instead of using AD, a simpler approach with some approximating assumptions is taken here. Taking advantage of the fact that the linearized discretized equations, e.g. by applying the finite volume method, have the form

$$\underbrace{\boldsymbol{R}(\boldsymbol{\alpha}, \boldsymbol{U})}_{4N\times 1} = \underbrace{\boldsymbol{A}}_{4N\times 4N} \underbrace{\boldsymbol{U}}_{4N\times 1} - \underbrace{\boldsymbol{b}}_{4N\times 1} = \underbrace{\boldsymbol{0}}_{4N\times 1}, \tag{14}$$

where $\boldsymbol{A}$ is the system matrix of the linearized discretized residual. Note that $\boldsymbol{A}$ is equal to the Jacobian $\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}}$, whereas in general (e.g. if OpenFOAM is used) $\boldsymbol{A}$ only is an approximation of $\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}}$.

Moreover, note that $\boldsymbol{A}$ is a $4N \times 4N$ matrix whose elements take into account the implicit influence of each state variable on the corresponding residual equations. Such terms also arise from the velocity-pressure coupling in the forward problem. Such a coupling often is treated by segregated methods such as the *SIMPLE* algorithm, where not all off-diagonal components of the Jacobian appear in $\boldsymbol{A}$. Here, a fully coupled solver is used for both forward and inverse problems, where all off-diagonal elements of $\boldsymbol{A}$ are computed [39]. This provides a more accurate approximation of $\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}}$. Another advantage of coupled solvers is faster convergence, since the iterations only are needed for the nonlinear convective terms, albeit at the cost of higher memory requirements.

Computing the remaining term $\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{\alpha}}$ can be tricky if neither AD nor FD approaches are used. To address this issue, having converged forward solutions, implicit operators provided by *OpenFOAM* [40] are used to discretize and linearize the residual equations with respect to $\boldsymbol{\alpha}$. One should note that $\boldsymbol{\alpha}$ only appears explicitly in the momentum equations and not in the pressure equation. Therefore, the effect of the term $\frac{\partial \boldsymbol{R}_4}{\partial \boldsymbol{\alpha}}$ on $\frac{\mathrm{d} f}{\mathrm{d} \boldsymbol{\alpha}}$ is neglected. Brenner et al. [4] showed that this approximation has some inaccuracies, but is generally good enough for successful gradient-based optimization.

2.2. Regularization

As discussed in the introduction, merely minimizing the distance function without any constraints may result in inaccurate solutions between the measurement points. With a linear eddy viscosity model, the inverse problem can at least be initialized with a reasonable solution. However, it can be shown that for the case where only sparse measurements are available, the correction due to DA is spiky and thus unphysical [4]. More precisely, this means that the optimized α field results in accurate velocities at the reference points, but is generally inaccurate between the reference points. For assessment, this can be quantified by introducing test points (cf. Fig. 4) at which the discrepancy between computed and reference velocities is measured.

The process of adding more information or constraints to reduce ambiguity can be called regularization. First, three common regularization methods are discussed and then the PLDR approach is proposed. It should be noted that all the regularization methods are straightforward to implement and do not affect the adjoint computations. The first two, L_2- and TV-regularization, affect only the first right-hand side term in equation (12). Sobolev gradient- and PLDR-regularization directly affect the sensitivity $\frac{\mathrm{d} f}{\mathrm{d} \boldsymbol{\alpha}}$, but in a step after the sensitivity is computed by the adjoint method.

The additional computational cost of L_2- and TV-regularization is negligible, since only one iteration over all grid cells and faces, respectively, has to be added. The Sobolev gradient-regularization requires solving a Helmholtz equation, which imposes an additional cost of the order of the forward or adjoint problem, making it more expensive than L_2- and TV-regularization. The computational cost of PLDR is negligible, similar to L_2 and TV regularization, since the most expensive part is the computation of the basis functions which only depends on the geometry and can be computed and stored before starting the optimization iterations.

2.2.1. L_2-regularization

The L_2-regularization relies on an additional term in the cost function that penalizes the deviation of $\boldsymbol{\alpha}$ from its base value $\boldsymbol{\alpha^b}$ (which is unity in the present case). This automatically enforces some smoothing of the optimized parameter field, since it minimizes the changes in $\boldsymbol{\alpha}$. At the same time, the ambiguity is reduced. To implement the method, the squared L_2 norm penalty term is added to the cost function in equation (8) as

$$f = \hat{\boldsymbol{V}} \cdot \left(\mathbf{H}\boldsymbol{u}_1 - \hat{\boldsymbol{u}}_1\right)^{\circ 2} + \omega_{L2} \left\| \boldsymbol{\alpha} - \boldsymbol{\alpha}^b \right\|^2 , \tag{15}$$

where ω_{L2} is a regularization hyperparameter. The first term penalizes the misfit of the model output with respect to the measurement values and the second term penalizes the deviation from the baseline parameter field.

In practice, the effect of regularization contributes to the cost function gradient (cf. equation (14)), since

$$\underbrace{\frac{\partial f}{\partial \boldsymbol{\alpha}}}_{1\times M} = 2\,\omega_{L2}\,\underbrace{\left(\boldsymbol{\alpha}-\boldsymbol{\alpha}^{b}\right)}_{1\times M}\,. \tag{16}$$

2.2.2. Total variation-regularization

Total variation-regularization (TV) is based on penalizing the local spatial gradient of α [41]. The cost function in equation (8) is replaced by

$$f = \hat{\mathbf{V}}\cdot\left(\mathbf{H}\boldsymbol{u}_1-\hat{\boldsymbol{u}}_1\right)^{\circ 2} + \omega_{TV}\sum_{m=1}^{M}\left[\frac{1}{|\mathcal{B}_m|}\sum_{o\in\mathcal{B}_m}(\alpha_m-\alpha_o)^2\right], \tag{17}$$

where ω_{TV} is the regularization hyperparameter, m is the index looping over all parameters, $\mathcal{B}_m$ is the set of parameters adjacent to the parameter with index m, o the index looping over the set $\mathcal{B}_m$, and $|\mathcal{B}_m|$ is the number of adjacent parameters. Similar to the L_2-regularization, the TV-regularization effectively influences the optimization, since

$$\frac{\partial f}{\partial \alpha_m} = 2\,\omega_{TV}\sum_{o\in\mathcal{B}_m}\left[\left(\frac{1}{|\mathcal{B}_m|}+\frac{1}{|\mathcal{B}_o|}\right)(\alpha_m-\alpha_o)\right]. \tag{18}$$

2.2.3. Sobolev gradient-regularization

The Sobolev gradient-regularization is based on the projection of the sensitivity field $\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{\alpha}}$ defined in L^2-space onto the more regular Sobolev space (H^1-space) [29]. This is achieved by solving the Poisson equation, that is,

$$\underbrace{\left(\frac{1}{1+{l_{sob}}^2}\left(\mathbb{I}-l_{sob}^2\nabla^2\right)\right)}_{M\times M}\underbrace{\left(\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{\alpha}}^{H^1}\right)^T}_{M\times 1} = \underbrace{\left(\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{\alpha}}\right)^T}_{M\times 1}, \tag{19}$$

for $\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{\alpha}}^{H^1}$, where ∇^2 is the Laplace operator, l_{sob} is a regularization parameter, and $\mathbb{I}$ is the $M\times M$ identity matrix. This equation can be solved by the same finite volume method as used for the forward problem. The parameter l_{sob} can be considered as a low-pass filter width below which the fluctuations of the sensitivity field are smoothed.

While ω_{L2} and ω_{TV} are hyperparameters with no physical meaning that must be adjusted by trial and error, the Sobolev gradient parameter l_{sob} has a physical meaning and thus can be defined more easily [29].

2.2.4. Piecewise linear dimensionality reduction (PLDR)

It will be shown that the regularization methods discussed above are unable to provide non-spiky wall shear stress profiles even for optimal hyperparameter values. For this purpose, the piecewise linear dimensionality reduction (PLDR) method is proposed.

