

***adjointOptimisationFoam*: AN OPENFOAM-BASED FRAMEWORK FOR ADJOINT-ASSISTED OPTIMISATION**

E.M. Papoutsis-Kiachagias¹, K.Th. Gkaragkounis¹, A.-S.I. Margetis¹, T. Skamagkis¹,
V.G. Asouti¹ and K.C. Giannakoglou¹

¹National Technical University of Athens,
Parallel CFD & Optimization Unit, School of Mechanical Engineering
Zografou Campus, 9 Iroon Polytechniou Str
e-mail: vpapout@mail.ntua.gr, kogkar@hotmail.com,
{amargetis, themistoklissk, vasouti, kgianna}@mail.ntua.gr

Abstract. *This paper overviews the capabilities of adjointOptimisationFoam, an OpenFOAM-based framework for adjoint-assisted, gradient-based optimisation that first appeared publicly in the open-source CFD toolbox, OpenFOAM, in v1906. Capabilities of the publicly available software and of the in-house version of it are separately discussed. The publicly available software is structured in a way that allows for an automated shape optimisation loop, including all steps from the solution of the flow equations to the update of the design variables in a single executable, avoiding thus the need for external scripting. The software has already been used in a number of industrial optimisation problems, some of which will briefly be presented herein. Additionally, adjointOptimisationFoam is currently developed and extended by the Parallel CFD & Optimization Unit of NTUA. Topics of active development include a) the continuous adjoint to unsteady flows, including data compression techniques to reduce the memory footprint, b) stabilisation techniques for the solution of the primal and adjoint equations based on the Recursive Bisection Method and the control of the Adjoint Transposed Convection term, c) a suite of tools for topology optimisation, with or without heat transfer, occasionally in the presence of two fluids, and d) a number of adjoint-assisted methods for Uncertainty Quantification and Robust Design Optimisation.*

Keywords: Aero/hydrodynamic Optimisation, Continuous Adjoint, OpenFOAM, open source, adjointOptimisationFoam

1 INTRODUCTION

The open-source CFD toolbox, OpenFOAM¹, is one of the largest open-source projects related to the solution of a number of computational problems pertaining to fluid mechanics, with solvers tackling compressible and incompressible, single and multi-phase flows, chemical reactions, heat transfer, species transport etc. Due to its versatility and capability to tackle a wide range of physical problems, it has attracted the attention of many industrial sectors, among which the automotive, energy and turbomachinery ones as well as industries related to combustion, plastics etc.

In the OpenFOAM release by OpenCFD, version v1906, OpenFOAM introduced a library supporting adjoint-based shape optimisation, developed by the Parallel CFD & Optimisation Unit of NTUA (PCOpt/NTUA) and contributed by the group of authors. This paper focuses on highlighting the capabilities of this publicly available software, presenting some indicative applications and discussing further developments that are currently being undertaken in-house. The main executable making use of the adjoint-based infrastructure is called *adjointOptimisationFoam*; this name will hereafter be used to refer to all adjoint-related OpenFOAM capabilities, both publicly available and in-house developed at PCOpt/NTUA. *adjointOptimisationFoam* has been continuously developed and maintained with publicly available contributions till v2012.

The publicly available software is structured in a way that allows for an automated shape optimisation loop, without the need for external scripting, incorporating a) the adjoint to steady-state incompressible flows, with full differentiation of the Spalart-Allmaras model with or without wall functions, b) a number of differentiated objective functions, like forces, moments and total pressure losses, c) a parameterisation scheme based on volumetric B-Splines that can also act as a grid displacement tool, d) two major families of methods for computing shape sensitivity derivatives based on either surface or field integrals and e) a number of methods for updating the design variables, like the one from Broyden-Fletcher-Goldfarb-Shanno (BFGS) and Sequential Quadratic Programming (SQP, used for constrained optimisation). The software has already been used in a number of industrial optimisation problems, some of which will be briefly presented herein.

At the same time, *adjointOptimisationFoam* is further developed and extended in-house by PCOpt/NTUA. Topics of active development include a) the continuous adjoint to unsteady flows, focusing on the usage of efficient compression algorithms to cut down on the sizable storage requirements associated with unsteady adjoint, b) stabilisation techniques, mainly focusing on the Recursive Projection Method, to tackle convergence challenges associated with small scale oscillations of the flow field in practical applications, c) a suite of tools for topology optimisation, including Conjugate Heat Transfer (CHT) and multiple fluids, for the design of heat exchangers and d) a number of adjoint-assisted methods for Uncertainty Quantification (UQ) and Robust Design Optimisation (RDO), including first-and second-order variants of the Method of Moments, an adjoint-assisted regression approach to the non-intrusive Polynomial Chaos Expansion (PCE) and the adjoint to the intrusive PCE variant. Some relevant applications features will be briefly presented herein.

2 THE ADJOINT METHOD IN BRIEF

The adjoint method is used to compute the sensitivity derivatives (SDs) $\delta J / \delta b_i$, where J is the objective function of the optimisation problem (e.g. lift/drag forces, total pressure losses,

¹www.openfoam.com

etc) and $b_i, i \in [1, N]$ the design variables, at a cost that does not scale with N . This allows for handling optimisation problems with a very large number of design variables, occasionally in the order of millions (see section 4.3).

The publicly available *adjointOptimisationFoam* solves the adjoint to incompressible, laminar and turbulent flows. Assuming turbulence is modelled by the Spalart–Allmaras model, [1], the flow (primal) equations read

$$R^p = -\frac{\partial v_j}{\partial x_j} = 0 \quad (1a)$$

$$R_i^v = v_j \frac{\partial v_i}{\partial x_j} - \frac{\partial \tau_{ij}}{\partial x_j} + \frac{\partial p}{\partial x_i} = 0, \quad i = 1, 2, 3 \quad (1b)$$

$$R^{\tilde{\nu}} = v_j \frac{\partial \tilde{\nu}}{\partial x_j} - \frac{\partial}{\partial x_j} \left[\left(\nu + \frac{\tilde{\nu}}{\sigma} \right) \frac{\partial \tilde{\nu}}{\partial x_j} \right] - \frac{c_{b2}}{\sigma} \left(\frac{\partial \tilde{\nu}}{\partial x_j} \right)^2 - \tilde{\nu} P(\tilde{\nu}) + \tilde{\nu} D(\tilde{\nu}) = 0 \quad (1c)$$

where v_i are the velocity components, p the static pressure divided by the constant density, ν and ν_t the constant bulk and turbulent viscosity, respectively, $\tau_{ij} = (\nu + \nu_t) \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right)$ the stress tensor components, $\tilde{\nu}$ is the turbulence model variable and Δ is the distance from the wall. Details about the terms and constants of eq. 1c can be found in [1]. To account for the distance variation with respect to (w.r.t) the design variables during the adjoint formulation for shape optimisation problems, [2], the Hamilton-Jacobi PDE is used for computing Δ

$$R^\Delta = \frac{\partial}{\partial x_j} \left(\frac{\partial \Delta}{\partial x_j} \Delta \right) - \Delta \frac{\partial^2 \Delta}{\partial x_j^2} - 1 = 0 \quad (2)$$

In industrial cases, the use of the law-of-the-wall (wall functions) is a common practice in order to avoid extremely stretched grids close to the solid walls and lower the computational cost. Working with a cell-centered finite volume discretisation scheme, Spalding's law is used to compute the friction velocity based on the velocity magnitude at the first cell-centre off the wall [3].

