Structural and Multidisciplinary Optimization (2022) 65:49
https://doi.org/10.1007/s00158-021-03150-4

RESEARCH PAPER

Topology optimisation of turbulent flow using data-driven modelling

James Hammond[1] · Marco Pietropaoli[1] · Francesco Montomoli[1]

Received: 13 May 2021 / Revised: 24 November 2021 / Accepted: 9 December 2021 / Published online: 18 January 2022

Abstract
Fluid topology optimisation has become a popular approach for optimisation of geometries in aero-thermal applications. However, one of the main limitations of current approaches considering turbulent flow is the fidelity of the Reynolds Averaged Navier–Stokes models employed. In response, this paper shows the development of the first data-driven fluid topology optimisation technique based on the continuous adjoint method. The technique first extracts data from a high fidelity simulation of a standard topology-optimised geometry. These data are fed through a symbolic regression-based machine learning algorithm called gene expression programming, to learn an explicit model for the anisotropy tensor. The novel aspect of the work is the derivation of the adjoint form of the generalised explicit algebraic stress model such that the developed turbulence model can be inserted directly into the primal and adjoint system of equations. This allows a second, data-driven optimisation to be performed. Finally, a high fidelity simulation of the resulting geometry is also conducted to allow comparison against the standard geometry. The method is first applied to the back-facing step to verify the approach and then to a 2D u-bend configuration. The data-driven optimisation was able to find geometries exhibiting significant reductions in pressure loss when compared with results from the standard optimisation.

Keywords Topology optimisation · Continuous adjoint · Data-driven · Machine learning · Turbulence modelling

1 Introduction

Fluid topology optimisation (FTO) is fast gaining traction within multiple industrial sectors including aerospace and automotive for use in the design of parts for additive manufacturing. The technique was originally introduced to structural mechanics by Bendsøe and Kikuchi (1988), who used a homogenisation method in terms of material density to identify areas in a domain where material should be added to minimise compliance. The idea was not introduced to fluids-based problems until Borrvall and Petersson (2003), who applied topology optimisation to Stokes flow by introducing a permeability field to the governing equations.

In general, permeability-based FTO starts with an initial guess for the permeability field and modifies this guess across the domain to optimise some flow-based objective function such as pressure loss. A variety of different gradient and non-gradient optimisation techniques can be used to solve these optimisation problems. Amongst the most common, and the approach adopted here is the adjoint method (either continuous or discrete). It is a gradient-based technique which operates by extracting topological sensitivities within a predefined design domain (Othmer 2008). These sensitivities are equivalent to the gradient of the objective function with respect to the chosen design variables. The adjoint method is popular in topology optimisation as it requires the evaluation of just two sets of partial differential equations (primal and adjoint) to compute these sensitivities (Giles and Pierce 2000).

Calculated sensitivities give information on favourable and counterproductive cells with respect to the chosen objective function. Favourable cells are then rewarded by increased permeability and counterproductive cells penalised by reducing their permeability. Over several iterations, the optimised fluid path is revealed as the regions in which sensitivities remained favourable and the permeability stayed high enough for the region to be considered fluid (Pietropaoli et al. 2016). The corresponding optimised solid geometry is taken as the opposing regions in

Responsible Editor: Emilio Carlos Nelli Silva

✉ James Hammond
j.hammond18@imperial.ac.uk

[1] Department of Aeronautics, Imperial College London, London, UK

which the permeability has been reduced to zero. Therefore, FTO provides a method of generating optimised geometries for associated fluid flows, where the design surface does not require a priori specification.

Currently, treatment of turbulence in FTO solvers remains limited by Reynolds Averaged Navier–Stokes (RANS) methods with linear constitutive laws (Yoon 2016; Dilgen et al. 2018b). Many recent works do not consider the formulation for turbulent flows and focus instead on laminar cases, limiting the applicability of methods to low Reynolds number situations (Alexandersen and Andreasen 2020). In many aerospace applications, this assumption is not valid and in the free-form geometries created by FTO, flows are often complex and turbulent. They can exhibit strong streamline curvature, rotational effects and separation, all of which are widely documented to be poorly predicted by RANS methods (Wilcox 2006; Spalart et al. 2015; Pichler et al. 2016). Unfortunately, computational expense renders higher fidelity turbulence treatment such as Direct Numerical Simulation and Large Eddy Simulation (DNS/LES) infeasible for direct use in FTO solvers.

In light of this, it can be argued that FTO solvers considering turbulent flow typically suffer on two accounts. The first is by the so called 'frozen turbulence' assumption, where variations in the turbulent viscosity field are neglected and are not reflected in the optimised geometries produced. Work has now been done to overcome this assumption and it effects are analysed in the context of topology optimisation by Kontoleontos et al. (2013) and Dilgen et al. (2018a). The former modified work by Zymaris et al. (2009), who derived a continuous adjoint form of the Spalart–Allmaras turbulence model for shape optimisation. The latter used automatic differentiation to produce discrete adjoint versions of the Spalart–Allmaras and k–ω models for FTO. The second is that on overcoming the frozen turbulence assumption, the turbulence model itself is still based on linear constitutive laws, so designs will be limited by the predictive accuracy of these laws. The focus of this work is overcoming this second limitation, by improving the predictive capabilities of the constitutive turbulence laws themselves.

Away from topology optimisation, much fluids-based research focuses on augmenting and developing RANS-based turbulence models, making use of the rapidly growing availability of high fidelity flow data. Machine learning techniques are being used in tandem with this data to develop 'data-driven' models. The aim is to help increase the predictive accuracy of RANS-based methods to be comparable to higher fidelity methods, whilst maintaining a computational cost in the region of the original RANS methods (Durbin 2018). These techniques have seen promising results especially in cases where the flows used in training and testing remain similar (Duraisamy et al. 2019).

Given the research trend and successes reported in data-driven turbulence modelling, it appears a natural solution to incorporate these ideas into the FTO framework, particularly given the inevitable similarity in training and testing flow cases. This paper outlines a newly developed technique for the introduction of data-driven turbulence models into FTO. To the authors knowledge, no work has yet been performed to tackle the solver fidelity limit in FTO. However recently, Zhang et al. (2021) developed a data-driven multi-fidelity shape optimisation method, making use of a hierarchical-Kriging surrogate framework.

Any machine learning method common to turbulence modelling may be chosen to enhance the optimisation, given a suitable implementation in the adjoint system is found. However here, as a first work and example of the approach, the data-driven aspect will rely on the Gene Expression Programming (GEP) algorithm. GEP was originally developed by Ferreira (2001) and has since been applied to turbulence problems by Weatheritt and Sandberg (2016). They combined GEP with explicit algebraic stress models (EASMs) to learn explicit functional forms for the anisotropic component of the Reynolds stress.

In this work, the general form of an EASM, originally proposed by Pope (1975), will be introduced to the FTO framework by deriving its continuous adjoint form, such that specific data-driven solutions learnt by GEP can be applied to different cases. Here, the frozen turbulence assumption has been retained to help ease the complexity of implementation and testing, and to isolate the effects of the new technique when comparing against standard adjoint methods. Two cases are then considered, both aiming to minimise total pressure loss: the first is a backward-facing step which acts as a good verification case for any turbulence based problem; the second is a 2D u-bend. This provides a more complex case to explore the potential of the method.

2 Methods

As stated previously, this work aims to break a boundary present in current state-of-the-art FTO. This boundary is present on account of using linear constitutive laws, usually realised by eddy viscosity models, to close the RANS equations. Here the turbulence model will be extended to a data-driven EASM, the general form of which will be presented along with a novel derivation of the continuous adjoint system of equations for the EASM. The method of developing specific data-driven closures from the EASM framework, by feeding high fidelity data into the GEP algorithm is also outlined.

2.1 Explicit algebraic stress model

When closing the RANS equations, the Reynolds stresses $\boldsymbol{\tau} = \overline{\boldsymbol{u}'\boldsymbol{u}'}$ must be modelled using mean flow characteristics. The Reynolds stress is often decomposed into an isotropic component described purely in terms of the turbulent kinetic energy k and an anisotropic component. This decomposition, rearranged for the anisotropic component and normalised by $2k$, is given as follows:

$$\mathbf{a} = \frac{\boldsymbol{\tau}}{2k} - \frac{1}{3}\mathbf{I}. \tag{1}$$

In linear eddy viscosity models, the anisotropy is simply modelled as proportional to the mean strain rate tensor $\mathbf{a} = -\nu_t\mathbf{S}/k$ (for incompressible flow). The introduced proportionality constant ν_t is a turbulent (eddy) viscosity which supplements the molecular viscosity to model the effects of the turbulent stresses. This linear relationship is the stem of many continuing problems with RANS modelling today (Wilcox 2006). Explicit algebraic stress models provide a natural extension from a linear stress–strain relationship to a nonlinear one. Introduced originally by Pope (1975) some 100 years later, he reasoned based on dimensional arguments that the anisotropy could be modelled purely as a function of the mean strain and rotation rate tensors, the turbulent kinetic energy, and the turbulent dissipation rate such that $\mathbf{a} = \mathbf{a}(\mathbf{S}, \boldsymbol{\Omega}, k, \epsilon)$. By doing so, the most general form for the anisotropy can be written as a 10 term sum

$$\mathbf{a} = \sum_{n=1}^{10} \beta_n \mathbf{V}^{(n)}, \tag{2}$$

where $\mathbf{V}^{(n)}$ is a set of linearly independent tensor bases discovered by means of a weak equilibrium hypothesis proposed by Rodi (1976), and application of the Cayley–Hamilton theorem. The coefficients β_n are themselves functions of a set of scalar invariants I_m which also arise from the Cayley–Hamilton theorem. Defining the anisotropy in this manner also guarantees the satisfaction of various physical constraints such as Galilean invariance (Spencer and Rivlin 1958). The tensor bases are defined in terms of the nondimensional strain $\mathbf{s} = \tau_t\mathbf{S}$ and rotation $\boldsymbol{\omega} = \tau_t\boldsymbol{\Omega}$ rate tensors where τ_t is a turbulent timescale defined as the inverse of the specific turbulent dissipation rate $\tau_t = 1/\omega$. For two-dimensional flows, which will be considered in this paper, the complete sum of ten bases reduces to just three (Weatheritt et al. 2017) and these are defined as follows:

$$\mathbf{V}^{(1)} = \mathbf{s}, \quad \mathbf{V}^{(2)} = \mathbf{s}\boldsymbol{\omega} - \boldsymbol{\omega}\mathbf{s}, \quad \mathbf{V}^{(3)} = \mathbf{s}^2 - \frac{1}{3}I_1\mathbf{I}. \tag{3}$$

The variable I_1 in the definition of $\mathbf{V}^{(3)}$ is from the set of scalar invariants of which the β_n coefficients are functions, and in 2D, these are defined as follows:

$$I_1 = \mathrm{tr}(\mathbf{s}^2), \quad I_2 = \mathrm{tr}(\boldsymbol{\omega}^2). \tag{4}$$

EASMs lend themselves well to data-driven methods as the inputs I_m, and $\mathbf{V}^{(n)}$ and target variable $\mathbf{a}$, are simple to extract from high fidelity simulation.

Once discovered by the chosen data-driven method, specific expressions for the anisotropy can be directly substituted into the RANS equations and run with little additional computational cost in comparison to a standard eddy viscosity model. Their non-linearity, however, allows the turbulent stresses to be modelled more accurately which should lead to better prediction of the mean velocity field.

2.2 Gene expression programming

In this work, the machine learning method chosen to discover specific expressions for the β_n coefficients, is gene expression programming (GEP). GEP is selected given that it is a symbolic regression technique, which naturally generates explicit functional forms. This not only reduces the tendency of machine learning algorithms to present a black-box style solution but also aids in implementation of the resulting expression into the primal and adjoint systems. However, it is highlighted again that GEP is just one possible choice from numerous machine learning options including Neural Networks and other regression techniques. An overview of its operation will be presented here following the schematic in Fig. 1, and for a full description of the algorithm, the reader is referred to Weatheritt and Sandberg (2016).

GEP is an evolutionary algorithm, and here is used to discover explicit expressions for $\mathbf{a} = \mathbf{a}(I_m, \mathbf{V}^{(n)})$, where the structure of the expressions is given by Eq. 2. First, a population of possible solutions is randomly generated within the solution space. This population is then iteratively subjected to computational imitations of natural selection and genetic operation, which result in a gradual drift of the population

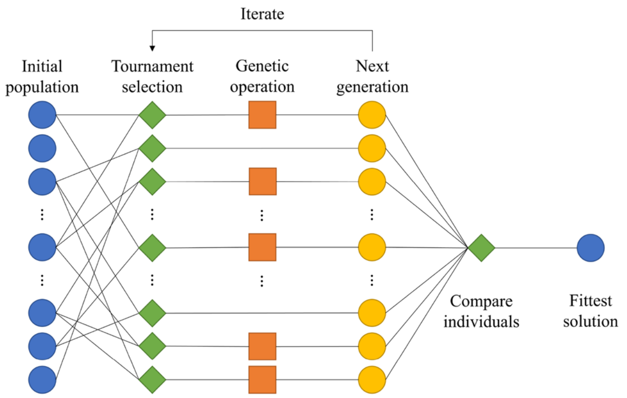

Fig. 1 Schematic of the gene expression programming (GEP) algorithm

towards more optimal solutions (Koza 1992). A large initial population is selected to span as much of the solution space as possible and avoid the tendency of optimisation algorithms to get trapped in local minima.

Natural selection is imitated by choosing m individuals with replacement out of the n individuals in the population to compete in tournaments. The fittest individual in each tournament secures a place in the next generation. Tournaments are repeated until all n places in the next generation are occupied. The surviving individuals are then stochastically subjected to genetic operations to introduce variation into the population. These operations include mutations of one symbol in an individual, and genetic crossover, where sequences of symbols from two individuals in the population are swapped.

The genetically operated population then forms the initial population for the next iteration, known in evolutionary algorithms as a generation. Generations are then progressed until either the best fitness in the population reaches some required tolerance, or a preset number of generations have evolved. The fittest member of the population at the end of this process is then taken to be the best approximation of the optimal solution.

2.3 Topology optimisation

The general form of the EASM will now be incorporated into the topology optimisation framework by deriving the corresponding adjoint system. This section details the computation of sensitivities and derives the complete system of adjoint equations required to perform data-driven topology optimisations.