The basic idea of the PLDR method is to reduce the noise and the high spatial variability in the parameter field by decreasing the degrees of freedom. More specifically, the α field is represented as a superposition of piecewise linear functions (of linear Lagrange elements), so that the number of unknown weights β is much smaller than the number of grid cells. For the two-dimensional case, the elements are defined as triangles whose vertices coincide with the measurement points, i.e., the α field is represented as

$$\boldsymbol{\alpha} = \boldsymbol{\Phi}\,\boldsymbol{\beta}\,, \tag{20}$$

where $\boldsymbol{\beta}$ is the new parameter vector and $\boldsymbol{\Phi}$ is the piecewise linear projection matrix. It can be written also in the vector form as $\alpha_m = \phi_{mr}\,\beta_r$ where subscripts $m = 1, \ldots, M$ and $r = 1, \ldots, R$ are the indices of the vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, respectively. Here, it is assumed that R equals to the number of measurement points. The components of each row of $\boldsymbol{\Phi}$ are all zero, except for three components corresponding to the three vertices of the associated triangular element. The sum of each row of $\boldsymbol{\Phi}$ is unity and every α_m is a linear combination of three corresponding nonzero nodal values β_r.

Note that the triangular elements may not cover the whole domain, e.g., some cells near the convex parts of the boundaries of the periodic hill test cases may lie outside a triangular element. To solve this issue, dummy nodes are added far away from the boundary outside of the domain. This acts as a Neumann boundary condition for the reconstructed α field. An example of piecewise linear basis functions is depicted in Fig. 1.

In practice, PLDR regularizes the inverse problem by reducing the number of dimensions M of α to R of β, where $R \ll M$. The chain rule yields

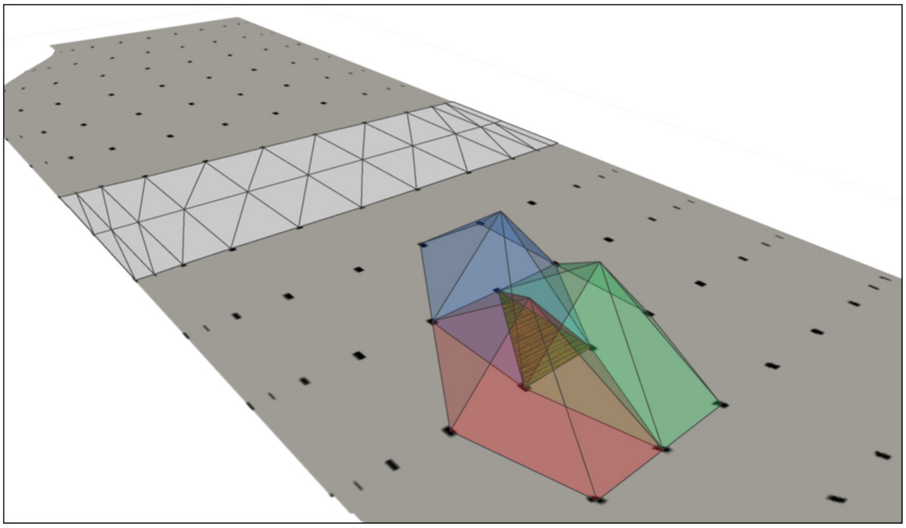


Fig. 1. The basis functions (blue, green, and red polyhedrons) span the solution within the dashed yellow triangle. The measurement points are depicted by black dots, and a few elements are shown as white triangles (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

$$\underbrace{\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{\beta}}}_{1\times R} = \underbrace{\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{\alpha}}}_{1\times M}\underbrace{\frac{\mathrm{d}\boldsymbol{\alpha}}{\mathrm{d}\boldsymbol{\beta}}}_{M\times R} = \underbrace{\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{\alpha}}}_{1\times M}\underbrace{\boldsymbol{\Phi}}_{M\times R} . \tag{21}$$

The values of $\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{\beta}}$ are normalized by the sum over each corresponding column of $\boldsymbol{\Phi}$ to be consistent with $\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{\alpha}}$. Based on $\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{\beta}}$, the parameter vector $\boldsymbol{\beta}$ is updated as

$$\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} - \Delta^{(t)} \left(\frac{\mathrm{d}f^{(t)}}{\mathrm{d}\boldsymbol{\beta}}\right)^T , \tag{22}$$

and equation (20) subsequently is used to reconstruct the parameter vector $\boldsymbol{\alpha}$. The optimization process continues by evaluating the forward and adjoint problems that yield the sensitivities, which in turn are used to update $\boldsymbol{\beta}$ again. The flowchart of the data assimilation procedure with PLDR is presented in Fig. 2.

2.3. Optimization

2.3.1. Positivity constraint of the eddy viscosity

Since the computed eddy viscosity has to be positive, it makes sense to incorporate this constraint into the optimization process. Brenner et al. [4] showed that imposing a hard constraint, e.g., by overwriting negative α values, impairs the performance of the optimizer. Instead, they suggested using a mapping function to avoid negative values. The mapping is

$$\alpha^m(\alpha) = \begin{cases} \alpha + 1, & \alpha > 0 \\ \exp(\alpha), & \alpha \leq 0 , \end{cases} \tag{23}$$

where α is the *intermediatory parameter* and α^m is the *model parameter*. The former can reach any real value, while the latter always is positive. Consequently, the sensitivities read

$$\frac{\mathrm{d}f}{\mathrm{d}\alpha} = \frac{\mathrm{d}f}{\mathrm{d}\alpha^m}\frac{\mathrm{d}\alpha^m}{\mathrm{d}\alpha} , \tag{24}$$

that is, the optimization directly affects the *intermediatory parameter* α and only indirectly affects the *model parameter* α^m. In practice, $\frac{\mathrm{d}f}{\mathrm{d}\alpha^m}$ is calculated first, then it is mapped to $\frac{\mathrm{d}f}{\mathrm{d}\alpha}$. Consequently $\frac{\mathrm{d}f}{\mathrm{d}\alpha}$ generates α through the optimizer. α^m is then obtained from α and is inserted in the RANS equations to solve the forward problem and to obtain sensitivities for the next minimization iteration.

2.3.2. Computation of the learning rate

Gradient-based optimization methods are widely used in data assimilation (DA) techniques. These methods can be divided into constrained and unconstrained methods, and penalty methods are commonly used to handle constraints. The

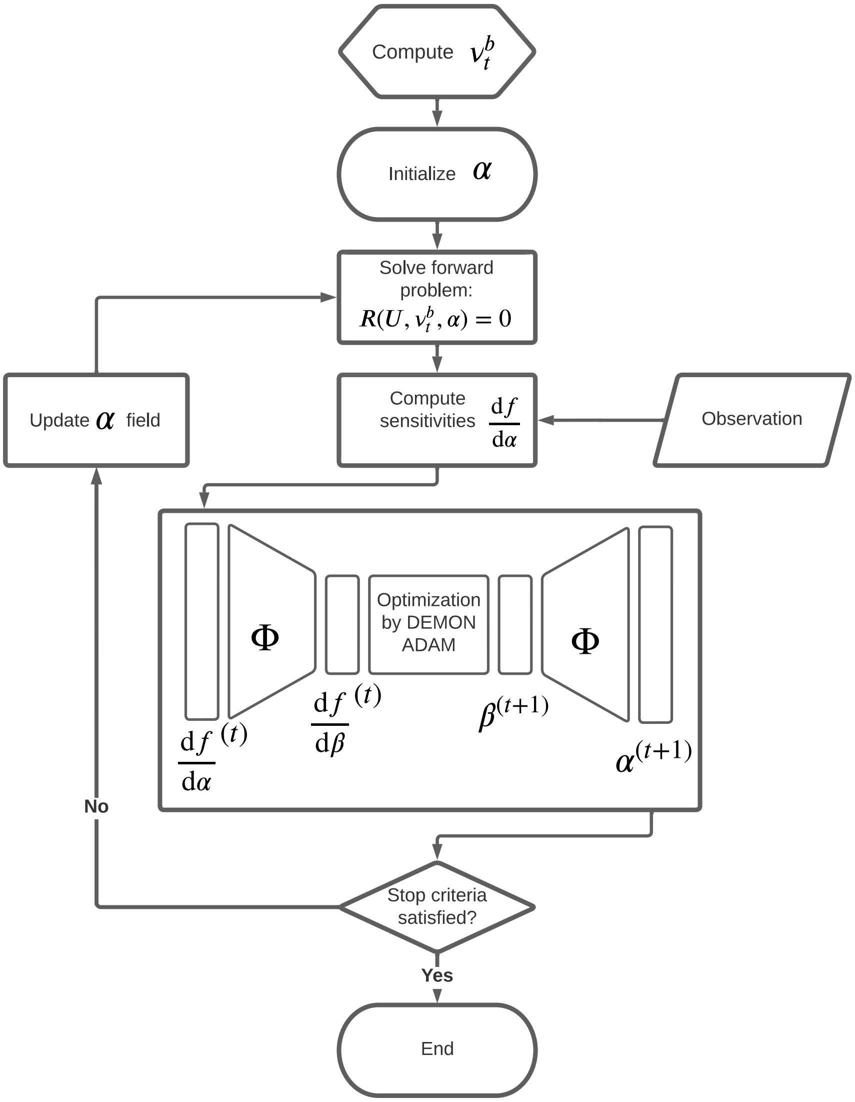


Fig. 2. Flowchart of the data assimilation procedure using the PLDR-regularization.