As already mentioned, the adjoint equations are formulated in a way that makes the final SD expression free of variations of the flow variables w.r.t. the design ones. To this end, an augmented objective function L is formed by adding the field integrals of the products of the flow PDEs and the adjoint variable fields to the objective function J , i.e.

$$L = J + \int_{\Omega} u_i R_i^v d\Omega + \int_{\Omega} q R^p d\Omega + \int_{\Omega} \tilde{\nu}_a R^{\tilde{\nu}} d\Omega + \int_{\Omega} \Delta_a R^\Delta d\Omega \quad (3)$$

where Ω is the computational domain, u_i the adjoint velocity components, q the adjoint pressure, $\tilde{\nu}_a$ the adjoint turbulence variable and Δ_a the adjoint distance. Given that, upon convergence, the residuals of the primal equations are zero, $L \equiv J$.

After differentiating eq. 3 w.r.t. b_n and developing it using the Gauss divergence theorem, the adjoint PDEs result by setting the multipliers of the derivatives of the flow variables w.r.t. b_n to

zero, [2], and read

$$R^q = -\frac{\partial u_j}{\partial x_j} = 0 \quad (4a)$$

$$R_i^u = u_j \frac{\partial v_j}{\partial x_i} - \frac{\partial(v_j u_i)}{\partial x_j} - \frac{\partial \tau_{ij}^a}{\partial x_j} + \frac{\partial q}{\partial x_i} + \tilde{\nu}_a \frac{\partial \tilde{\nu}}{\partial x_i} - \frac{\partial}{\partial x_l} \left(\tilde{\nu}_a \tilde{\nu} \frac{\mathcal{C}_Y}{Y} e_{mjk} \frac{\partial v_k}{\partial x_j} e_{mli} \right) = 0, \quad i = 1, 2, 3 \quad (4b)$$

$$R^{\tilde{\nu}_a} = -\frac{\partial(v_j \tilde{\nu}_a)}{\partial x_j} - \frac{\partial}{\partial x_j} \left[\left(\nu + \frac{\tilde{\nu}}{\sigma} \right) \frac{\partial \tilde{\nu}_a}{\partial x_j} \right] + \frac{1}{\sigma} \frac{\partial \tilde{\nu}_a}{\partial x_j} \frac{\partial \tilde{\nu}}{\partial x_j} + 2 \frac{c_{b2}}{\sigma} \frac{\partial}{\partial x_j} \left(\tilde{\nu}_a \frac{\partial \tilde{\nu}}{\partial x_j} \right) + \tilde{\nu}_a \tilde{\nu} \mathcal{C}_{\tilde{\nu}} + \frac{\partial \nu_t}{\partial \tilde{\nu}} \frac{\partial u_i}{\partial x_j} \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) + (-P + D) \tilde{\nu}_a = 0 \quad (4c)$$

$$R^{\Delta_a} = -2 \frac{\partial}{\partial x_j} \left(\Delta_a \frac{\partial \Delta}{\partial x_j} \right) + \tilde{\nu} \tilde{\nu}_a \mathcal{C}_{\Delta} = 0 \quad (4d)$$

where $\tau_{ij}^a = (\nu + \nu_t) \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right)$ are the adjoint stress tensor components. The $\mathcal{C}_{\tilde{\nu}}$, \mathcal{C}_Y and \mathcal{C}_{Δ} expressions can be found in [4, 5].

The adjoint boundary conditions are defined after zeroing the multipliers of the derivatives of the flow variables w.r.t. b_n in the boundary surface integrals emerging after differentiating eq. 3 and applying the Gauss divergence theorem. The detailed presentation of the adjoint boundary conditions can be found in [2], including the adjoint law of the wall for cases using wall functions.

The final SD expression depends on the design variables b_n (see, for instance, [6] for shape and [7] for topology optimisation). It is important to note that the eqs. 4a to 4c do not depend on b_n and can, thus, be used with any design variable, supporting for instance shape [2], flow control [8], topology optimisation [7] (after minor additions), etc.

The publicly available version of *adjointOptimisationFoam* includes the adjoint to the Spalart–Allmaras turbulence model and with or without wall functions, [4, 2]. The adjoint to a number of other turbulence models has also been developed in-house and can be found in [9, 8] for high- and low-Re variants of the $k - \epsilon$ turbulence model and in [10] for the $k - \omega$ SST one.

3 SHAPE OPTIMISATION

3.1 Publicly Available Functionality

In shape optimisation, the publicly available *adjointOptimisationFoam* can be used to compute only sensitivity maps or perform shape optimisation loops.

Sensitivity maps, like the one presented in fig. 1, are plots of the derivatives of the objective function w.r.t. the normal displacement of boundary wall nodes and may become a very useful tool for designers, since they offer insight into areas with great aero/hydrodynamic optimisation potential and mark the direction of favorable surface displacement. Hence, at the cost of a single flow and adjoint solution, the designer can obtain useful information about favorable shape deformations without even running an optimisation loop.

To create an automated shape optimisation loop, apart from solving the flow and adjoint equations, a number of additional steps need to be performed within each cycle, fig. 2. *adjointOptimisationFoam* incorporates all these steps within a single executable, without external scripts for managing the optimisation process.

Regarding parameterisation (which, among other, determines the design variables controlling the aero/hydrodynamic shape), *adjointOptimisationFoam* makes use of volumetric (i.e. trivari-

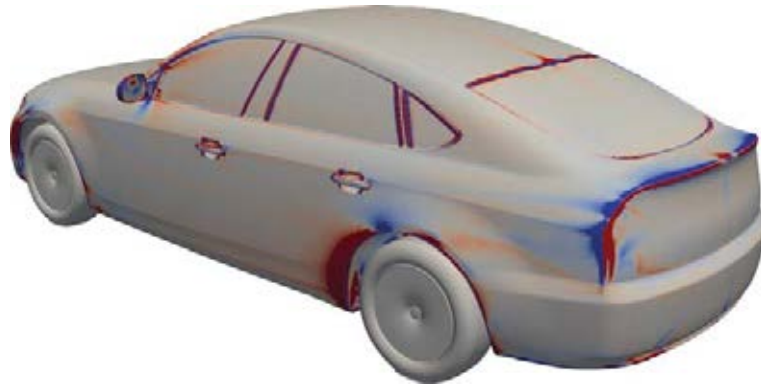


Figure 1: Drag sensitivity map computed on the surface of the DrivAer car model. Blue/red areas should be displaced inwards/outwards to reduce drag. All other areas have a negligible effect on drag.

ate) B-Splines; this is based on structured grids of control points, the coordinates of which act as the design variables, fig. 3. More details about the mathematical background of the parameterisation tool can be found in [11].

