



HAL
open science

Large-scale parallel topology optimisation of three-dimensional incompressible fluid flows in a level set, anisotropic mesh adaptation framework

W Abdel Nour, A Larcher, D Serret, P Meliga, E Hachem

► To cite this version:

W Abdel Nour, A Larcher, D Serret, P Meliga, E Hachem. Large-scale parallel topology optimisation of three-dimensional incompressible fluid flows in a level set, anisotropic mesh adaptation framework. *Computer Methods in Applied Mechanics and Engineering*, 2023, 416, pp.116335. 10.1016/j.cma.2023.116335 . hal-04244684

HAL Id: hal-04244684

<https://hal.science/hal-04244684v1>

Submitted on 16 Oct 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Large-scale parallel topology optimisation of three-dimensional incompressible fluid flows in a level set, anisotropic mesh adaptation framework

W. Abdel Nour^a, A. Larcher^a, D. Serret^b, P. Meliga^a, E. Hachem^{a,*}

^a*Mines Paris, PSL University, Centre for material forming (CEMEF), UMR CNRS, 06904 Sophia Antipolis, France*

^b*TEMISTh SAS, Technocentre des Florides, 13700 Marignane, France*

Abstract

This paper considers the topology optimization of duct flows governed by the three-dimensional steady state Navier–Stokes equations, using anisotropic mesh adaptation to achieve a high-fidelity description of all fluid–solid interfaces. The numerical framework combines an immersed volume method solving stabilized, linear equal-order finite element formulations cast in the Variational Multiscale (VMS) framework, and level set representations of the interface, used as a posteriori anisotropic error estimator to minimize the interpolation error under the constraint of a prescribed number of nodes in the mesh. Both the resolution and remeshing steps are performed in a massively parallel framework allowing for the optimization of large-scale systems. In particular, an original parallelization strategy is used for mesh adaptation, that combines local remeshing performed sequentially and independently on each subdomain with blocked interfaces, and constrained repartitioning to optimally move the interfaces between subdomains in an optimal way, both iterated until a satisfying mesh and partition are obtained. **The proposed approach reduces the computational burden related to the call of the finite element solver, compared to classical optimization schemes working on uniform grids with similar mesh refinement. For a given number of nodes, it improves the accuracy in the geometric description of all layouts. Finally, it has the potential to alleviate the end user from most of the post-processing step aiming at extracting the final layout, due to the ability of anisotropic adapted meshes to generate intrinsically smooth designs.**

Numerical results are provided for several three-dimensional problems of power dissipation minimization involving several dozen million state degrees of freedom, for which the optimal designs agree well with reference results from the literature, while providing superior accuracy over prior studies solved on isotropic meshes (**in the sense that the flow is better resolved, especially in the near-wall regions, and the layouts are more smooth**). The potential of the method for engineering problems of practical interest is eventually exposed by optimizing the distributor section conveying the cold fluid within the plates of a plate fin heat exchanger.

Keywords: Topology Optimization, Fluid mechanics, Large-scale systems, Parallelization, Level Set Method, Anisotropic mesh adaptation

1. Introduction

Fluid flow topology optimization aims at maximizing a measure of performance by identifying the best path for a fluid to flow in a design domain, subject to a set of design constraints, a typical example being to minimize dissipation under a constant volume of fluid constraint. The approach was initiated for mechanical design problems [1, 2], but has since spread to a variety of other physics modeled after partial differential equations, including fluids, acoustics, electromagnetics, optics and combinations thereof; see especially Ref. [3] for a recent survey dedicated to fluid flow problems. The mathematical foundation builds on iterative analysis and design update steps, often steered

*Corresponding author

Email address: elie.hachem[@]mines-paristech.fr (E. Hachem)

9 by gradient evaluations. Compared to the size and shape optimization methods it has emerged
10 from, the main advantage of topology optimization is in the increased design freedom, that allows
11 starting from arbitrary initial guesses and generating non-intuitive designs, even under conflicting
12 requirements and complex correlations between design parameters and system response.

13 There has been a significant leap in terms of methods and approaches used for topology opti-
14 mization. Those include, but are not limited to, methods of density, level set, topological derivative,
15 phase field, and evolutionary; see Ref. [4] for a comparison and critical review of the different ap-
16 proaches. Leaving aside explicit boundary methods, that represent the fluid-solid interface by
17 edges or faces of a body-fitted mesh, and have limited flexibility to handle complicated topological
18 changes, the prevalent classes of methods for fluid flow topology optimization are the density and
19 the level set methods. Density methods rely on Brinkman penalization to weakly enforce fluid-solid
20 no-slip conditions in a unified domain [1, 5, 6]. They manage drastic topological changes, as the
21 gradient (or sensitivity) information is distributed over a large part of the domain, but require a
22 well-tuned penalization factor, small enough to ensure numerical stability of the solution and op-
23 timisation algorithms, but large enough to prevent the flow from leaking inside the solid domain.
24 Level set methods conversely manage to capture the solid boundaries by iso-contours of a level set
25 function [7-10]. Due to the sensitivities being located only at the solid-fluid interface, they are
26 without a nucleation mechanism, hence the common practice to start from a design with many
27 holes. Meanwhile, they easily handle complicated topological changes (e.g., merging or cancella-
28 tion of holes), and yield well defined, crisp interface representations while avoiding the intermediate
29 material phases (grayscale) and mesh-dependent spatial oscillations of the interface (staircasing)
30 often encountered in density methods, unless they are combined with cut element techniques [11].

31 **A classical topology optimization problem resorts to the discretization of a levelset function/a**
32 **density field to tackle the minimization setting, along with the adoption of finite element methods**
33 **to cope with the approximation of the relevant governing equations. Both discretizations generally**
34 **employ identical meshes** with close-to-uniform element size, small enough that all relevant phys-
35 ical phenomena are accurately described, but not so small that the computational cost becomes
36 prohibitive. A recent trend in this regards has been to incorporate adaptive remeshing techniques,
37 where one starts from a coarse base grid, then adds recursively finer and finer subgrids in the regions
38 requiring higher resolution. **This proceeds** either until a maximum level of refinement is reached,
39 or the local truncation error drops below a certain tolerance, for more sophisticated implementa-
40 tions endowed with error estimation routines. Within the context of fluid flow problems, particular
41 emphasis has been put on adaptive meshing refinement schemes, using both density [12, 13] and
42 level set methods [14, 15]; see also [16] for an application to phase field methods and [17-19] for
43 recent efforts applying a different remeshing scheme to a combination of level set functions and
44 adaptive body-conforming meshes. **Other approaches include the one used in [20-22], where the**
45 **virtual element method is used to solve topology optimization problems governed by Stokes and**
46 **Navier-Stokes equations on unstructured polygonal finite element meshes (see also [23] for a com-**
47 **prehensive domain-agnostic survey) and the surface-capturing extended finite element method, in**
48 **which cut elements are integrated using a special scheme and the interface boundary conditions**
49 **are imposed using stabilised Lagrange multipliers or a stabilised Nitsche’s method, although the**
50 **related solutions are shown in [24] to possibly exhibit penalty- and mesh-dependent mass losses**
51 **through the interface.**

52 While fluid dynamics deals with convection dominated problems **typified by the presence of**
53 **strongly directional features**, e.g., boundary layers where the fluid velocity exhibits steep gradients
54 in the wall-normal direction and skin-friction plays a defining role, the adaptive algorithms applied
55 so far to fluid flow topology optimization support almost exclusively isotropic size maps. **The**
56 **approach proposed in the present work consists of anisotropic mesh adaptation based on a theo-**
57 **retically sound tool, i.e., a recovery-based a posteriori error analysis. It enjoys several advantages.**
58 **First, it substantially reduces the computational burden related to the call of the finite element**
59 **solver, compared to classical optimization schemes working on uniform grids with similar mesh**
60 **refinement. Second, for a given number of nodes, it improves the accuracy in the geometric de-**
61 **scription of the optimal designs while naturally conveying said accuracy to the numerical solutions**
62 **, which is all perfectly in line with the recommendations made in [3] to improve upon the current**
63 **state of the art. Finally, the use of anisotropic adapted meshes yields intrinsically smooth final**
64 **layouts, meaning that the post-processing phase can be strongly reduced (or even skipped), and**

65 the structure can directly move on to the production manufacturing phase, unlike most traditional
66 algorithms, where a strong post-processing can be required to extract the final layout. Similar
67 approaches towards free form design has been taken in [25, 26] to solve compliance minimization
68 problems in topology optimization for structural applications. Nonetheless, our literature review
69 did not reveal any other study combining anisotropic mesh adaptation and fluid flow topology
70 optimization, besides the density-based optimisation of Stokes flow in Ref. [13], possibly because
71 the notorious difficulty of finding spatial discretization schemes that meet the level of robustness
72 required by automatic anisotropic mesh adaptation.

73 The scope of this article is to present and provide basic verifications of a novel level set,
74 anisotropic mesh adaptation framework for topology optimisation of large-scale, three-dimensional
75 (3-D) steady and laminar flow. In recent years, an increasing number of studies have dealt with
76 efficient large-scale topology optimization, to which the reader is referred to for further information
77 regarding the use of parallel programming using the message parsing interface (MPI) and parallel
78 resolution of partial differential equations with scalable and high performance algorithms [27-31].
79 Particular emphasis is thus put here on the parallel adaptive meshing technique, whose implemen-
80 tation in the context of large-scale fluid flow topology optimization makes for the main novelty of
81 this study. The latter combines local remeshing performed independently on each subdomain with
82 fixed interfaces, and constrained repartitioning to move the interfaces between subdomains in an
83 optimal way, both iterated until a satisfying mesh and partition are obtained. The metric map
84 providing both the size and the stretching of mesh elements in a very condensed information data
85 is derived from the level set. A posteriori anisotropic error estimator is then used to minimize the
86 interpolation error under the constraint of a prescribed number of nodes in the mesh. The latter
87 can be adjusted over the course of optimization, meaning that the base grid can be either refined
88 or coarsened on demand: this is expected to achieve further speed-ups, as it reduces the cost of
89 modelling the solid material away from the interface, and also to help improve manufacturability of
90 the optimal design, which remains an issue as most classical topology optimization methods render
91 organic designs that can be difficult to translate into computer-aided design models.

92 The paper organization is as follows: the governing equations for the gradient-based opti-
93 mization model are formulated in Sec. 2. The stabilized finite element numerical framework and
94 anisotropic mesh adaptation algorithm used to perform the design update step are described in
95 Secs. 3 and 4, respectively, with additional details regarding the parallel implementation provided
96 in Sec. 5. The implementation details of the optimization algorithm are given in Sec. 6. Fi-
97 nally, numerical experiments assessing relevance on a series of large-scale, three-dimensional power
98 dissipation minimization problems (including a simplified industrial case study) are presented in
99 section 7, with particular attention paid to highlight the improved accuracy of the obtained solu-
100 tions.

101 2. Problem setting

102 In the following, we denote by Ω a fixed, open bounded domain in \mathbb{R}^d (with d the space
103 dimension), with boundary $\partial\Omega$ oriented with inward-pointing normal vector \mathbf{n} . Throughout this
104 study, $\Omega = \Omega_f \cup \Omega_s$ is the disjoint reunion of two domains Ω_f and Ω_s . For simplicity, we refer to
105 Ω_f as the fluid domain, and to Ω_s as the solid domain, although we also fill Ω_s with a fluid for
106 numerical convenience, as further explained in the following. The two domains are separated by
107 an interface $\Gamma = \Omega_f \cap \Omega_s$, whose position we seek to optimize with respect to a certain measure of
108 performance, here a cost function J to minimize.

109 2.1. State equations

Mathematically, the problem is characterized by a set of physical variables determined as the
solutions of partial differential equations, themselves derived from modeling considerations. Here,
the flow motion in the fluid domain Ω_f is modeled after the steady incompressible Navier-Stokes
equations

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega_f, \quad (1)$$

$$\rho \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \nabla \cdot (2\mu \varepsilon(\mathbf{u})) \quad \text{in } \Omega_f, \quad (2)$$

where \mathbf{u} is the velocity, p is the pressure, $\boldsymbol{\varepsilon}(\mathbf{u}) = (\nabla\mathbf{u} + \nabla\mathbf{u}^T)/2$ is the rate of deformation tensor, and we assume constant fluid density ρ and dynamic viscosity μ . The fluid domain boundary $\partial\Omega_f$ is split into (wall) interface Γ , inlet Γ_i (defined as the combined boundary of all surfaces where fluid enters the domain), outlet Γ_o (the combined boundary of all surfaces where fluid leaves the domain). Open flow boundary conditions are appended under the form of a prescribed velocity at the inlet, zero velocity at the wall

$$\mathbf{u} = \mathbf{u}_i \quad \text{on } \Gamma_i, \quad (3)$$

$$\mathbf{u} = \mathbf{0} \quad \text{on } \Gamma, \quad (4)$$

and a convenient outflow condition at the outlet, either a prescribed velocity

$$\mathbf{u} = \mathbf{u}_o \quad \text{on } \Gamma_o, \quad (5)$$

adjusted to ensure mass conservation, or a more natural zero pressure/zero viscous stress condition

$$p\mathbf{n} = \mu\boldsymbol{\varepsilon}(\mathbf{u}) \cdot \mathbf{n} = \mathbf{0} \quad \text{on } \Gamma_o. \quad (6)$$

110 2.2. Adjoint-based sensitivity analysis

111 We assume in the following that the cost function (i) can be formulated as a surface integral
 112 over the domain boundary, rather than its interior, and (ii) does not depend on the flow quantities
 113 on the wall, which is most often true in topology optimization. It is thus expressed as integrals
 114 over all or any part of inlet and/or outlet, i.e.,

$$J_s = \int_{\Gamma_i \cup \Gamma_o} J ds. \quad (7)$$

115 The problem of minimizing the cost function subject to Navier–Stokes as state equations is tackled
 116 using the continuous adjoint method. The reader interested in the technicalities of the method is
 117 referred to [32]. One first forms the Lagrangian

$$\mathcal{L} = \int_{\Gamma_i \cup \Gamma_o} J ds - \int_{\Omega_f} \tilde{p} \nabla \cdot \mathbf{u} dv - \int_{\Omega_f} \tilde{\mathbf{u}} \cdot (\rho \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p - \nabla \cdot (2\mu \boldsymbol{\varepsilon}(\mathbf{u}))) dv, \quad (8)$$