selection and adjustment of the penalty term, if applicable, are crucial aspects of the optimization process. Gradient-based optimization algorithms commonly used in DA include gradient descent, first-order Newton methods such as BFGS, and stochastic gradient descent inspired by machine learning approaches. However, it is important to acknowledge that high-dimensional problems present challenges due to the impracticality of storing large Hessian matrices. To address this, Sequential Quadratic Programming (SQP) techniques can be employed [42]. Moreover, gradient-enhanced surrogate-based methods (GEK) can be considered for low-dimensional problems [43].

In equation (22), $\boldsymbol{\Delta}$ can be either a scalar or a $R \times R$ matrix depending on the optimization method. The simplest case is the vanilla gradient descent method, where Δ is a constant whose value should be tuned by trial and error.

If the optimization method uses second-order derivatives (or a Hessian matrix), like Newton's method, $\boldsymbol{\Delta}$ is a matrix. Such methods lead to fewer optimization iterations. However, in general, Hessian matrices are not computed directly, due to high computational and memory cost, even when adjoint methods are used to compute the second-order derivatives. Another approach would be a quasi-Newton method such as BFGS, where the Hessian matrix is approximated using the gradient information [22].

In this work, an adaptive learning rate method is used where $\boldsymbol{\Delta}$ is a diagonal matrix, i.e., for each element of the parameter vector, a specific optimization step size is used at each iteration step, which is determined by the DEMON Adam method [44]. No correlations between parameters are assumed (zero off-diagonals). As an extension of the default Adam method, which uses larger learning rates for those parameter elements that were less frequently updated with larger gradients, a decaying momentum (DEMON) term is added to accelerate convergence [45]. In summary, equation (22) is replaced by

$$\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} - \boldsymbol{\Delta}^{(t)}\, \boldsymbol{\gamma}^{(t)}\,, \tag{25}$$

where

$$\boldsymbol{\gamma}^{(t)} = \left(\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{\beta}}\right)^T + \kappa^{(t)}\, \boldsymbol{\gamma}^{(t-1)}\,. \tag{26}$$

The variable $\kappa^{(t)}$ is the weight of the decaying exponential moving average of past gradients (momentum) computed for each iteration as

$$\kappa^{(t)} = \kappa_0 \frac{\left(1-\frac{t}{T}\right)}{(1-\kappa_0)+\kappa_0\left(1-\frac{t}{T}\right)}\,, \tag{27}$$

where κ_0 is its initial value and T is the maximum number of iterations. The optimization step size for the r-th component of vector $\boldsymbol{\beta}$ is computed as

$$\boldsymbol{\Delta}^{(t)}_{r,r} = \frac{\eta}{\sqrt{\boldsymbol{\varepsilon}_r^{(t+1)}} + \delta} \tag{28}$$

where η is the nominal step size, $\delta > 0$ is a small constant, and

$$\boldsymbol{\varepsilon}^{(t+1)} = \kappa' \boldsymbol{\varepsilon}^{(t)} + \left(1-\kappa'\right)\left(\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{\beta}}^{(t)} \circ \frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{\beta}}^{(t)}\right)^T \tag{29}$$

acts as a diagonal preconditioning matrix. The parameter κ' is the discount factor and $\circ$ denotes the Hadamard product (element-wise multiplication).

2.4. Case setup

The framework was developed in *foam-extend* 4.0, a version of *OpenFOAM* with the option of coupled solvers [40]. The proposed method was tested for flows over periodic hills, for which numerous numerical and experimental reference data are available at various Reynolds numbers. Although it is a rather simple 2D geometry (see Fig. 3), it features separation and reattachment which are challenging to predict for eddy viscosity models. Here it is of particular interest to obtain accurate wall shear stress profiles through data assimilation. The mean flow direction is from left to right, applying periodic boundary conditions with an effective pressure drop from the inflow to the outflow boundary. The latter is determined such that a specific mean bulk velocity U_b is maintained. No-slip boundary conditions are applied at the top and bottom walls. A flow with a Reynolds number of $Re = 10595$ (based on the hill crest height H and the bulk velocity U_b at the inflow boundary) is considered, and averaged LES data from Gloerfelt and Cinnella [46] is used as reference data.

The computational grid consists of 180×130 cells in the streamwise (x) and spanwise (y) directions, respectively. A dimensionless wall distance of $y^+ \approx 1$–5 was maintained for the wall-adjacent cell centers. The baseline eddy viscosity was computed using the standard k-ϵ model with default coefficient values. Due to the wide range of eddy viscosity values, ranging from very tiny values close to walls to larger values in the free shear flow region, a realistic initialization of the eddy viscosity is very important to avoid that the gradient-based optimizer gets stuck in local minima.

The eddy viscosity was set to zero at the wall boundaries (low Reynolds boundary condition). Second-order schemes were used to discretize the convective and diffusive terms, and a bi-conjugate gradient stabilized linear solver with a Cholesky preconditioner was used to solve the coupled system.

For data assimilation, 15×11 measurement points were considered as the reference data, and together with the ghost points, the length of the vector $\boldsymbol{\beta}$ is $17 \times 13 = 221$. The length of vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}'$ are equal to the number of grid cells, which is $180 \times 130 = 23400$. This means that PLDR reduces the number of tuning parameters by more than 99%. The hyperparameter values that led to the fastest convergence in this case study are summarized in Table 1. The hyperparameters of the L_2-, TV-, and PLDR-TV-regularization methods are chosen based on trial and error. Too large values would result in overly smooth results with little variations in the eddy viscosity field. On the other hand, small values lead to better agreement with measurements, but cannot eliminate jagged profiles. For Sobolev gradient-regularization, the value should be chosen close to the distance of nodal points such that variations below this distance are dampened. One should note that such a process is inevitable when dealing with the ambiguity of solving such an inverse problem.

It should be noted that increasing the number of iterations T (cf. Table 1), say from 500 to 1000, does not significantly increase the overall computational cost, since α changes only slightly between optimization steps after an initial phase, and thus it is cheap to obtain the next forward solution from the previous step.

The verification of gradient computations $\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{\alpha}}$ is presented in [4], where it was shown that the approximations have a negligible effect on the data assimilation process. Since the computation of $\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{\beta}}$ is only a multiplication of $\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{\alpha}}$ with $\frac{\mathrm{d}\boldsymbol{\alpha}}{\mathrm{d}\boldsymbol{\beta}}$, which

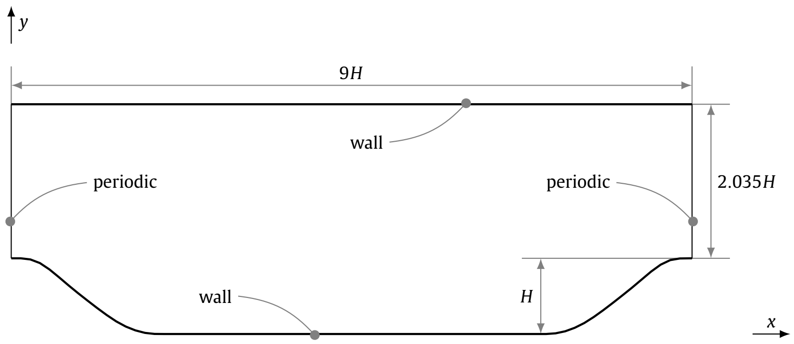


Fig. 3. Periodic hill geometry with boundary conditions. All length scales are expressed with respect to the hill height H.

Table 1
Hyperparameter values for regularization and optimization methods.