Regarding the available objective functions, *adjointOptimisationFoam* can handle forces, moments, total pressure losses and a surrogate objective for noise minimisation, see [11].

Sensitivity derivatives for shape optimisation can be computed using two families of methods, based on either surface or field integrals, namely the (E-)SI and FI adjoints developed in [6]. The E-SI approach includes also the adjoint to the grid displacement model, taking into consideration the effect of the so-called grid sensitivities.

The design variables can be updated using a number of methods, including among other the conjugate gradient and (L)BFGS, [12], ones for unconstrained optimisation problems as well as Rosen's projection method, [13], and SQP, [12], for optimisation problems including constraints. All the above-mentioned objective functions can also act as constraints (for instance, by setting target or threshold values for them). Additionally, some geometric constraints, like the area/volume of an aerodynamic shape, can also be imposed.

Finally, all grid displacement methods already available in OpenFOAM can be used to adapt the internal mesh points to the new boundary designed by the optimisation loop. Experience with a number of industrial optimisation problems has shown that the volumetric B-Splines used to parameterise surfaces can also act as a robust grid displacement method and are, hence, usually preferred to other methods.

An indicative application of *adjointOptimisationFoam*, studied in [5], deals with the multi-point, multi-objective optimisation of the "FP01" concept car designed by the Toyota aerodynamics department, depicted in fig. 3. To investigate the trade-off between drag and side-wind sensitivity, a two operating point design problem is defined and solved. The two operating points correspond to two flow directions (0° and 30° side-wind); each case is associated with its own objective function. The two functions being minimised are the drag coefficient at 0° and the yaw moment coefficient at 30° side-wind. The parameterisation of the spoiler and diffuser regions with volumetric B-Splines boxes is depicted in fig. 3. The Pareto front of non-dominated solutions, computed by combining the two objective functions with different weight value-sets and optimising anew, is presented in fig. 4, along with the optimised car geometries corresponding to a number of Pareto front members. The optimisation was based on the (steady-state) RANS equations for the CPU cost to be affordable. However, some of the optimised geometries were

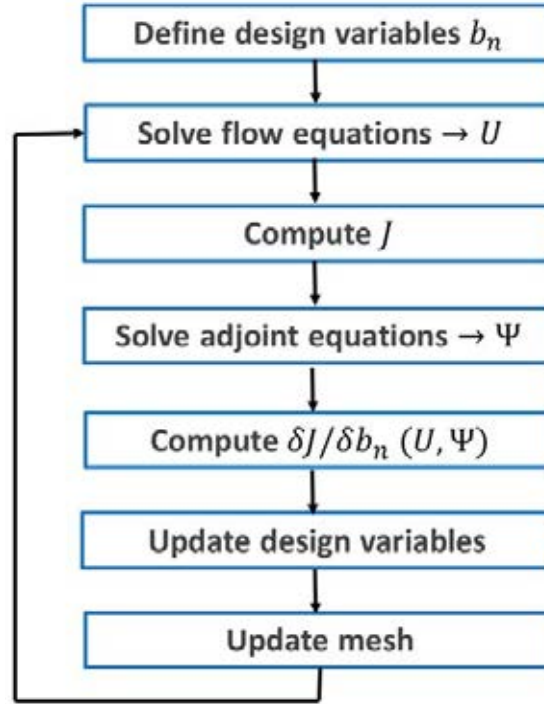


Figure 2: Flowchart of a typical shape optimisation loop. U stands for the primal variables (unknowns in eqs. 1b and 1c). Ψ are the adjoint variable fields, i.e. solutions to eqs. 4.

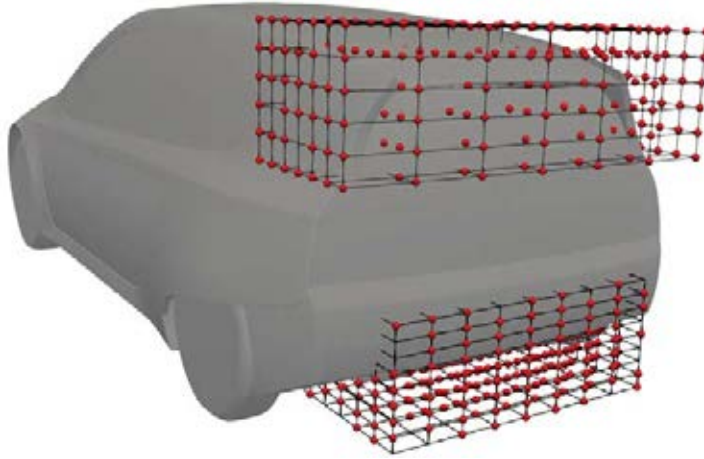


Figure 3: Two morphing boxes parameterising the spoiler and diffuser areas of a concept car. Case studied in [5]. The location of the morphing boxes has been found after computing sensitivity maps for the two objectives.

re-evaluated using DDES, [5]; it was observed that even though the quantitative reduction of the objective functions was different for the RANS- and DDES-based evaluations, the geometries designed using the RANS equations were indeed better than the baseline car even when re-evaluated with DDES.