2.3.1 System of primal equations

First, the primal state equations (which are the steady state incompressible RANS equations with generalised EASM, built on a baseline k–ω turbulence model) are expressed as follows:

$$R_p = -\boldsymbol{\nabla} \cdot \boldsymbol{u}, \tag{5}$$

$$\begin{aligned}\boldsymbol{R_u} = (\boldsymbol{u} \cdot \boldsymbol{\nabla})\boldsymbol{u} + \boldsymbol{\nabla} p - \boldsymbol{\nabla} \cdot (\nu_{\text{eff}}(\boldsymbol{\nabla} \boldsymbol{u} + \boldsymbol{\nabla}^{\mathrm{T}} \boldsymbol{u}) \\ - 2k\mathbf{a}^{\text{ex}}) + \alpha \boldsymbol{u},\end{aligned} \tag{6}$$

$$R_k = (\boldsymbol{u} \cdot \boldsymbol{\nabla})k - P + \beta^* \omega k - \boldsymbol{\nabla} \cdot ((\nu + \sigma_k \nu_t)\boldsymbol{\nabla} k), \tag{7}$$

$$\begin{aligned}R_\omega = (\boldsymbol{u} \cdot \boldsymbol{\nabla})\omega - \gamma \frac{\omega}{k} P + \beta \omega^2 \\ - \boldsymbol{\nabla} \cdot ((\nu + \sigma_\omega \nu_t)\boldsymbol{\nabla} \omega),\end{aligned} \tag{8}$$

where $P = -\boldsymbol{\tau} : \boldsymbol{\nabla} \boldsymbol{u}$ is the production term, $\nu_t = k/\omega$ is the eddy viscosity, and $\gamma = 0.52$, $\sigma_k = \sigma_\omega = 0.5$, $\beta^* = 0.09$ and $\beta = 0.072$ are the standard coefficients of the Wilcox (1998) k–ω model. The k–ω model is selected here on account of its low Reynolds number behaviour and relative simplicity of implementation (although any model for computing ν_t could be directly substituted here). The first tensor basis is equal to the nondimensional strain rate and is linear. As discussed by Wu et al. (2019), it can, therefore, be implemented implicitly to improve the conditioning and stability of the primal solver. The viscous term in the momentum equations is already implicitly treated, so this is achieved through a simple redefinition of the effective viscosity to include the coefficient of the first basis as $\nu_{\text{eff}} = \nu + \nu_t - k\tau_t \beta_1$. The remaining nonlinear part of the anisotropy $\mathbf{a}^{\text{ex}}$ must be treated explicitly and is defined as follows:

$$\mathbf{a}^{\text{ex}} = \sum_{n=2}^{3} \beta_n \mathbf{V}^{(n)} + \frac{\nu_t \mathbf{S}}{k}. \tag{9}$$

The eddy viscosity added implicitly in ν_{eff} is cancelled out by the corresponding eddy viscosity term in the explicit anisotropy, leaving just the EASM. This approach is taken as including ν_t in the implicitly treated viscous term allows much more stable convergence.

The Darcy penalisation term $\boldsymbol{\alpha u}$ is also included where α is a spatially varying continuous variable taking values in the range $0 \leq \alpha \leq \alpha_{\max}$ representing impermeability. A high value of α (limited to $\alpha_{\max}$ for numerical considerations) forces the velocity to zero to satisfy Eq. 6, simulating solid material. A value of $\alpha = 0$ reduces Eq. 6 to the standard RANS momentum equations simulating the region as fluid.

2.3.2 Statement of constrained optimisation problem

Given the set of state equations above, the subset $\mathcal{R} = [\boldsymbol{R_u}, R_p]^{\mathrm{T}}$ is taken as is standard practice when considering the frozen turbulence assumption. The general optimisation problem can then be stated as follows:

$$\begin{aligned}&\text{minimise } \mathcal{J} = \mathcal{J}(\boldsymbol{X}, \alpha), \\ &\text{subject to } \mathcal{R}(\boldsymbol{X}, \alpha) = 0, \\ &\text{and } \mathcal{C}_i(\alpha) = 0,\ i = 1, \ldots, N,\end{aligned} \tag{10}$$

where $\mathcal{J}$ is a cost function, $\mathcal{R}$ is the set of state equations acting as a constraint and $\mathcal{C}_i$ are an additional set of constraints to be satisfied. $\boldsymbol{X}$ is the state vector, and α is the design variable which in this case is impermeability. Here, a volume constraint $\mathcal{C}(\alpha)$ is the only additional constraint considered. This can be optionally enforced to prescribe a fluid volume fraction in the final design and is defined as follows:

$$\mathcal{C}(\alpha) = c^2, \quad c = \frac{1}{\mathcal{V}}\int_{\mathcal{V}} (1 - \frac{\alpha}{\alpha_{\max}}) \mathrm{d}\mathcal{V} - f, \tag{11}$$

where $\mathcal{V}$ is the total domain volume, f is the prescribed fluid volume fraction and c is defined for convenience in expressing the sensitivity. The constrained optimisation can then be refactored into an unconstrained problem:

$$\mathcal{L} = \mathcal{J} + \int_{\mathcal{V}} (qR_p + \boldsymbol{w} \cdot \boldsymbol{R_u}) \, \mathrm{d}\mathcal{V} - \gamma\mathcal{C} + w_{\mathcal{C}}\mathcal{C}^2, \tag{12}$$

where the Lagrangian functional $\mathcal{L}$ is the new subject of minimisation. q and $\boldsymbol{w}$ are Lagrange multipliers (termed the adjoint pressure and velocity respectively) which set penalisation weightings for deviations of the state equations from their constraints $\mathcal{R}(\boldsymbol{X}, \alpha) = 0$. The volume constraint is included through an augmented Lagrangian where γ is an estimate to the Lagrange multiplier and $w_{\mathcal{C}}$ is a scalar weight.

The primal equations R_k and R_ω are not included in the formulation of Eq. 12. This constitutes the commonly used frozen turbulence assumption and is equivalent to considering the derivatives of the turbulent quantities with respect to the state variables negligible. The effects of this assumption have been analysed in topology optimisation by Kontoleontos et al. (2013) and Dilgen et al. (2018a).

2.3.3 Sensitivity analysis

To calculate sensitivity information, it is necessary to take the total variation of the Lagrange functional $\mathcal{L} = \mathcal{L}(\boldsymbol{X}, \alpha)$ such that

$$\mathrm{d}\mathcal{L} = \delta_{\boldsymbol{X}}\mathcal{L}\mathrm{d}\boldsymbol{X} + \delta_\alpha\mathcal{L}\mathrm{d}\alpha \tag{13}$$

where $\delta_{\boldsymbol{X}}$ and δ_α refer to variations with respect to the state vector and design variable, respectively. The expression $\delta_{\boldsymbol{X}}\mathcal{L}$ is non-trivial to evaluate. However, by choosing the adjoint variables such that $\delta_{\boldsymbol{X}}\mathcal{L} = 0$, which from Eq. 12 is equivalent to setting

$$\delta_{\boldsymbol{X}}\mathcal{J} + \int_{\mathcal{V}} (q\delta_{\boldsymbol{X}}R_p + \boldsymbol{w} \cdot \delta_{\boldsymbol{X}}\boldsymbol{R_u}) \, \mathrm{d}\mathcal{V} = 0, \tag{14}$$

the total variation of $\mathcal{L}$ is reduced to $\mathrm{d}\mathcal{L} = \delta_\alpha\mathcal{L}\mathrm{d}\alpha$. It is noted that $\mathcal{C}$ does not appear in Eq. 14 as it has no dependence on the state variables. Again making use of Eq. 12, the total variation can be written as follows:

$$\begin{aligned}\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\alpha} =& \delta_\alpha\mathcal{L} = \delta_\alpha\mathcal{J} + \int_{\mathcal{V}} (q\delta_\alpha R_p + \boldsymbol{w} \cdot \delta_\alpha\boldsymbol{R_u}) \, \mathrm{d}\mathcal{V} \\ &+ \delta_\alpha(-\gamma\mathcal{C} + w_{\mathcal{C}}\mathcal{C}^2).\end{aligned} \tag{15}$$

The state equation R_p has no dependence on α such that $\delta_\alpha R_p = 0$ and the only term in $\boldsymbol{R_u}$ with dependence on α is the penalisation term $\alpha\boldsymbol{u}$ such that $\delta_\alpha\boldsymbol{R_u} = \boldsymbol{u}$. Further, for ducted flows, the cost function $\mathcal{J}$ generally has no explicit dependence on the design variable α such that $\delta_\alpha\mathcal{J} = 0$. The sensitivity can then be reduced to

$$\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\alpha} = \int_{\mathcal{V}} \boldsymbol{w} \cdot \boldsymbol{u}\mathrm{d}\mathcal{V} + (-\gamma + 2w_{\mathcal{C}}\mathcal{C})\delta_\alpha\mathcal{C}. \tag{16}$$

The quantity $\delta_\alpha\mathcal{C}$ is simple to evaluate and yields

$$\delta_\alpha\mathcal{C} = -\frac{2}{\alpha_{\max}}\left[\frac{1}{\mathcal{V}}\int_{\mathcal{V}} (1 - \frac{\alpha}{\alpha_{\max}})\mathrm{d}\mathcal{V} - f\right] = -\frac{2c}{\alpha_{\max}}. \tag{17}$$

Discretising to give local sensitivities for each cell j finally gives

$$\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\alpha_j} = \boldsymbol{w}_j \cdot \boldsymbol{u}_j\mathcal{V}_j - \frac{2c}{\alpha_{\max}}(-\gamma + 2w_{\mathcal{C}}\mathcal{C}), \tag{18}$$

where $\mathcal{V}_j$ is the volume of cell j. The design variable α is then updated over successive iterations n by use of the steepest descent method such that

$$\alpha_j^{n+1} = \min\left[\alpha_{\max}, \max\left[0, \alpha_j^n - \lambda\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\alpha_j}\bigg|^n\right]\right], \tag{19}$$

where λ is a step length which must be set, and the impermeability α is limited such that it cannot be negative or exceed a maximum value $\alpha_{\max}$ which is considered solid. The volume constraint parameters are also updated with every design variable update following the rules:

$$\begin{aligned}\gamma^{n+1} &= \gamma^n - 2w_{\mathcal{C}}^n\mathcal{C}^n, \\ w_{\mathcal{C}}^{n+1} &= \min(\eta w_{\mathcal{C}}^n, w_{\mathcal{C}}^{\max}),\end{aligned} \tag{20}$$

where η and $w_{\mathcal{C}}^{\max}$ are user-defined values to control the growth and maximum value of $w_{\mathcal{C}}$.

2.3.4 Derivation of adjoint system

The task now is to determine the set of equations for the adjoint variables such that when solved the choice that $\delta_{\boldsymbol{X}}\mathcal{L} = 0$ is satisfied. The majority of this derivation has been presented exhaustively in the literature, and for a full description, the reader is referred to Othmer (2008). However, the explicit anisotropy term has not been considered in an adjoint derivation before and will be presented here. As a starting point, the standard adjoint equation obtained from Eq. 14 is

$$
\begin{aligned}
&\int_{\mathcal{V}} \underbrace{\delta p(-\boldsymbol{\nabla}\cdot\boldsymbol{w})}_{\text{(a)}}\,\mathrm{d}\mathcal{V}\\
&+\int_{\mathcal{V}} \underbrace{\delta\boldsymbol{u}\cdot(-2\boldsymbol{u}\cdot\mathbf{S}^{a}+\boldsymbol{\nabla}q-\boldsymbol{\nabla}\cdot(2\nu_{\text{eff}}\mathbf{S}^{a})+\alpha\boldsymbol{w})}_{\text{(b)}}\,\mathrm{d}\mathcal{V}\\
&+\int_{\Gamma} \underbrace{\delta p(w_{n}+\partial_{p}\mathcal{J}_{\Gamma})}_{\text{(c)}}\,\mathrm{d}\Gamma\\
&+\int_{\Gamma} \underbrace{\delta\boldsymbol{u}\cdot(-q\boldsymbol{n}+(\boldsymbol{w}\cdot\boldsymbol{u})\boldsymbol{n}+\boldsymbol{w}(\boldsymbol{u}\cdot\boldsymbol{n})}_{\text{(d)}} \qquad (21)\\
&\quad\underbrace{+\nu_{\text{eff}}(\boldsymbol{n}\cdot\boldsymbol{\nabla})\boldsymbol{w}+\partial_{\boldsymbol{u}}\mathcal{J}_{\Gamma})}_{\text{(d)}}\,\mathrm{d}\Gamma\\
&-\int_{\Gamma} \underbrace{\nu_{\text{eff}}(\boldsymbol{n}\cdot\boldsymbol{\nabla}\delta\boldsymbol{u})\cdot\boldsymbol{w}}_{\text{(e)}}\,\mathrm{d}\Gamma\\
&+\int_{\mathcal{V}} \underbrace{\boldsymbol{w}\cdot(\boldsymbol{\nabla}\cdot(2k\delta\mathbf{a}^{\text{ex}}))}_{\text{(f)}}\,\mathrm{d}\mathcal{V}=0,
\end{aligned}
$$

where δ is used to represent variations of the following variable with respect to the state vector, and $\mathbf{S}^{a}=(\boldsymbol{\nabla}\boldsymbol{w}+\boldsymbol{\nabla}^{T}\boldsymbol{w})/2$ is the symmetric strain rate tensor for the adjoint velocity. Partial derivatives of the cost function at the boundaries with respect to primal pressure and velocity respectively are represented by $\partial_{p}\mathcal{J}_{\Gamma}$ and $\partial_{\boldsymbol{u}}\mathcal{J}_{\Gamma}$. Terms (a–e) comprise the standard adjoint equation and (f) is the additional explicit anisotropy term that will be considered here.