Hyperparameter values			
Regularization		Optimization	
ω_{L2}	$1e-12$	κ_0	0.900
ω_{TV}	$5e-12$	κ'	0.999
l_{sob}	$5e-1$	η	0.010
$\omega_{TV,\beta}$	$1e-13$	T	1000

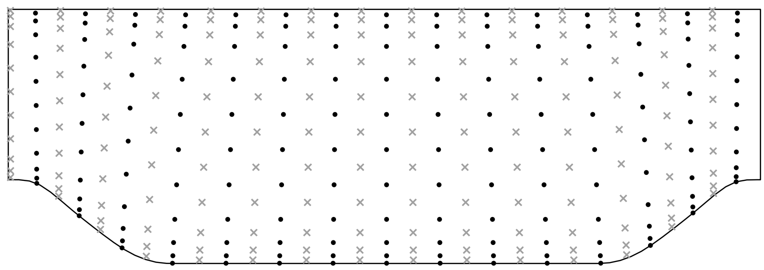


Fig. 4. Distributions of measurement (black dots) and test data (gray crosses) locations. The test data are not used for data assimilation, but to evaluate the generalization error.

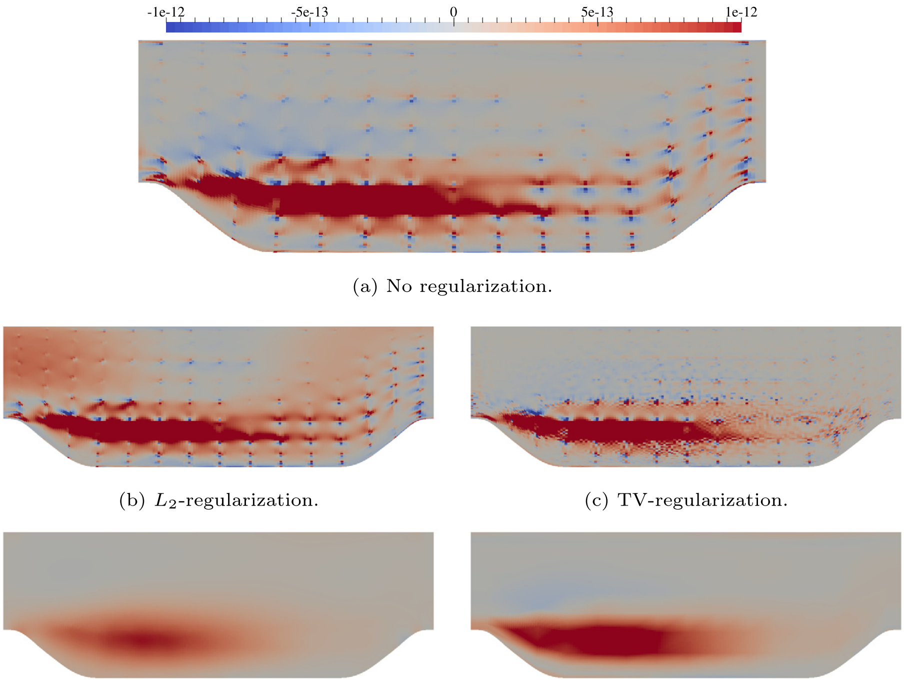
is equal to the basis functions and are exact, the computed $\frac{\mathrm{d}f}{\mathrm{d}\boldsymbol{\beta}}$ values are accurate enough for the present data assimilation problem.

3. Results and discussion

In this section, results obtained with the proposed PLDR-regularization are presented and compared to those obtained with more conventional methods. In particular, streamwise velocity and wall shear stress profiles are of interest here [4]. Fig. 4 shows the staggered spatial distributions of measurement and test locations, respectively. The choice of measurement and test data is arbitrary, since the data is sampled from a full LES simulation. The proper choice of training and test data is an important task that is not in the scope of this work.

3.1. Regularization effect on a single optimization step

Before presenting the results obtained with PLDR and the other regularization methods, the effect of each regularization method on the sensitivities $\frac{\mathrm{d}f}{\mathrm{d}\alpha}$ for a single initial optimization step are discussed. The L_2-, TV-, and Sobolev gradient-regularization act directly on $\frac{\mathrm{d}f}{\mathrm{d}\alpha}$, while PLDR acts indirectly. To compare the PLDR sensitivity $\frac{\mathrm{d}f}{\mathrm{d}\beta}$ with those of the other methods, it is projected to $\frac{\mathrm{d}f}{\mathrm{d}\alpha}$ by multiplying by the linear projection matrix. Fig. 5 shows the corresponding results. It can be seen that PLDR is the most successful method for removing noise and peaks at the measurement points (cf. Fig. 4), especially near the walls. While regularization should eliminate undesired peaks in the parameter and solution fields, excessive smoothing has to be avoided. The effect of penalty-based regularization such as TV- and L_2-regularization is gradual during optimization and cannot be compared with the other two methods. However, comparing PLDR- with Sobolev gradient-regularization shows that PLDR is able to remove the undesired noise and peaks while preserving the magnitudes



(a) No regularization.

(b) L_2-regularization.

(c) TV-regularization.

(d) Sobolev gradient-regularization.

(e) PLDR-regularization.

Fig. 5. Sensitivity $\frac{\mathrm{d}f}{\mathrm{d}\alpha}$ maps after an intermediate optimization step without (a) and with (b-e) regularization. With PLDR the sensitivities are obtained by projecting $\frac{\mathrm{d}f}{\mathrm{d}\beta}$ using the projection matrix.

of the sensitivity field in the wake and near the walls. Sobolev gradient-regularization also removes the undesired noise and peaks, but tends to excessively smooth the sensitivity field.

3.2. PLDR-regularization

First, the data-driven simulation results without regularization are examined and then discussed along with those resulting from PLDR-regularization. Fig. 6 shows the corresponding horizontal velocity profiles and the wall shear stresses at the bottom wall. Without regularization, the velocity profiles and wall shear stresses exhibit unphysical wiggles and spikes compared to the averaged LES reference. While the solution is quite accurate at the measurement points (cf. Fig. 4), it severely deviates at some other locations. This is also reflected in Table 2, where for the unregularized results the training error (discrepancy at the measurement locations) is very low, but the generalization error (discrepancy at the test locations) is high.

PLDR produces results that are smoother and in better agreement with the reference profiles. The improvement is more pronounced for the wall shear stresses (cf. Fig. 7, bottom). For example, the spikes completely disappeared and the reattachment length is closer to the reference value. However, an oscillation appears near the first hill that is not present in the reference profile. In the velocity profiles, a substantial improvement is observed in all regions. Overpredictions are observed near the upper wall and slight deviations are seen near the crest of the hill. However, the oscillations in the wall shear stress profile and the deviations in the velocity profiles motivate further improvements in the PLDR method, which are discussed below.

3.3. PLDR-TV-regularization

Looking at the velocities obtained with the PLDR-regularization (Fig. 7), deviations can be seen near the top wall, which could originate from the high gradients of α in those regions. To reduce these discrepancies, a concept similar to TV-regularization is applied to the β field. This leads to improved predictions of the quantities of interest as shown below. The modified cost function reads

$$f = \hat{\boldsymbol{V}} \cdot \left(\mathbf{H}\boldsymbol{u}_1 - \hat{\boldsymbol{u}}_1\right)^{\circ 2} + \omega_{TV,\beta} \sum_{r=1}^{R} \left[\frac{1}{|\mathcal{B}_r|} \sum_{l \in \mathcal{B}_r} (\beta_r - \beta_l)^2 \right] , \tag{30}$$

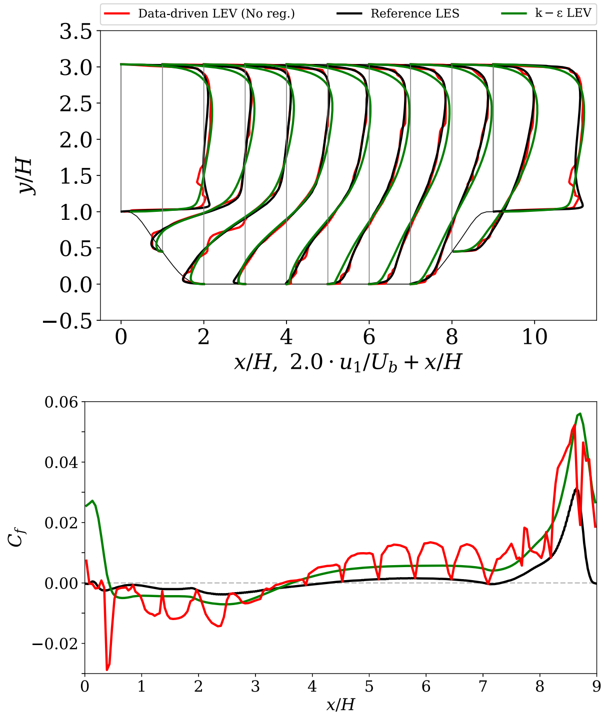


Fig. 6. Horizontal velocity component u_1 profiles (top) and wall shear stress (friction coefficient C_f) profiles at the lower wall (bottom plot). Results of the baseline k-ε model (green), the unregularized data-driven method (red), and the averaged LES reference (black) are shown.