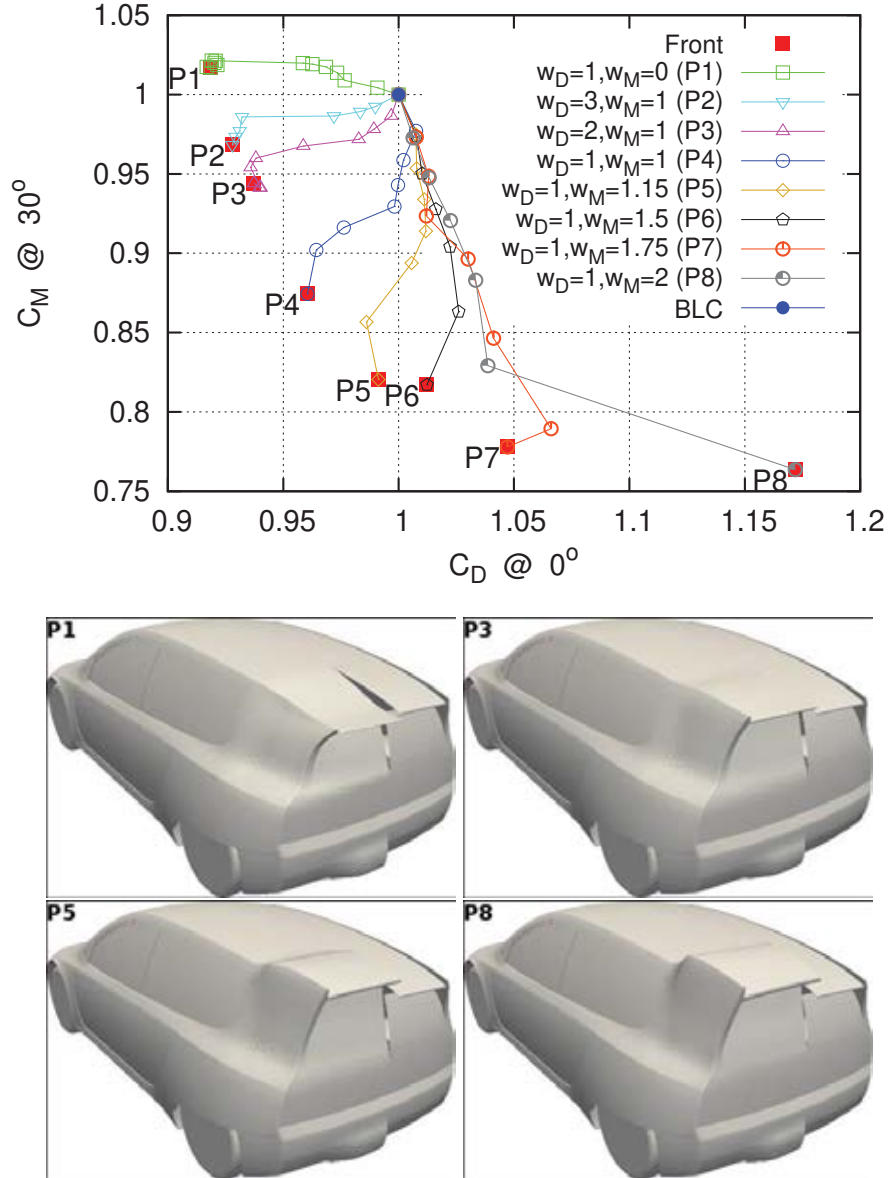


Figure 4: Shape optimisation of the FP01 concept car: Top: Front of non-dominated solutions (filled, red squares) and convergence paths of the optimisation runs carried out using different sets of weights for the drag coefficient (plotted in the abscissa) and the yaw moment coefficient (shown in the ordinate). All values have been normalised w.r.t. the baseline geometry. Mid and bottom: optimised geometries (port side) compared to the baseline car (starboard side), corresponding to four Pareto front members. Case studied in [5].

3.2 In-house shape optimisation developments

The publicly available shape optimisation infrastructure is enriched in-house with a number of additional objectives and parameterisation schemes/design variables. Indicatively, shape optimisation can be performed using the displacement of all wall nodes as the design variables. To ensure the surface smoothness while using such a high number of design variables, the boundary displacement is smoothed using a Helmholtz-type PDE, as in [14].

An indicative application pertaining to the optimisation of a Kaplan-type turbine runner is presented in fig. 5. The objective function is to minimise the area of the runner surface with a pressure below a certain threshold, to avoid cavitation. Since, in gradient-based optimisation, objective functions must be differentiable, the minimum of the pressure field is replaced by a differentiable sigmoid function, as in [15]. As seen from fig. 5, the cavitation-prone areas have practically disappeared after 40 cycles.

The in-house functionality of *adjointOptimisationFoam* has also been extended to include the adjoint method supporting CHT shape optimisation, see [16].

4 ADDITIONAL FEATURES - EXTENSIONS

adjointOptimisationFoam is actively being developed at PCOP/NTUA, extending the publicly available features and adding new ones. In this section, some of these features are briefly discussed and indicative applications are showcased.

4.1 Compression methods for unsteady adjoint simulations

In gradient-based optimisation with unsteady flows, the adjoint equations must be integrated backwards in time; this requires the instantaneous flow fields to be available at each time-step of the adjoint solver. Storing the entire flow history is usually infeasible for practical applications due to the immense memory requirements. The latter can be alleviated using the so-called check-pointing approach, [17], which stores a number of flow solutions (the check-points) along the simulation time span; then, the flow solution at any time-step other than a check-point is retrieved by integrating the flow equations starting from the nearest check-point, while re-using check-points that become idle. Though check-pointing can reduce storage requirements, the additional CPU cost for recomputing intermediate flow fields might not be negligible at all.

Lately, a compressed full storage strategy for unsteady adjoint-based optimisation problems has been implemented and proved to exhibit great benefits compared to the memory requirements of a (uncompressed) full storage, by avoiding at the same time the flow recomputations of check-pointing and maintaining the SD accuracy. The compression of the computed flow fields at each time-step is performed using a combination of the ZFP lossy compression algorithm, [18], and an incremental variant of Proper Generalised Decomposition (iPGD) [19]. Initial results show that an efficient combination of the two approaches can lead to a compression ratio of the order of 100-1000, [20], allowing the compressed full storage of the flow fields and avoiding, thus, the use of check-pointing and the associated extra CPU cost.

Indicatively, fig. 6 presents an adjoint-based optimisation of the fairing of the motorbike tutorial of OpenFOAM, based on the URANS equations. The objective to be minimised is the time-averaged drag exerted on the whole motorbike, fig. 6. The solver runs with $\Delta t = 2.5 \cdot 10^{-4} \text{ sec}$ for a total time of 7 sec on a grid of $\sim 1.1 \cdot 10^6$ cells. The optimisation is performed on two computational nodes with 128 GB memory each. A reduction in the memory footprint by a factor of 480 was achieved, with a negligible error in the computed sensitivities, fig. 7.