To start, $\boldsymbol{w}$ is brought inside the divergence operator and the product rule applied. Gauss' theorem is then applied to the divergence term to give

$$
+\int_{\Gamma}\underbrace{2k(\boldsymbol{n}\cdot\delta\mathbf{a}^{\text{ex}})\cdot\boldsymbol{w})}_{\text{(a)}}\,\mathrm{d}\Gamma-\int_{\mathcal{V}}\underbrace{2k\delta\mathbf{a}^{\text{ex}}:(\boldsymbol{\nabla}\boldsymbol{w})}_{\text{(b)}}\,\mathrm{d}\mathcal{V}. \qquad (22)
$$

Term (a) cannot be manipulated further so attention shifts to term (b). Recalling the definition of $\mathbf{a}^{\text{ex}}$ from Eq. 9, we start by considering only the final term in the sum, that is $\beta_{3}\mathbf{V}^{(3)}$, as its analysis is the simplest to follow. Further decomposing given that $\mathbf{V}^{(3)}=\tau_{t}^{2}(\mathbf{SS}-\frac{1}{3}\mathrm{tr}(\mathbf{SS})\mathbf{I})$, term (b) may be written as follows:

$$
-\int_{\mathcal{V}}2k\tau_{t}^{2}\beta_{3}(\delta\mathbf{SS}:\boldsymbol{\nabla}\boldsymbol{w}+\mathbf{S}\delta\mathbf{S}:\boldsymbol{\nabla}\boldsymbol{w}-\frac{2}{3}\mathrm{tr}(\delta\mathbf{SS})\mathbf{I}:\boldsymbol{\nabla}\boldsymbol{w})\,\mathrm{d}\mathcal{V}, \qquad (23)
$$

where variations of $\mathbf{V}^{(3)}$ have been taken. It should be highlighted here that this is where the extension of the frozen turbulence assumption has been applied. Variations with respect to the basis coefficients β_{n} are not considered, and the effect of this assumption will not be investigated further in this work. From adjoint continuity, it follows that $\mathbf{I}:\boldsymbol{\nabla}\boldsymbol{w}=\boldsymbol{\nabla}\cdot\boldsymbol{w}=0$ and expanding the variations of the strain rate tensors $\delta\mathbf{S}=(\boldsymbol{\nabla}\delta\boldsymbol{u}+\boldsymbol{\nabla}^{T}\delta\boldsymbol{u})/2$ gives the expression:

$$
\begin{aligned}
&-\int_{\mathcal{V}}k\tau_{t}^{2}\beta_{3}(\boldsymbol{\nabla}\delta\boldsymbol{u}\mathbf{S}:\boldsymbol{\nabla}\boldsymbol{w}+\boldsymbol{\nabla}^{T}\delta\boldsymbol{u}\mathbf{S}:\boldsymbol{\nabla}\boldsymbol{w}\\
&+\mathbf{S}\boldsymbol{\nabla}\delta\boldsymbol{u}:\boldsymbol{\nabla}\boldsymbol{w}+\mathbf{S}\boldsymbol{\nabla}^{T}\delta\boldsymbol{u}:\boldsymbol{\nabla}\boldsymbol{w})\,\mathrm{d}\mathcal{V}.
\end{aligned} \qquad (24)
$$

Making use of the fact that $\mathbf{S}$ is a symmetric tensor to isolate $\boldsymbol{\nabla}\delta\boldsymbol{u}$ and $\boldsymbol{\nabla}^{T}\delta\boldsymbol{u}$ in their respective double inner products, and transposing terms involving the latter results in

$$
\begin{aligned}
&-\int_{\mathcal{V}}k\tau_{t}^{2}\beta_{3}\boldsymbol{\nabla}\delta\boldsymbol{u}:(\boldsymbol{\nabla}\boldsymbol{w}\mathbf{S}+\mathbf{S}\boldsymbol{\nabla}^{T}\boldsymbol{w}+\mathbf{S}\boldsymbol{\nabla}\boldsymbol{w}+\boldsymbol{\nabla}^{T}\boldsymbol{w}\mathbf{S})\,\mathrm{d}\mathcal{V}\\
&=-\int_{\mathcal{V}}k\tau_{t}^{2}\beta_{3}\boldsymbol{\nabla}\delta\boldsymbol{u}:((\boldsymbol{\nabla}\boldsymbol{w}+\boldsymbol{\nabla}^{T}\boldsymbol{w})\mathbf{S}+\mathbf{S}(\boldsymbol{\nabla}\boldsymbol{w}+\boldsymbol{\nabla}^{T}\boldsymbol{w}))\,\mathrm{d}\mathcal{V}.
\end{aligned} \qquad (25)
$$

Noticing that the term $\boldsymbol{\nabla}\boldsymbol{w}+\boldsymbol{\nabla}^{T}\boldsymbol{w}$ is twice the symmetric strain rate tensor for the adjoint velocity, we let the whole expression on the right of the double inner product be the adjoint form of the third tensor basis $\mathbf{W}^{(3)}$ such that $\mathbf{W}^{(3)}=\tau_{t}^{2}(\mathbf{S}^{a}\mathbf{S}+\mathbf{SS}^{a})=\mathbf{s}^{a}\mathbf{s}+\mathbf{ss}^{a}$. The term can then be simply written as follows:

$$
-\int_{\mathcal{V}}2k\beta_{3}\boldsymbol{\nabla}\delta\boldsymbol{u}:\mathbf{W}^{(3)}\,\mathrm{d}\mathcal{V}. \qquad (26)
$$

Applying the same process to the term $\beta_{2}\mathbf{V}^{(2)}$ in $\mathbf{a}^{\text{ex}}$, results in the ability to define a corresponding adjoint form of its primal tensor basis. For brevity's sake, the derivation is not presented here. However, the resulting set of adjoint tensor bases $\mathbf{W}^{(1)}$–$\mathbf{W}^{(3)}$ are found as follows:

$$
\begin{aligned}
&\mathbf{W}^{(1)}=\mathbf{s}^{a},\quad \mathbf{W}^{(2)}=-\mathbf{s}^{a}\boldsymbol{\omega}+\mathbf{ss}^{a}-\mathbf{s}^{a}\mathbf{s}+\boldsymbol{\omega}\mathbf{s}^{a},\\
&\mathbf{W}^{(3)}=\mathbf{s}^{a}\mathbf{s}+\mathbf{ss}^{a}.
\end{aligned} \qquad (27)
$$

This result presents the opportunity to define an adjoint explicit anisotropy $\mathbf{b}^{\text{ex}}$, analogous to its primal counterpart, which is given as follows:

$$
\mathbf{b}^{\text{ex}}=\sum_{n=2}^{3}\beta_{n}\mathbf{W}^{(n)}+\frac{\nu_{t}\mathbf{S}^{a}}{k}. \qquad (28)
$$

The full term given in Eq. 22 can then be written as follows:

$$
+\int_{\Gamma}2k(\boldsymbol{n}\cdot\delta\mathbf{a}^{\text{ex}})\cdot\boldsymbol{w})\mathrm{d}\Gamma-\int_{\mathcal{V}}2k\boldsymbol{\nabla}\delta\boldsymbol{u}:\mathbf{b}^{\text{ex}}\mathrm{d}\mathcal{V}. \qquad (29)
$$

To finish the derivation, $\mathbf{b}^{\text{ex}}$ is brought inside the gradient operator and the product rule applied. Gauss' theorem is once again applied to the resulting divergence term to give

$$+\int_{\Gamma} 2k(\boldsymbol{n}\cdot\delta\mathbf{a}^{\mathrm{ex}})\cdot\boldsymbol{w})\mathrm{d}\Gamma - \int_{\Gamma}\delta\boldsymbol{u}\cdot(2k(\mathbf{b}^{\mathrm{ex}}\cdot\boldsymbol{n}))\mathrm{d}\Gamma + \int_{\mathcal{V}}\delta\boldsymbol{u}\cdot(\boldsymbol{\nabla}\cdot(2k\mathbf{b}^{\mathrm{ex}}))\mathrm{d}\mathcal{V}. \tag{30}$$

This expression can be directly substituted into Eq. 21 for term (f). The whole adjoint equation, written in a compact form, is then

$$\int_{\mathcal{V}}\delta pR_q + \delta\boldsymbol{u}\cdot\boldsymbol{R}_w\mathrm{d}\mathcal{V} + \int_{\Gamma}\delta pBC_q + \delta\boldsymbol{u}\cdot\boldsymbol{BC}_w - BC_1 + BC_2\mathrm{d}\Gamma = 0, \tag{31}$$

where the terms R_q, $\boldsymbol{R}_w$, BC_q, $\boldsymbol{BC}_w$, BC_1 and BC_2 are defined as follows:

$$R_q = -\boldsymbol{\nabla}\cdot\boldsymbol{w}, \tag{32}$$

$$\begin{aligned}\boldsymbol{R}_w = &-\boldsymbol{u}\cdot(\boldsymbol{\nabla}\boldsymbol{w} + \boldsymbol{\nabla}^{\mathrm{T}}\boldsymbol{w}) + \boldsymbol{\nabla}q\\ &-\boldsymbol{\nabla}\cdot(\nu_{\mathrm{eff}}(\boldsymbol{\nabla}\boldsymbol{w} + \boldsymbol{\nabla}^{\mathrm{T}}\boldsymbol{w}) - 2k\mathbf{b}^{\mathrm{ex}}) + \alpha\boldsymbol{w},\end{aligned} \tag{33}$$

$$BC_q = w_n + \partial_p\mathcal{J}_{\Gamma}, \tag{34}$$

$$\begin{aligned}\boldsymbol{BC}_w = &-q\boldsymbol{n} + (\boldsymbol{w}\cdot\boldsymbol{u})\boldsymbol{n} + \boldsymbol{w}(\boldsymbol{u}\cdot\boldsymbol{n})\\ &+\nu_{\mathrm{eff}}(\boldsymbol{n}\cdot\boldsymbol{\nabla})\boldsymbol{w} - 2k(\mathbf{b}^{ex}\cdot\boldsymbol{n}) + \partial_{\boldsymbol{u}}\mathcal{J}_{\Gamma},\end{aligned} \tag{35}$$

$$BC_1 = \nu_{\mathrm{eff}}(\boldsymbol{n}\cdot\boldsymbol{\nabla}\delta\boldsymbol{u})\cdot\boldsymbol{w}, \tag{36}$$

$$BC_2 = 2k(\boldsymbol{n}\cdot\delta\mathbf{a}^{\mathrm{ex}})\cdot\boldsymbol{w}. \tag{37}$$

To force the equality of Eq. 31 to zero in the domain, given that δp and $\delta\boldsymbol{u}$ are not zero in general, requires $R_q = 0$ and $\boldsymbol{R}_w = 0$. From this, we arrive at the adjoint RANS continuity and momentum equations with generalised EASM turbulence closure.

2.3.5 Treatment of boundary conditions

Similarly we require on the boundaries of the domain that each of the terms δpBC_q, $\delta\boldsymbol{u}\cdot\boldsymbol{BC}_w$, BC_1 and BC_2 vanish within the integral. To simplify this analysis considerably, we enforce that the EASM is switched off on all boundaries reverting to the standard k–ω model. The effect of this is that in Eq. 37, $\delta\mathbf{a}^{\mathrm{ex}} = 0$ on all boundaries and this condition is automatically satisfied. In Eq. 35, we also have that $\mathbf{b}^{\mathrm{ex}} = 0$ on all boundaries, reducing the set of adjoint boundary conditions to those of the standard continuous adjoint equations seen in Othmer (2008). For the pressure loss cost function

$$\mathcal{J} = -\int_{\Gamma}(p + \frac{1}{2}|\boldsymbol{u}|^2)u_n\mathrm{d}\Gamma, \tag{38}$$

which represents the power dissipation calculated as the net inward flux of energy through the boundaries, the adjoint boundary conditions at the inlet and walls are

$$\boldsymbol{w_t} = 0,\quad w_n = \begin{cases} 0 & \text{walls}\\ u_n & \text{inlet}\end{cases},\ \text{and}\ \boldsymbol{\nabla}q = 0. \tag{39}$$

The corresponding boundary conditions on the outlet, obtained by decomposing Eq. 35 into normal and tangential directions are

$$\begin{aligned}&q = \boldsymbol{w}\cdot\boldsymbol{u} + w_nu_n + \nu_{\mathrm{eff}}(\boldsymbol{n}\cdot\boldsymbol{\nabla})w_n - \frac{1}{2}|\boldsymbol{u}|^2 - u_nu_n,\\ &\boldsymbol{w_t}u_n + \nu_{\mathrm{eff}}(\boldsymbol{n}\cdot\boldsymbol{\nabla})\boldsymbol{w_t} - \boldsymbol{u_t}u_n = 0.\end{aligned} \tag{40}$$

It is noted here that if the EASM is not switched off on all boundaries, then the task of determining appropriate boundary conditions for the complete set of adjoint equations has not yet been solved. Attempts so far lead to continuity violating solutions where the adjoint velocity must be set to zero at the outlet, but not at the inlet.

2.3.6 Optimisation algorithm implementation

The optimisation algorithm follows a one-shot method in which the primal and adjoint systems and the sensitivities and design variables are converged together. Such methods, which leverage the fact that the primal and adjoint systems are solved in a pseudo-transient manner, were introduced by Hazra (2008) for shape optimisation. More recently, they have been applied to topology optimisation by Papoutsis-Kiachagias and Giannakoglou (2016) and shown to be highly efficient for fluid topology optimisation problems.

The one-shot approach used in the present work is outlined in Fig. 2 and is implemented in the finite volume package OpenFOAM. Instead of obtaining a fully converged primal and adjoint solution between sensitivity and optimisation update calculations, a prescribed number of pseudo-timesteps of the primal and adjoint system (Eqs. 5–8 and Eqs. 32–33, respectively) are performed between each optimisation update (n^{upd} in Fig. 2). Sensitivities are then calculated based on the current values of $\boldsymbol{u}^i$ and $\boldsymbol{w}^i$ via Eq. 18, where i is the current pseudo-timestep. The optimisation variables can finally be updated based on these sensitivities through Eqs. 19 and 20. This process is iterated to convergence of the primal and adjoint systems, the sensitivities, and the optimised design simultaneously. Although the number of optimisation updates is often large compared with traditional gradient methods, the total computational cost can be reduced as the time to calculate the sensitivities for each update is small. Hazra (2008) showed that converged optimisations can be achieved with less than two times the effort of converging the primal system alone (See Sects. 3.9 and 4.3.5 for more details on the computational cost of the present method).