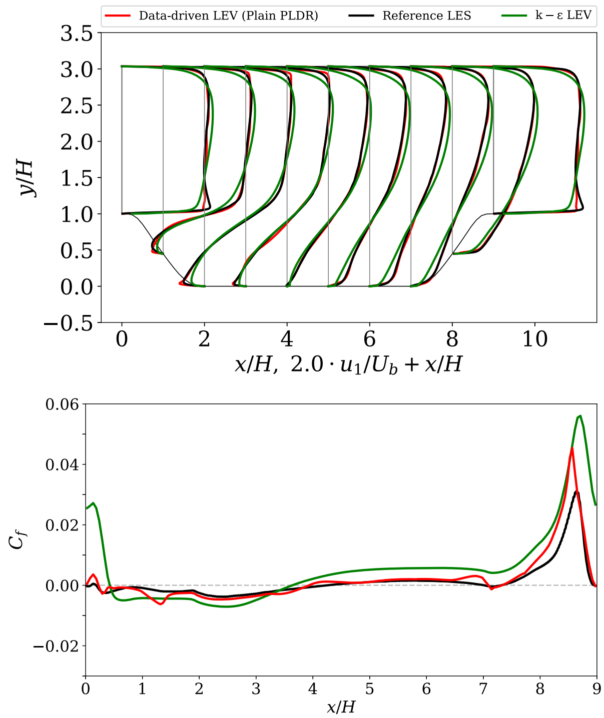


Fig. 7. Horizontal velocity component u_1 profiles (top) and wall shear stress (friction coefficient C_f) profiles at the lower wall (bottom plot). Results of the baseline k-ε model (green), the data-driven method with PLDR-regularization (red), and the averaged LES reference (black) are shown.

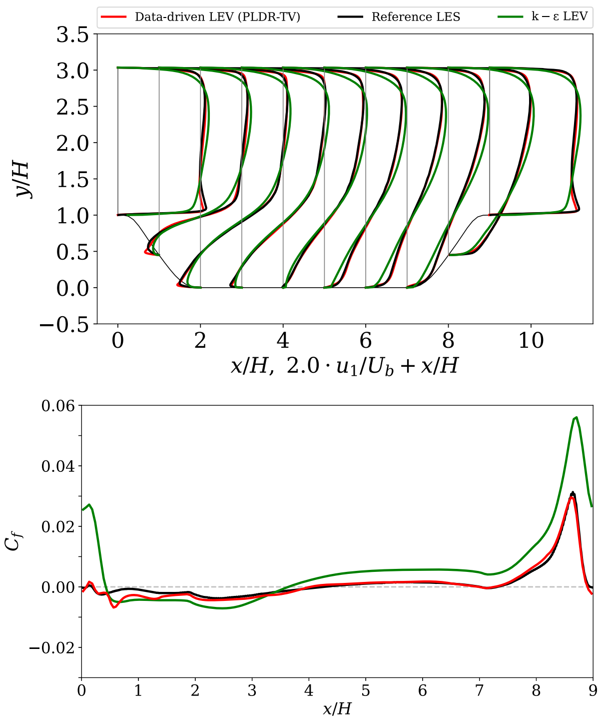


Fig. 8. Horizontal velocity component u_1 profiles (top) and wall shear stress (friction coefficient C_f) profiles at the lower wall (bottom plot). Results of the baseline k-ε model (green), the data-driven method with PLDR-TV-regularization (red), and the averaged LES reference (black) are shown.

where $\omega_{TV,\beta}$ is the weight factor of the β-TV-regularization, $\mathcal{B}_r$ the set of coarse neighborhood nodes, and l denotes the index of the adjacent coarse nodes.

Moreover, the estimation of the wall shear stress can be improved by adding additional weights c_w to the discrepancy function near the wall, i.e.,

$$f = \left(\boldsymbol{c_w} \circ \hat{\boldsymbol{V}}\right) \cdot \left(\mathbf{H}\boldsymbol{u}_1 - \hat{\boldsymbol{u}}_1\right)^{\circ 2} + \omega_{TV,\beta} \sum_{r=1}^{R} \left[\frac{1}{|\mathcal{B}_r|} \sum_{l \in \mathcal{B}_r} (\beta_r - \beta_l)^2 \right]. \tag{31}$$

In the present case, the weight factor vector $\boldsymbol{c_w}$ was set to 10 at reference locations near the wall and to 1 otherwise. This combination of PLDR- and TV-regularization is termed PLDR-TV-regularization.

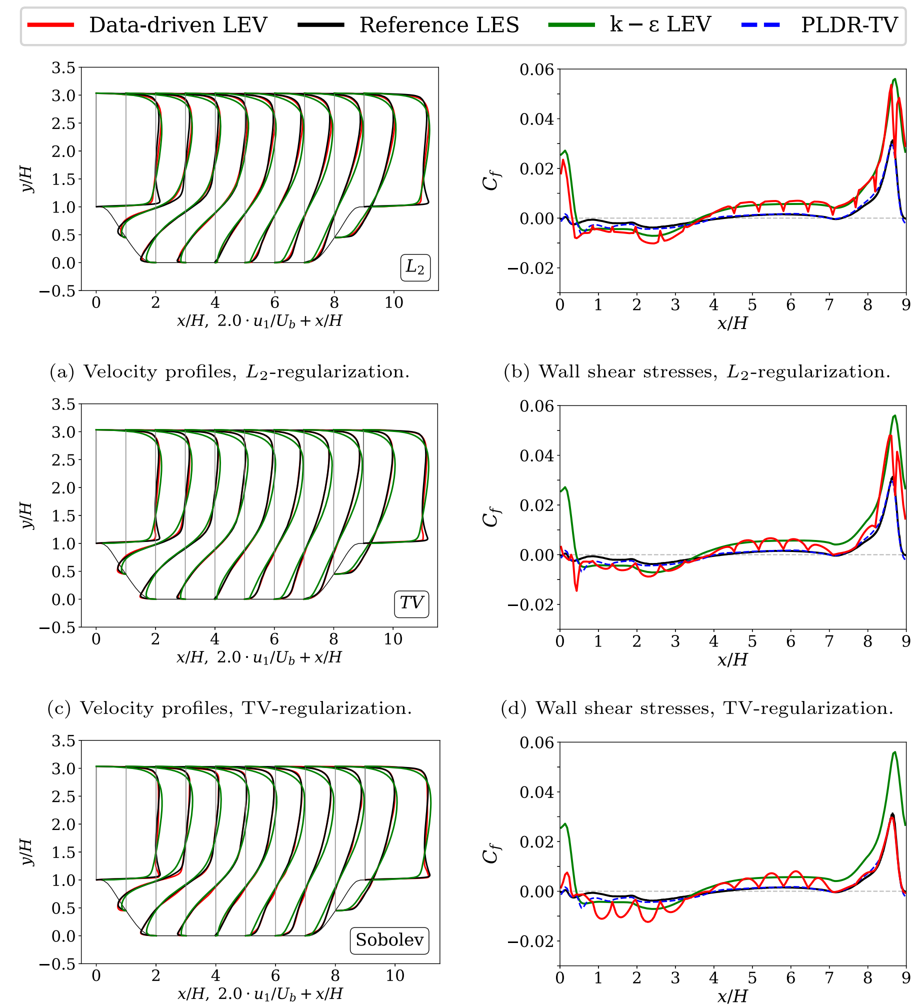
Fig. 8 shows that the above modifications reduce the discrepancies in both the optimized velocity and wall shear stress profiles. Both quantities show a small undershoot near the inlet, which may be explained by the absence of reference points in that region. Note that the measurement locations here are somewhat arbitrary and there exists room for improvement in this regard.