118 featuring the adjoint velocity $\tilde{\mathbf{u}}$ as the Lagrange multiplier for the momentum equations (2) and
 119 the adjoint pressure \tilde{p} as the Lagrange multiplier for the continuity equation (1), then seeks to
 120 decompose the variation of \mathcal{L} due to a change in the interface position into individual variations
 121 with respect to the adjoint, state and design variables. The variation with respect to the adjoint
 122 variables

$$\delta_{(\tilde{\mathbf{u}}, \tilde{p})} \mathcal{L} = - \int_{\Omega_f} \delta \tilde{p} \nabla \cdot \mathbf{u} dv - \int_{\Omega_f} \delta \tilde{\mathbf{u}} \cdot (\rho \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p - \nabla \cdot (2\mu \boldsymbol{\varepsilon}(\mathbf{u}))) dv, \quad (9)$$

is trivially zero as long as (\mathbf{u}, p) is solution to the above Navier–Stokes equations, in which case $\mathcal{L} = J_s$. After integrating by parts, the variation with respect to the state variables is

$$\begin{aligned} \delta_{(\mathbf{u}, p)} \mathcal{L} &= \int_{\Omega_f} (\nabla \cdot \tilde{\mathbf{u}}) \delta p dv + \int_{\Omega_f} (-\rho \mathbf{u} \cdot \nabla \tilde{\mathbf{u}} + \rho \nabla \mathbf{u}^T \cdot \tilde{\mathbf{u}} - \nabla \tilde{p} - \nabla \cdot (2\mu \boldsymbol{\varepsilon}(\tilde{\mathbf{u}}))) \cdot \delta \mathbf{u} dv \\ &+ \int_{\Gamma_i \cup \Gamma_o} \partial_{\mathbf{u}} J \cdot \delta \mathbf{u} ds + \int_{\partial\Omega_f} (\tilde{p} \mathbf{n} + 2\mu \boldsymbol{\varepsilon}(\tilde{\mathbf{u}}) \cdot \mathbf{n} + \rho(\mathbf{u} \cdot \mathbf{n}) \tilde{\mathbf{u}}) \cdot \delta \mathbf{u} ds \\ &- \int_{\Gamma_i \cup \Gamma_o} \partial_p J \mathbf{n} \cdot (-\delta p \mathbf{n} + 2\mu \boldsymbol{\varepsilon}(\delta \mathbf{u}) \cdot \mathbf{n}) ds - \int_{\partial\Omega_f} \tilde{\mathbf{u}} \cdot (-\delta p \mathbf{n} + 2\mu \boldsymbol{\varepsilon}(\delta \mathbf{u}) \cdot \mathbf{n}) ds, \end{aligned} \quad (10)$$

on behalf of the viscous stress being purely tangential in incompressible flows. At this stage, adjoint equations and boundary conditions are designed to ensure $\delta_{(\mathbf{u}, p)} \mathcal{L} = 0$, which requires the domain and boundary integrals to vanish individually in (10). Keeping in mind that we work here under the assumption of a fixed interface (since the design variable is constant), we obtain the linear, homogeneous problem

$$\nabla \cdot \tilde{\mathbf{u}} = 0 \quad \text{in } \Omega_f, \quad (11)$$

$$-\rho \mathbf{u} \cdot \nabla \tilde{\mathbf{u}} + \rho \nabla \mathbf{u}^T \cdot \tilde{\mathbf{u}} = \nabla \tilde{p} + \nabla \cdot (2\mu \boldsymbol{\varepsilon}(\tilde{\mathbf{u}})) \quad \text{in } \Omega_f, \quad (12)$$

Algorithm 1 Simplified update scheme

Require: Anisotropic mesh adapted to initial interface position

- 1: **loop**
 - 2: Compute state, adjoint and cost function sensitivity
 - 3: Set displacement in the direction of steepest slope
 - 4: Update interface position
 - 5: Generate anisotropic mesh adapted to new interface position
-

driven by the non-homogeneous boundary conditions

$$\tilde{\mathbf{u}} = -\partial_p J \mathbf{n} \quad \text{on } \Gamma_i, \quad (13)$$

$$\tilde{\mathbf{u}} = \mathbf{0} \quad \text{on } \Gamma, \quad (14)$$

associated to (3)-(4), with adjoint outflow condition

$$\tilde{\mathbf{u}} = -\partial_p J \mathbf{n} \quad \text{on } \Gamma_o, \quad (15)$$

if the prescribed velocity outflow condition (5) is used, or

$$\tilde{p} \mathbf{n} + 2\mu_f \boldsymbol{\varepsilon}(\tilde{\mathbf{u}}) \cdot \mathbf{n} + \rho_f (\mathbf{u} \cdot \mathbf{n}) \tilde{\mathbf{u}} = -\partial_{\mathbf{u}} J \quad \text{on } \Gamma_o, \quad (16)$$

123 if the zero pressure/zero viscous stress outflow condition (6) is used. Expressing the interface
124 normal deformation after (33) as

$$\delta \mathbf{u} = \beta \nabla \mathbf{u} \cdot \mathbf{n}, \quad (17)$$

125 the variation with respect to the design variable, now encompassing the domain deformation, is
126 ultimately computed as

$$\delta_\beta J_s = \delta_\beta \mathcal{L} = \int_\Gamma \beta (\tilde{p} \mathbf{n} + 2\mu_f \boldsymbol{\varepsilon}(\tilde{\mathbf{u}}) \cdot \mathbf{n}) \cdot (\nabla \mathbf{u} \cdot \mathbf{n}) \, ds = \int_\Gamma \beta \mu_f (\nabla \tilde{\mathbf{u}} \cdot \mathbf{n}) \cdot (\nabla \mathbf{u} \cdot \mathbf{n}) \, ds, \quad (18)$$

127 where the second equality stems from the incompressibility of the state and adjoint solutions (32).
128 This enables efficient design update schemes via first-order gradient descent methods, as the second
129 term in the integrand is the desired sensitivity to a displacement β at some specific point of the
130 interface. For instance the simplest steepest-descent algorithm implemented herein moves down
131 the cost function, in the direction of the steepest slope using

$$\beta = -\mu (\nabla \tilde{\mathbf{u}} \cdot \mathbf{n}) \cdot (\nabla \mathbf{u} \cdot \mathbf{n}), \quad (19)$$

132 up to a positive multiplicative factor to control the step taken in the gradient direction.

133 3. Computational methods

134 A primitive pseudo-code of the procedure for solving the above topology optimization problem is
135 provided in Alg. 1 to repeat until a maximum number of iterations or a convergence threshold has
136 been reached. In a nutshell, this is done here using a finite element immersed numerical framework
137 combining implicit representation of the different domains, level set description of the interface,
138 and anisotropic remeshing capabilities. For the sake of readability, the mesh adaptation algorithm
139 and parallel computational framework, whose implementation in the context of fluid flow topology
140 optimization makes for the main novelty of this study, are presented in the following as stand-alone
141 sections. In the remainder of this section, we walk through each of the other steps and review the
142 various problems involved and the numerical methods for solving them.

143 *3.1. Level set representation of the interface*

144 The level set method is used here to localize and capture the interface between the fluid and
 145 solid domains from the zero iso-value of a smooth level set function, classically the signed distance
 146 function defined as

$$\varphi(\mathbf{x}) = \begin{cases} -\text{dist}(\mathbf{x}, \Gamma) & \text{if } \mathbf{x} \in \Omega_f, \\ 0 & \text{if } \mathbf{x} \in \Gamma, \\ \text{dist}(\mathbf{x}, \Gamma) & \text{if } \mathbf{x} \in \Omega_s, \end{cases} \quad (20)$$

147 with the convention that $\varphi < 0$ in the fluid domain. Once the sensitivity analysis has output
 148 a displacement β in the direction of the steepest slope, the position of the level set is updated
 149 solving a transport equation with normal velocity $\beta \mathbf{n} / \Delta\tau$, where $\Delta\tau$ is a pseudo-time step to
 150 convert from displacement to velocity, that has no physical relevance since we are not concerned
 151 by the absolute displacement of a given point on the interface, only by its relative displacement
 152 with respect to its neighbors. This equation is posed in the whole domain Ω , which is because
 153 the normal vector recovered at the interface as $\mathbf{n} = \nabla\varphi / \|\nabla\varphi\|$ is easily extended to Ω using (20).
 154 The main problem with this approach is that the level set after transport is generally no longer a
 155 distance function, which is especially problematic when a specific remeshing strategy depending on
 156 the distance property is used at the interface (as is the case in this study). As a result, the distance
 157 function needs to be reinitialized, which is done here using a coupled convection-reinitialization
 158 method wherein the level set function is automatically reinitialized during the resolution of the
 159 transport equation. In practice, the signed distance function is cut off using a hyperbolic tangent
 160 filter, as defined by

$$\phi = E \tanh\left(\frac{\varphi}{E}\right), \quad (21)$$

161 with E the cut-off thickness, so the metric property is asymptotically satisfied in the vicinity of
 162 the zero iso-value. This filtered level set is then evolved solving the auto-reinitialization equation

$$\partial_\tau \phi + \mathbf{a}_\tau \cdot \nabla \phi = S, \quad (22)$$

163 where we note

$$\mathbf{a}_\tau = \frac{\beta}{\Delta\tau} \mathbf{n} + \frac{\lambda}{\Delta\tau} \text{sgn}(\phi) \frac{\nabla\phi}{\|\nabla\phi\|}, \quad S = \frac{\lambda}{\Delta\tau} \text{sgn}(\phi) \left(1 - \left(\frac{\phi}{E}\right)^2\right), \quad (23)$$

164 and λ is a parameter homogeneous to a length, set to the mesh size h_\perp in the direction normal to
 165 the interface. Such an approach is shown in [34–36] to reduce the computational cost and to ensure
 166 a better mass conservation compared to the classical Hamilton–Jacobi method in which both steps
 167 are performed in succession). Moreover, since the filtered level set defined in (21) is bounded,
 168 Dirichlet boundary conditions $\phi = \pm E$ are easily appended to Eq. (22) to explicitly design fluid
 169 and solid sub-regions of $\partial\Omega$. In practice, we impose here fluid at the inlet and outlet, and solid
 170 everywhere else.

171 *3.2. Immersed volume method*

172 The immersed volume method (IVM) [37–38] is used to combine the fluid and solid phases of
 173 the problem into a single fluid with variable material properties (density and viscosity). Simply
 174 put, this amounts to solving the state and adjoint equations identical to (1)–(2) and (11)–(12),
 175 but formulated on a unique mesh of the domain Ω in which the fluid and solid domains Ω_f and
 176 Ω_s are immersed, and featuring phase-dependent density and viscosity adequately interpolated
 177 over a small layer around the interface and otherwise equal to their fluid and solid values. One
 178 particularity is that the thickness of the interpolation layer is user-defined and thus, it does not
 179 increase in size during the optimization, unlike the homogenization method or any other generalized
 180 material method. Using the level set function (20) as criterion for anisotropic mesh adaptation
 181 (more details provided in Sec. 3) ensures that individual material properties can be distributed
 182 accurately and smoothly as possible over the smallest possible thickness around the interface. This

183 is classically done using the the arithmetic mean of the solid and fluid values, using a smooth
184 Heaviside function computed from the level set to avoid discontinuities by creating an interface
185 transition with a thickness of a few elements. Such an approach is especially relevant to thermal
186 coupling problems, as having composite conductivity and specific heat means that the amount of
187 heat exchanged at the interface then proceeds solely from the individual material properties on
188 either side of it, and removes the need for a heat transfer coefficient. For the pure flow problems
189 tackled here, though, it suffices to use constant density and viscosity equal to the fluid values, and
190 to set the velocity to zero at all grid nodes located inside the solid domain Ω_s . Compared to using
191 a very high solid to fluid viscosity ratio to ensure that the velocity is zero in the solid domain,
192 this can be seen as a hard penalty preventing the fluid from leaking across the immersed interface.
193 The latter holds numerically because anisotropic mesh adaptation ensures that the interface does
194 not intersect arbitrarily the mesh elements (as it precisely aims at aligning the mesh element edges
195 along the interface), which may otherwise compromise the accuracy of the finite element approach.

196 3.3. Variational multiscale modeling

197 The convective terms in the incompressible Navier–Stokes and level set transport equations
198 may cause spurious node-to-node velocity oscillations. Furthermore, the equal order linear/linear
199 approximations used for the velocity and pressure variables may give rise to spurious pressure oscil-
200 lations, albeit very desirable due to its simplicity of implementation and affordable computing cost,
201 especially for 3-D applications. To prevent these numerical instabilities, we solve here stabilized
202 formulations cast in the Variational Multiscale (VMS) framework, that enhance the stability of the
203 Galerkin method via a series of additional integrals over element interior. The basic idea is to split
204 all quantities into coarse and fine scale components, corresponding to different levels of resolution,
205 with a coarse scale resolved by the finite element mesh and the effect of the fine scale onto the large
206 scale approximated via consistently derived residual based terms. For the sake of simplicity in the
207 notations, and as long as it does not lead to ambiguity, we omit in what follows the distinction
208 between all continuous variables (e.g., domains, solutions, operators) and their discrete finite ele-
209 ment counterparts, as well as the dependency of all variables on the iteration of the optimization
210 process.

211 3.3.1. Navier–Stokes equations

212 In practice, the state solution is computed by time-stepping the unsteady Navier–Stokes equa-
213 tions with large time steps to accelerate convergence towards a steady state. The stopping criterion
214 is here for two consecutive time steps to differ by less than 10^{-6} in L^∞ norm. In order to deal with
215 the time-dependency and non-linearity of the momentum equation, the transport time of the time
216 scale is assumed much smaller than that of the coarse scale. In return, the fine scale contribution to
217 the transport velocity is neglected, and the fine scale is not tracked in time, although it is driven by
218 the coarse-scale, time-dependent residuals and therefore does vary in time in a quasi-static man-
219 ner. In-depth technical and mathematical details together with extensive discussions regarding
220 the relevance of the approximations can be found in [39]. Ultimately, the coarse scale variational
221 problem is formulated as

$$\int_{\Omega} (\rho \partial_t \mathbf{u} + \rho \mathbf{u} \cdot \nabla \mathbf{u}) \cdot \mathbf{w} \, dv + \int_{\Omega} 2\mu \boldsymbol{\varepsilon}(\mathbf{u}) : \boldsymbol{\varepsilon}(\mathbf{w}) \, dv - \int_{\Omega} p(\nabla \cdot \mathbf{w}) \, dv + \int_{\Omega} (\nabla \cdot \mathbf{u}) q \, dv - \sum_{k=1}^{N_e} \int_{\Omega_k} \tau_1 \mathbf{r}_1 \cdot (\rho \mathbf{u} \cdot \nabla \mathbf{w}) \, dv - \sum_{k=1}^{N_e} \int_{\Omega_k} \tau_1 \mathbf{r}_1 \cdot \nabla q \, dv - \sum_{k=1}^{N_e} \int_{\Omega_k} \tau_2 r_2 (\nabla \cdot \mathbf{w}) \, dv = 0, \quad (24)$$

222 where we have considered a discretization of Ω into N_e non-overlapping elements (triangles or
223 tetrahedrons), Ω_k is the domain occupied by the k th element, and \mathbf{r}_1 and r_2 are the momentum
224 and continuity residuals

$$-\mathbf{r}_1 = \rho \partial_t \mathbf{u} + \rho \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p, \quad -r_2 = \nabla \cdot \mathbf{u}, \quad (25)$$

225 whose second derivatives vanish since we use linear interpolation functions. Finally, τ_1 and τ_2 are
226 ad-hoc stabilization coefficients, computed on each element after [38, 40] as

$$\tau_1 = \frac{1}{\rho (\tau_t^2(u) + \tau_d^2)^{1/2}}, \quad \tau_2 = \frac{h^2}{\tau_1}, \quad (26)$$

227 with convection (transport) and diffusion-dominated limits defined as

$$\tau_t(u) = c_t \frac{u}{h}, \quad \tau_d = c_d \frac{\mu}{\rho h^2}. \quad (27)$$

228 Here, u is a characteristic norm of the velocity on the element, computed as the average L^2 norm
 229 of the nodal element velocities, h is the element size, computed as its diameter in the direction
 230 of the velocity to support using anisotropic meshes with highly stretched elements [41], and $c_{t,d}$
 231 are algorithmic constants taken as $c_t = 2$ and $c_d = 4$ for linear elements [40]. Equation (24) is
 232 discretized with a first-order-accurate time-integration scheme combining semi-implicit treatment
 233 of the convection term, implicit treatment of the viscous, pressure and divergence terms, and
 234 explicit treatment of the stabilization coefficients.