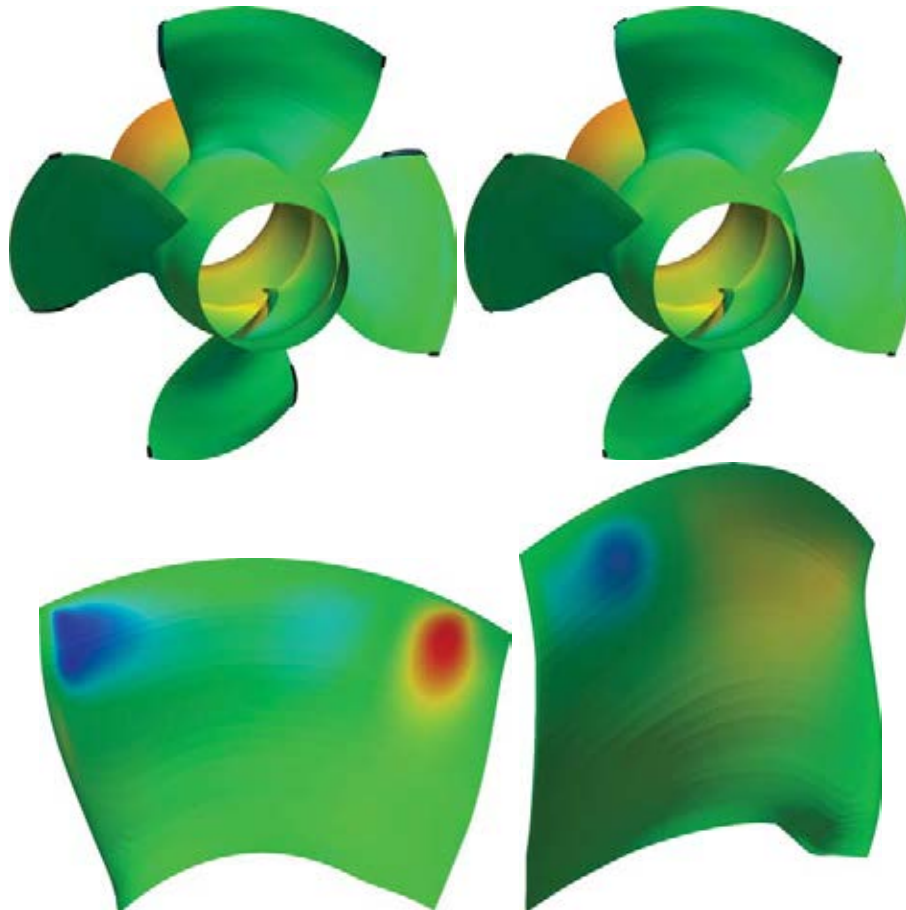


Figure 5: Kaplan-type turbine runner: optimisation targeting suppression of cavitation. Top: pressure distribution over the initial (left) and optimised (right) geometries, as seen from the outlet of the runner/suction side of the runner blades. The runner areas with a pressure below the defined threshold are contained within the black lines (areas close to the leading edge of the suction side and the trailing edge of the pressure side). The cavitation-prone areas have practically disappeared in the optimised geometry. Bottom: cumulative normal displacement of the optimised blade, plotted over the suction (left) and pressure (right) sides. Red areas have been displaced “inwards” while blue areas have moved in the opposite direction. It is interesting to note that even the pressure side has been slightly displaced in an attempt to eliminate the cavitation-prone area close to the trailing edge. The geometry is a courtesy of Andritz Hydro.



Figure 6: Case 4. Initial (left) and optimised (right) shape of the motorbike's fairing.

<i>Compression Method</i>	<i>CPU cost</i>		CR	<i>Memory Size(GB)</i>	ϵ
	<i>abs(h)</i>	<i>rel</i>			
<i>check-pointing</i>	50.6	100%		218	
<i>iPGDZ</i>	35.6	70.4%	479.1	4.25	0.59%

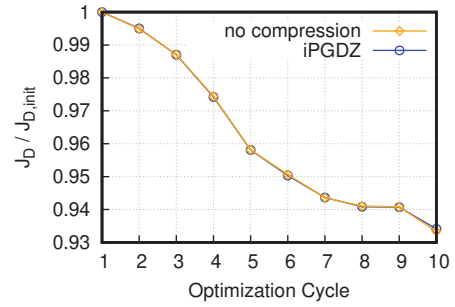


Figure 7: Motorbike optimisation using unsteady adjoint: Left: CPU cost, compression ratio (CR) and error in SDs (ϵ) at the first optimisation cycle, using a combination of ZFP and iPGD (abbreviated as iPGDZ) to compress the entire flow field series, compared with check-pointing. Right: evolution of the normalised mean drag coefficient during the course of the optimisation. J_D is reduced by 6.7%. By using iPGDZ, the memory footprint can be divided by a factor of 480 (from $\sim 2TB$ for the uncompressed full storage to 4.3GB for the iPGDZ-compressed full storage) with a negligible error in SDs (less than 0.6%). Additionally, the computational cost is reduced by $\sim 30\%$ (from $\sim 51h$ to $\sim 36h$) compared to that of a binomial check-pointing approach which stores approximately 10% of the total number of flow fields.

4.2 Stabilisation of the adjoint equations

Despite the great cost benefits that can be achieved using the compressed full-storage strategy presented in section 4.1, optimisation using unsteady flow solvers can be quite expensive due to the high cost of each transient flow solution. In many industrial cases, steady-state solvers are used even in the presence of mild, occasionally a bit stronger, flow unsteadiness in an attempt to reduce the CPU cost of CFD evaluations and optimisations. However, the use of steady-state solvers in cases with mild unsteadiness (or, even, cases with vortex shedding), often encountered in flows past bluff bodies, usually leads to significant convergence difficulties for both the primal and adjoint solvers. The use of a steady flow solver does not allow convergence of the flow equations, and the adjoint solution may be led to stalling or divergence [21, 22]. For the stabilisation of iterative procedures, the Recursive Projection Method (RPM), [23], has been implemented within *adjointOptimisationFoam* to deal with the aforementioned convergence difficulties, [20]. The RPM splits the solution space into two subspaces, containing the unstable and stable modes of the Jacobian matrix of the iterative scheme, where the one is the orthogonal complement of the other. It, then, performs an additional Newton step within the unstable subspace while retaining the original iterative scheme on its complement. An indicative application of the RPM for the stabilisation of the adjoint equations in a problem where vortex-shedding is predicted by a steady-state solver in fig. 8.