3.4. Comparison with other regularization methods

The significance of PLDR-TV-regularization in generating sound profiles is highlighted by comparing it with conventional regularization methods. Here, PLDR-TV is compared with L_2, Total Variation, and Sobolev gradient regularization. Fig. 9 shows the horizontal velocity and wall shear stress profiles resulting from data assimilation with these three other regularization approaches.

Regarding the horizontal velocity profiles, all regularization methods were able to prevent spiky profiles, while L_2-regularization method shows more inaccuracies in the hill wake region. To examine the methods more quantitatively, the training errors (discrepancy at the measurement point) and the generalization errors (discrepancy at testing points) are reported in Table 2. The training error is one to two orders of magnitude smaller when data assimilation is applied. The smallest training error is observed for data-driven LEV-RANS simulations without regularization, but the trade-off is a very large generalization error. Among the other three regularization methods, TV-regularization exhibits the lowest generalization error, while DA without regularization exhibits the highest. Regarding both types of errors, PLDR-TV-regularization proved to be quite successful (cf. Fig. 8).

It should be noted that the optimal values for the regularization hyperparameters were determined by trial and error. While there exists an optimal value for TV-regularization, where both the cost function and the generalization errors are



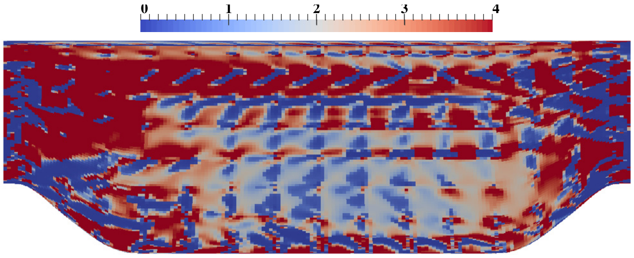
(a) Velocity profiles, L_2-regularization.
(b) Wall shear stresses, L_2-regularization.
(c) Velocity profiles, TV-regularization.
(d) Wall shear stresses, TV-regularization.
(e) Velocity profiles, Sobolev gradient-regularization.
(f) Wall shear stresses, Sobolev gradient-regularization.

Fig. 9. Horizontal velocity component u_1 profiles (left plots) and wall shear stress (friction coefficient C_f) profiles at the lower wall (right plots). Results of the baseline k-ε model (green), the data-driven method with PLDR-TV-regularization (blue), the averaged LES reference (black), and either L_2 (top), TV (middle) or Sobolev gradient-regularization (bottom) are shown.

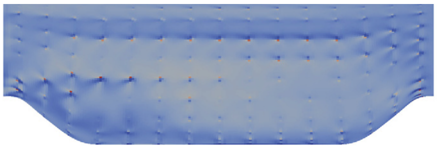
small, no optimal hyperparameter value could be found for L_2- and Sobolev gradient-regularization. Small parameter values resulted in noisy and spiky solutions, while large values resulted in excessively smooth solutions.

Regarding the wall shear stresses, it can be seen that, with the exception of PLDR-TV-regularization, all methods are unable to prevent spikes. Table 2 shows that PLDR-TV-regularization is particularly effective in reconstructing wall shear stresses. PLDR-TV- and TV-regularization result in the lowest C_f training errors, and no regularization and L_2-regularization lead to the highest generalization errors. TV-regularization, which performs best for the internal velocity field, does not perform well in estimating wall shear stresses, and thus the skin friction drag (with an error of around 200 percent). The corresponding error for the Sobolev gradient-regularization is only 8.74 percent, mainly because the peak of the wall shear stress near the right boundary is more accurately captured. PLDR-TV-regularization results in a skin friction drag error of only 1.38 percent, which is impressive considering the limitations of linear eddy viscosity models.

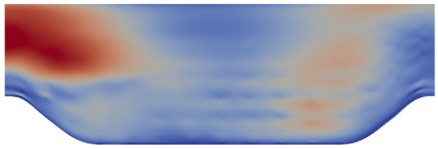
Moreover, PLDR-TV-regularization results in a smooth and accurate wall shear stress profile, unlike the spiky and unphysical profiles obtained with Sobolev gradient-regularization. Fig. 10 displays the optimized α parameter fields obtained with different regularization methods and without regularization. As can be seen, the smoothing effects of the Sobolev gradient- and L_2-regularization methods are not sufficient to remove the spikes at the near-wall measurement locations. TV-regularization results in less noisy but highly smoothed α fields. PLDR-TV-regularization, on the other hand, proved to be successful in removing all spikes in the near-wall region, mainly due to the constraint that the parameter variations are piecewise linear.



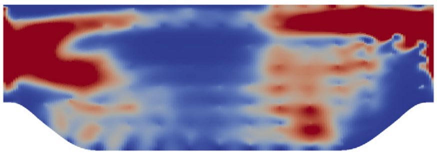
(a) No regularization; data range [0.0, 85.6].



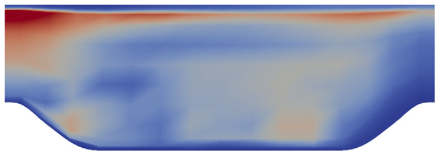
(b) L_2-regularization; data range [0.003, 8.94].



(c) TV-regularization; data range [0.001, 4.2].



(d) Sobolev gradient-regularization; data range [0.0, 11.2].



(e) PLDR-TV-regularization; data range [0.0, 5.1].

Fig. 10. Optimal α fields without (a) and with (b-e) regularization. The same color bar scaling is used for all plots.

Table 2
Summary of the errors for different regularization methods. The training error is the final cost function value f normalized by the number of measurement points. The generalization error is the final cost function value at the test points (f_{test}) normalized by the number of test points. The C_f training error is the mean absolute error of C_f at wall-adjacent measurement points (bottom wall). The drag error (skin friction drag error) is the relative error with respect to the skin friction drag of LES data for the bottom wall.

Method	Errors			
	Training	Generalization	C_f training	Drag [%]
k-ϵ model	$5.04e-08$	$4.45e-08$	$5.12e-06$	454.87
Data-driven, no reg.	$1.17e-10$	$1.47e-08$	$1.60e-07$	404.65
Data-driven, L_2	$3.75e-09$	$8.12e-09$	$1.99e-06$	356.87
Data-driven, TV	$6.89e-10$	$2.60e-09$	$1.38e-06$	200.06
Data-driven, Sobolev	$3.68e-10$	$8.01e-09$	$1.98e-06$	8.74
Data-driven, PLDR-TV	$3.06e-09$	$6.26e-09$	$1.39e-06$	1.38

Other scenarios with different distributions and numbers of measurement points were considered in this study. In principle, the PLDR nodes can be chosen independently, but it was found that the most accurate results are obtained when the linear element nodes coincide with the reference data locations.

Finally, the computational aspects of the methods are compared. In general, for all methods, most of the computational costs are caused by solving the forward problem. This is because the Jacobian matrix for adjoint equations is constructed in this step and the task of the adjoint equation solver is merely to solve the already constructed algebraic equations. The additional computational cost of the TV-regularization, L_2-regularization, and PLDR-TV-regularization methods is negligible, since no iterative process is involved. For Sobolev gradient-regularization, the additional cost of solving the Poisson equation is of the same order as the forward problem, but still smaller. Assuming that the computational cost for each optimization iteration is almost the same for all methods, the convergence rate of the methods can be compared. Fig. 11 compares the evolution of the cost function and test function (the discrepancy at the test points). As can be seen, the convergence rate of the cost function for the case without regularization and with Sobolev gradient-regularization is slower. The other methods achieve their final values after less than 200 iterations. All methods reach flat values due to the effect of the decaying momentum term in the DEMON-Adam optimization method.