235 3.3.2. Adjoint Navier–Stokes equations

236 If the prescribed velocity condition (5) is used, application of the stabilized formulation, as
 237 described above, to the adjoint Navier–Stokes equations yields the following coarse scale variational
 238 problem

$$\begin{aligned} & \int_{\Omega} (-\rho \mathbf{u} \cdot \nabla \tilde{\mathbf{u}} + \rho \nabla \mathbf{u}^T \cdot \tilde{\mathbf{u}}) \cdot \mathbf{w} \, dv + \int_{\Omega} 2\mu \varepsilon(\tilde{\mathbf{u}}) : \varepsilon(\mathbf{w}) \, dv + \int_{\Omega} \tilde{p}(\nabla \cdot \mathbf{w}) \, dv + \int_{\Omega} (\nabla \cdot \tilde{\mathbf{u}}) q \, dv \\ & - \sum_{k=1}^{N_e} \int_{\Omega_k} \tilde{\tau}_1 \tilde{\mathbf{r}}_1 \cdot (-\rho \mathbf{u} \cdot \nabla \mathbf{w}) \, dv - \sum_{k=1}^{N_e} \int_{\Omega_k} \tilde{\tau}_1 \tilde{\mathbf{r}}_1 \cdot \nabla q \, dv - \sum_{k=1}^{N_e} \int_{\Omega_k} \tilde{\tau}_2 \tilde{r}_2 (\nabla \cdot \mathbf{w}) \, dv = 0. \end{aligned} \quad (28)$$

239 The associated momentum and continuity residuals read

$$-\tilde{\mathbf{r}}_1 = -\rho \mathbf{u} \cdot \nabla \tilde{\mathbf{u}} + \rho \nabla \mathbf{u}^T \cdot \tilde{\mathbf{u}} - \nabla \tilde{p}, \quad -\tilde{r}_2 = \nabla \cdot \tilde{\mathbf{u}}. \quad (29)$$

240 The stabilization coefficients are computed on each element after [42] as

$$\tilde{\tau}_1 = \frac{1}{(\tau_t^2(u) + \tau_d^2 + \tau_r^2)^{1/2}}, \quad \tilde{\tau}_2 = \tau_2, \quad (30)$$

241 with additional reaction-dominated limit, due to the $\rho \nabla \mathbf{u}^T \cdot \tilde{\mathbf{u}}$ term associated with the production
 242 of adjoint perturbations, defined as

$$\tau_r = \rho \nabla u, \quad (31)$$

243 where ∇u is a characteristic norm of $\nabla \mathbf{u}$ on the element, computed as the average L^2 norm of the
 244 nodal velocity gradients. It is important to note that the adjoint stabilization coefficients depend
 245 solely on \mathbf{u} , not $\tilde{\mathbf{u}}$, which is because the adjoint flow field is transported at (minus) the state
 246 velocity.

247 If the zero pressure/zero viscous stress condition (6) is used, then the adjoint coarse scale
 248 variational problem becomes

$$\begin{aligned} & \int_{\Omega} (-\rho \mathbf{u} \cdot \nabla \tilde{\mathbf{u}} + \rho \nabla \mathbf{u}^T \cdot \tilde{\mathbf{u}}) \cdot \mathbf{w} \, dv + \int_{\Omega} 2\mu \varepsilon(\tilde{\mathbf{u}}) : \varepsilon(\mathbf{w}) \, dv + \int_{\Omega} \tilde{p}(\nabla \cdot \mathbf{w}) \, dv + \int_{\Omega} (\nabla \cdot \tilde{\mathbf{u}}) q \, dv \\ & - \sum_{k=1}^{N_e} \int_{\Omega_k} \tilde{\tau}_1 \tilde{\mathbf{r}}_1 \cdot (-\rho \mathbf{u} \cdot \nabla \mathbf{w}) \, dv - \sum_{k=1}^{N_e} \int_{\Omega_k} \tilde{\tau}_1 \tilde{\mathbf{r}}_1 \cdot \nabla q \, dv - \sum_{k=1}^{N_e} \int_{\Omega_k} \tilde{\tau}_2 \tilde{r}_2 (\nabla \cdot \mathbf{w}) \, dv \\ & - \int_{\Gamma_o} \rho(\mathbf{u} \cdot \mathbf{n})(\tilde{\mathbf{u}} \cdot \mathbf{w}) \, ds = \int_{\Gamma_o} \partial_{\mathbf{u}} J \cdot \mathbf{w} \, ds, \end{aligned} \quad (32)$$

249 because the integration by part of the pressure and viscous terms unveils a boundary term

$$\int_{\partial\Omega} (\tilde{p} \mathbf{n} + 2\mu \varepsilon(\tilde{\mathbf{u}}) \cdot \mathbf{n}) \cdot \mathbf{w} \, ds = - \int_{\Gamma_o} (\rho(\mathbf{u} \cdot \mathbf{n}) \tilde{\mathbf{u}} + \partial_{\mathbf{u}} J) \cdot \mathbf{w} \, ds, \quad (33)$$

250 evaluated at the outlet due to the adjoint boundary condition (16).

251 Both problems are fully implicitly integrated, except the outflow boundary term in (32) that
 252 needs be treated explicitly for implementation convenience. Even though the last computed adjoint
 253 solution (hence pertaining to the previous design) is used to evaluate the boundary term, this simple
 254 scheme has been found to converge to identical shapes and cost function minimum, compared to
 255 solving iteratively with relaxed sub-iterations. Due to the linearity of Eqs. (11)-(12), this in turn
 256 cuts down the numerical effort, as only one single linear system needs be solved at each update
 257 step, regardless of the state outflow condition.

258 *3.3.3. Interface update scheme using the convective level set method*

259 The auto-reinitialization level set problem (22) is solved with an SUPG method, whose stabi-
 260 lization proceeds from that of the ubiquitous convection-diffusion-reaction equation (43, 44). The
 261 associated variational problem is formulated as

$$\int_{\Omega} (\partial_{\tau} \phi + \mathbf{a}_{\tau} \cdot \nabla \phi) \xi \, dv - \int_{\Omega_k} \tau_3 r_3 \mathbf{a}_{\tau} \cdot \nabla \xi \, dv = \int_{\Omega} S \xi \, dv, \quad (34)$$

262 with residual

$$-r_3 = \partial_{\tau} \phi + \mathbf{a}_{\tau} \cdot \nabla \phi - S, \quad (35)$$

263 and stabilization coefficient

$$\tau_3 = \frac{1}{\tau_t(a_{\tau})}. \quad (36)$$

264 It is easily checked that all terms scale as $1/\Delta\tau$, so we can set $\Delta\tau = 1$ without any loss of generality
 265 because the solution is ultimately independent on the pseudo-time step value. Equation (34) is
 266 solved with semi-implicit treatment of the convection term (as the convection velocity \mathbf{a}_{τ} depends
 267 on main unknown ϕ) and explicit treatment of the source term and stabilization coefficients.

268 4. Anisotropic mesh adaptation

269 4.1. Construction of an anisotropic mesh

270 The main idea of anisotropic, metric-based mesh adaptation is to generate a uniform mesh (with
 271 unit length edges and regular elements) in a prescribed Riemannian metric space, but anisotropic
 272 and well adapted (with highly stretched elements) in the Euclidean space. Assuming that, in the
 273 context of metric-based adaptation methods, controlling the interpolation error suffices to master
 274 the global approximation error, the objective can be formulated as finding the mesh, made up
 275 of at most N_n nodes, that minimizes the linear interpolation error in the L^1 norm. Following
 276 the lines of (45, 46), an edge-based error estimator combined to a gradient recovery procedure is
 277 used to compute, for each node, a metric tensor that prescribes a set of anisotropic directions and
 278 stretching factors along these directions, without any direct information from the elements, nor any
 279 underlying interpolation. The optimal stretching factor field is obtained by solving an optimization
 280 problem using the equi-distribution principle under the constraint of a fixed number of nodes in
 281 the mesh, after which a new mesh is generated using the parallel procedure described in Sec. 5

282 4.2. Edge error estimate

283 Given a mesh Ω_h of the domain Ω , we denote by \mathbf{x}^{ij} the edge connecting a given node \mathbf{x}^i to
 284 $\mathbf{x}^j \in \Sigma(i)$, where $\Sigma(i)$ is the set of nodes connected to \mathbf{x}^i , and the number of such nodes is noted
 285 as $|\Sigma(i)|$. Also, given a regular analytical scalar function ψ defined on Ω , and its P1 finite element
 286 approximation ψ_h computed on Ω_h , we follow (45) and estimate the interpolation error along the
 287 edge \mathbf{x}^{ij} as the projection along the edge of the second derivative of ψ . This is obtained projecting
 288 along the edge a Taylor expansion of the gradient of ψ at \mathbf{x}^j to give

$$\varepsilon_{ij} = |\mathbf{g}^{ij} \cdot \mathbf{x}^{ij}|, \quad (37)$$

289 where the i and j superscripts indicate nodal values at nodes \mathbf{x}^i and \mathbf{x}^j , respectively, $\mathbf{g}^i = \nabla\psi(\mathbf{x}^i)$
 290 is the exact value of the gradient at \mathbf{x}^i , and $\mathbf{g}^{ij} = \mathbf{g}^j - \mathbf{g}^i$ is the variation of the gradient along
 291 the edge. Although Eq. (37) involves only values of the gradient at the edge extremities and can
 292 thus be evaluated without resorting to resource expensive Hessian reconstruction methods, this
 293 however requires the gradient of ψ to be known and continuous at the nodes, which in turn requires
 294 full knowledge of ψ . Meanwhile, only the linear interpolate ψ_h is known in practice, whose gradient
 295 is piecewise constant and discontinuous from element to element, although its projection along the
 296 edges is continuous since it depends only on the nodal values of the field.

297 A recovery procedure is thus used to build a continuous gradient estimator defined directly at
 298 the nodes. It is shown in [45] that a suitable error estimate preserving second-order accuracy is
 299 obtained substituting the reconstructed gradient for the exact gradient in (37), to give

$$\varepsilon_{ij} = |\bar{\mathbf{g}}^{ij} \cdot \mathbf{x}^{ij}|, \quad (38)$$

300 where $\bar{\mathbf{g}}^{ij} = \bar{\mathbf{g}}^j - \bar{\mathbf{g}}^i$ and we denote by $\bar{\mathbf{g}}^i$ the recovered gradient of ψ_h at node \mathbf{x}^i . The latter is
 301 defined in a least-square sense as

$$\bar{\mathbf{g}}^i = \operatorname{argmin}_{\mathbf{g} \in \mathbb{R}^d} \sum_{j \in \Sigma(i)} |(\mathbf{g} - \nabla \psi_h) \cdot \mathbf{x}^{ij}|^2, \quad (39)$$

302 for which an approximate solution using the nodal values as sole input is shown in [45] to be

$$\bar{\mathbf{g}}^i = (\mathbf{X}^i)^{-1} \cdot \sum_{j \in \Sigma(i)} (\psi_h(\mathbf{x}^j) - \psi_h(\mathbf{x}^i)) \mathbf{x}^{ij}, \quad (40)$$

303 where \mathbf{X}^i is the length distribution tensor defined as

$$\mathbf{X}^i = \frac{1}{|\Sigma(i)|} \sum_{j \in \Sigma(i)} \mathbf{x}^{ij} \otimes \mathbf{x}^{ij}, \quad (41)$$

304 that gives an average representation of the distribution of the edges sharing an extremity.

305 4.3. Metric construction

306 In order to relate the error indicator ε_{ij} defined in (38) to a metric suitable for mesh adaptation
 307 purposes, we introduce the stretching factor s_{ij} as the ratio between the length of the edge \mathbf{x}^{ij}
 308 after and before the adaptation. The metric at node \mathbf{x}^i is sought to generate unit stretched edge
 309 length in the metric space, that is,

$$(s_{ij} \mathbf{x}^{ij})^T \cdot \mathbf{M}^i \cdot (s_{ij} \mathbf{x}^{ij}) = 1, \quad \forall j \in \Sigma(i), \quad (42)$$

310 for which an approximate least-square solution is shown in [45] to be

$$\mathbf{M}^i = \left(\frac{d}{|\Sigma(i)|} \sum_{j \in \Sigma(i)} s_{ij}^2 \mathbf{x}^{ij} \otimes \mathbf{x}^{ij} \right)^{-1}, \quad (43)$$

311 provided the nodes in $\Sigma(i)$ form at least d non co-linear edges with \mathbf{x}^i , which holds if the mesh is
 312 valid. The metric solution of (43) is ultimately computed setting a target total number of nodes
 313 N_n . Assuming a total error equi-distributed among all edges, the stretching factor is shown in [46]
 314 to be

$$s_{ij} = \left(\frac{\sum_i N_i(1)}{N_n} \right)^{\frac{2}{d}} \varepsilon_{ij}^{-1/2}, \quad (44)$$

315 where $N_i(1)$ is the number of nodes generated in the vicinity of node \mathbf{x}^i for a unit error, given by

$$N_i(1) = \left(\det \left(\frac{d}{|\Sigma(i)|} \sum_{j \in \Sigma(i)} \varepsilon_{ij}^{1/2} \frac{\mathbf{x}^{ij}}{|\mathbf{x}^{ij}|} \otimes \frac{\mathbf{x}^{ij}}{|\mathbf{x}^{ij}|} \right) \right)^{-1/2}. \quad (45)$$

Algorithm 2 Anisotropic mesh adaptation algorithm

Require: Anisotropic adapted mesh

- 1: Set number of nodes N_n
 - 2: Compute ψ_h on current mesh
 - 3: **for** each node \mathbf{x}^i **do**
 - 4: Compute length distribution tensor \mathbf{X}^i using (41)
 - 5: Compute nodal recovered gradient $\bar{\mathbf{g}}^i$ using (40)
 - 6: **for** all edges \mathbf{x}^{ij} **do**
 - 7: Compute edge recovered gradient $\bar{\mathbf{g}}^{ij}$
 - 8: Compute edge-based error ε_{ij} using (38)
 - 9: Compute stretching factor s_{ij} using (44)
 - 10: Compute metric \mathbf{M}^i using (43)
 - 11: Generate new mesh by local improvement in the neighborhood of the nodes and edges (47)
 - 12: Interpolate ψ_h on new mesh using classical linear interpolation
-

316 4.4. Level set-based adaptation criteria

317 In order to simplify and clarify the presentation, the main steps needed for metric construction
 318 at the nodes is summarized in algorithm 2. In practice, the sole variable used for error estimation
 319 purpose is the filtered level set defined in (21), as it satisfies the metric property in a thin layer
 320 around the interface (in particular it preserves the zero iso-value of φ , which is the only relevant
 321 information for mesh adaptation purposes), but avoids unnecessary adaption of the mesh further
 322 away from the interface, where the interpolation error is close-to-zero, due to $\|\nabla\phi\| \sim 0$. This means
 323 that the criterion for mesh adaptation is purely geometric, i.e., the same mesh is pre-adapted around
 324 the fluid-solid interface, then used to compute all quantities needed to perform the next design
 325 update step. Nonetheless, it is worth mentioning that the approach also supports more complex
 326 adaptation criteria featuring physical quantities, thus providing the ability to dynamically adapt
 327 the mesh during the simulations. The common method to adapt a mesh to several variables is to
 328 combine the metrics corresponding to each individual variable using metric intersection algorithms,
 329 which is known to incur a relatively high computational cost and to have potentially non-unique,
 330 suboptimal outcome. Conversely, the present approach allows building directly a unique metric
 331 from a multi-component error vector combining level set and any relevant flow quantity of interest,
 332 as definition (38) is easily extended to account for several sources of error (48). Indeed, if we consider
 333 $\boldsymbol{\psi} = (\psi_1, \psi_2, \dots, \psi_p)$ a vector consisting of p scalar variables, it comes out straightforwardly that
 334 the error is now a vector $\boldsymbol{\varepsilon}_{ij} = (\varepsilon_{ij,1}, \varepsilon_{ij,2}, \dots, \varepsilon_{ij,p})$, whose L^2 norm can serve as simple error value
 335 for the edge from which to compute the stretching factor (44) and ultimately, the metric solution
 336 of (43). For instance, the $2d + 3$ sized nodal vector field defined as

$$337 \quad \boldsymbol{\psi}_h(\mathbf{x}^i) = \left(\frac{\phi_h^i}{\max_{j \in \Sigma(i)} \phi_h^j}, \frac{u_{h_{k \in \{1 \dots d\}}}^i}{\|\mathbf{u}_h^i\|}, \frac{\|\mathbf{u}_h^i\|}{\max_{j \in \Sigma(i)} \|\mathbf{u}_h^j\|}, \frac{\tilde{u}_{h_{k \in \{1 \dots d\}}}^i}{\|\tilde{\mathbf{u}}_h^i\|}, \frac{\|\tilde{\mathbf{u}}_h^i\|}{\max_{j \in \Sigma(i)} \|\tilde{\mathbf{u}}_h^j\|} \right), \quad (46)$$

338 can be used to combine adaptivity with respect to the norm and direction of the state and adjoint
 339 velocity vectors, in addition to the level set. Because all fields are normalized by their respective
 340 global maximum, a field much larger in magnitude cannot dominate the error estimator, meaning
 341 that the variations of all variables are fairly taken into account. This benefits problems involving
 342 more complex physics, e.g., turbulence, heat transfer, fluid-structure interaction, multiple phases,
 343 possibly in interaction with one another, all the more so in the context of topology optimization, as
 344 the difference in the spatial supports of the state and adjoint quantities (that stems from the non-
 345 normality of the linearized evolution operator (49)) may otherwise yield conflicting requirements in
 terms of the regions of the computational domain most in need of refinement.