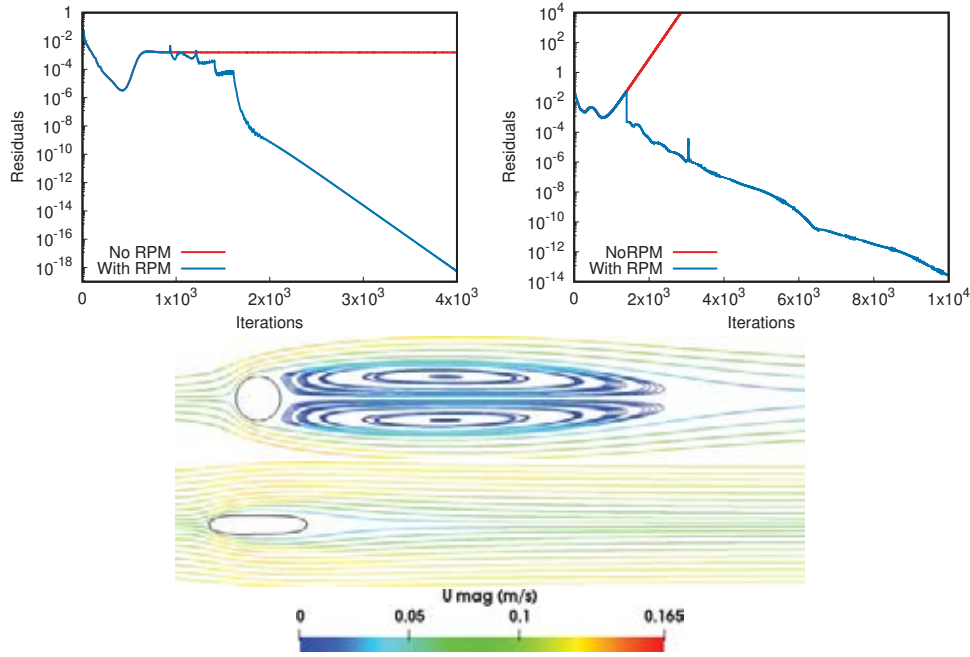


Figure 8: Shape optimisation of a cylinder with a $Re = 140$, targeting min. drag under the constraint of maintaining the cylinder area. Despite the high Re number and the presence of vortex-shedding in the initial geometry, a steady-state solver is used for the solution of the primal and adjoint equations. Top: Residuals of the primal (left) and adjoint (right) pressure equations. Blue/red colors correspond to solutions with/without the RPM. In both cases plotted on the right, the adjoint equations were solved after having stabilised the primal equations by means of the RPM. The RPM manages to stabilise the solution of both the primal and the adjoint equations. Bottom: streamlines of the flow velocity, colored by the velocity magnitude, plotted around the initial and optimised geometries. Using the RPM, the optimisation converged to a geometry in which vortex-shedding has been suppressed, justifying the use of a steady-state solver.

4.3 Topology optimisation

Topology Optimisation (TopO) is nowadays a popular method for the preliminary design of industrial duct systems with multiple inlets and outlets [24]. Though a number of variants exist for formulating the TopO problem, such as the density- (or porosity) based approach [25] or level-set methods, they all follow the idea of artificially blocking part of an initial flow domain to penalize its counter-productive areas, in an attempt to minimize J . This blockage (or porosity) field acts as the field of the design variables in TopO problems. Usually, one design variable exists per grid cell, formulating optimisation problems with thousands or millions of design variables. This particular feature of TopO makes the utilisation of adjoint methods for computing $\delta J / \delta b_n$ the only computationally feasible approach.

PCOpt/NTUA has recently incorporated a number of best practices related to porosity-based TopO, such as the regularisation and projection of the porosity field to mitigate the effects of a grid-dependent solution and alleviate the checkerboard effect [26], to the in-house version of *adjointOptimisationFoam*, [7]. Additionally, a number of peripheral tools supporting TopO, like the Method of Moving Asymptotes (MMA, [27]) used to update the design variables variables in TopO problems with inequality and bound constraints, have also been implemented. Finally, the adjoint to Conjugate Heat Transfer (CHT) problems, with one or more working fluids, has

been implemented in *adjointOptimisationFoam* for tackling the optimisation/design of heat exchangers. An indicative application of TopO for CHT problems, initially presented in [7], is showcased in fig. 9.

4.4 Adjoint methods for uncertainty quantification and robust design optimisation

Most of the optimisation applications assume fixed boundary conditions and do not consider manufacturing imperfections. However, this is not the case in real life and industrial problems. For instance, operating conditions may vary within a certain range, creating an uncertain environment. If these variations are not taken into account, it is likely that the designed/optimised geometries will perform efficiently solely at a single operating point [28]. Hence, there is a need to develop alternative optimisation methods that account for uncertainties [29], a.k.a. robust design optimisation (RDO) methods. In order to measure the impact of uncertainties on the performance of an aero/hydrodynamic shape, statistical moments (usually the mean value and variance) of the Quantity of Interest (QoI, the objective function in optimisation problems without uncertainties) have to first be quantified. The process of computing these statistical moments is referred to as Uncertainty Quantification (UQ). In an RDO problem, a weighted combination of the statistical moments of the QoI is minimised. A recent review of UQ methods and their application in the RDO of air vehicles can be found in [30].

Here, we will focus on how adjoint methods, programmed within *adjointOptimisationFoam*, can reduce the cost of some UQ approaches and/or drive the RDO loop. Non-intrusive Polynomial Chaos Expansion (niPCE) [31, 32] is a popular UQ method that approximates the QoI as a function of weighted polynomials of the uncertain variables. The weights of the polynomials can be computed using either Gauss Quadrature rules or a regression approach, [33]. Adjoint methods within *adjointOptimisationFoam* have been used to either compute the gradient of the RDO objective function evaluated using the former approach, to drive the RDO loop, or provide entries for the regression system of the latter approach at the cost of a single adjoint solution, [34], reducing its UQ cost by a factor of M , where M is the number of the uncertain variables. Additionally, an intrusive PCE (iPCE) variant for incompressible flows and its adjoint counterpart have also been implemented within *adjointOptimisationFoam*, [35], further reducing the cost of RDO for this particular type of problems.

Another UQ method that heavily utilises adjoint is the Method of Moments (MoM) [36, 37]. According to the MoM, the QoI is expanded into a Taylor series in terms of the uncertain variables. By keeping only the first-order term in the Taylor expansion and computing the first two statistical moments of the QoI, namely its mean and standard deviation, a First-Order Second-Moment (FOSM) UQ method is formulated. Since the FOSM-based RDO objective already includes first-order derivatives w.r.t. the uncertain variables, second-order mixed ones (w.r.t. both design and uncertain variables) are needed to compute the gradient driving the RDO. All first-order gradients, w.r.t. either the design or the uncertain variables, are computed based on the continuous adjoint method presented in section 2. To avoid the computation of the second-order mixed derivatives, with a cost that scales with the min. of M and N , its projection to a certain vector is computed at a CPU cost of 2 Equivalent Flow Solutions, formulating the only RDO approach known to the authors with a cost that is independent from both M and N . The latter stands for the projected FOSM (pFOSM) method proposed by the group of authors, [38], and implemented within *adjointOptimisationFoam*. Hessian computations used to support the Second-Order Second-Moment (SOSM) variant of the MoM have also been implemented within *adjointOptimisationFoam* and are discussed in [34].