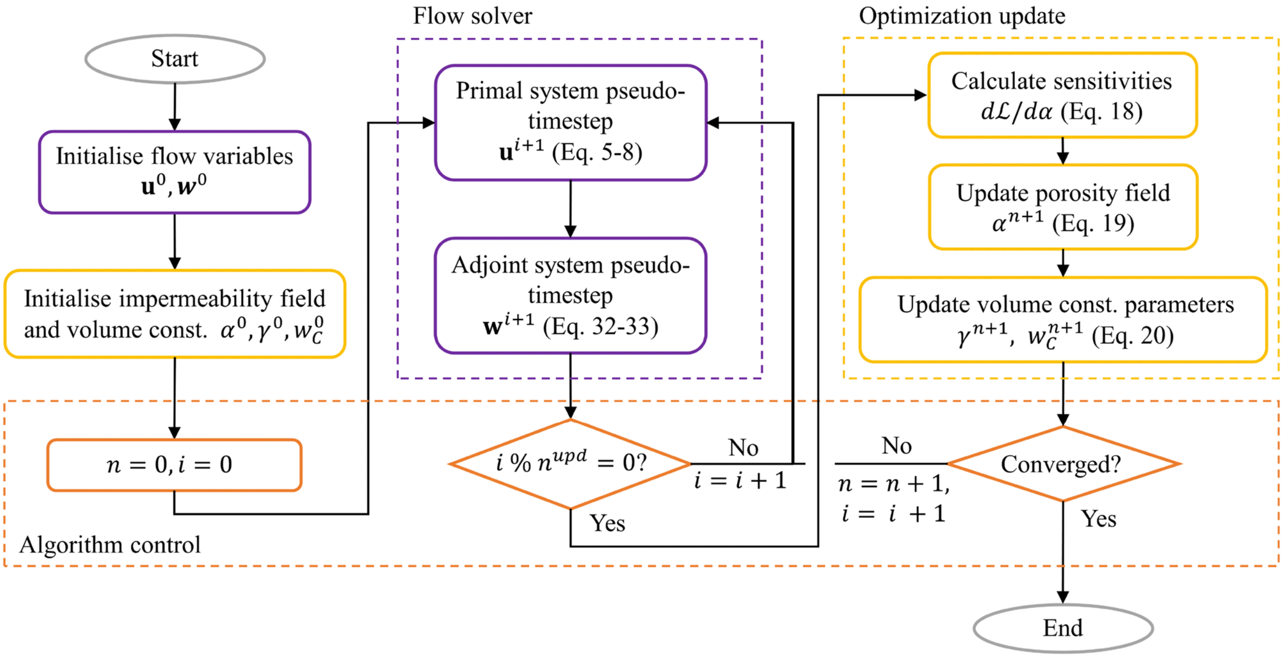
Flow solver
Optimization update
Start
Initialise flow variables
$\mathbf{u}^0, \boldsymbol{w}^0$
Initialise impermeability field
and volume const. $\alpha^0, \gamma^0, w_C^0$
$n = 0, i = 0$
Primal system pseudo-timestep
$\mathbf{u}^{i+1}$ (Eq. 5-8)
Adjoint system pseudo-timestep
$\mathbf{w}^{i+1}$ (Eq. 32-33)
$i \,\%\, n^{upd} = 0?$
No
$i = i + 1$
Yes
Calculate sensitivities
$d\mathcal{L}/d\alpha$ (Eq. 18)
Update porosity field
α^{n+1} (Eq. 19)
Update volume const. parameters
γ^{n+1}, w_C^{n+1} (Eq. 20)
No
$n = n + 1,$
$i = i + 1$
Converged?
Yes
End
Algorithm control

Fig. 2 Flow diagram of the one-shot optimisation algorithm. The purple boxes relate to primal and adjoint flow solver steps, yellow boxes show steps related to the optimisation updates and orange boxes show steps to control the flow of the algorithm. (Color figure online)

2.4 Verification and testing procedure

In the present work, the developed method is applied to two cases. The first is a verification on the backward-facing step of Le et al. (1997). This is a simple test case to evaluate, and the physics is reduced to a level allowing some physical intuition of the problem. As such, it can act as a sanity check to show that the method is operating in the correct manner. The second case is a u-bend, which aims to test the method by presenting it with a more complex optimisation.

The general procedure follows that shown in Fig. 3. An overview will be given here and details for each case are explained in the following sections (backward-facing step in Sect. 3 and u-bend in Sect. 4). First, a high fidelity LES of the baseline geometry is performed, against which subsequent optimisations can be compared. In parallel, a standard optimisation is run using the k–ω model.

To prepare this standard optimised geometry for high fidelity simulation, it must be extracted from the topology optimization, smoothed and remeshed. First, the raw optimised fluid region is obtained by isolating the region in which $\alpha < \alpha_{\max}/2$. Given the nature of the fixed grids used, this region will not have a smooth surface. Thus, to obtain a smooth geometry, the surface is passed through a smoothing filter. The filter operates by repeatedly moving each vertex in the triangulated surface to the average position of its neighbours (directly connected vertices). Once the smooth optimised geometry is obtained, it can be remeshed with a body-fitted mesh ready for high fidelity simulation.

The next step is to run the data obtained from the high fidelity simulation of the standard optimised geometry through GEP, to obtain a model which can be implemented in the data-driven optimisation. The training dataset is generated from the LES of the standard geometry (details in Sect. 3.3). For the u-bend case, it will be seen that the topology resulting from the standard optimisation differs from the baseline design. Therefore, there is no guarantee that a model learned on the baseline design will remain applicable when run in the data-driven optimisation. So, by training on data from the LES of the standard geometry, we allow a rough initial design to be drafted and a turbulence model more representative of the optimised geometry to be developed.

The GEP model is then used to run a second, data-driven optimisation, from which the resulting geometry is again extracted, smoothed and remeshed with a body fitted mesh allowing a third LES to be conducted. By comparing the average total pressure loss across the domain $\mathcal{J}^{\mathrm{bl}}$, $\mathcal{J}^{\mathrm{std}}$ and $\mathcal{J}^{\mathrm{dd}}$ (Eq. 38) for the baseline, standard and data-driven geometries, respectively, it can be determined whether the data-driven method has successfully generated an improved design.

Details of the computational cost of the method are presented for the backward-facing step and u-bend in Sects. 3.9 and 4.3.5, respectively. In these sections, we demonstrate the

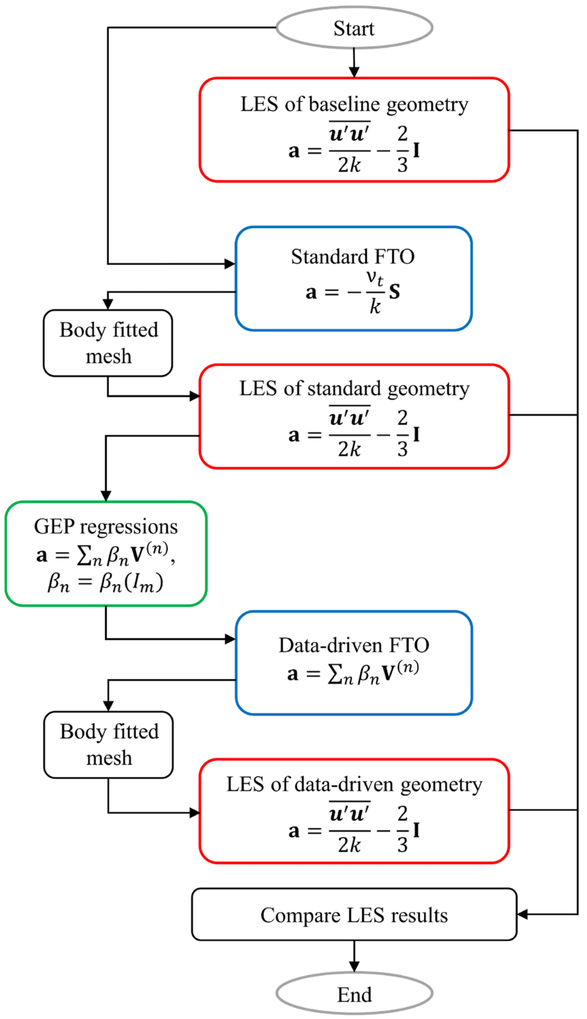
Start
LES of baseline geometry
$\mathbf{a} = \frac{\overline{\boldsymbol{u}'\boldsymbol{u}'}}{2k} - \frac{2}{3}\mathbf{I}$
Standard FTO
$\mathbf{a} = -\frac{\nu_t}{k}\mathbf{S}$
Body fitted mesh
LES of standard geometry
$\mathbf{a} = \frac{\overline{\boldsymbol{u}'\boldsymbol{u}'}}{2k} - \frac{2}{3}\mathbf{I}$
GEP regressions
$\mathbf{a} = \sum_n \beta_n \mathbf{V}^{(n)}$,
$\beta_n = \beta_n(I_m)$
Data-driven FTO
$\mathbf{a} = \sum_n \beta_n \mathbf{V}^{(n)}$
Body fitted mesh
LES of data-driven geometry
$\mathbf{a} = \frac{\overline{\boldsymbol{u}'\boldsymbol{u}'}}{2k} - \frac{2}{3}\mathbf{I}$
Compare LES results
End

Fig. 3 Flow diagram of the verification and testing procedure for the data-driven topology optimisation method. Red, blue, and green boxes correspond to LES, topology optimisation, and GEP simulation steps, respectively. Black boxes relate to intermediary data-processing steps. (Color figure online)

significant cost advantage over attempting to implement high fidelity methods in optimisation routines directly.

Finally, it should be noted that in the backward-facing step case, the baseline geometry is used to perform the GEP regressions (rather than the standard geometry). This is so the operation of GEP can be assessed independently from any optimisation procedure. As will be seen in the results, the topology of the standard optimisation remains unchanged from the baseline geometry and the high fidelity flows have very similar structures. Therefore, the GEP model can be considered applicable as it is still presented with similar flow physics during optimisation to the case it was trained on.

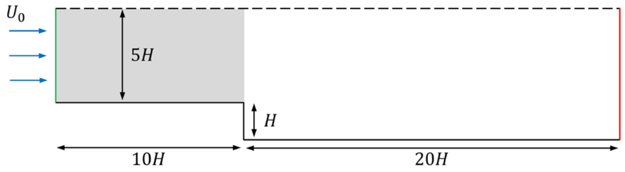

Fig. 4 Schematic of the backward-facing step domain. The inlet is coloured green and the outlet red. No-slip walls are indicated by a black solid line and slip walls by black dashed lines. The area shaded grey is excluded from the optimisation domain. (Color figure online)

3 Verification using backward-facing step

3.1 Backward-facing step case details

Figure 4 shows a schematic of the backward-facing step case used to verify the method. The domain consists of an entry region of length $10H$ before the step and an exit region of length $20H$ after the step, the entry region has a height of $5H$ and the step height is H giving an expansion ratio of 1.2. The mean velocity boundary layer profile obtained by Spalart (1988) is imposed at the inlet with $Re_\theta = 667$ and the Reynolds no. based on the step height H and mean inlet velocity U_0 is $Re_H = 5100$. When running high fidelity simulations, random fluctuations $\boldsymbol{u}'$ are superimposed on the inlet velocity profile to match the Reynolds stresses and an advective boundary condition is applied at the outlet. For RANS-based calculations (including optimisations), the mean profile alone is used, and a Neumann condition for velocity is imposed at the outlet. In all simulations, the bottom wall is set as no-slip and a slip condition representing free-stream flow is imposed at the top wall. For the pressure, an atmospheric condition is set at the outlet with Neumann conditions at the inlet and walls.

3.2 Initial high fidelity simulation

The initial high fidelity simulation is run for the baseline geometry presented in Fig. 4 with boundary conditions as specified in the previous section. The chosen method for all high fidelity simulations is large eddy simulation (LES) which was run with a wall-adapting local eddy viscosity (WALE) sub-grid scale (SGS) turbulence model and solved using the pressure implicit with splitting of operator (PISO) algorithm. The mesh was generated such that at the lower wall, the first cell is placed within the range $y^+ < 1$, where $y^+ = yu_\tau/\nu$ is the nondimensional wall unit and u_τ is the friction velocity at the wall defined as $u_\tau = \sqrt{\nu \partial_y u|_w}$. The mesh consists of 1,140,000 cells with 150 cells placed in the streamwise direction before the step, 300 cells in the streamwise direction after the step, 40 cells in the spanwise direction behind the step and 100 cells in the spanwise direction above the step. Mesh refinement was used in the spanwise direction at the lower wall and in both streamwise and spanwise directions in the wake region behind the step. 20 cells are also placed normal to the streamwise, spanwise plane and cyclic boundary conditions are imposed on the front and back walls. The generated mesh is shown in Fig. 5, highlighting the refinement around the step. The total simulation time is $t_{\text{total}} = 1500H/U_0$. Assuming a mean convective speed of $U_c \approx 0.8U_0$, the first 20 pass-throughs of the post-expansion region with length $20H$ are discarded to ignore the effect of any initial transient behaviour. This leaves 40 complete fluid passes ($\approx 1000H/U_0$) from which to acquire the converged statistical dataset.

3.3 Training data extraction

The statistical quantities obtained from averaging of the LES data are the time averaged velocity $\boldsymbol{u}$ and the Reynolds stress $\overline{\boldsymbol{u}'\boldsymbol{u}'}$ at each grid location in the domain. These raw data require some post-processing to extract the training data for GEP which comprises $(\boldsymbol{a}, I_m, \mathbf{V}^{(n)})$. The first step is to average $\boldsymbol{u}$ and $\overline{\boldsymbol{u}'\boldsymbol{u}'}$ in the z-direction as all RANS-based calculations are conducted in 2D. The nondimensional anisotropy $\boldsymbol{a}$ is then calculated from its definition in Eq. 1, where k is calculated by definition as $k = \text{tr}(\overline{\boldsymbol{u}'\boldsymbol{u}'})/2$. To obtain I_m and $\mathbf{V}^{(n)}$ the nondimensional strain and rotation rate tensors are required which themselves require the specific dissipation rate ω. Although a value for ω can be calculated directly from LES, it will not be consistent with the ω predicted in RANS. As such, a 'correct' value of ω is found by solving the ω-transport equation (Eq. 8) such that $R_\omega = 0$, whilst holding the LES quantities $\boldsymbol{u}$, $\overline{\boldsymbol{u}'\boldsymbol{u}'}$ and k constant. The scalar invariants and tensor bases are then simple to compute from Eqs. 3 and 4.