346 5. Parallel computational framework

347 The numerical resolution framework relies on the in-house, parallel, finite element library Cim-
 348 LIB_CFD (48), whose organization relies on fundamental choices allowing an efficient implementa-

349 tion of high-level parallel algorithms. We discuss below the efficient tools to generate and adapt the
350 meshes, and to solve the large-scale linear systems arising from the finite element discretization,
351 steps where most computational time is spent. Another key aspect of the method is the ability of
352 the stabilized finite element formulations to support using anisotropic adapted meshes in both the
353 fluid and solid domains, regardless of the problem dimensionality, On the one hand, using linear
354 approximations for all variables drastically reduces the size of the systems that need be solved. To
355 give a taste, the meshes used herein are make up of about 5 million elements yielding a total of
356 3.5 million degrees of freedom, but 20 million degrees of freedom using quadratic approximations
357 for the velocities, hence a reduction by nearly 80%. On the other hand, using anisotropic meshes
358 decreases the cost of improving the numerical precision, as the number of nodes needs be increased
359 only in the direction of interest. This makes a huge difference in 3-D calculations, as the accuracy
360 can be improved by a factor of 2 (in the best case scenario) using only 2 times as many nodes,
361 instead of 8.

362 5.1. Parallel resolution

363 Computing the numerical solutions to the governing **Navier–Stokes, adjoint Navier–Stokes and**
364 **level-set advection equations** considered herein requires solving large-scale linear systems (or non-
365 linear systems that may lead to the resolution of several linear systems if an implicit discretization
366 scheme is used). To this end, the resolution step makes a clear distinction between those large-
367 scale systems that need be stored and solved, and their local contributions at the element levels.
368 Namely, all finite element formulations are only implemented sequentially at the element level,
369 then assembled and solved in parallel using the PETSc library [50], that offers a wide range
370 of parallel data structures (linear and non-linear solvers as well as preconditioners) and can be
371 run on large computing clusters. Here, only semi-implicit and explicit discretization schemes are
372 used, and the associated linear systems are sufficiently well conditioned to be solved by iterative
373 methods. We thus use the Generalized Minimal Residual algorithm with block Jacobi incomplete
374 LU preconditioning, and consider the solutions to be converged if the absolute residuals are less
375 than 10^{-6} .

376 5.2. Parallel adaptive remeshing

377 Although most numerical solvers have embraced parallel computing as a way to continue to
378 improve performance, it is less common to see massively parallel computation using anisotropic
379 adapted unstructured meshes, let alone if the mesh is dynamically adapted to track the interface
380 deformations. An original parallelization strategy is used here for the mesh adaptation step, based
381 on an independent subdomain remeshing under the constraint of blocked interfaces. **An initial**
382 **mesh is partitioned into several submeshes using a parallel graph/mesh partitioning/repartitioning**
383 **algorithm that allows to balance well the number of mesh entities (vertices or elements) per proces-**
384 **sor [51–53]. Remeshing operations are then performed with a sequential mesh adaptator on each**
385 **subdomain with an extra treatment of the interfaces, using the procedure described in [47], based**
386 **on a topological representation of the computational domain. In practice, a level-set based error**
387 **estimate is computed for each subdomain. An iterative approach is used, in which remeshing is**
388 **performed concurrently on each processor while the interfaces between sub-domains are locked to**
389 **avoid any communication between processors. Then, to obtain a satisfactory final mesh regarding**
390 **the quality function, a repartitioning step is performed to move the interface inside the domain in**
391 **order to enable re-meshing in a next phase. As illustrated in Figure 1, the algorithm iterates until**
392 **all items have been re-meshed. Finally, the new mesh is repartitioned over the allocated CPUs to**
393 **take into account for the changes of mesh topology in the computational loads distribution. Note,**
394 **the constraint on the number of mesh elements could be considered as local to each subdomain. In**
395 **this case, solving the error estimate problem is straightforward, as all computations are local and**
396 **there is no need to exchange data between the processors. The local constraint on the number of**
397 **elements implies the generation of a new mesh with the same number of elements per processor.**
398 **This allows avoiding heavy load balancing cost after each mesh adaptation, but tends towards an**
399 **overestimate of the mesh density on subdomains where flow activity is almost neglected. From a**
400 **scaling point of view, such an approach leads to a weak scalability model for which the problem size**
401 **grows linearly with respect to the number of processors. To derive a hard scalability model with**

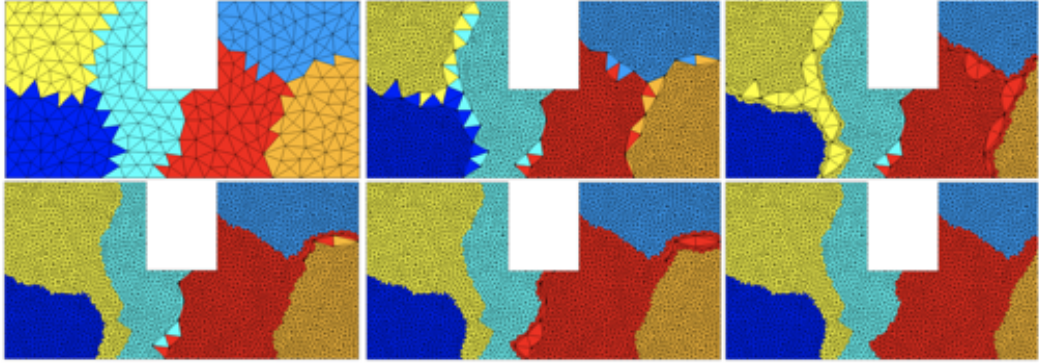


Figure 1: Illustration of the iterative parallel remeshing steps on a model 2-D distributed mesh.

402 good parallel performances, the constraint on the number of elements for the new generated mesh is
 403 thus handled globally, with the global number of elements over the entire domain distributed with
 404 respect to the mesh density prescribed by the error estimator. In doing so, the parallel behavior of
 405 the mesh adaptation is very close to the serial one and the error analysis is still the same, although
 406 load rebalancing is required after each mesh adaptation stage.

407 Because the parallel remesher is made of nested iterations between remeshing and repartitioning,
 408 the metric map providing both the size and the stretching of mesh elements needs to be transported
 409 after each repartitioning step. Indeed, it is given as a nodal field, hence one scalar value per node,
 410 and must be updated after each migration or renumbering. The parallel adaptive remeshing is
 411 thus the combination of three iteratively nested steps: (i) independent adaptive remeshing per
 412 subdomain, (ii) constrained repartitioning and (iii) updating the metric map. Depending on the
 413 problem dimensionality, three to five iterations of remeshing and repartitioning are typically needed
 414 to build the optimal mesh, but the time spent per iteration decreases drastically as fewer and fewer
 415 elements and nodes need to be moved and migrated across processors, as we only need to move
 416 bad quality zones inside the domain in order to remesh them. For several test cases in two and
 417 three dimensions, this simple approach is shown in [54] to yield close-to-optimal parallel remeshing
 418 speed-up up to 32 cores.

419 5.3. Dynamic load balancing

420 In this work, we follow the same load balancing strategy than in Ref. [51], to which the inter-
 421 ested reader is referred for technical details. A cost function is defined and takes into account
 422 the theoretical computation and communication time of the allocated resources. Then, the load
 423 balancing process is realized using two major steps: (i) forming disjoint pairs of processors that
 424 are susceptible to minimize the cost function, and (ii) optimizing the cartography on each pair.
 425 This optimization is done by transferring mesh nodes or mesh cells from a processor to the other
 426 using the notion of strip migration. These two steps are repeated as long as the global cost of the
 427 partition can be optimized. The results from [103] show that the use of this method on various
 428 system architectures allows accelerating the mesh partitioning process. In terms of scalability, a
 429 linear behavior is observed. An example of load balancing is given in Fig. 2 for the same example
 430 than in Fig. 1 after the parallel remeshing procedure. One notices that the partition in Fig. 2(a)
 431 is not optimal, as the size of the interfaces is too large, which could have a damaging impact on
 432 the communication costs. The cost function is thus optimized using the previously described load
 433 balancing procedure by transferring nodes from one processor to the other, in order to obtain the
 434 final optimal partition presented on the right side of Fig. 2(b).

435 6. Numerical implementation

436 6.1. Geometrical constraints

437 Fluid flow topology optimization is generally performed under geometrical constraints, typi-
 438 cally, constant or upper bounded surfaces and/or volumes to avoid the two extreme cases of the

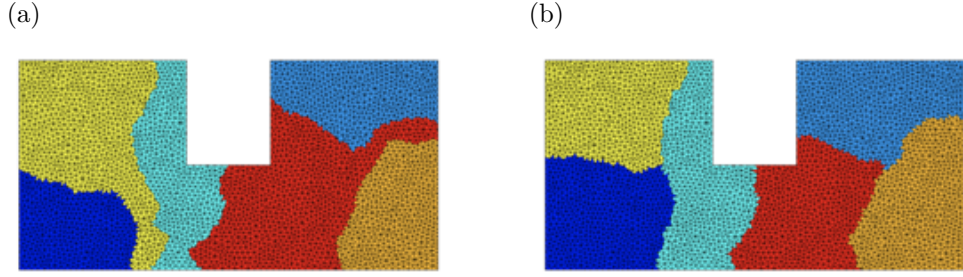


Figure 2: (a) Before and (b) after load balancing for the same model 2-D distributed mesh as in Fig. 1

439 solid domain clogging the entire design domain or disappearing altogether, as in pressure drop
 440 minimization and drag minimization problems, respectively. This is usually done adding penalty
 441 terms to the Lagrangian, each of which consists of an empirical penalty parameter multiplied by
 442 a measure of violation of the constraint, whose variations with respect to the state and design
 443 variables snowballs into the derivation of the adjoint problem and of the cost function sensitivity.
 444 Here, the constraint of a constant volume of fluid V_{target} is applied a posteriori, i.e., we solve the
 445 unconstrained problem presented in Sec. 2 with no penalty term added to the Lagrangian (although
 446 the optimization remains subject to Navier–Stokes as state equations). Once the convective level
 447 set method presented in Sec. 3.3.3 has updated the interface position, a first pass of anisotropic
 448 mesh adaptation is performed, after which the volume of the fluid domain is computed as

$$V_\varphi = \int_{\Omega} H_\epsilon(\varphi) \, dv, \quad (47)$$

449 where H_ϵ is the smoothed Heaviside function on the fluid domain defined as

$$H_\epsilon(\varphi) = \begin{cases} 1 & \text{if } \varphi < -\epsilon, \\ \frac{1}{2} \left(1 - \frac{\varphi}{\epsilon} - \frac{1}{\pi} \sin\left(\pi \frac{\varphi}{\epsilon}\right) \right) & \text{if } |\varphi| \leq \epsilon, \\ 0 & \text{if } \varphi > \epsilon, \end{cases} \quad (48)$$

450 and ϵ is a regularization parameter set to $2h_1$. A simple dichotomy approach is then used to
 451 optimize a constant deformation $\delta\varphi$ meant to enlarge ($\delta\varphi < 0$) or shrink ($\delta\varphi > 0$) the fluid domain,
 452 until the difference $|V_{\varphi+\delta\varphi} - V_{target}|$ between the actual and target volumes drops below a certain
 453 tolerance, at which point we cut off $\varphi + \delta\varphi$ and perform a second pass of mesh adaptation. Two
 454 points are worth mentioning: first, because each offset changes the min-max values of the trunca-
 455 tion, the above procedure requires knowledge of the level set φ , not just the filtered level set ϕ .
 456 A brute force algorithm therefore performs beforehand a complete reconstruction of the distance
 457 function from the zero iso-value of ϕ , as only the filtered level set (not the level set) is evolved
 458 during the convection-reinitialization step. Second, only small deformations are considered so that
 459 no intermediate mesh adaptation passes are required. By doing so, the total cost is essentially that
 460 of performing the second pass of mesh adaptation, as further discussed in the following.

461 6.2. Steepest descent update rule

462 In practice, the displacement used to perform the update step is defined as

$$\beta = -\theta \frac{\mu(\nabla \tilde{\mathbf{u}} \cdot \mathbf{n}) \cdot (\nabla \mathbf{u} \cdot \mathbf{n}) \chi_\Gamma(\mathbf{x})}{\max_{\Omega} \mu(\nabla \tilde{\mathbf{u}} \cdot \mathbf{n}) \cdot (\nabla \mathbf{u} \cdot \mathbf{n}) \chi_\Gamma(\mathbf{x}) \prod_l \zeta(\|\mathbf{x} - \mathbf{X}_s^l\|)}, \quad (49)$$

463 where $\theta > 0$ is a descent factor controlling the step taken in the gradient direction, and χ_Γ and
 464 ζ are activation functions between 0 and 1 ensuring that the design is fittingly updated only in
 465 relevant regions of the computational domain. More details are as follows:

- 466 • χ_Γ is a binary filter returning a value of 1 only at nodes within a distance E of the interface.
 467 This is because the normal vector in a level set framework is recovered as $\mathbf{n} = \nabla\phi/\|\nabla\phi\|$, so

the displacement is non-zero in the whole fluid domain, even far from the interface where \mathbf{n} has unit norm because $\|\nabla\phi\|$ only tends asymptotically to zero. In return, the update step can break down numerically at nodes nearly equidistant from two subparts of the interfaces, which can occur for instance at the centerline of a channel.

- ζ is a smooth filter assigning 0 value to some subset $\mathbf{X}_s^l \in \partial\Omega$ that can be either a point or a curve, and $\|\mathbf{x} - \mathbf{X}_s^l\|$ is the shortest-path distance to \mathbf{X}_s^l . Such subsets are singled out prior to optimization, because the flow there may be driven to a singularity, and ill-defined velocity gradients may cause large, unphysical displacements. Such singularities can be dealt with numerically by appending fluid/solid Dirichlet boundary conditions to the level set convection-reinitialization problem. Nonetheless, they must not be included in the normalization step to avoid forcing excessively small displacements along the remaining part of the interface, and thereby considerably slowing down the convergence rate of the iterative optimization process. We use here hyperbolic tangent filters

$$\zeta(r) = \frac{1}{2} + \frac{1}{2} \tanh\left(\alpha_s \tan\left(-\frac{\pi}{2} + \frac{\pi}{2} \frac{r}{r_s + \epsilon_{s1}} + \epsilon_{s2}\right)\right), \quad (50)$$

increasing from 0 to 1 within a distance of $2r_s$ from the singularity, with r_s a transition radius such that

$$4r_s < \min_{l,m} \|\mathbf{X}_s^l - \mathbf{X}_s^m\|, \quad (51)$$

to prevent overlaps, α_s a steepness parameter controlling the sharpness of the transition, and $\epsilon_{s1,2}$ small regularization parameters to avoid local discontinuities.