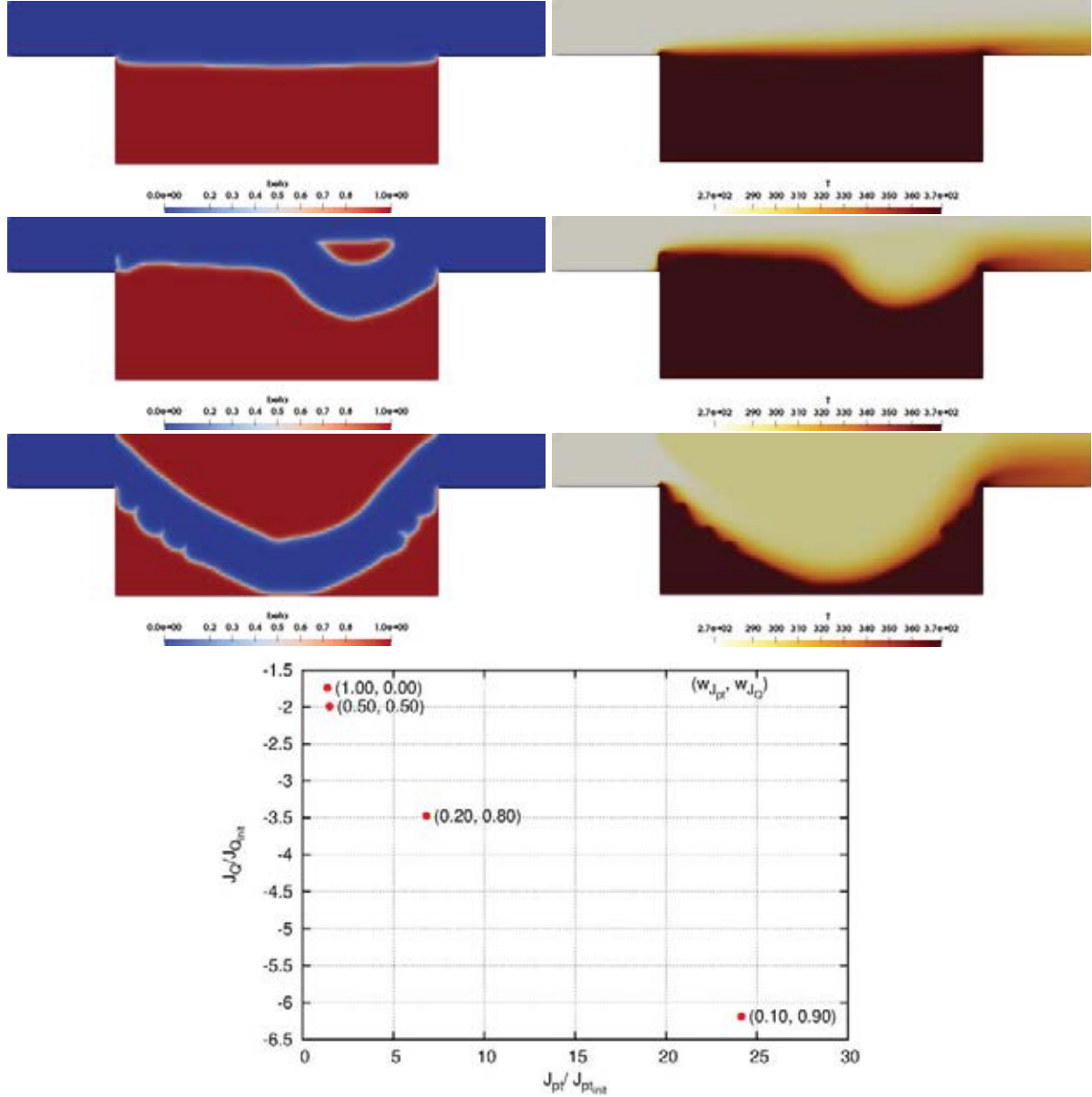


Figure 9: CHT optimisation of a micro-channel. The bottom and side walls of the channel have a temperature of $373K$ while the flow enters the domain from the left with $273K$ and a $Re = 166$. The walls adjacent to the inlet and outlet are adiabatic and the top boundary has symmetry conditions. Weighted combinations of total pressure losses (J_{pt}) and the temperature difference between the outlet and the inlet (J_Q) of the domain are formulated and minimised, using different weight value-sets. The fluid and solid distributions (blue and red areas, respectively) obtained for the (1,0), (0.5,0.5) and (0.1,0.9) weight-value sets of J_{pt} and J_Q , along with the corresponding temperature fields are shown in the top three rows. Increasing the weight of J_Q computes ducts which are approaching the heated bottom wall and even create some fin-like structures to increase heat transfer. The front of non-dominated solutions obtained for the various weight value sets is plotted (bottom). All objective values have been normalised with those corresponding to a completely fluidised domain. Case studied in [7].

Acknowledgments

This project has received funding from the Hellenic Foundation for Research and Innovation (HFRI) and the General Secretariat for Research and Technology (GSRT), under grant agree-

ment No 603, for the development of UQ and RDO methods.

REFERENCES

- [1] P. Spalart and S. Allmaras. A one-equation turbulence model for aerodynamic flows. In *AIAA Paper 1992-0439, 30th Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, 6-9 January 1992.
- [2] E.M. Papoutsis-Kiachagias and K.C. Giannakoglou. Continuous adjoint methods for turbulent flows, applied to shape and topology optimization: Industrial applications. *Archives of Computational Methods in Engineering*, 23(2):255–299, 2016.
- [3] NT. Frink. Assessment of an unstructured-grid method for predicting 3-D turbulent viscous flows. In *AIAA Paper 1996-0292, 34th Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, 15-18 January 1996.
- [4] A.S. Zymaris, D.I. Papadimitriou, K.C. Giannakoglou, and C. Othmer. Continuous adjoint approach to the Spalart-Allmaras turbulence model for incompressible flows. *Computers & Fluids*, 38(8):1528–1538, 2009.
- [5] E.M. Papoutsis-Kiachagias, V.G. Asouti, K.C. Giannakoglou, K. Gkagkas, S. Shimokawa, and E. Itakura. Multi-point aerodynamic shape optimization of cars based on continuous adjoint. *Structural and Multidisciplinary Optimization*, 59(2):675–694, 2019.
- [6] I.S. Kavvadias, E.M. Papoutsis-Kiachagias, and K.C. Giannakoglou. On the proper treatment of grid sensitivities in continuous adjoint methods for shape optimization. *Journal of Computational Physics*, 301:1–18, 2015.
- [7] E.M. Papoutsis-Kiachagias and K.C. Giannakoglou. An adjoint-based topology optimization framework for fluid mechanics and conjugate heat transfer in OpenFOAM. In *8th OpenFOAM Conference*, Digital Event, October 13-15 2020.
- [8] E.M. Papoutsis-Kiachagias, A.S. Zymaris, I.S. Kavvadias, D.I. Papadimitriou, and K.C. Giannakoglou. The continuous adjoint approach to the $k-\epsilon$ turbulence model for shape optimization and optimal active control of turbulent flows. *Engineering Optimization*, 47(3):370–389, 2015.
- [9] A.S. Zymaris, D.I. Papadimitriou, K.C. Giannakoglou, and C. Othmer. Adjoint wall functions: A new concept for use in aerodynamic shape optimization. *Journal of Computational Physics*, 229(13):5228–5245, 2010.
- [10] I.S. Kavvadias, E.M. Papoutsis-Kiachagias, G. Dimitrakopoulos, and K.C. Giannakoglou. The continuous adjoint approach to the $k-\omega$ SST turbulence model with applications in shape optimization. *Engineering Optimization*, 47(11):1523–1542, 2015.
- [11] E.M. Papoutsis-Kiachagias, N. Magoulas, J. Mueller, C. Othmer, and K.C. Giannakoglou. Noise reduction in car aerodynamics using a surrogate objective function and the continuous adjoint method with wall functions. *Computers & Fluids*, 122:223–232, 2015.
- [12] J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer, New York, 1999.