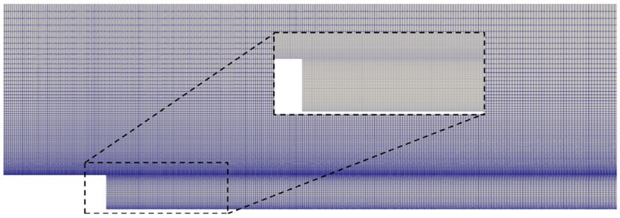

Fig. 5 Mesh used for the high fidelity simulation of the baseline backward-facing step geometry

The final step in preparing the training data set for GEP is to condition the data on the magnitude of the turbulent kinetic energy. When k is low, its value is dominated by noise in the high fidelity data, as we train on the anisotropy normalised by $2k$ this noise can be carried through to the training data and adversely affect the performance of the machine learning. As such we discount these regions of the domain from the training data. A threshold value $k_{\text{threshold}} = 0.05k_{\text{max}}$ is used and the resulting region is visualised in Fig. 6. The choice of threshold is case dependent and has been set heuristically. Data in the coloured region where $k > k_{\text{threshold}}$ is taken as the training dataset for GEP. This results in training data consisting of 28 000 sets of values for $(\boldsymbol{a}, I_m, \mathbf{V}^{(n)})$, one from each grid point inside the conditioned region.

3.4 Application of GEP to high fidelity dataset

From the acquired training dataset for GEP, $\mathbf{a}$ is the target variable and the scalar invariants and tensor bases are the independent variables used to form specific solutions $\mathbf{a}^{\text{GEP}} = \mathbf{a}^{\text{GEP}}(I_m, \mathbf{V}^{(n)})$. These solutions are constrained by the general EASM structure given in Eq. 2. As the flow is effectively 2D, the z components of the anisotropy and tensor bases are not considered in the regressions. The cost function used to evaluate the fitness of individuals in the population is, thus, given as the mean square error $\mathcal{J}^{\text{GEP}}$ of the x and y components averaged over the N training points

$$\mathcal{J}^{\text{GEP}} = \frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{2}\sum_{j=1}^{2}(\text{a}_{ij} - \text{a}_{ij}^{\text{GEP}})^2. \tag{41}$$

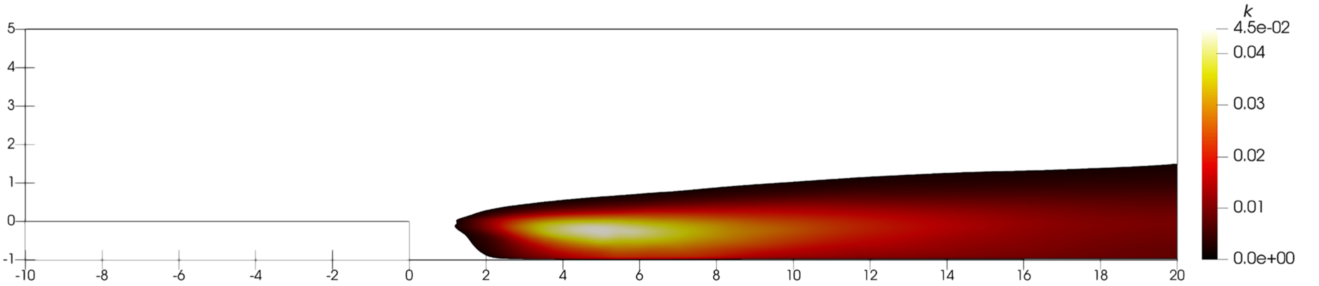

Fig. 6 Data conditioning for the backward-facing step. Data from the wake region where $k > k_{\text{threshold}}$ is used as the GEP data set. Data in the white region are discarded

Table 1 Parameters for GEP algorithm

Parameter	Value
Population size	200
Generations	300
Tournament size	2
Random constant range	$\|r\| < 0.2$
No. random constants	5
Host terminals	$\mathbf{V}^{(1)}, \mathbf{V}^{(2)}, \mathbf{V}^{(3)}$
Host functions	$p, +, -$
Plasmid terminals	$1.0, 0.01, r, I_1, I_2$
Plasmid functions	$+, -, \times$

The parameters used in GEP are detailed in Table 1. Further details on the definitions of stated parameters can be found in Weatheritt and Sandberg (2016).

GEP was run 50 times and given the stochastic nature of the algorithm, each run produced a unique model. To mitigate against the risk of any single model over fitting the anisotropy, the complete set of 50 candidate solutions was ensemble averaged to produce the final data-driven closure. Although the full expression is too long to present here, to first order in the β_n coefficients, it is given as follows:

$$\begin{aligned}\mathbf{a}^{\mathrm{GEP}} &= (-1.021 + 0.651 I_1 - 1.540 I_2)\mathbf{V}^{(1)} \\ &+ (1.019 - 0.125 I_1 + 0.971 I_2)\mathbf{V}^{(2)} \\ &+ (1.697 - 0.903 I_1 + 1.094 I_2)\mathbf{V}^{(3)}.\end{aligned} \tag{42}$$

The EASM structure should is easy to discern as each basis $\mathbf{V}^{(n)}$ is premultiplied by some expression containing the scalar invariants I_m. These expressions are the β_n coefficients and their specification is what makes the data-driven method realisable.

3.5 Verification of the GEP closure

Analysing Eq. 42, it is seen that GEP has found a coefficient of the first tensor basis of −1.021 to leading order which is very close to the standard EVM as $-\mathbf{V}^{(1)} = -\mathbf{S}/\omega = -\nu_t\mathbf{S}/k$. This is encouraging as it attests to the ability of GEP to discover physically relevant closures. However, GEP also finds strong dependency on the higher-order bases and scalar invariants, and this is where the advantage of the EASM eddy viscosity models such as k–ω is introduced.

The ensembled model found a mean square error calculated by Eq. 41 of 0.0273 in comparison to a mean square error of 0.0527 when using the EVM relationship $\mathbf{a} = -\mathbf{V}^{(1)}$. To investigate where this improvement is obtained, we look at Fig. 7 which plots the β_1 coefficient over the training region. Values range between −1 (recovering the linear EVM) and close to 0. It can be seen in the central wake region and the near wall region that $|\beta_1| < 1$ indicating a lower turbulent viscosity in comparison to the linear EVM. This suggests that the linear EVM over predicts the additional viscosity effect induced by turbulent fluctuations in these areas.

RANS calculations using the k–ω and data-driven models were also performed such that the resulting velocity fields could be compared against the time averaged LES. Figure 8 shows results for the streamwise and spanwise velocity profiles of each model plotted at several x/H locations. For both the streamwise and spanwise profiles, the GEP closure is able to more accurately recreate the LES profile than the k–ω model. One key aspect is the ability of GEP to better predict the length of the re-circulation bubble, which is slightly longer than predicted in the k–ω calculation. This may have been expected when remembering that the GEP model reduced the effective viscosity in the wake region, the flow over the step, thus, travels further before being brought back to the wall by viscous effects.

Across all considered metrics, the GEP closure was found to outperform the k–ω model in the RANS calculations. The MSE in the anisotropy improved 24.8% from 0.0433 to 0.0325. These values are a slight deterioration of the values seen when purely considering the high fidelity data, although this is not unexpected as a full RANS calculation is a much tougher test for the closure than simply performing well in the regression. The improvements in anisotropy prediction propagated through to a 65.2% reduction in MSE for the velocity field which improved from 2.06×10^{-3} to 7.18×10^{-4}.

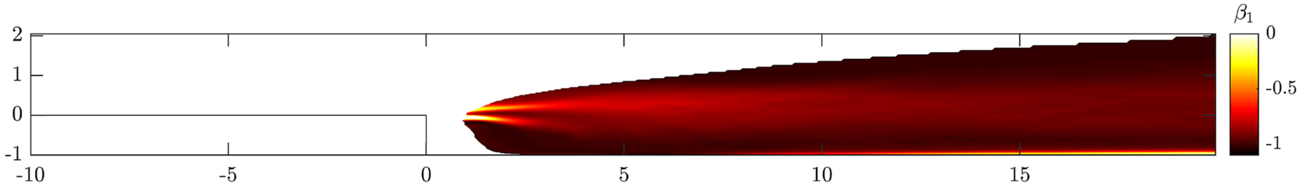

Fig. 7 Plot of the β_1 coefficient across the training domain showing reduced values in the wake region compared with a linear eddy viscosity model

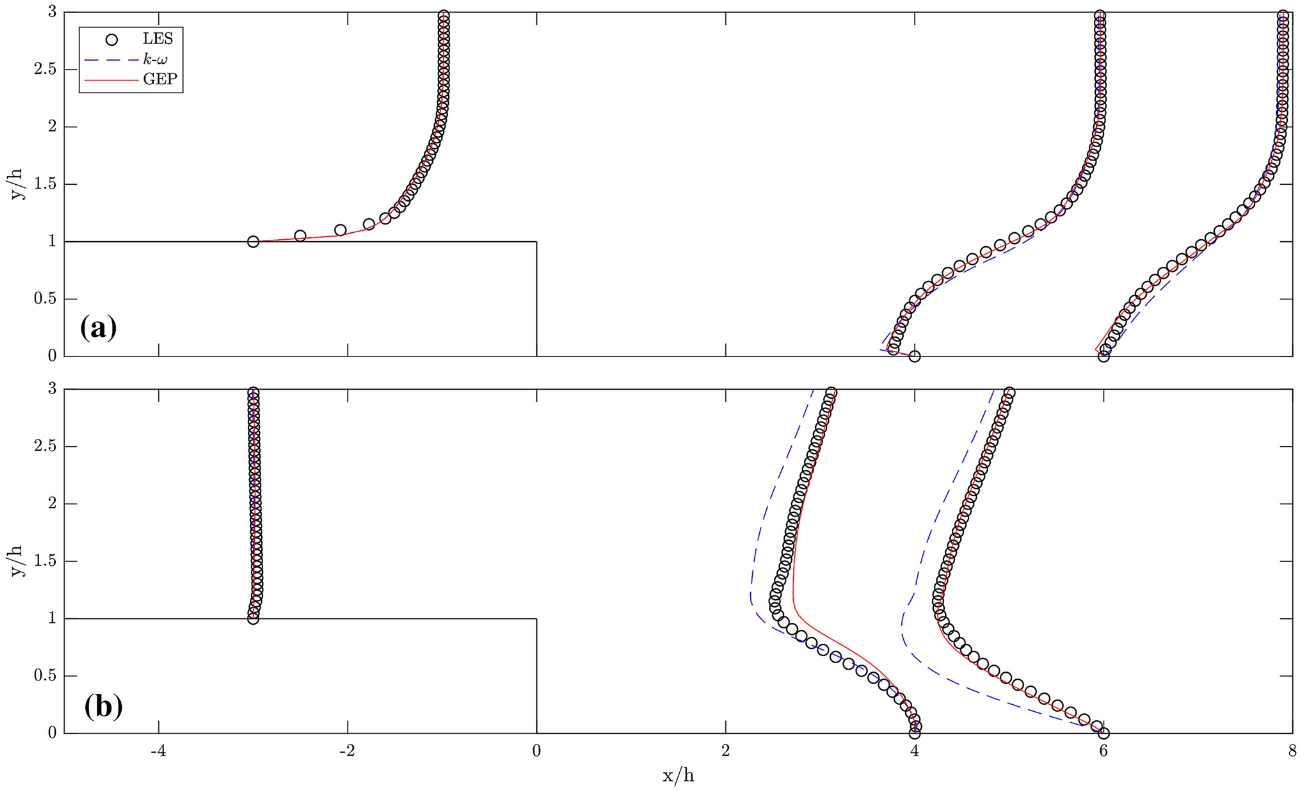

Fig. 8 Wall normal profiles of **a** streamwise velocity and **b** spanwise velocity at various streamwise locations for the LES data, k–ω and GEP models

3.6 Topology optimisations

Now we are confident the GEP model is operating correctly, we proceed to consider the topology optimisation. Two optimisations are run according to Fig. 3. The first is a standard optimisation with the k–ω turbulence model, and the second is a full data-driven optimisation complete with data-driven closure. For each optimisation, the same grid was used as for the previous RANS verification. This is a 2D grid with the same streamwise and spanwise cell locations as the high fidelity LES but with just 1 cell placed in the z-direction, resulting in a grid size of 57,000. The optimisation domain corresponds to the white area on the schematic of the case in Fig. 4 with the grey-shaded region excluded by turning off impermeability updates (Eq. 19). The maximum impermeability was set to $\alpha_{\max} = 2000$ with a step size $\lambda = 1 \times 10^6$. The optimisation algorithm was run using the one-shot method detailed in Sect. 2.3.6 with $n^{\mathrm{upd}} = 10$, meaning that 10 pseudo-timesteps of the primal and adjoint system are performed between optimisation updates. Volume constraints were not enforced in these optimisations, achieved by excluding the second term in the sensitivity calculation in Eq. 18.

Each optimisation was run with the converged RANS solutions obtained in the previous section as initial conditions. The optimisations were run for 300 iterations (3000 pseudo-timesteps) to allow convergence of α and the primal and adjoint systems. Optimised geometries are shown in Fig. 9 with the corresponding optimisation histories presented in Fig. 10. Despite the similarities in the optimised geometries, the data-driven optimisation generates an elongated ‘ramp’ length in comparison to the standard optimisation. This is a promising result as the improvement that GEP found over the eddy viscosity model was attributed to an increase in the length of the separation bubble. An increased ‘ramp’ length would, therefore, seem an appropriate adjustment for the data-driven optimisation to make.

3.7 Geometry extraction and high fidelity simulations

The aim is to produce optimised geometries reflective of high fidelity turbulence treatment. So, to test if the geometry produced by the data-driven optimisation is an improvement over the standard geometry, they must both be extracted from their optimisations, remeshed with body fitted meshes and run with LES.