Ultimately, the above filtering and normalization steps ensure that the level set is updated using a displacement that is non-zero only in a thin layer of thickness E about the interface, minus a certain number of singular subsets.

6.3. Descent factor

It follows from Eq. (49) that the descent factor θ physically represents the maximum displacement amplitude over the update region of interest. In practice, though, the actual numerical displacement, estimated from the difference between zero iso-value of the filtered level set before and after transport, has been found to be well below its theoretical value. This is because the state and adjoint velocities are forced to zero on the solid domain. The displacement, being driven by the velocity gradients, is thus also zero everywhere in the solid, except in a very narrow region about the interface, typically a couple of elements thick. As a result, it is not possible to explicitly control the displacement achieved numerically at each iteration. A simple scheme to do so would have been to repeatedly evolve the interface with a small descent factor until the difference between the cumulated and target displacement drops below a certain tolerance. However, the interface can be evolved only once per update step, as the gradient information is lost if the displacement happens to be in the direction of the solid, for the same reason mentioned above. We thus tune the descent factor manually on a case by case basis, for the achieved displacement to be slightly smaller than the cut-off thickness. This has been found to be a satisfactory trade-off between accuracy and numerical effort, as the number of iterations required for convergence remains affordable, and the position of the evolved interface is accurately tracked. Displacements larger than the cut off thickness conversely move the level set into regions of the computational domain lacking the proper mesh refinement, which has been found to ultimately affect the accuracy of the interface representation.

6.4. General algorithm

Figure 3 shows the flowchart of the implemented topology optimization algorithm, in which anisotropic mesh adaptation is key to capture the interface with the highest precision possible. The necessary algorithmic parameters common to all examples documented in the following are given in Tab. 1. Note, as a consequence of the level set-based technique used to enforce the volume

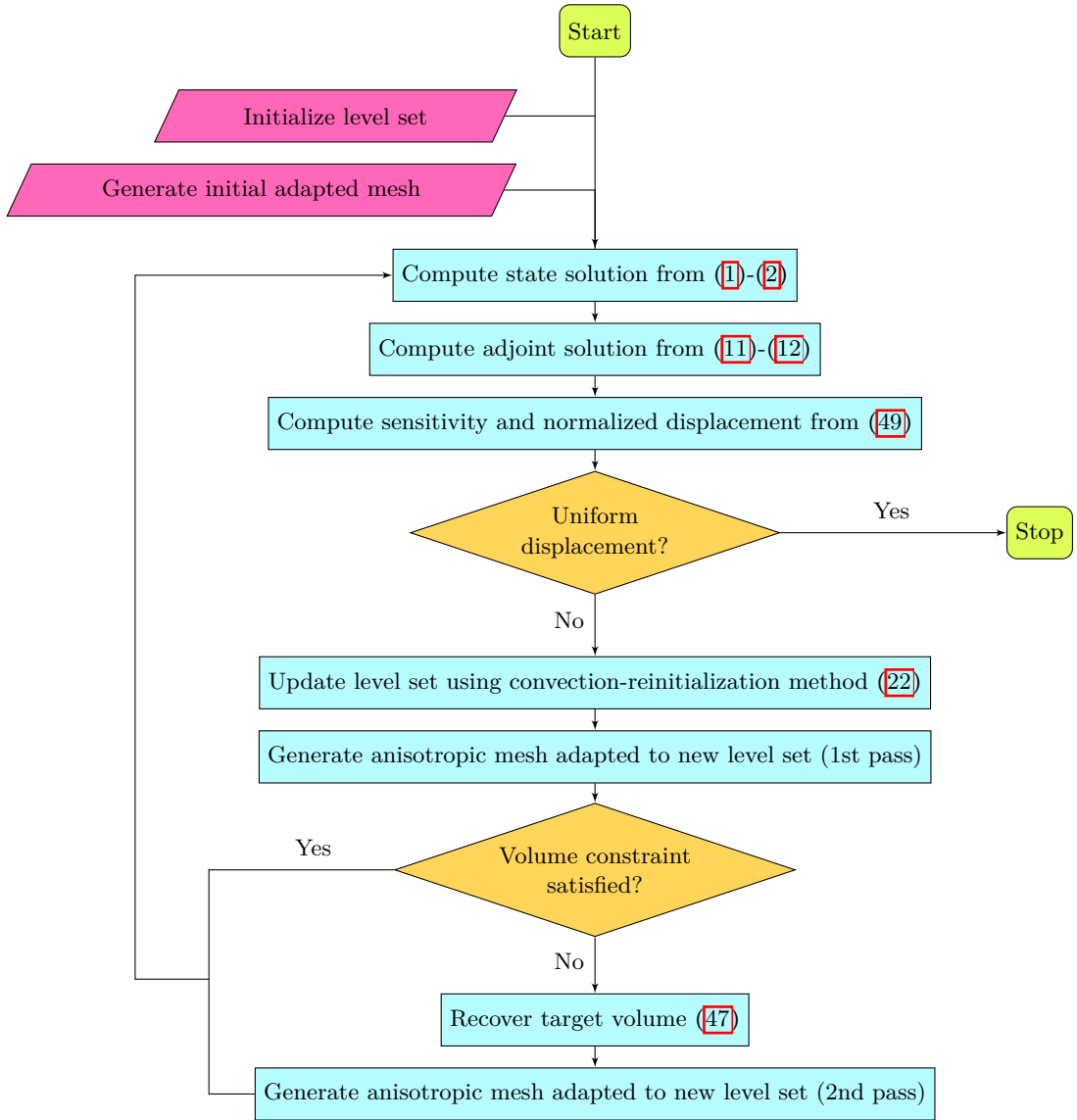


Figure 3: Flowchart of performance topology optimization procedure.

513 of fluid constraint, convergence is achieved not when the displacement is identically zero, as would
 514 be the case using a penalized Lagrangian approach, but when the displacement is uniform along
 515 the interface. This is not easily done on the fly, though, so we rather iterate until a maximum
 516 number of iterations has been reached and estimate convergence a posteriori.

517 7. Numerical benchmarks

518 This section assesses the accuracy and efficiency of the numerical framework through a series
 519 of topology optimization problems, for which the novelty lies not necessarily in the associated
 520 optimal designs themselves, but in the accuracy to which the optimal interfaces are captured in
 521 the simulation model. Each problem is tackled on 64 cores of a cluster of AMD Rome EPYC 7502
 522 bi-processors. The cost function to minimize is the net inward flux of total pressure through the
 523 boundaries, taken as a measure of the total power dissipated by a fluid dynamic device. Since the
 524 orientation of the normal \mathbf{n} yields $\mathbf{u} \cdot \mathbf{n}|_{\Gamma_i} > 0$ and $\mathbf{u} \cdot \mathbf{n}|_{\Gamma_o} < 0$, this can be expressed in the form

$h_{\perp} = 0.0001$	Min. interface normal mesh size
$\Delta t = 0.1$	CFD Numerical time step
$E = 0.005$	Level set cut off thickness
$ \delta\varphi = 0.0005$	Initial volume recovery offset
$r_s = 0.0125$	Transition radius
$\alpha_s = 2.1$	Sharpness parameter
$(\epsilon_{s1}, \epsilon_{s2}) = (0.0005, 0.005)$	Regularization parameters

Table 1: Algorithmic parameters.

of (7) using

$$J = p_{tot}(\mathbf{u} \cdot \mathbf{n}) = (p + \frac{1}{2}\rho(\mathbf{u} \cdot \mathbf{u}))(\mathbf{u} \cdot \mathbf{n}). \quad (52)$$

All examples aim at finding the best path for a fluid to flow in a reference design domains under the form of cubic or cuboid (parallelepipedic) cavities, with either a single or multiple identical inlets, all cylindrical, and a single or multiple identical outlets, either cylindrical or rectangular. For each case, the sole control parameter is the Reynolds number, built here on inlet diameter and maximum inlet velocity (the same for all inlets).

The remainder of the practical implementation details are as follows:

- All design domains are initialized with solid inclusions coming in various shapes and sizes. **No new holes are created over the course of optimization, in the absence of a dedicated mechanism for seeding solid occlusions, but** from experience, all problems tackled in the following are essentially insensitive to the initial design provided a sufficiently large number of inclusions is used.
- The admissible error on the target volume is set to 1% in two dimensions, and 5% in three dimensions.
- The fluid is systematically conveyed into and out of the design domain using leads of length l_i (the same at all inlets) and l_o (the same at all outlets) appended normal to the boundary. This is for numerical consistency, as the exact problem formulation in the literature may vary depending on the case, and it is not always clear whether such leads should be included in the design domain. **This is the case here,** although the leads are not considered in the volume constraint, neither in definition of the target volume nor in the computation of the volume of fluid.
- The singular subsets excluded from the displacement normalization step are the sharp intersections between the leads and the boundary of the cavities, hence each smooth filter ζ transitions from 0 to 1 over either a torus of minor radius $2r_s$ (for all inlets and cylindrical outlets) or a set of intersecting cylinders of radius $2r_s$ (for all rectangular outlets). Note, this is not a consequence of explicitly representing the leads, as the exact same procedure has been found suitable without such appendage.
- The leads are excluded from the displacement normalization step, for which we simply add to the max argument of (49) a binary filter returning a value of 0 at all nodes located inside the pipes. This is again to avoid slowing down the convergence rate of the iterative optimization process, as the maximum displacement is otherwise located in the leads, because the easiest way to minimize the dissipated power is to suppress the flow by having the solid entirely clogging the leads.
- Without seeking to optimize the performance, all optimization runs have been found to converge within a few hundreds iterations, which is essentially the number of steps used to fulfill the fluid volume constraint while ensuring that the displacement achieved at each iteration remains below the level set cut-off thickness (more details in the following).

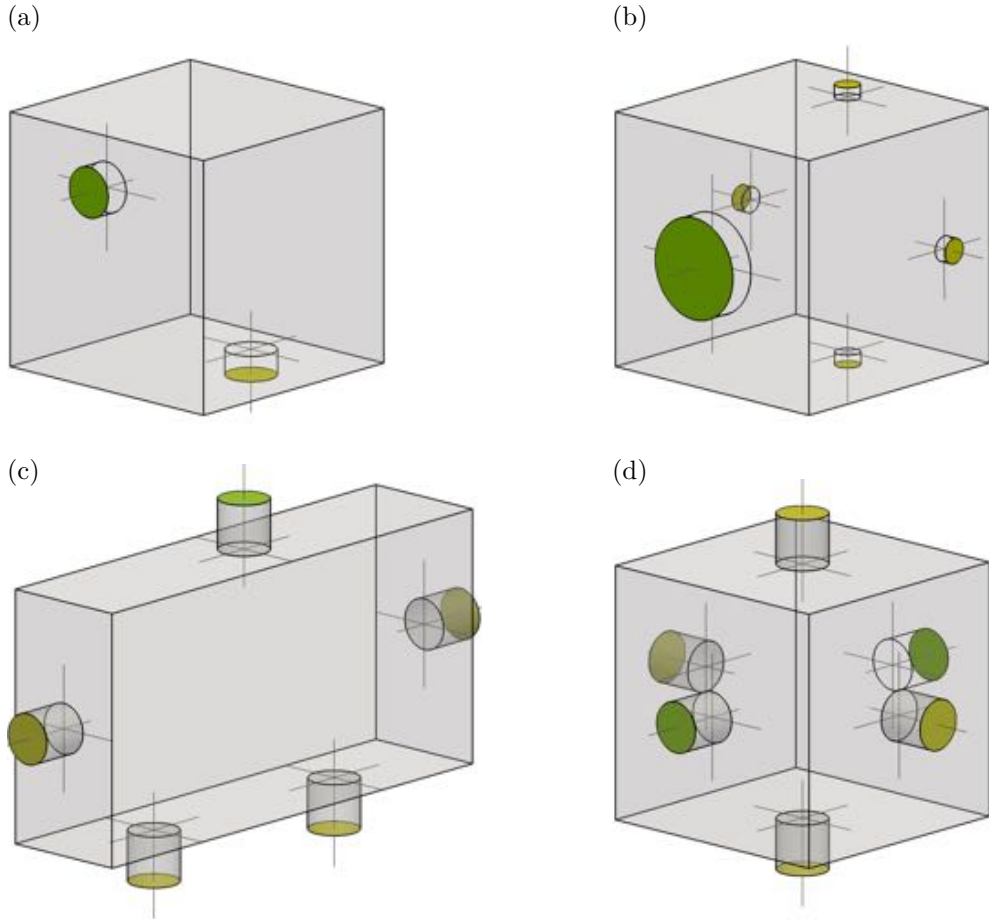


Figure 4: Problem set-up for the (a) single inlet/single outlet (b-c) single inlet/multiple outlets and (d) multiple inlets/multiple outlets examples.

- All 3-D meshes have been checked to have an element-to-node ratio close to 5, as should be for dens meshes made up of tetrahedral elements. In order to ease the comparison with the available literature, the mesh information is thus documented in the following in terms of its equivalent number of elements, defined as $N_{el} = 5N_n$.

Finally, all systems considered in the following have from 1 up to 3 reflectional symmetries. Nonetheless, we do not reduce the computational cost by modeling only a half (or a quarter/eighth) of the domain together with symmetry boundary conditions, which is feasible [24] but would somehow contradict the objective of assessing the method in the context of large-scale CFD systems. The entire domain is thus discretized, and we let symmetry arise as a result of the optimization process, even though this likely increases the number of iterations needed to achieve convergence.

7.1. Single inlet/single outlet duct flow

We optimize first the single inlet / single outlet duct flow whose setup is shown in Fig. 4(a). The design domain is a cubic cavity of unit length, that has one circular inlet on the left side, one circular outlet at the bottom, and reflectional symmetry with respect to the inlet/outlet plane. The aim is to determine the optimal design of the pipe bend that connects the inlet to the outlet and minimizes the dissipated power subject to the constraint that the fluid must occupy a given fraction of the total volume. The boundary conditions for this case consist of a normal to the boundary, parabolic inlet velocity profile and a zero pressure/zero viscous stress condition. This is a classical benchmark for 2-D topology optimization in fluid dynamics [12, 15, 55-58], hence the line of thought here is to provide first a verification and characterization of the method in two dimensions, then extend the analysis in three dimensions, for which the amount of available literature is much more scarce [24, 58].

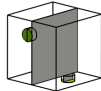
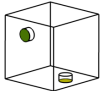
		
$\Omega = [0; 1] \times [0; 1]$	$[0; 1] \times [0; 1] \times [0; 1]$	Design domain
$d = 2$	3	Problem dimensionality
$V_{target} = 0.25$	0.039	Target volume of fluid
$V_{\varphi,0} = 0.25$	0.76	Initial volume of fluid
$Re = 2$	>	Reynolds number
$\mathbf{x}_i = (-0.1, 0.8)$	$(-0.1, 0.8, 0.5)$	Inlet center coordinates
$u_i = 0.2$	>	Inlet centerline velocity
$e_i = 0.2$	>	Inlet diameter
$l_i = 0.1$	>	Inlet leads length
$\mathbf{x}_o = (0.8, -0.1)$	$(0.8, 0.5, -0.1)$	Outlet center coordinates
$e_o = 0.2$	>	Outlet diameter
$l_o = 0.1$	>	Outlet leads length
$N_n = 20000$	1M	Nb. mesh nodes
$N_{el} = 40000$	5M	Nb. mesh elements

Table 2: Numerical parameters for the single inlet/single outlet duct flow problem.