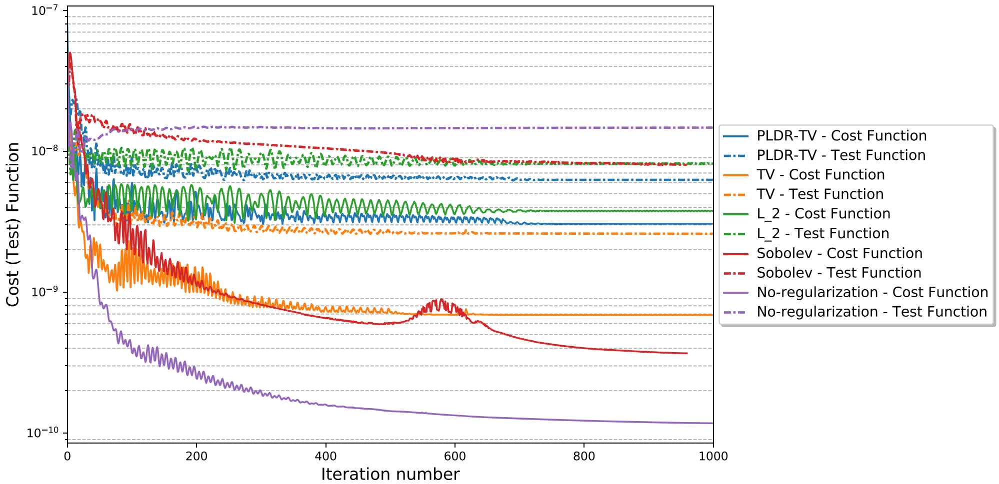


Fig. 11. Cost (test) function evolution of different data-driven LEV models with and without regularization methods.

4. Conclusions and outlook

A variational data-driven RANS simulation framework was developed based on solving adjoint problems and a novel regularization approach to obtain solutions consistent with sparsely distributed measurements within the simulation domain, allowing accurate interpolation between these measurement locations. The RANS equations are closed by the linear eddy viscosity model, and the data assimilation task was to correct the modeled eddy viscosity by a spatially varying correction factor. It is obvious that regularization is needed to avoid unphysical peaks in the obtained velocity fields due to the ambiguity of such inverse problems, but all tested classical regularization approaches could not remove all peaks satisfactorily, especially in the near-wall regions.

The objective of the new PLDR-TV-regularization method is to constrain the corrective parameter by a piecewise linear variation to reduce ambiguity and avoid unphysical spikes in the solution field. As shown, this regularization approach not only provides accurate velocity fields in the free shear flow region, but is also suitable for accurate wall shear stress predictions; in contrast to e.g. TV or Sobolev gradient regularization.

An open question regarding PLDR-TV-regularization is the placement of the degrees of freedom, which in our study coincide with the reference points. In principle, they can be chosen arbitrarily, but it is important that the local resolution can account for strong spatial variations, while it still has to provide smoothing. A more systematic approach to ensure this is the subject of future research. Another future task is to extend the current PLDR framework for 3D cases where tetrahedral rather than triangular elements will be used. In addition, advanced techniques in machine learning for automatically tuning the user-defined hyperparameters of regularization or optimization methods can be investigated for such a data assimilation problem [47,48]. Optimal placement of measurement points and proper decisions about training and test sets would be another attractive research topic [49]

List of symbols

Abbreviations

Re	Reynolds number
AD	automatic differentiation
DA	data assimilation
LEV	linear eddy viscosity
PLDR	piecewise linear dimension reduction
RANS	Reynolds–averaged Navier-Stokes
TV	total variation

Greek letters

α	intermediatory corrective parameter
α^m	model parameter
$\boldsymbol{\beta}$	reduced dimension corrective parameter
$\boldsymbol{\Phi}$	piecewise linear projection matrix
Δ	learning rate or the optimization step size

δ	very small number for DEMON Adam
δ_{ij}	Kronecker delta, second-order identity tensor
ϵ	turbulent dissipation rate
η	the nominal step size in DEMON Adam
γ	DEMON Adam variable
κ	weight of the decaying momentum in DEMON Adam
κ'	discount factor in DEMON Adam
κ_0	the momentum initial value in DEMON Adam
λ	adjoint variables
ν	kinematic viscosity
ν_t	turbulent eddy viscosity
ω_{L2}	the regularization hyperparameter of L_2 method
$\omega_{TV,\beta}$	the regularization hyperparameter of TV for β parameter
ω_{TV}	the regularization hyperparameter of TV method
τ_{ij}	Reynolds stress tensor
ε	diagonal preconditioning matrix in DEMON Adam

Roman letters

$\bar{k}$	turbulent kinetic energy
$\bar{S}_{ij}$	mean flow strain rate
$\hat{\boldsymbol{V}}$	the vector of grid cells' volume
$\mathbf{H}$	Linear observation operator
$\boldsymbol{A}$	block system matrix of the residual equations
$\boldsymbol{b}$	the right hand side of the residual equations
$\boldsymbol{c_w}$	weight factor for wall-adjacent measurement
$\boldsymbol{J}$	matrix of ones
$\boldsymbol{R}$	vector of flow residuals
$\mathcal{B}_m$	the set of neighboring cells of the grid cell m
$\mathcal{B}_r$	the set of neighboring nodes of the measurement point m
$\mathcal{I}$	the identity matrix
C_μ	RANS model coefficient
C_f	skin friction coefficient
f	cost function
H^1	Sobolev space
i, j, k, l, m, o, r, t	indices
l_{sob}	the filtering length scale of the Sobolev gradient regularization method
M	number of the parameters
N	the size of the computational grid
p	pressure
R	number of measurement points
t	the iteration number of the optimization
T_{max}	the maximum number of the optimization iterations
U	state variables
U_b	bulk velocity
u_i	velocity
u'_i	velocity fluctuation
x_i	spatial coordinates
y^+	dimensionless wall distance

CRediT authorship contribution statement

Pasha Piroozmand: Conceptualization, Formal analysis, Methodology, Software, Writing – original draft. **Oliver Brenner:** Conceptualization, Formal analysis, Methodology, Writing – review & editing. **Patrick Jenny:** Conceptualization, Funding acquisition, Methodology, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

The authors would like to thank Daniel Oppliger for an initial version of the discrete adjoint implementation and Heng Xiao and Daniel W. Meyer-Massetti for valuable discussions. We also thank Francesca Ferrara who was involved in parts of this work in the context of her thesis project. Financial support from ETH Zurich is gratefully acknowledged.