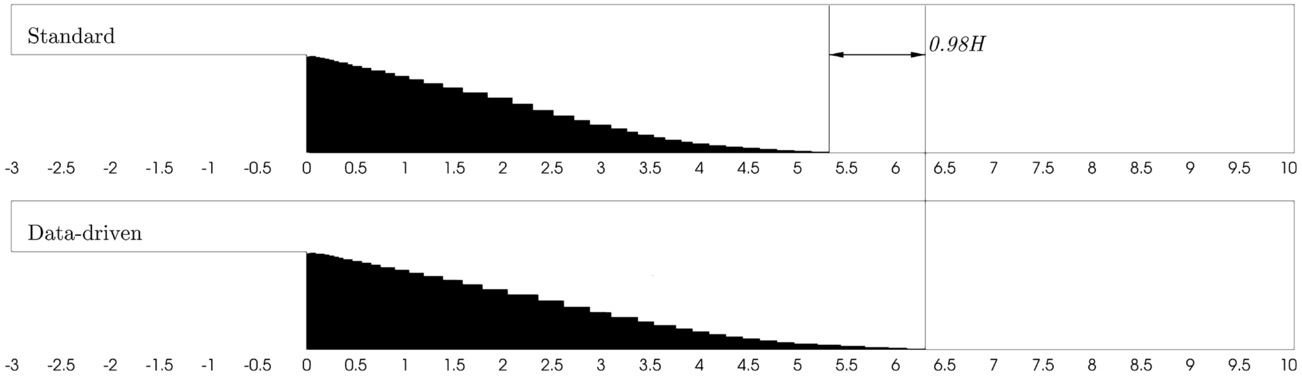

Fig. 9 Resulting geometries from the standard FTO (top) and data-driven FTO (bottom). The increase in ramp length for the data-driven geometry is highlighted

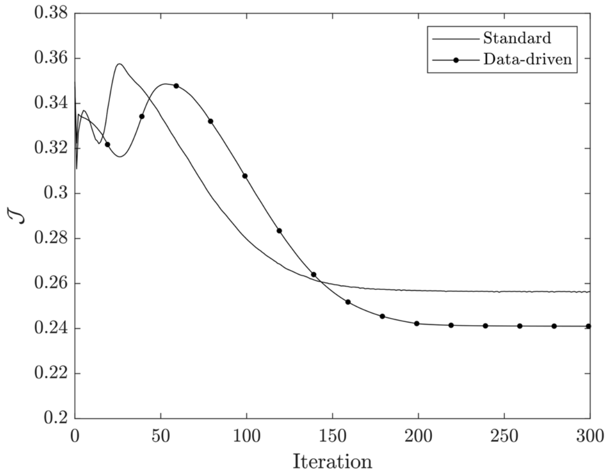

Fig. 10 Convergence of the standard and data-driven topology optimisations for the backward-facing step case

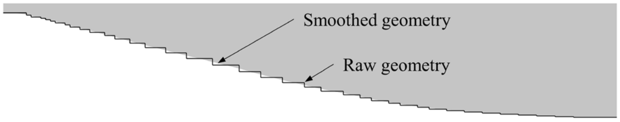

Fig. 11 Raw geometry extracted directly from the standard optimisation of the backward-facing step and the resulting smoothed geometry after application of the smoothing filter

This process follows the steps outlined in Sect. 2.4. First, the raw geometry is extracted by taking the region where $\alpha < \alpha_{\max}/2$. Given the nature of the fixed grids used in optimisation, the generated surface will not be smooth and must be smoothed before remeshing. The smoothing operation of moving each vertex to the average position of it's neighbours is repeated 30 times to produce the smoothed geometry. The raw and smoothed geometry are shown for the ramp section of the standard optimisation in Fig. 11.

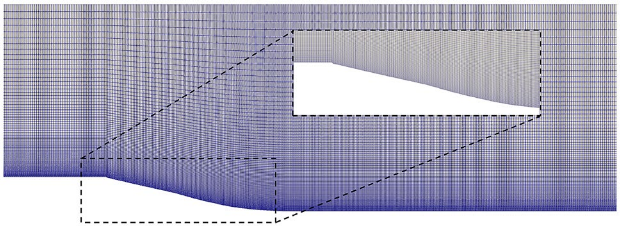

Fig. 12 Body-fitted mesh generated for the high fidelity simulation of the standard backward-facing step geometry

This smoothed geometry can now be remeshed and the body fitted mesh for the standard geometry is shown in Fig. 12. The meshes for each geometry are given the same structure, consisting of $1,200,000$ hexahedral cells with 600, 100 and 20 elements in the x (streamwise), y (spanwise) and z directions, respectively. Cyclic boundary conditions are again imposed on the front and back walls. Other than the differing geometries, the simulations are set up with exactly the same boundary and initial conditions as the initial high fidelity simulation. All models and solution methods are also identical. Converged average pressure loss values are then calculated from the baseline, standard and data-driven geometries, the results of which will be presented in the next section.

3.8 Verification results

Values for the average pressure drop of the fluid as it flows through the domain obtained: (1) directly from the topology optimisations, (2) from RANS calculations (using the corresponding closure) on the body-fitted meshes, and (3) from LES for each of the baseline, standard and data-driven geometries are detailed in Table 2.

Comparing the results of the optimisations with the baseline RANS the standard optimisation improves by 22.5%,

Table 2 Pressure drop results for the backward-facing step optimisation

Geometry	RANS (TO)	RANS (body fit)	LES
Baseline	–	0.3310	0.3454
Standard	0.2564	0.2840	0.3950
Data driven	0.2410	0.2628	0.3326

whilst the data-driven optimisation finds an improvement of 27.2%. When running RANS calculations on the body-fitted geometries with their respective closures, the results deteriorate slightly with the standard RANS finding an improvement over the baseline of 14.2% and the data-driven RANS improving on the baseline by 20.6%. Looking finally at the LES results which is the true test of the method, it is sen that the pressure drop for the standard geometry is actually higher than the baseline at 0.3950 in comparison to 0.3454 whilst the data-driven geometry finds a slight improvement over the baseline of 3.7%.

The reason for the standard geometry failing to improve on the baseline can be explained when looking at the separation predicted in Fig. 13. The standard RANS model fails to accurately predict the separation over the generated 'ramp', and in the optimisation, this suggests that ramp is incorrectly calibrated for the LES flow where a much larger degree of separation is predicted. The data-driven RANS provides an improved prediction of the separation over the ramp so the optimisation is able to generate a ramp more representative of the high fidelity flow (by removing more of the separation region) allowing an improvement over the baseline to be achieved.

These results highlight the necessity of the developed data-driven optimisation technique and the importance of testing geometries with high fidelity methods. They demonstrate that running an optimisation using an inaccurate flow model can result in a solution that fails to perform well in practice. Although this is unlikely to be true for all test cases, it serves as an important demonstration of the issues associated with current topology optimisation solvers. The data-driven technique, however, has been shown to operate as expected for the backward-facing step test case.

3.9 Computational cost

It is important to consider the computational cost of the full data-driven procedure as one of the benefits of the method is the feasibility of achieving designs representative of high fidelity methods. To frame this discussion, a breakdown of the computational cost for each step in the process of the backward-facing step verification is presented in Table 3.

The optimisations, GEP regressions and RANS calculations are all run on a standard desktop using 4 cores whilst the high fidelity LES is run on a high performance computing cluster using 32 cores. The computational cost of the standard optimisation is just 1.5 times that of the baseline RANS calculation, highlighting the efficiency of the one-shot method in generating optimised designs. Even though

Table 3 Details of computational cost for the backward-facing step data-driven optimisation verification

Optimisation step	Core-h	No. cores	CPU model	Clock speed/ GHz
Standard FTO	0.3	4	Intel Core i7-9700	3.0
Data-driven FTO	0.7	4	Intel Core i7-9700	3.0
GEP regressions	1.7	4	Intel Core i7-9700	3.0
RANS (baseline)	0.2	4	Intel Core i7-9700	3.0
LES (baseline)	930	32	AMD EPYC 7724	2.8

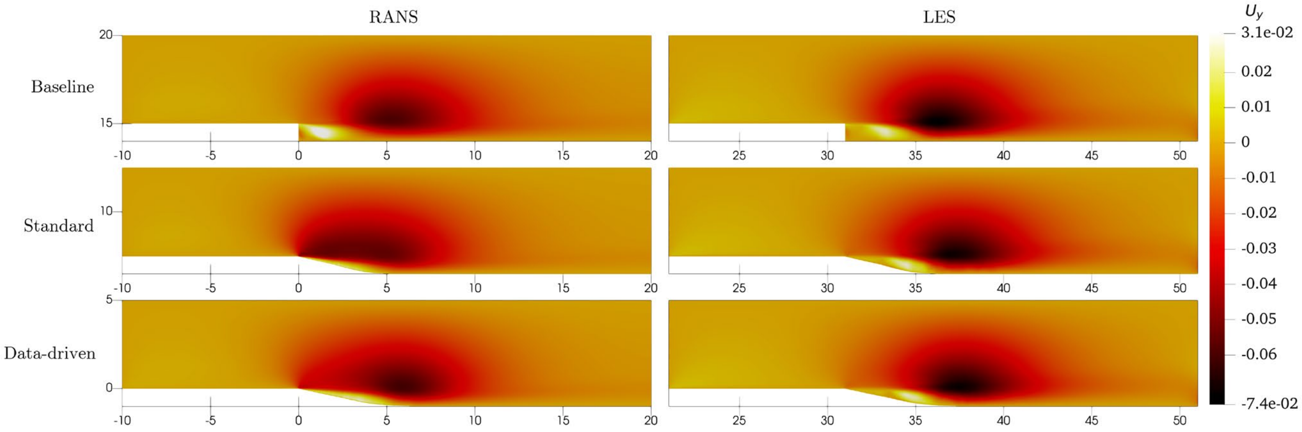

Fig. 13 Plots of the *y*-component of velocity for each of the baseline, standard and data-driven geometries when run using both body fitted RANS and LES

300 optimisation iterations are required, the converged design is obtained in just 5 min. Adding in the data-driven terms slows down the optimisation, but solutions are still generated in around 3.5 RANS times. The GEP regressions take slightly more effort as 50 solutions are required for ensemble averaging, but these are still obtained in just 1.7 core-h of training (25 min using 4 cores). The bottleneck in the method is running the high fidelity simulations, as the computational costs of the other steps are negligible in comparison. The total cost of the method is then approximated to the order of 3 high fidelity simulations (for the baseline, standard, and data-driven geometries).

Attempting to implement a high fidelity method directly into the optimisation framework would require a fully converged, time averaged simulation with each iteration of the optimisation solver. This would result in N high fidelity simulations, where N is the number of optimisation steps required for convergence, which can easily be of the order $O(10^2)$. The present method then provides a computational saving of around two orders of magnitude, demonstrating why data-driven turbulence models are required. They enable high fidelity level turbulence treatment to be introduced to optimisation methods for the first time.

4 Testing using U-bend

4.1 U-bend case details

The chosen test case to evaluate the performance gains of the data-driven optimisation method is a 2D u-bend configuration. Figure 14 shows a schematic of the case. The dashed line depicts the outside bend for the baseline geometry which forms a standard u-bend shape. It is against this baseline geometry that the optimisations will be compared, rather than the full-square optimisation region. Further, the inlet and outlet ducts are excluded from the optimisation domain.

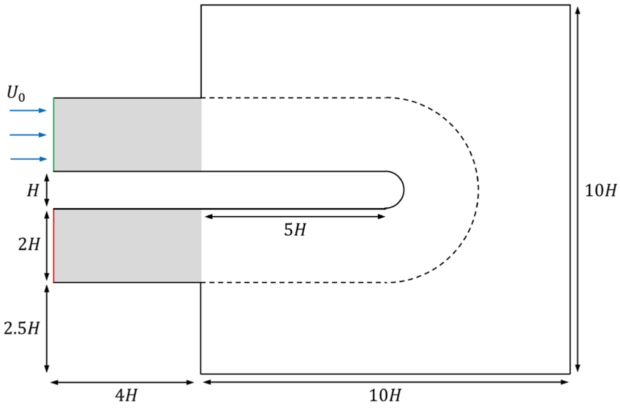

Fig. 14 Schematic of the u-bend test case. The inlet is coloured green and the outlet coloured red. No-slip walls are indicated by a black solid line. The dashed line shows the shape of the baseline u-bend against which optimisation results are compared. The area shaded grey is not considered as part of the optimisation domain. (Color figure online)

The turbulent inlet velocity profile is adapted from the DNS channel flow profile obtained by Kim et al. (1987) to fit an inlet Reynolds no. based on half the inlet region height H and the bulk inlet velocity U_0 of $Re_H = 5000$. In the same manner as for the backward-facing step, random fluctuations $\boldsymbol{u}'$ are superimposed on the mean velocity profile for all high fidelity simulations such that they match the DNS Reynolds stresses and a no-slip condition is applied at all walls. For high fidelity and RANS-based calculations, respectively, an advective and a zero gradient condition are applied at the outlet. The outlet pressure is atmospheric and zero pressure gradient is imposed at the inlet and walls. All models and solution methods for LES and RANS are the same as for the backwards-facing step.

In the optimisations, $\alpha_{\max}$ was kept at 2000 as for the backward-facing step but λ was increased to 2×10^6 to accelerate the convergence. All high fidelity data post-processing steps and the configuration and running of GEP regressions are the same as for the backward-facing step and will not be reiterated here.

The domain for the optimisations is meshed using a structured mesh consisting of $106,000$ hexahedral elements. At the walls in the inlet and outlet section and around the inside of the u-bend, the first cell is again placed within the range $y^+ < 1$. Walls on the outer section are not refined as far as they are not expected to feature in the optimised designs. The baseline geometry is meshed with structured hexahedral elements and consist of $190,000$ elements in the xy-plane. For the high fidelity LES, a further 40 elements are also given in the z-direction with cyclic boundary conditions imposed at the front and back walls, resulting in a total element count of $7,600,000$.

4.2 Volume constraint study

Before testing the data-driven method on the u-bend case a volume constraint study is performed. It is considered important for real-world applications that the method is compatible with volume constraints. The data-driven method needs to be able to find improved designs even when constrained to use the same fluid fraction as the standard optimisation. Five values of prescribed fluid fraction are tested using the standard optimisation, and the data-driven method is then applied to the fluid fraction giving the optimum results for pressure drop in the following section.

The fluid fractions and corresponding volume constraint parameters used are presented in Table 4. The fluid fraction

Table 4 Volume constraint parameters

Fluid fraction, f	γ^0	w_C^0	$w_C^{\max}$	η
0.41	0	10.0	10000	1.1
0.50	0	10.0	10000	1.1
0.60	0	10.0	10000	1.1
0.70	0	1.0	10000	1.03
0.80	0	1.0	10000	1.03

$f = 0.41$ corresponds to the fluid fraction present in the baseline design and then increments of 0.1 are considered up to $f = 0.8$. A lower initial weighting and growth rate was used for the fluid fractions $f = 0.7$ and 0.8 as less forcing was required to satisfy the volume constraints in these cases. Higher weighting early in the optimisation also leads to deteriorated performance for these fluid fractions.

The resulting optimisation histories for the objective functions and fluid fractions are presented in Fig. 15. The corresponding optimised geometries plotted with streamlines coloured by velocity magnitude are shown in Fig. 16. Here, a RANS calculation on the optimisation grid with an impermeability field the same as the baseline design is also presented.