584 7.1.1. Two-dimensional case

585 The 2-D analysis is carried out in the symmetry plane slicing the 3-D domain through the
586 inlet/outlet centerlines; see Tab. 2 for provision of all relevant numerical parameters. The design
587 domain reduces to a square cavity of unit length, with one inlet on the left side and one outlet
588 at the bottom, discretized here with 40000 elements¹. The fluid is set to occupy 25% of the
589 cavity, which is the same volume² as the quarter annulus fitting exactly to the inlet and outlet.
590 The initial design shown in Fig. 5 consists of spherical occlusions arranged for the initial fluid
591 to match the target within the desired tolerance, meaning that the volume constraint is satisfied
592 right away. The optimization run is illustrated in Fig. 5 by the anisotropic adapted mesh and
593 zero level set of a selected sample. It can be seen that the method easily handles the topological
594 changes occurring over the course of optimization, e.g., merging or cancellation of holes. Also,
595 all adapted meshes exhibit the expected orientation and deformation of the mesh elements, whose
596 longest edges are parallel to the solid boundaries. The elements are naturally and automatically
597 coarsened in smooth regions where the filtered level set is constant, all the more so in the solid
598 domain where only a few ten elements are used. Meanwhile, they are extremely refined near the
599 interface for the velocity to smoothly transition to zero across the boundary layer. In return, the
600 interfaces are sharply captured, not only at optimality but during all stages of the optimization.
601 This represents a major improvement in accuracy of the geometric representation with respect to
602 the available recent literature, as even traditional (isotropic) adaptive mesh refinement techniques
603 have been shown to yield quality issues (staircase effects) in smoothly curved regions. Ultimately,
604 we obtain an almost straight channel nearly identical to that documented in [55], which is because
605 most energy is dissipated by shear at low Reynolds numbers, so an optimal flow pipe is preferably
606 as short and wide as possible.

607 7.1.2. Three-dimensional case

608 The problem is now tackled in three dimensions, with the entire cubic cavity discretized into
609 5000000 (5M) mesh elements. The fluid is now set to occupy 3.9% of the cavity, which is the
610 same as the quarter torus fitting exactly to the inlet and outlet. All other numerical parameters

¹The mesh information is documented in terms of the equivalent number of elements $N_{el} = 2N_n$ if $d = 2$, as the meshes then have an element-to-node ratio close to 2, as should be for dens meshes made up of triangular elements.

²Actually cross-sectional area or volume per unit length in the third dimension, in which case we choose to keep the volume terminology for the sake of generality.

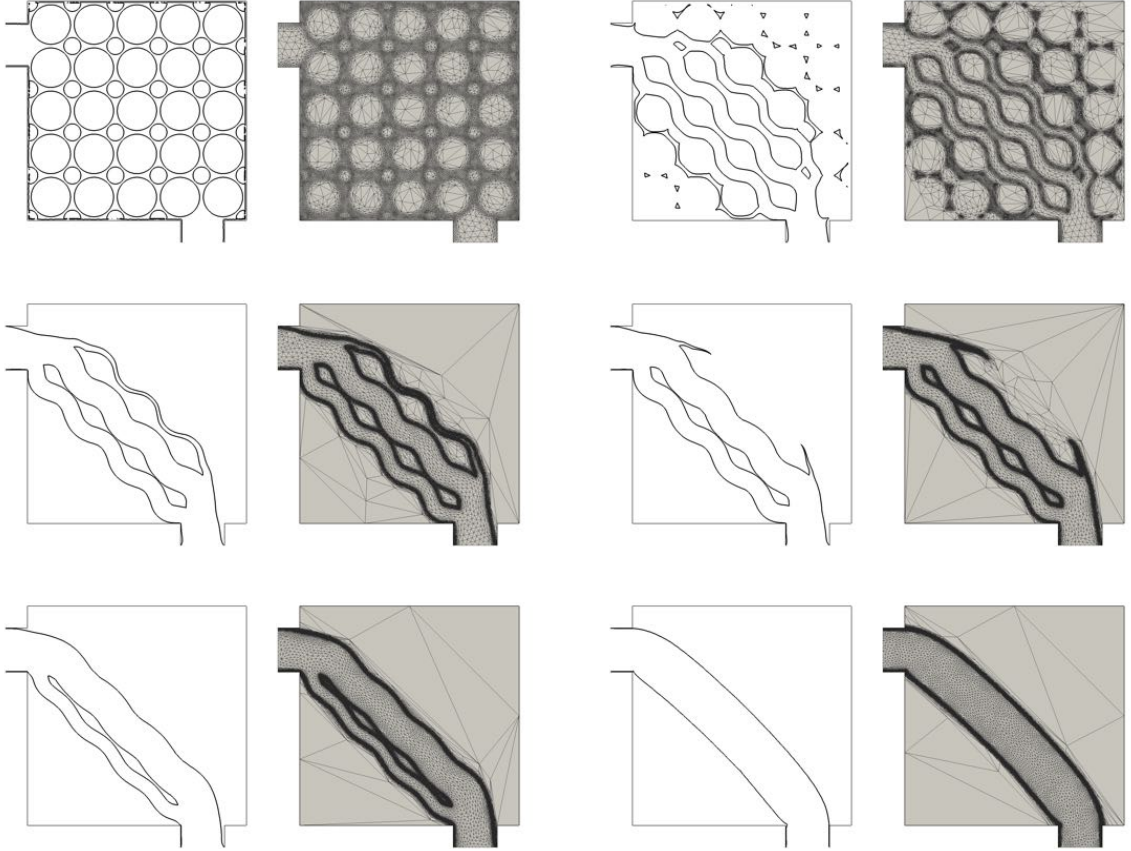


Figure 5: Two-dimensional optimization of the single inlet/single duct flow presented in Fig. 4(a). The zero iso-value of the level set function and associated anisotropic adapted meshes are sampled over the course of optimization using the parameters given in Tab. 2. The associated volume of fluid of all samples matches the target (25%) within the desired tolerance.

611 are documented in Tab. 2. The initial design in Fig. 6 consists of spherical occlusions occupying
612 about 24% of the cavity. The volume of fluid therefore initially fills about 76% of the cavity, in
613 violation of the volume constraint. This is because many more smaller inclusions are needed to
614 recover the proper volume, which in turn would either dramatically increase the surface of the
615 interfaces that needs be captured (and thus the number of mesh elements needed to maintain the
616 numerical accuracy), or risk clogging the fluid path due to insufficient mesh refinement. As shown
617 in the convergence history presented in Fig. 7, there is thus an initial transient during which the
618 cost function, albeit low, has little physical meaning, as the constraint value is decreased up to the
619 point where it reaches the target within the desired tolerance. Once the constraint is satisfied, the
620 cost function adjusts until a feasible minimum is found, that corresponds to the almost straight
621 pipe shown in Fig. 6, that closely resembles that in [58], but with vastly superior accuracy. To
622 give a taste, the element size is about 6×10^{-4} at the interface and 0.01 in the fluid domain,
623 with up to 40-50 elements distributed across a pipe diameter. In comparison, the problem in the
624 aforementioned reference is tackled with a uniform grid made up of 162000 tetrahedral elements,
625 hence an element size of about 0.0375, which is insufficient to claim accuracy of the numerical
626 solutions since only 5-6 grid points can be distributed across a pipe diameter. Again, the method
627 handles well the various topological changes occurring over the course of optimization, and all
628 adapted meshes exhibit extremely stretched elements regardless of the interface complexity, that
629 allow sharply representing the fluid and solid domains and accurately computing the fluid solutions
630 during all stages of optimization.

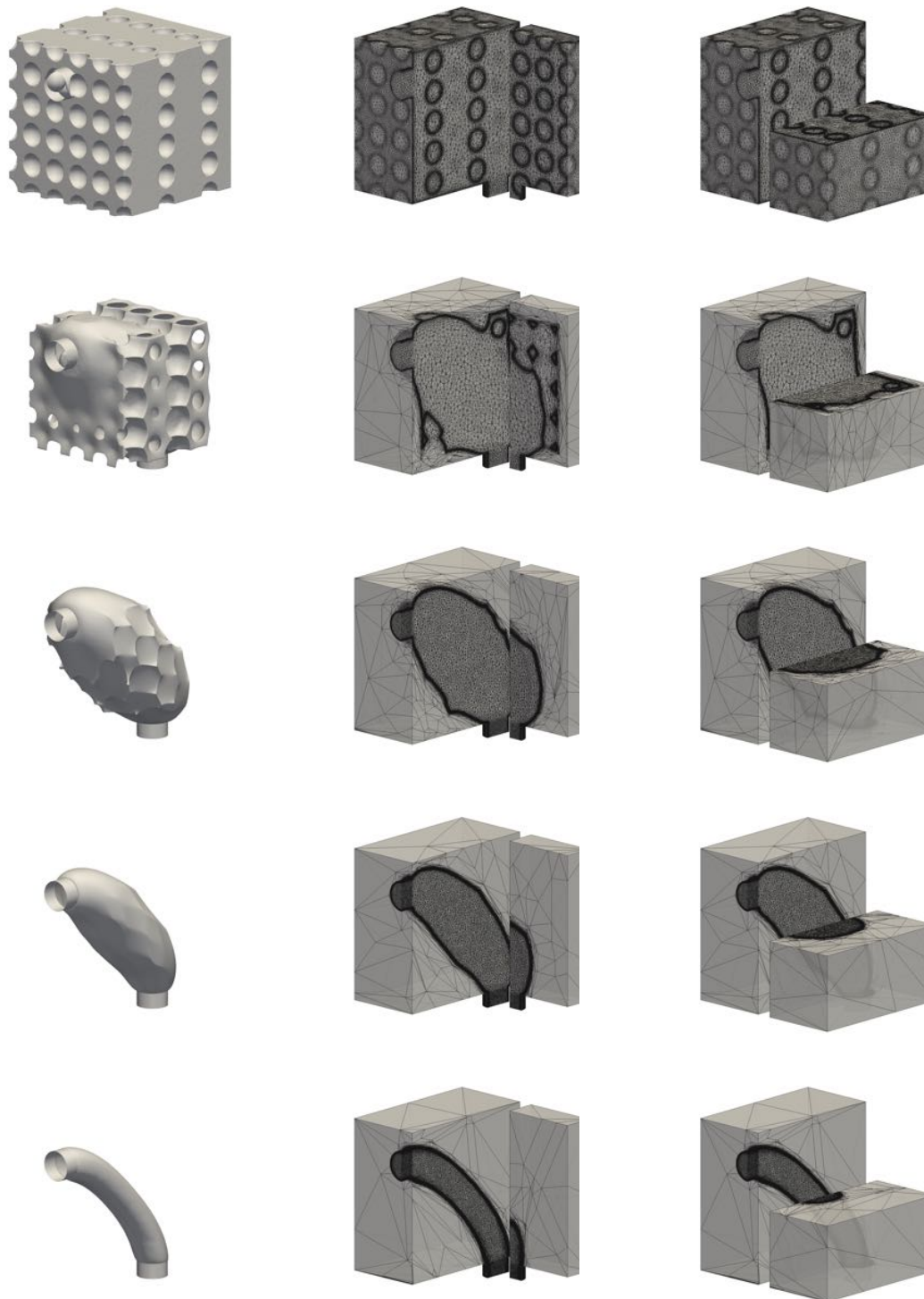


Figure 6: Optimization of the single inlet/single outlet duct flow presented in Fig. 4(a). The zero iso-value of the level set function and associated anisotropic adapted meshes are sampled at intermediate iterations 1, 402, 548, 700 and 900 (from top to bottom) using the parameters given in Tab. 2. The associated volumes of fluid are 77.0%, 40.8%, 27.5%, 13.7% and 3.94%, respectively.

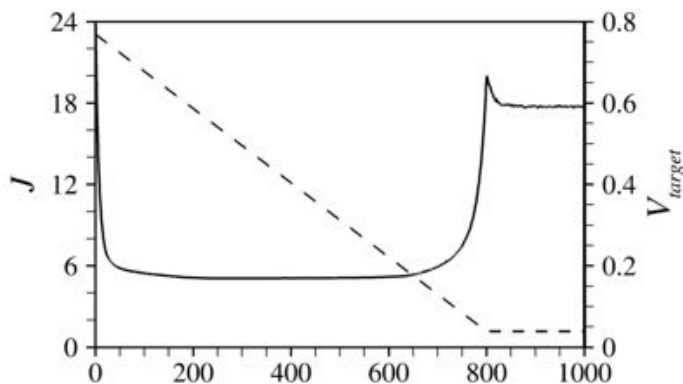


Figure 7: Convergence history for the 3-D bend pipe problem with $5M$ elements. All cost function values made non dimensional using the inlet diameter and maximum inlet velocity (equivalently, using $\rho v_i^3 e_i^2$ as reference cost functional value). The dashed line shows the decrease in the target volume.

631 7.1.3. Discussion

632 The parallel remeshing strategy for the 3-D case is illustrated in Fig. 8 showing for the same
633 iterations already sampled in Fig. 6 the 64 submeshes generated by the graph/mesh partitioning/
634 repartitioning algorithm, each shown by a different color and handled sequentially by a differ-
635 ent processor. The various submeshes are initially uniformly distributed in the whole domain, due
636 to the presence of the multiple solid occlusions. Nonetheless, they quickly reorganize to cover the
637 vicinity of the interface, where the mesh refinement is maximum, with only a handful of submeshes
638 needed to handle the coarse solid domains, meaning that the load is well balanced between the
639 processors.

640 It is worth noticing that the large number of nodes used here is mostly useful during the early
641 stage of optimization. This is because the surface of the interfaces (perimeter in two dimensions)
642 that needs be captured is initially dramatically large to the many solid inclusions, then decreases
643 substantially after the first dozens of iterations, as has been found computing the surface area

$$S_\varphi = \int_{\Omega} \delta_\epsilon(\varphi) dv, \quad (53)$$

644 where δ_ϵ is the Dirac function

$$\delta_\epsilon(\varphi) = \begin{cases} \frac{1}{2\epsilon} \left(1 + \cos\left(\pi \frac{\varphi}{\epsilon}\right) \right) & \text{if } |\varphi| \leq \epsilon, \\ 0 & \text{if } |\varphi| > \epsilon, \end{cases} \quad (54)$$

645 smoothed with the same regularization parameter ϵ as the Heaviside function (48). Also, the
646 anisotropic mesh adaptation algorithm refines the mesh in hierarchical importance of the level set
647 gradient. If new geometrical features associated with high gradients appear in the solution, the
648 mesh is automatically coarsened in regions with lower gradient and refined near the newly emerging
649 features. If the number of nodes is large, as has been the case so far, the decrease in the interface
650 surface area allows resolving finer, more complex patterns without degrading the accuracy in other
651 parts of the design domain, because the coarsened regions are actually over-resolved. This shows
652 through the progressive mesh refinement in the fluid domain in Figs. 5 and 6 as more and more
653 elements become available to improve the mesh in other regions of the domain.