- [13] J.B. Rosen. The gradient projection method for nonlinear programming. Part I. Linear constraints. *Journal of the Society for Industrial and Applied Mathematics*, 8(1):181–217, 1960.
- [14] A. Jameson and J. Vassberg. Studies of alternate numerical optimization methods applied to the brachistochrone problem. In *OptiCON '99 Conference*, Newport Beach, CA, USA, 14–15 October 1999.
- [15] E.M. Papoutsis-Kiachagias, S.A. Kyriacou, and K.C. Giannakoglou. The continuous adjoint method for the design of hydraulic turbomachines. *Computer Methods in Applied Mechanics and Engineering*, 278:612–639, 2014.
- [16] K.T. Gkaragkounis, E.M. Papoutsis-Kiachagias, and K.C. Giannakoglou. The continuous adjoint method for shape optimization in conjugate heat transfer problems with turbulent incompressible flows. *Applied Thermal Engineering*, 140:351–362, 2018.
- [17] A. Griewank and A. Walther. Algorithm 799: Revolve: an implementation of checkpointing for the reverse or adjoint mode of computational differentiation. *ACM Transactions on Mathematical Software (TOMS)*, 26(1):19–45, 2000.
- [18] P. Lindstrom. Fixed-Rate Compressed Floating-Point Arrays. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2674–2683, 2014.
- [19] F. Chinesta, R. Keunings, and A. Leygue. *The proper generalized decomposition for advanced numerical simulations: A primer*. Springer, 2014.
- [20] T. Skamagkis, A.-S.I Margetis, E.M. Papoutsis-Kiachagias, and K.C. Giannakoglou. On the efficiency and robustness of the adjoint method: Applications in steady and unsteady shape optimization in fluid mechanics. In *8th OpenFOAM Conference*, Digital Event, October 13-15 2020.
- [21] R. Dwight and D. Vollmer. Efficient algorithms for solution of the adjoint compressible navier-stokes equations with applications. In *Proceedings of the ONERA-DLR Aerospace Symposium (ODAS)*, Toulouse, France, 2006. Proceedings of the ONERA-DLR Aerospace Symposium (ODAS).
- [22] T. Albring, T. Dick, and N. Gauger. Assessment of the Recursive Projection Method for the stabilization of discrete adjoint solvers. In *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Denver, Colorado, 2017. 18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference.
- [23] G.M. Shroff and H. Keller. Stabilization of unstable procedures: The Recursive Projection Method. *SIAM Journal of Numerical Analysis*, 30(4):1099–1120, 1993.
- [24] C. Othmer. Adjoint methods for car aerodynamics. *Journal of Mathematics in Industry*, 4(6), 2014.
- [25] C.B. Dilgen, S.B. Dilgen, D.R. Fuhrman, O. Sigmund, and B.S. Lazarov. Topology optimization of turbulent flows. *Computer Methods in Applied Mechanics and Engineering*, 331:363 – 393, 2018.

- [26] B.S. Lazarov and O. Sigmund. Filters in topology optimization based on Helmholtz-type differential equations. *International Journal for Numerical Methods in Engineering*, 86(6):765–781, 2011.
- [27] K. Svanberg. The method of moving asymptotes – a new method for structural optimization. *International Journal for Numerical Methods in Engineering*, 24(2):359–373, 1987.
- [28] G.K. Kenway and J.R.R. Martins. Aerodynamic shape optimization of the CRM configuration including buffet-onset conditions. In *54th AIAA Aerospace Sciences Meeting*, San Diego, California, USA, January 2016.
- [29] G.I. Schuëller and H.A. Jensen. Computational methods in optimization considering uncertainties - an overview. *Computer Methods in Applied Mechanics and Engineering*, 198(1):2–13, 2008.
- [30] Z. Huan, G. Zhenghong, X. Fang, and Z. Yidian. Review of robust aerodynamic design optimization for air vehicles. *Archives of Computational Methods in Engineering*, 26:685–732, 2019.
- [31] D. Xiu and G.M. Karniadakis. Modeling uncertainty in flow simulations via generalized polynomial chaos. *Journal of Computational Physics*, 187:137–167, 2003.
- [32] C. Dinescu, S. Smirnov, C. Hirsch, and C. Lacor. Assessment of intrusive and non-intrusive non-deterministic CFD methodologies based on polynomial chaos expansion. *International Journal of Engineering Systems Modeling and Simulations*, 2:87–98, 2010.
- [33] D. Xiu. Fast numerical methods for stochastic computations: A review. *Communications in computational physics*, 5:242–222, 2009.
- [34] E.M. Papoutsis, V.G. Asouti, and K.C. Giannakoglou. Assessment of variants of the method of moments and polynomial chaos approaches to aerodynamic uncertainty quantification. In *4th ECCOMAS Thematic Conference on Uncertainty Quantification in Computational Sciences and Engineering (UNCECOMP 2021)*, Virtual Event, Streamed from Athens, Greece, June 27-30 2021.
- [35] A.K. Papageorgiou, K.B. Fragkos, E.M. Papoutsis-Kiachagias, and K.C. Giannakoglou. Uncertainty quantification and robust design for aerodynamic applications, using continuous adjoint methods. In *6th European Conference on Computational Mechanics (ECCM 6)-7th European Conference on Computational Fluid Dynamics (ECFD 7)*, Glasgow, UK, June 11-15 2018.
- [36] R.W. Waters and L. Huyse. Uncertainty analysis for fluid mechanics with applications. *NASA/CR 2002*, 211449, 2002.
- [37] M.M. Putko, P.A. Newman, A.C. Taylor, and L.L. Green. Approach for uncertainty propagation and robust design in CFD using sensitivity derivatives. In *AIAA Paper 2001-2528, 15th Computational Fluid Dynamics Conference*, Anaheim, CA, 2001.
- [38] K.B. Fragkos, E.M. Papoutsis-Kiachagias, and K.C. Giannakoglou. pFOSM: An efficient algorithm for aerodynamic robust design based on continuous adjoint and matrix-vector products. *Computers & Fluids*, 181:57–66, 2019.