Compared to this baseline design, it is apparent from examining the streamlines that all optimisations act to remove separated regions in the flow, and from the objective function histories, it is clear that improved designs are obtained with a larger fluid fraction. This appears to be in part due to the generation of splitters which are present on all optimised designs other than the baseline fluid fraction where $f = 0.41$. The splitters help turn the flow around the u-bend and also reduce the velocity gradients around the inside of the bend helping reduce the pressure loss. Streamlines are also shown on each of the designs to confirm that the maximum impermeability used is high enough and that the fluid does not penetrate the solid regions. This is especially important in the designs where splitters are generated to check that there is not flow passing through the centre of the splitters.

The fluid fraction $f = 0.8$ marginally achieves the lowest value of pressure drop. However, when considering the final design in Fig. 16, it is apparent that the impermeability in the solid part of the domain has not been forced to $\alpha_{\max}$. This is a result of the optimisation attempting to generate a design with a lower fluid fraction. To satisfy the volume constraint, the optimisation acts by reducing the impermeability across the whole solid region, rather than by expanding the fluid region. The extra fluid fraction then comes from partially solidified material rather than true fluid region. The unconverged fluid fraction history for $f = 0.8$ in Fig. 15 confirms this observation. As such, this optimisation is discarded and we proceed with the data-driven method using a fluid fraction $f = 0.7$.

4.3 U-bend results

4.3.1 Geometry extraction and remeshing

To proceed with the data-driven optimzation, a high fidelity simulation of the standard optimised geometry with a prescribed fluid fraction $f = 0.7$ needs to be run. The geometry extraction and smoothing steps are the same as for the backward-facing step and the raw and smoothed

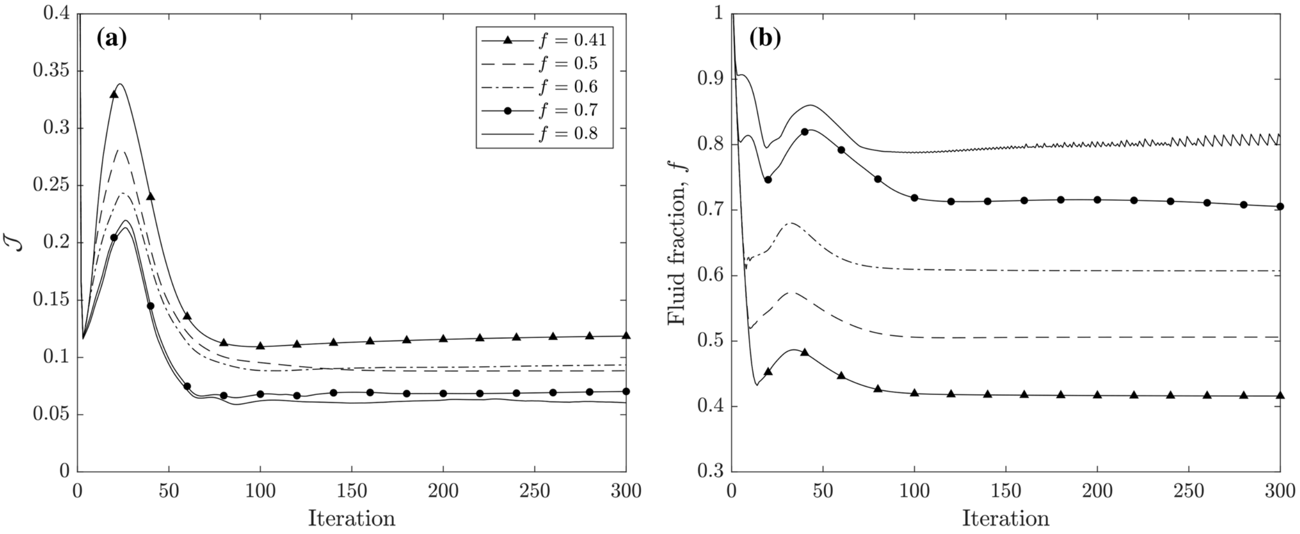

Fig. 15 Convergence histories of **a** the objective functions and **b** fluid fractions, for each of the prescribed fluid fractions considered in the volume constraint study

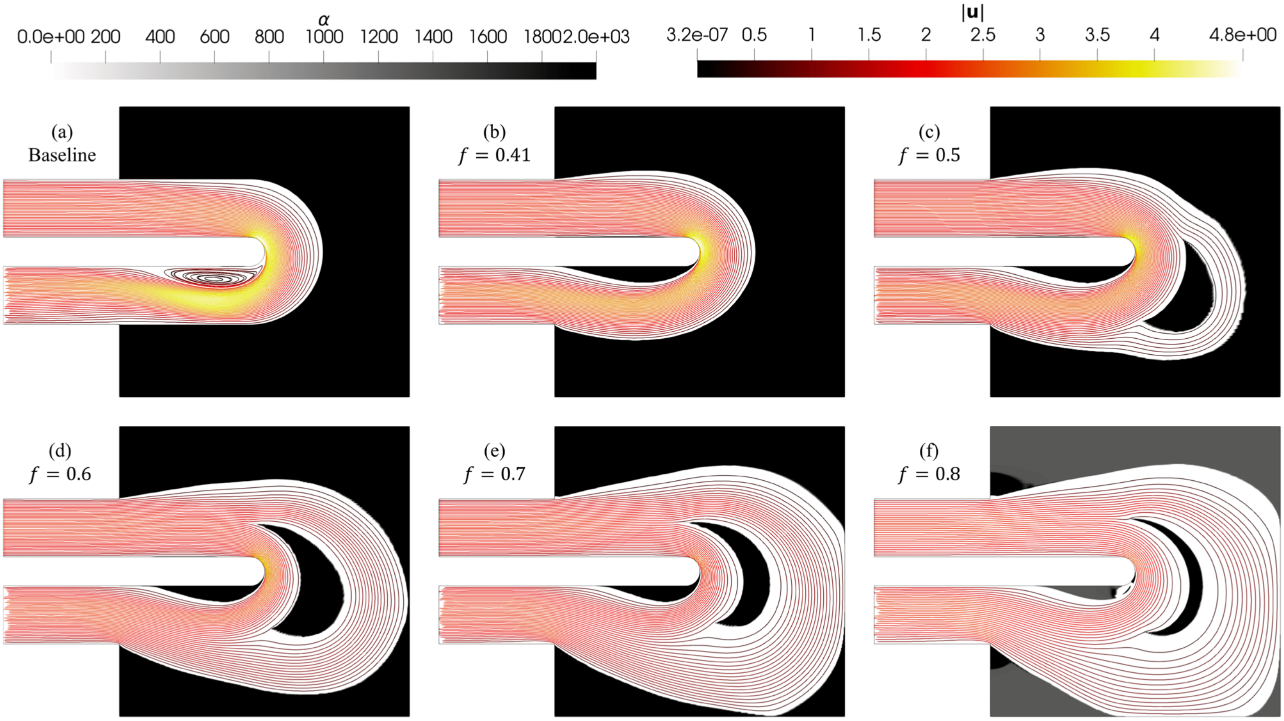

Fig. 16 Optimised geometries resulting from standard optimisations with volume constraints enforced, with streamlines coloured by velocity magnitude plotted. Each optimisation is labelled with it is corresponding prescribed fluid fraction. A RANS calculation with the impermeability field set to the baseline design is also presented

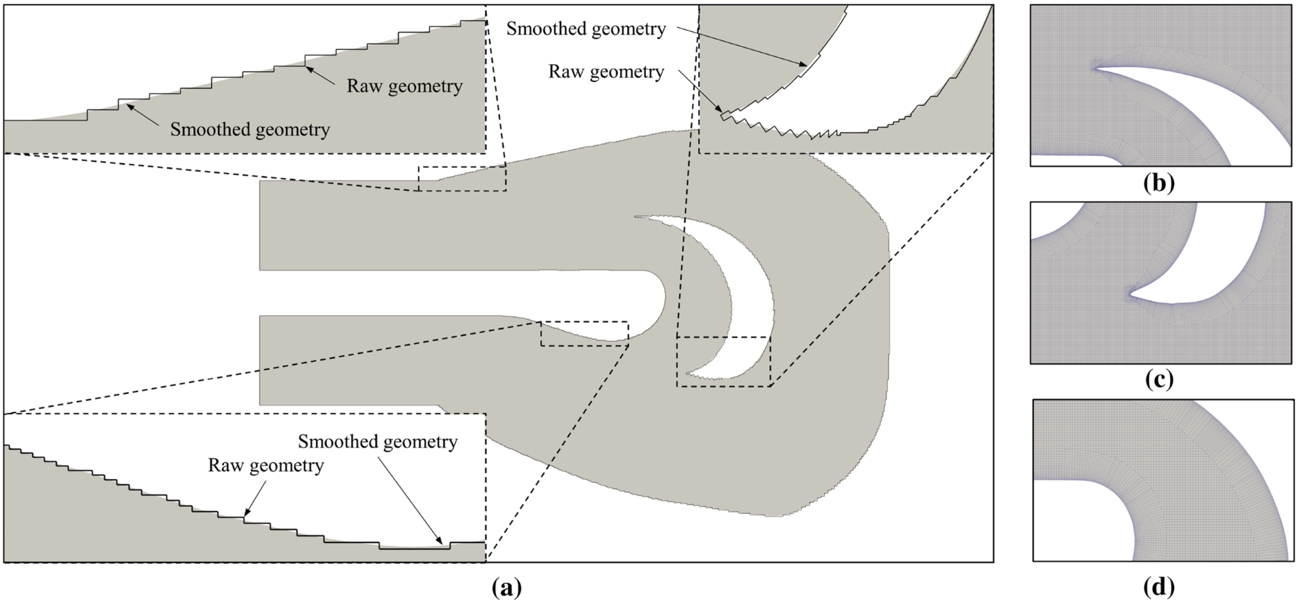

Fig. 17 **a** Details of the raw and smoothed geometries resulting from the geometry extraction and remeshing operations for the standard optimisation. **b**–**d** Plots of the body-fitted mesh showing the structured hexahedral mesh in the bulk and prism layer refinement at the walls

geometries resulting from this process are shown in Fig. 17a. The smoothing operation is repeated 30 to produce the smoothed geometry. The generated mesh consists of structured hexahedral elements in the bulk of the domain with 25 prism layer cells used at the walls to resolve the boundary layer and ensure $y^+ < 1$ at the wall. This resulted in a 2D mesh with 168, 000 cells, where 40 cells are also place in the z-direction when running LES, giving a total of 6, 720, 000 cells. Fig. 17b–d shows the mesh around the splitter and the inner wall.

4.3.2 GEP regressions

Training data from the time averaged results of the high fidelity LES was then extracted to learn the data-driven model for the u-bend case. Applying GEP to this training data generated the following model for the anisotropy (here presented to first order only)

$$\begin{aligned}\mathbf{a}^{\mathrm{GEP}} = &(-0.465 + 0.213I_1 - 0.005I_2)\mathbf{V}^{(1)}\\ &+ (0.896 - 0.403I_1 + 0.273I_2)\mathbf{V}^{(2)}\\ &+ (0.820 - 0.290I_1 - 0.018I_2)\mathbf{V}^{(3)}.\end{aligned} \tag{43}$$

It is interesting to compare this model with the GEP closure found for the backward-facing step case given in Eq. 42. The main point of note is the magnitude of the constant coefficient of $\mathbf{V}^{(1)}$. Whilst the backward-facing step model had a constant coefficient of close to −1, the u-bend has a value of −0.465. This suggests that the eddy viscosity model over predicts the additional turbulent viscosity across the domain, and the GEP model developed here acts to reduce it.

4.3.3 Topology optimisations

Applying Eq. 43 in the data-driven optimisation results in the geometry shown on the right in Fig. 18 and the convergence of the standard and data-driven optimisations is presented in Fig. 19. The first point to note is that the topologies generated by both optimisations are the same, this is an important feature as it helps give confidence that the data-driven closure is still presented with the same flow physics and the model is not being extrapolated beyond its capabilities. Beyond this, there are several key differences in the generated geometries. First, the data-driven geometry provides a flatter expansion of the outer wall for the flow entering from the inlet duct. The splitter itself on the data-driven geometry is more evenly distributed around the bend and presents a slight elongation of the solid region cutting out the separated region on the inside of the bend. These differences lead to the reduced converged value of pressure drop obtained in Fig. 19.

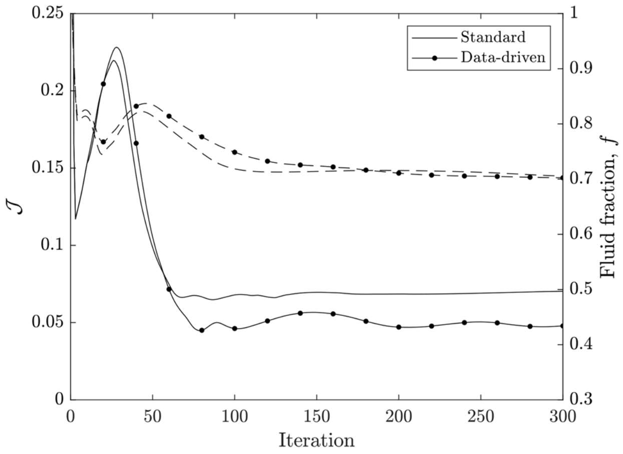

Fig. 19 Objective function (solid) and fluid fraction (dashed) optimisation histories for the standard and data-driven optimisations

4.3.4 High fidelity results

The effects of these differences in geometry are realised by looking at the pressure drop results in Table 5. The direct output from the optimisations predict improvements of 36.1% and 56.5% for the standard and data-driven respectively over the baseline u-bend. When RANS calculations were run with body fitted meshes the results become even more promising with improvements of 57.5% and 64.2%, respectively. Finally, when the geometries were tested with LES the standard geometry improved on the baseline by

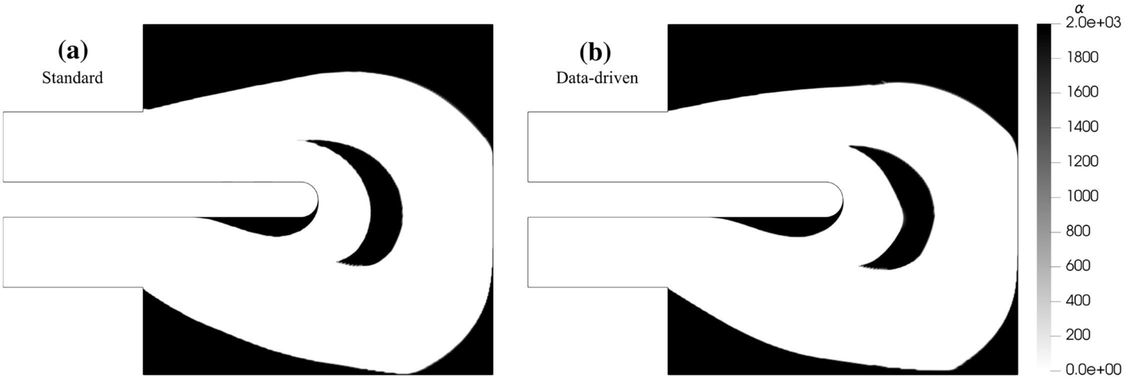

Fig. 18 Resulting geometries from **a** the standard FTO and **b** the data-driven FTO of the u-bend test case

Table 5 Pressure drop results for the u-bend optimisation

Geometry	RANS (TO)	RANS (body fit)	LES
Baseline	–	0.1100	0.1470
Standard	0.0703	0.0468	0.0526
Data-driven	0.0478	0.0394	0.0455

64.2% and the data-driven optimisation improved even further by 69.0%.