654 7.2. Single inlet/multiple outlets duct flow

655 This section is devoted to a series of more complex duct flow problems with one inlet and
656 multiple outlets. All cases are tackled with boundary normal, parabolic velocity profiles prescribed
657 at the inlets and the outlets, with outlet centerline velocities adjusted for the the total amount
658 of mass flow exiting through the outlets to match exactly that entering through the inlet. The
659 objective for doing so is twofold: first, it forces the inlet to connect to all the outlets, and thereby

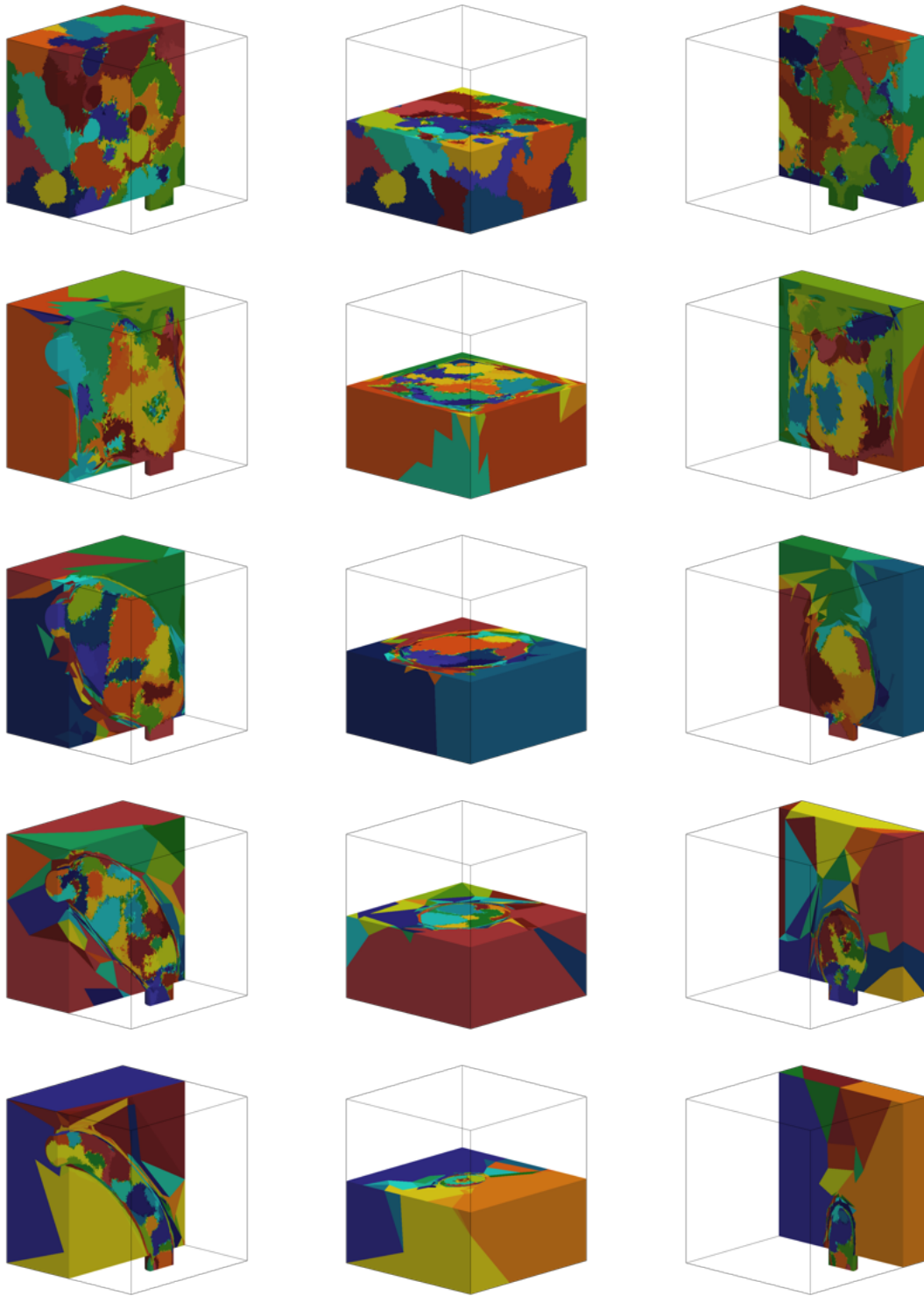


Figure 8: Illustration of mesh partitioning for the three dimensional, single inlet/single outlet duct flow. The colors in these plots represent the 64 subdomains generated by the graph/mesh partitioning/repartitioning for the same iterations sampled in Fig. 6.

660 emphasizes the ability of the numerical framework to engineer complex designs, as a more natural
 661 zero pressure/viscous stress condition may allow the flow to exit via a single outlet to save the
 662 cost of pipe splitting. Second, it reduces the computational cost, as the possibility of having

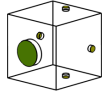
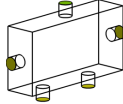
		
$\Omega = [0; 1] \times [0; 1] \times [0; 1]$	$[0; 2] \times [0; 1] \times [0; 0.5]$	Design domain
$d = 3$	3	Problem dimensionality
$V_{target} = 0.15$	0.10	Target volume of fluid
$V_{\varphi,0} = 0.76$	0.37	Initial volume of fluid
$Re = 2$	2	Reynolds number
$u_i = 0.2$	0.2	Inlet centerline velocity
$e_i = 0.4$	0.2	Inlet diameter
$l_i = 0.1$	0.2	Inlet leads length
$\mathbf{x}_i = (-0.1, 0.5, 0.5)$	$(1, 1.2, 0.25)$	Inlet center coordinates
$u_{o1} = 0.8$	0.6667	Outlet 1 centerline velocity
$u_{o2} = 0.8$	0.6667	Outlet 2 centerline velocity
$u_{o3} = 0.8$	0.3333	Outlet 3 centerline velocity
$u_{o4} = 0.8$	0.3333	Outlet 4 centerline velocity
$e_o = 0.1$	0.2	Outlet diameter
$l_o = 0.05$	0.2	Outlet leads length
$\mathbf{x}_{o1} = (0.75, 0.5, -0.05)$	$(-0.2, 0.5, 0.25)$	Outlet 1 center coordinates
$\mathbf{x}_{o2} = (0.75, 0.5, 1.05)$	$(2.2, 0.5, 0.25)$	Outlet 2 center coordinates
$\mathbf{x}_{o3} = (0.75, -0.05, 0.5)$	$(0.5, -0.2, 0.25)$	Outlet 3 center coordinates
$\mathbf{x}_{o4} = (0.75, 1.05, 0.5)$	$(1.5, -0.2, 0.25)$	Outlet 4 center coordinates
$N_n = 1M$	\gg	Nb. mesh nodes
$N_{el} = 5M$	\gg	Nb. mesh elements

Table 3: Numerical parameters for the single inlet/multiple outlets duct flow problems.

different number of pipes connect to an outlet may give existence to multiple local minimizers, whose basin of attraction can slow down the convergence; see for instance Ref. [59] for an example of competition between the single and double-ended wrench minimizers to the 2-D double pipe problem. In practice, our literature review did not reveal any study tackling multiple outlets 3-D topology optimization problems with zero pressure/viscous stress or zero stress conditions, except for a few cases in Ref. [24] adding mass flow rate constraints to the Lagrangian to similarly force the flow to exit via all outlets.

For the first example whose setup is shown in Fig. 4(b), the design domain is a cubic cavity of unit length, that has one inlet on the left side and four identical outlets at the top/bottom and on the front/back sides, each having 1/4 of the fluid flow entering through the inlet. For the second example whose setup is presented in Fig. 4(c), the design domain is a cuboid cavity of unit height and aspect ratio 2:1:0.5, that has one inlet at the top, and four identical outlets: two on the left and right sides, each having 1/3 of the fluid flow entering through the inlet, and two at the bottom, each having 1/6 of the inflow. Both domains have two reflectional symmetries with respect to the two inlet/outlet planes, but are discretized in their entirety using 5M mesh elements. For the first case, the initial design in Fig. 9 is made up of spherical solid occlusions occupying about 24% of the cavity. For the second case, we take advantage of the fact that all inlet and outlets are in the same plane and initialize the design with cylindrical solid occlusions filling about 63% of the cavity. The volume of fluid in each case is thus initially about 76% (first case) and 37% (second case), and decreases over the course of optimization until it reaches the target within the desired tolerance. The latter is set low to 15% in the first case, and 10% in the second case, to avoid trivial solutions and promote the formation of separate fluid channels.

For both cases, the optimization goes through several complex stages all accurately represented on anisotropic adapted meshes, as evidenced by the selected samples shown in Figs. 9-10. Similarly to what could be observed in the bend case, all mesh elements are coarse and regular away from the interface but fine and elongated on either side of the interface, to allow accurately representing

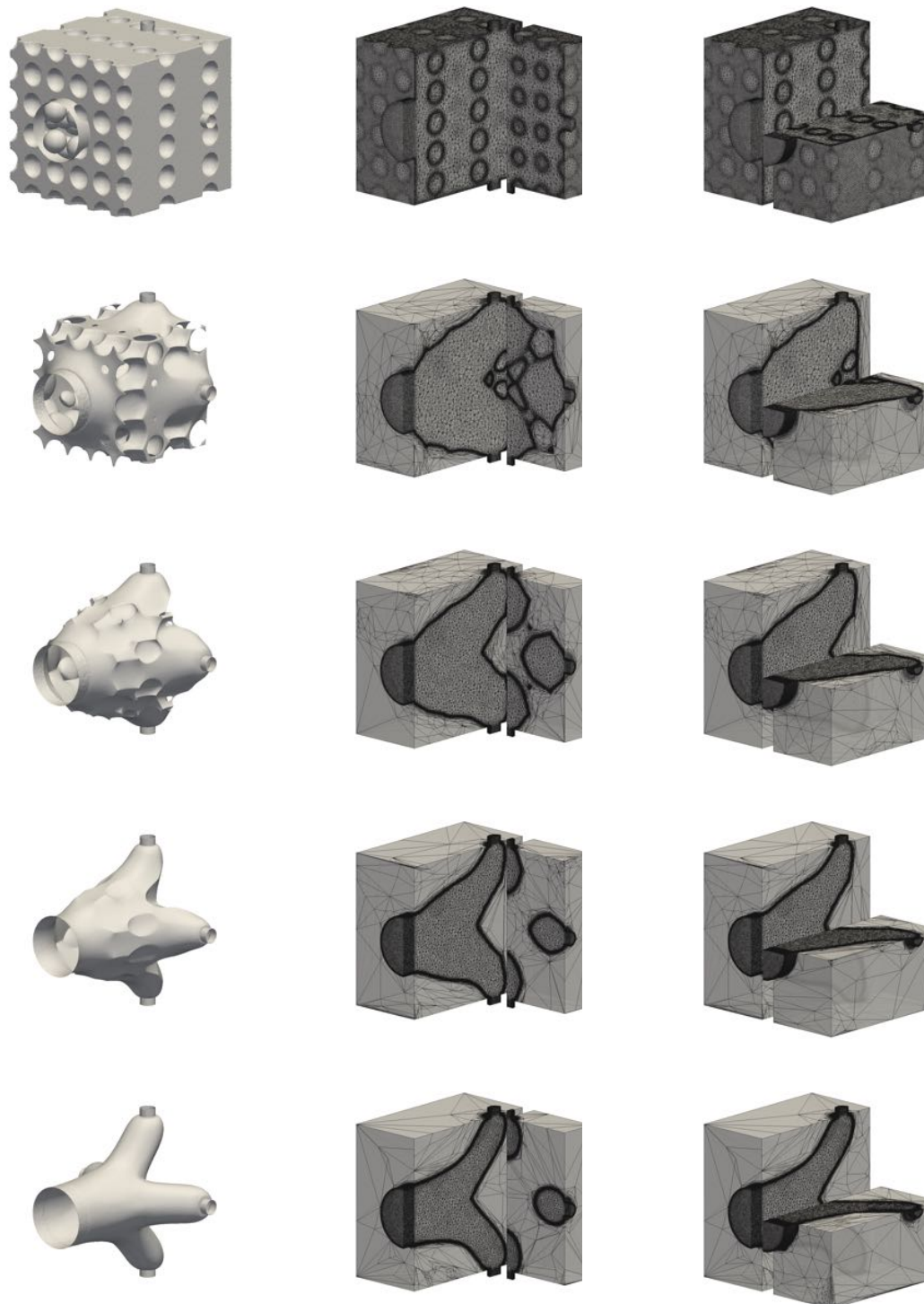


Figure 9: Optimization of the single inlet/multiple outlet duct flow presented in Fig. 4(b). The zero iso-value of the level set function and associated anisotropic adapted meshes are sampled at intermediate iterations 1, 505, 624, 750 and 1050 (from top to bottom) using the parameters given in Tab. 3. The associated volumes of fluid are 75.6%, 37.8%, 28.9%, 19.5% and 15.0%, respectively.

689 the boundary layers regardless of topology complexity, even in the leads. The optimal duct for the
 690 first case is a wide pipe splitting at mid length into four identical, thinner pipes, each connecting
 691 to an outlet. This layout stands as the better trade-off between transporting fluid the shortest

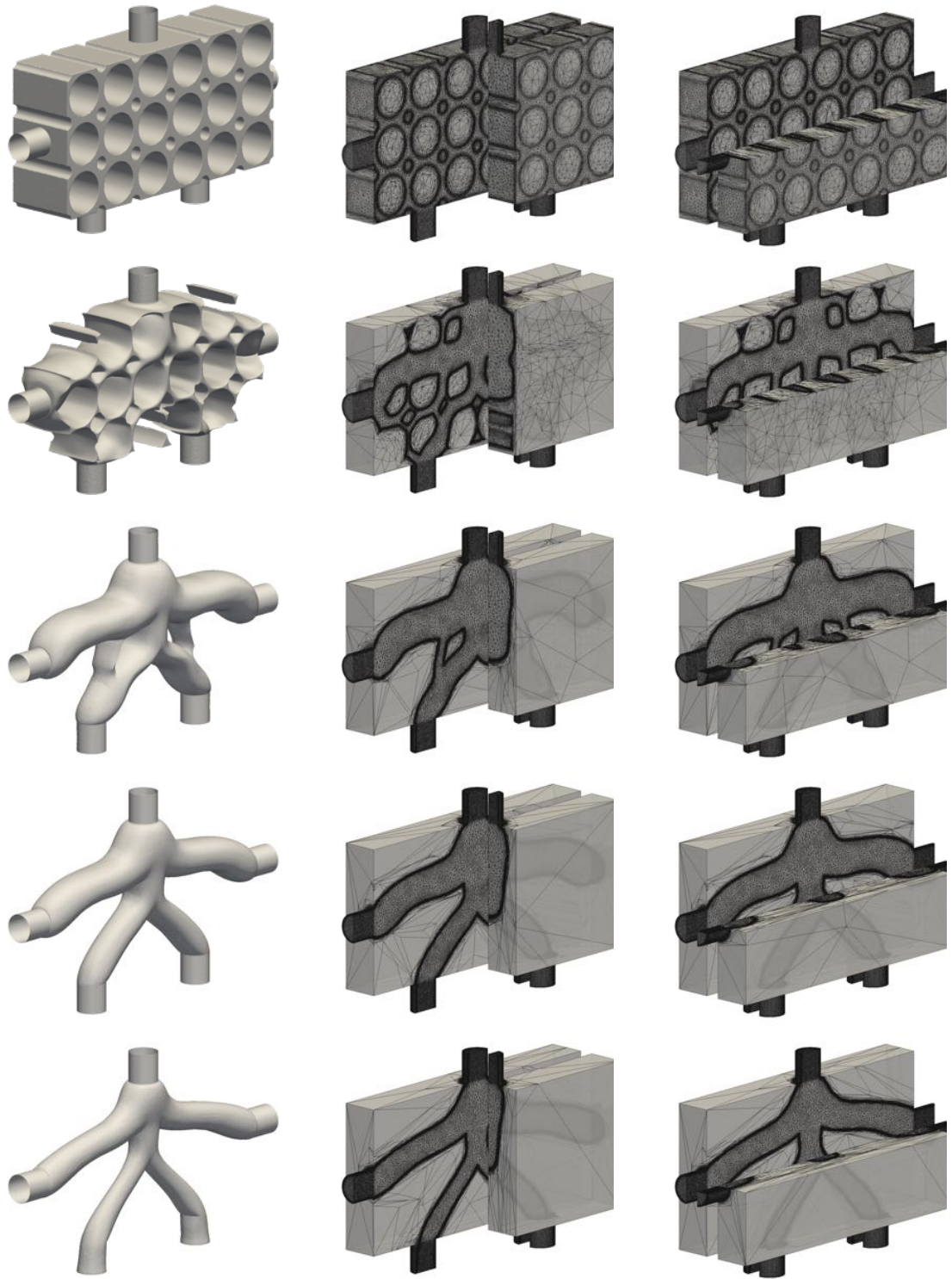
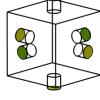


Figure 10: Optimization of the single inlet/multiple outlet duct flow presented in Fig. 4(c). The zero iso-value of the level set function and associated anisotropic adapted meshes are sampled at intermediate iterations 1, 60, 110, 180 and 300 (from top to bottom) using the parameters given in Tab. 3. The associated volumes of fluid are 36.2%, 30.0%, 24.8%, 17.5% and 9.9%, respectively.