References

[1] C. Tropea, A.L. Yarin, J.F. Foss (Eds.), Springer Handbook of Experimental Fluid Mechanics, vol. 1, Springer, 2007.
[2] H. Xiao, P. Jenny, A consistent dual-mesh framework for hybrid LES/RANS modeling, J. Comput. Phys. 231 (4) (2012) 1848–1865, https://doi.org/10.1016/J.JCP.2011.11.009.
[3] J.-L. Wu, J.-X. Wang, H. Xiao, J. Ling, A priori assessment of prediction confidence for data-driven turbulence modeling, Flow Turbul. Combust. 99 (1) (2017) 25–46, https://doi.org/10.1007/s10494-017-9807-0.
[4] O. Brenner, P. Piroozmand, P. Jenny, Efficient assimilation of sparse data into RANS-based turbulent flow simulations using a discrete adjoint method, J. Comput. Phys. 471 (2022) 111667.
[5] L. Franceschini, D. Sipp, O. Marquet, Mean-flow data assimilation based on minimal correction of turbulence models: application to turbulent high Reynolds number backward-facing step, Phys. Rev. Fluids 5 (9) (2020) 094603, https://doi.org/10.1103/PhysRevFluids.5.094603.
[6] C. He, P. Wang, Y. Liu, Data assimilation for turbulent mean flow and scalar fields with anisotropic formulation, Exp. Fluids 62 (5) (2021) 1–25.
[7] S. Li, C. He, Y. Liu, A data assimilation model for wall pressure-driven mean flow reconstruction, Phys. Fluids 34 (1) (2022) 015101.
[8] W. Liu, J. Fang, S. Rolfo, C. Moulinec, D.R. Emerson, An iterative machine-learning framework for RANS turbulence modeling, Int. J. Heat Fluid Flow 90 (2021) 108822, https://doi.org/10.1016/j.ijheatfluidflow.2021.108822.
[9] T.J. Craft, B.E. Launder, K. Suga, Development and application of a cubic eddy-viscosity model of turbulence, Int. J. Heat Fluid Flow 17 (2) (1996) 108–115, https://doi.org/10.1016/0142-727x(95)00079-6.
[10] T.B. Gatski, C.G. Speziale, On explicit algebraic stress models for complex turbulent flows, J. Fluid Mech. 254 (1993) 59–78, https://doi.org/10.1017/S0022112093002034.
[11] K. Hanjalić, Advanced turbulence closure models: a view of current status and future prospects, Int. J. Heat Fluid Flow 15 (3) (1994) 178–203, https://doi.org/10.1016/0142-727X(94)90038-8.
[12] J. Wu, H. Xiao, R. Sun, Q. Wang, Reynolds-averaged Navier–Stokes equations with explicit data-driven Reynolds stress closure can be ill-conditioned, J. Fluid Mech. 869 (2019) 553–586, https://doi.org/10.1017/jfm.2019.205.
[13] W.P. Jones, B.E. Launder, The prediction of laminarization with a two-equation model of turbulence, Int. J. Heat Mass Transf. 15 (2) (1972) 301–314, https://doi.org/10.1016/0017-9310(72)90076-2.
[14] D.C. Wilcox, Turbulence Modeling for CFD, vol. 2, DCW Industries, La Canada, CA, 1998.
[15] P. Spalart, S. Allmaras, A one-equation turbulence model for aerodynamic flows, in: 30th Aerospace Sciences Meeting and Exhibit, American Institute of Aeronautics and Astronautics, 1992, p. 439.
[16] F.R. Menter, Two-equation eddy-viscosity turbulence models for engineering applications, AIAA J. 32 (8) (1994) 1598–1605, https://doi.org/10.2514/3.12149.
[17] H. Xiao, J.-L. Wu, S. Laizet, L. Duan, Flows over periodic hills of parameterized geometries: a dataset for data-driven turbulence modeling from direct simulations, Comput. Fluids 200 (2020) 104431, https://doi.org/10.1016/j.compfluid.2020.104431.
[18] M. Asch, M. Bocquet, M. Nodet, Data Assimilation: Methods, Algorithms, and Applications, SIAM, 2016.
[19] A.P. Singh, A framework to improve turbulence models using full-field inversion and machine learning, Ph.D. thesis, University of Michigan, Ann Arbor, 2018.
[20] E. Hou, E. Lawrence, A.O. Hero, Penalized ensemble Kalman filters for high dimensional non-linear systems, PLoS ONE 16 (3) (2021) e0248046.
[21] M. Zauner, V. Mons, O. Marquet, B. Leclaire, Nudging-based data assimilation of the turbulent flow around a square cylinder, J. Fluid Mech. 937 (2022) A38, https://doi.org/10.1017/jfm.2022.133.
[22] D.C. Liu, J. Nocedal, On the limited memory BFGS method for large scale optimization, Math. Program. 45 (1) (1989) 503–528, https://doi.org/10.1007/BF01589116.
[23] R. Epp, F. Schmid, P. Jenny, Fast convergence strategy for ambiguous inverse problems based on hierarchical regularization, J. Comput. Phys. (2023) 112264.
[24] G.K.W. Kenway, C.A. Mader, P. He, J.R.R.A. Martins, Effective adjoint approaches for computational fluid dynamics, Prog. Aerosp. Sci. 110 (2019) 100542, https://doi.org/10.1016/j.paerosci.2019.05.002.
[25] D.P. Foures, N. Dovetta, D. Sipp, P.J. Schmid, A data-assimilation method for Reynolds-averaged Navier–Stokes-driven mean flow reconstruction, J. Fluid Mech. 759 (2014) 404–431.
[26] S. Symon, N. Dovetta, B.J. McKeon, D. Sipp, P.J. Schmid, Data assimilation of mean velocity from 2D PIV measurements of flow over an idealized airfoil, Exp. Fluids 58 (2017) 1–17.
[27] J.-L. Wu, R. Sun, S. Laizet, H. Xiao, Representation of stress tensor perturbations with application in machine-learning-assisted turbulence modeling, Comput. Methods Appl. Mech. Eng. 346 (2019) 707–726.
[28] B.P. Brener, M.A. Cruz, R.L. Thompson, R.P. Anjos, Conditioning and accurate solutions of Reynolds average Navier–Stokes equations with data-driven turbulence closures, J. Fluid Mech. 915 (2021), https://doi.org/10.1017/jfm.2021.148.
[29] M.Y.B. Ali, G. Tissot, D. Heitz, S. Aguinaga, E. Mémin, An adjoint approach for the analysis of RANS closure using pressure measurements on a high-rise building, preprint, https://hal.archives-ouvertes.fr/hal-03076369, Dec. 2020.
[30] W.N. Edeling, G. Iaccarino, P. Cinnella, Data-free and data-driven RANS predictions with quantified uncertainty, Flow Turbul. Combust. 100 (3) (2018) 593–616, https://doi.org/10.1007/s10494-017-9870-6.
[31] R.L. Thompson, A.A. Mishra, G. Iaccarino, W. Edeling, L. Sampaio, Eigenvector perturbation methodology for uncertainty quantification of turbulence models, Phys. Rev. Fluids 4 (4) (2019) 044603.
[32] P. Piroozmand, Sparse data-driven simulation of turbulent flows, Ph.D. thesis, ETH Zurich, 2022.
[33] A.P. Singh, S. Medida, K. Duraisamy, Machine-learning-augmented predictive modeling of turbulent separated flows over airfoils, AIAA J. 55 (7) (2017) 2215–2227, https://doi.org/10.2514/1.J055595.

[34] P. He, C.A. Mader, J.R.R.A. Martins, K. Maki, An object-oriented framework for rapid discrete adjoint development using OpenFOAM, in: AIAA Scitech 2019 Forum, American Institute of Aeronautics and Astronautics, 2019, p. 1210.
[35] J.R.R.A. Martins, P. Sturdza, J.J. Alonso, The complex-step derivative approximation, ACM Trans. Math. Softw. (TOMS) 29 (3) (2003) 245–262, https://doi.org/10.1145/838250.838251.
[36] N.R. Gauger, A. Walther, C. Moldenhauer, M. Widhalm, Automatic differentiation of an entire design chain for aerodynamic shape optimization, in: New Results in Numerical and Experimental Fluid Mechanics VI, Springer, 2007, pp. 454–461.
[37] P. Gomes, R. Palacios, Pitfalls of discrete adjoint fixed-points based on algorithmic differentiation, AIAA J. 60 (2) (2021) 1–6, https://doi.org/10.2514/1.J060735.
[38] B. Fleischli, L. Mangani, A. Del Rio, E. Casartelli, A discrete adjoint method for pressure-based algorithms, Comput. Fluids 227 (2021) 105037, https://doi.org/10.1016/j.compfluid.2021.105037.
[39] I. Clifford, Block-coupled simulations using OpenFOAM, Presentation, 2011, pp. 1–47.
[40] K. Jareteg, Block coupled calculations in OpenFOAM, Project within course: CFD with OpenSource software, Chalmers University of Technology.
[41] Y. Huang, M.K. Ng, Y.-W. Wen, A fast total variation minimization method for image restoration, Multiscale Model. Simul. 7 (2) (2008) 774–795, https://doi.org/10.1137/070703533.
[42] P.E. Gill, E. Wong, Sequential quadratic programming methods, in: Mixed Integer Nonlinear Programming, Springer, 2011, pp. 147–224.
[43] M.A. Bouhlel, J.R. Martins, Gradient-enhanced Kriging for high-dimensional problems, Eng. Comput. 35 (1) (2019) 157–173.
[44] J. Chen, A. Kyrillidis, Decaying momentum helps neural network training, https://openreview.net/forum?id=rJeA_aVtPB, Dec. 2019. (Accessed 1 March 2022).
[45] J. Chen, An updated overview of recent gradient descent algorithms, https://johnchenresearch.github.io/demon/, Feb. 2020. (Accessed 1 March 2022).
[46] X. Gloerfelt, P. Cinnella, Large eddy simulation requirements for the flow over periodic hills, Flow Turbul. Combust. 103 (1) (2019) 55–91, https://doi.org/10.1007/s10494-018-0005-5.
[47] B. Sun, D. Wu, M. Shang, Y. He, Toward auto-learning hyperparameters for deep learning-based recommender systems, in: Database Systems for Advanced Applications: 27th International Conference, DASFAA 2022, Virtual Event, April 11–14, 2022, Proceedings, Part II, Springer, 2022, pp. 323–331.
[48] A.H. Victoria, G. Maragatham, Automatic tuning of hyperparameters using Bayesian optimization, Evol. Syst. 12 (2021) 217–223.
[49] V. Mons, O. Marquet, Linear and nonlinear sensor placement strategies for mean-flow reconstruction via data assimilation, J. Fluid Mech. 923 (2021) A1.