To investigate how the data-driven optimisation has been able to discover this low resistance pathway around the u-bend, we first look at Fig. 20a which shows the time averaged velocities obtained from LES of the standard (left) and data-driven (right) geometries. Separation on the suction side of the splitter for the data-driven geometry is significantly delayed in comparison to the standard. This results in a much smaller separation bubble close to the trailing edge and also allows the flow to exit the splitter section at an angle closer to that of the outlet duct. This reduced angle then allows the separated flow on the inside of the u-bend to reattach sooner, reducing the size of this separated region, and also reduces the small separation region as the flow enters the outlet duct. Each of these flow features have the effect of reducing the drag and, thus, the pressure loss of the fluid in the data-driven geometry.

To visualise more clearly the reduction in turbulence produced by minimising separation in the data-driven geometry, Fig. 20b shows the turbulent kinetic energy. Clearly in the regions mentioned, the kinetic energy is at much higher levels in the standard geometry than the data-driven geometry. This effect is most pronounced around the splitters trailing edge separation bubble and the inside wall of the u-bend.

Figure 20c shows the effect the previously described features have on the total pressure. The reduced loss regions can be seen in the separation behind the splitter and around the inside wall of the u-bend and these lead to a lower value of total pressure at the inlet, representing the observed reduced total pressure loss.

4.3.5 Computational cost

Finally, a breakdown of the computational cost of each stage of the data-driven optimisation method is presented for the u-bend case in Table 6. The conclusions drawn are the same as for the backward-facing step; the bottleneck in the method

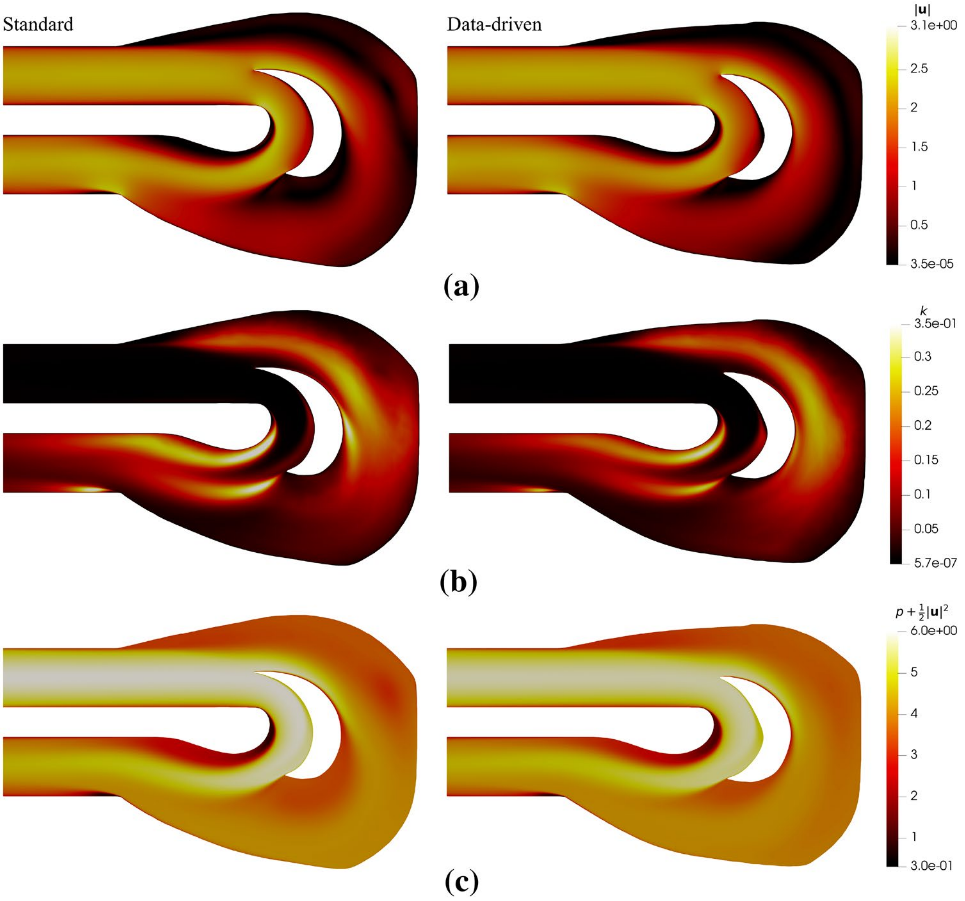

Fig. 20 Plots of time averaged data for the high fidelity simulations on the standard (left) and data-driven (right) geometries for **a** velocity magnitude, **b** turbulent kinetic energy and **c** total pressure

Table 6 Details of computational cost for the u-bend data-driven optimisation testing

Optimisation step	Core-h	No. cores	CPU model	Clock speed/ GHz
Standard FTO	1.0	4	Intel Core i7-9700	3.0
Data-driven FTO	1.8	4	Intel Core i7-9700	3.0
GEP regressions	1.9	4	Intel Core i7-9700	3.0
RANS (standard)	0.7	4	Intel Core i7-9700	3.0
LES (standard)	2085	32	AMD EPYC 7724	2.8

is the running of high fidelity LES simulations, whilst the optimisations and model training are computationally cheap in comparison. The total computational effort can again be approximated as that of 3 high fidelity simulations, demonstrating the ability to generate optimised designs representative of high fidelity turbulence treatment for the first time.

5 Conclusions

A novel data-driven technique has been developed to improve the treatment of turbulence in fluid topology optimisation. The data-driven aspect allows the optimisation solver to mimic high fidelity simulations whilst maintaining low computational cost. This was achieved by deriving the full adjoint system, for the corresponding primal RANS equations closed with a fully general explicit algebraic stress model.

The technique was first verified on a pressure drop optimisation across a backward-facing step geometry. The case presents a situation where the physics is simplified and some intuition of the expected result was possible, allowing confirmation of the correct operation of the method. This verification showed the potential performance gain available from the technique when tested using high fidelity simulations and also highlighted some of the limitations of standard fluid topology optimisation solvers. In fact, in this case, standard optimisations were shown to have detrimental performance when tested in high fidelity simulations, compared to the baseline design.

The technique was then tested on a u-bend optimisation for pressure drop which presents a more complex case where optimised geometries are hard to predict by physical intuition and standard turbulence models are more likely to suffer lower fidelity prediction. For this case, the data-driven optimisation was able to reduce the pressure drop through the domain by 69.0% over the baseline u-bend in comparison to 64.2% for the standard optimisation.

It is hoped that this work may serve as an example of how high fidelity methods can be utilised when considering optimisation problems previously limited to lower fidelity, computationally cheap methods.

Acknowledgements This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) Grant EP/L016230/1. The authors would also like to thank R. D. Sandberg for his continued advice and support.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Replication of results The implementation in the OpenFOAM libraries is taken directly from the description provided of the equations and numerical implementation in this article. This is possible as OpenFOAM provides a high level interface through which the governing equations may be written.

References

Alexandersen J, Andreasen CS (2020) A review of topology optimisation for fluid-based problems. Fluids 5(1):1–32. https://doi.org/10.3390/fluids5010029

Bendsøe MP, Kikuchi N (1988) Generating optimal topologies in structural design using a homogenization method. Comput Methods Appl Mech Eng 71:197–224

Borrvall T, Petersson J (2003) Topology optimization of fluids in Stokes flow. Int J Numer Methods Fluids 41(1):77–107. https://doi.org/10.1002/fld.426

Dilgen CB, Dilgen SB, Fuhrman DR, Sigmund O, Lazarov BS (2018a) Topology optimization of turbulent flows. Comput Methods Appl Mech Eng 331:363–393. https://doi.org/10.1016/j.cma.2017.11.029

Dilgen SB, Dilgen CB, Fuhrman DR, Sigmund O, Lazarov BS (2018b) Density based topology optimization of turbulent flow heat transfer systems. Struct Multidisc Optim 57(5):1905–1918. https://doi.org/10.1007/s00158-018-1967-6

Duraisamy K, Iaccarino G, Xiao H (2019) Turbulence modeling in the age of data. Annu Rev Fluid Mech 51(1):357–377. https://doi.org/10.1146/annurev-fluid-010518-040547

Durbin PA (2018) Some recent developments in turbulence closure modeling. Annu Rev Fluid Mech 50(1):77–103. https://doi.org/10.1146/annurev-fluid-122316-045020

Ferreira C (2001) Gene expression programming: a new adaptive algorithm for solving problems. Complex Syst 13(2):87–129

Giles MB, Pierce NA (2000) An introduction to the adjoint approach to design. Flow Turbulence Combust 65(3–4):393–415. https://doi.org/10.1023/A:1011430410075

Hazra SB (2008) Multigrid one-shot method for aerodynamic shape optimization. J Sci Comput 30(3):1527–1547

Kim J, Moin P, Moser R (1987) Turbulence statistics in fully developed channel flow at low reynolds number. J Fluid Mech 177:133–166. https://doi.org/10.1017/S0022112087000892

Kontoleontos EA, Papoutsis-Kiachagias EM, Zymaris AS, Papadimitriou DI, Giannakoglou KC (2013) Adjoint-based constrained topology optimization for viscous flows, including heat transfer. Eng Optim 45(8):941–961. https://doi.org/10.1080/0305215X.2012.717074

Koza JR (1992) Genetic programming: on the programming of computers by means of natural selection. MIT Press, Cambridge

Le H, Moin P, Kim J (1997) Direct numerical simulation of turbulent flow over a backward-facing step. J Fluid Mech 330:349–374. https://doi.org/10.1017/S0022112096003941

Othmer C (2008) A continuous adjoint formulation for the computation of topological and surface sensitivities of ducted flows. Int J Numer Methods Fluids 58:861–877. https://doi.org/10.1002/fld.1770

Papoutsis-Kiachagias EM, Giannakoglou KC (2016) Continuous adjoint methods for turbulent flows, applied to shape and topology optimization: industrial applications. Arch Comput Methods Eng 23(2):255–299. https://doi.org/10.1007/s11831-014-9141-9

Pichler R, Sandberg RD, Michelassi V, Bhaskaran R (2016) Investigation of the accuracy of RANS models to predict the flow through a low-pressure turbine. J Turbomach. https://doi.org/10.1115/1.4033507

Pietropaoli M, Ahlfeld R, Montomoli F, Ciani A, D'Ercole M (2016) Design for additive manufacturing: internal channel optimization. Heat Transf. https://doi.org/10.1115/GT2016-57318

Pope SB (1975) A more general effective-viscosity hypothesis. J Fluid Mech 72:331–340. https://doi.org/10.1017/S0022112075003382

Rodi W (1976) A new algebraic relation for calculating the Reynolds stresses. Gesellschaft Angewandte Mathematik und Mechanik. https://doi.org/10.1002/zamm.19765613093

Spalart PR (1988) Direct simulation of a turbulent boundary layer up to R$\theta = 1410$. J Fluid Mech 187:61–98. https://doi.org/10.1017/S0022112088000345

Spalart PR, Shur ML, Strelets MK, Travin AK (2015) Direct simulation and RANS modelling of a vortex generator flow. Flow Turbulence Combust 95(2–3):335–350. https://doi.org/10.1007/s10494-015-9610-8

Spencer AJ, Rivlin RS (1958) The theory of matrix polynomials and its application to the mechanics of isotropic continua. Arch Ration Mech Anal 2(1):309–336. https://doi.org/10.1007/BF00277933

Weatheritt J, Sandberg RD (2016) A novel evolutionary algorithm applied to algebraic modifications of the RANS stress–strain relationship. J Comput Phys 325:22–37. https://doi.org/10.1016/j.jcp.2016.08.015

Weatheritt J, Pichler R, Sandberg RD, Laskowski G, Michelassi V (2017) Machine learning for turbulence model development using a high-fidelity HPT cascade simulation. In: Turbomachinery technical conference and exposition, Charlotte, NC, pp 1–12. https://doi.org/10.1115/gt2017-63497

Wilcox DC (1998) Turbulence modeling for CFD, 2nd edn. DCW Industries, La Canada

Wilcox DC (2006) Turbulence modeling for CFD, 3rd edn. DCW Industries, La Canada

Wu JL, Xiao H, Sun R, Wang Q (2019) Reynolds-averaged Navier–Stokes equations with explicit data-driven Reynolds stress closure can be ill-conditioned. J Fluid Mech 869:553–586. https://doi.org/10.1017/jfm.2019.205

Yoon GH (2016) Topology optimization for turbulent flow with Spalart–Allmaras model. Comput Methods Appl Mech Eng 303:288–311. https://doi.org/10.1016/j.cma.2016.01.014

Zhang Y, Dwight RP, Schmelzer M, Gómez JF, hua Han Z, Hickel S (2021) Customized data-driven RANS closures for bi-fidelity LES-RANS optimization. J Comput Phys 432:110153. https://doi.org/10.1016/j.jcp.2021.110153

Zymaris AS, Papadimitriou DI, Giannakoglou KC, Othmer C (2009) Continuous adjoint approach to the Spalart–Allmaras turbulence model for incompressible flows. Comput Fluids 38:1528–1538. https://doi.org/10.1016/j.compfluid.2008.12.006