692 way, and transporting it in the widest possible pipe, and is consistent with the results documented
 693 in [58], although the optimal shapes therein exhibit quality issues (staircase effects) and anisotropic



$\Omega = [0; 1] \times [0; 1] \times [0; 1]$	Design domain
$d = 3$	Problem dimensionality
$V_{target} = 0.05$	Target volume of fluid
$V_{\varphi,0} = 0.73$	Initial volume of fluid
$Re = 2$	Reynolds number
$u_{i1} = 0.2$	Inlet 1 centerline velocity
$u_{i2} = 0.2$	Inlet 2 centerline velocity
$e_i = 0.2$	Inlet diameter
$l_i = 0.2$	Inlet leads length
$\mathbf{x}_{i1} = (-0.2, 0.5, 0.5)$	Inlet 1 center coordinates
$\mathbf{x}_{i2} = (1.2, 0.5, 0.5)$	Inlet 2 center coordinates
$u_{o1} = 0.1$	Outlet 1 centerline velocity
$u_{o2} = 0.1$	Outlet 2 centerline velocity
$u_{o3} = 0.1$	Outlet 3 centerline velocity
$u_{o4} = 0.1$	Outlet 4 centerline velocity
$e_o = 0.2$	Outlet diameter
$l_o = 0.2$	Outlet leads length
$\mathbf{x}_{o1} = (0.5, 0.5, -0.2)$	Outlet 1 center coordinates
$\mathbf{x}_{o2} = (0.5, 0.5, 1.2)$	Outlet 2 center coordinates
$\mathbf{x}_{o3} = (0.5, -0.2, 0.5)$	Outlet 3 center coordinates
$\mathbf{x}_{o4} = (0.5, 1.2, 0.5)$	Outlet 4 center coordinates
$N_n = 1M$	Nb. mesh nodes
$N_{el} = 5M$	Nb. mesh elements

Table 4: Numerical parameters for the multiple inlet/multiple outlets duct flow problem.

694 mesh adaptation represents a tremendous improvement in this regards. For the second case, the
 695 optimal duct comprises a wide pipe immediately splitting into three pipes: two symmetrical pipes
 696 connecting to the lateral outlets, and a central pipe quickly splitting into two symmetrical pipes
 697 connecting to the bottom outlets. All pipes are reasonably straight, and the lateral pipes are
 698 wider than their bottom counterparts. This is because most of the flow exits through the left/right
 699 outlets and optimal pipes at low Reynolds numbers are preferably short and wide, as splitting
 700 the fluid stream further away from the inlet would require complex bending patterns to connect
 701 the outlets, which in turn would increase the transport distance and would thus be detrimental in
 702 terms of cost function.

703 7.3. Multiple inlets/multiple outlets duct flow

704 In this example, the focus is on a duct flow problem with multiple inlet and outlets, whose
 705 setup is shown in Fig. 4(d). The design domain has two identical inlets on the left and right sides,
 706 and four identical outlets on all other sides, hence three reflectional symmetries with respect to the
 707 outlet and the two inlet/outlet planes. Parabolic inflow/outflow conditions are formulated in the
 708 same way as above, with outflow velocities adjusted for each outlet to have 1/4 of the fluid flow
 709 entering through the inlets. The entire cavity is discretized using 5M mesh elements. The initial
 710 design in Fig. 11 consists of spherical occlusions occupying about 27% of the cavity, after which
 711 the volume of fluid is progressively decreased, starting at 73% of the cavity, down until it reaches
 712 a 5% target within the desired tolerance. The optimization documented in Fig. 11 occurs within
 713 two different steps, all involving crisp interfaces represented on extremely stretched mesh elements:
 714 first, the design looks to be converging to a series of 8 straight pipes connecting each inlet to all
 715 four outlets, a duct arrangement that has the same reflectional symmetries as the design domain.
 716 This agrees well with the optimal documented in [24] while resembling conceptually the 2-D results

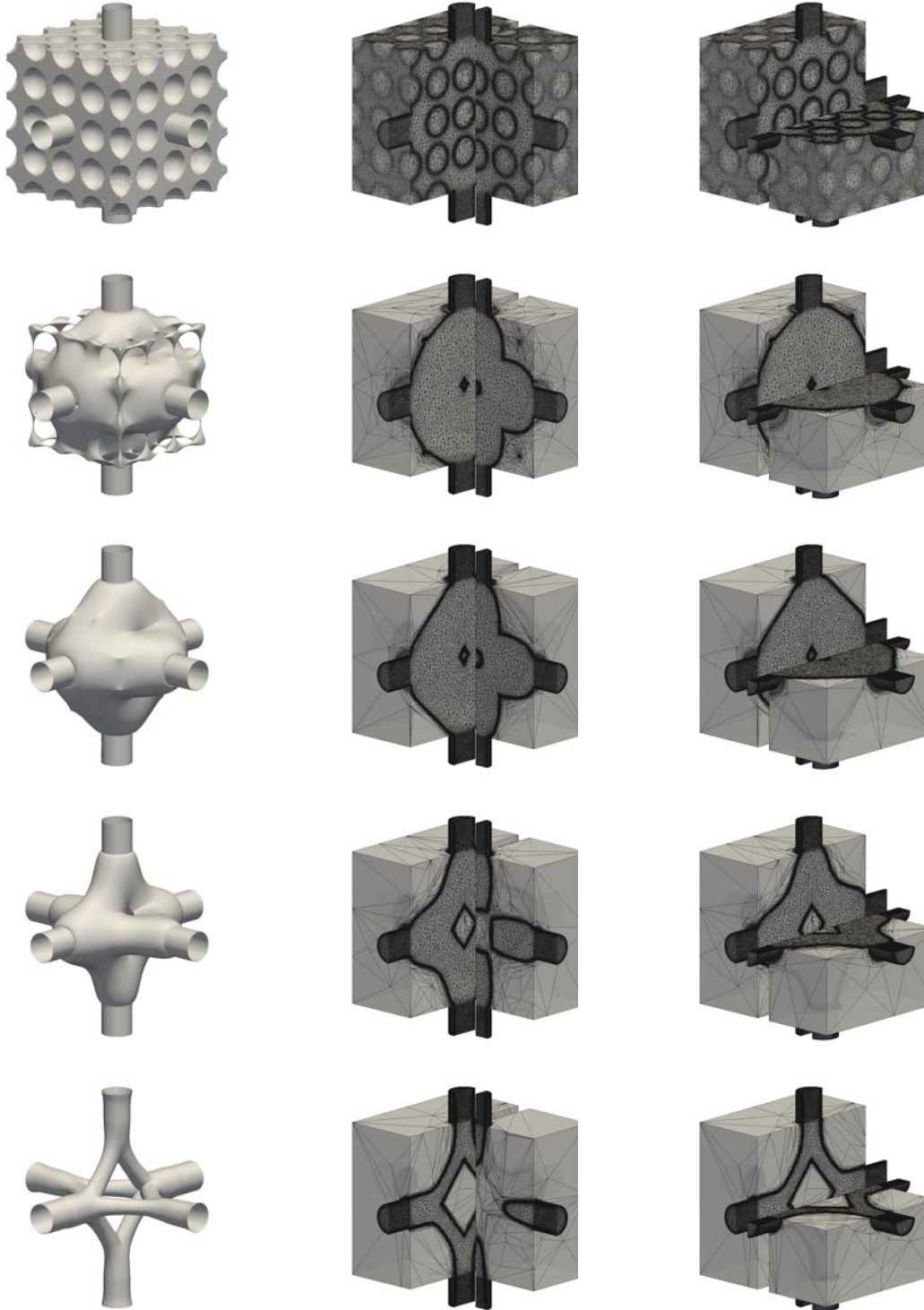


Figure 11: Optimization of the multiple inlet/multiple outlet duct flow presented in Fig. 4(d). The zero iso-value of the level set function and associated anisotropic adapted meshes are sampled at intermediate iterations 1, 201, 283, 400 and 550 (from top to bottom) using the parameters given in Tab. 2. The associated volumes of fluid are 72.8%, 45.9%, 34.8%, 18.9% and 4.9%, respectively.

717 from [60]. Nonetheless, this turns to be only a local minimizer, as we show in Fig. 12 that the
 718 optimization carries on under constant volume of fluid, and progressively wipes off 4 out of the 8
 719 pipes, for the optimal duct to ultimately consist of 4 straight but wider pipes connecting each inlet

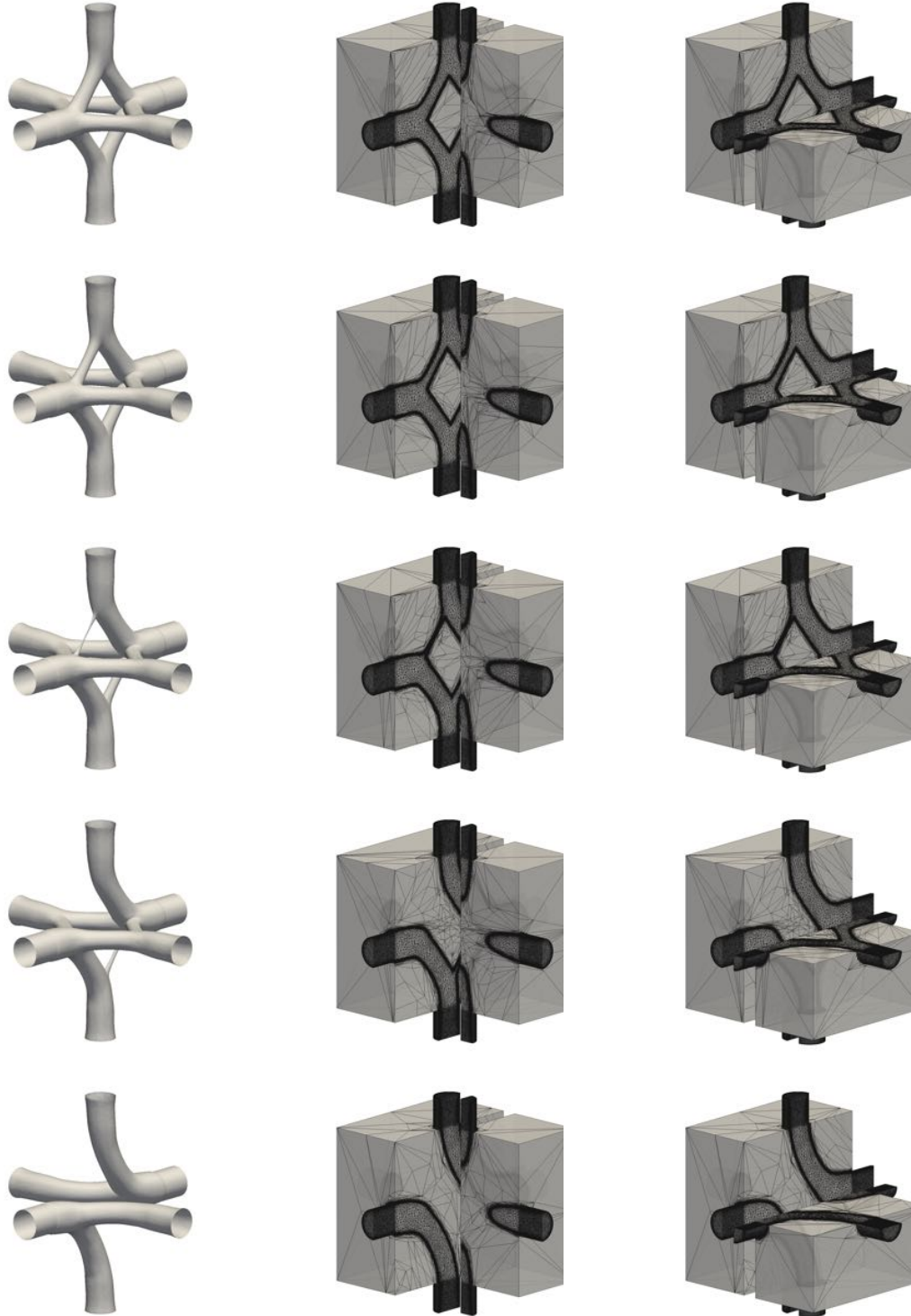


Figure 12: Optimization of the multiple inlet/multiple outlet duct flow presented in Fig. 4(d). The zero iso-value of the level set function and associated anisotropic adapted meshes are sampled at intermediate iterations 700, 800, 830, 850 and 1000 (from top to bottom) using the parameters given in Tab. 2. The associated volume of fluid of all samples matches the target (5%) within the desired tolerance.

720 to two outlets two-by-two perpendicular to one another (the whole arrangement being symmetric
 721 with respect to one of the bisector planes). This stresses the importance of performing full-scale
 722 optimization, as the cost function of the asymmetric design is noticeably lower (by almost 30%),

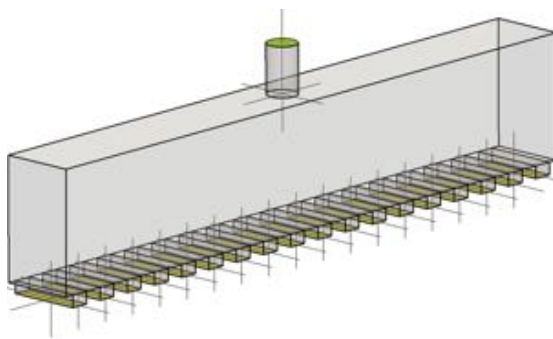
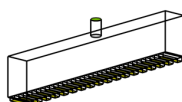


Figure 13: Problem set-up for the simplified plate fin heat exchangers (PFHE) case study.



$\Omega = [0; 2.7] \times [0; 0.7] \times [0; 0.3]$	Design domain
$d = 3$	Problem dimensionality
$V_{target} = 0.10$	Target volume of fluid
$V_{\varphi,0} = 0.20$	Initial volume of fluid
$Re = 4$	Reynolds number
$u_{i1} = 0.4$	Inlet centerline velocity
$e_i = 0.12$	Inlet diameter
$l_i = 0.2$	Inlet leads length
$\mathbf{x}_{i1} = (1.35, 0.7, 0.15)$	Inlet center coordinates
$u_{ok} = 0.00591$	Outlet k centerline velocity
$e_o = (0.075; 0.3)$	Outlet width
$l_o = 0.05$	Outlet leads length
$\mathbf{x}_{ok} = (0.075 + 0.15k, 0.35, 0.15)$	Outlet k center coordinates
$N_n = 1M$	Nb. mesh nodes
$N_{el} = 5M$	Nb. mesh elements

Table 5: Numerical parameters for the plate fin heat exchanger distributor problem.

723 and relying on simple problem symmetries to reduce the cost can thus yield suboptimal results;
 724 see [24] where only 1/8 of the cavity is simulated.

725 7.4. Application to a simplified industrial case study

726 In this section, we consider application of the numerical framework to an engineering problem
 727 of practical interest. The focus is on plate fin heat exchangers (PFHE), a specific type of compact,
 728 lightweight heat exchangers widely used in air conditioning and petro-chemical industries (also in
 729 railway engines and motor cars). Most PFHEs consist of layers of plates with the space between
 730 two adjacent plates forming the channel in which the hot fluid flows through. Fins are placed
 731 between the flat plates to both hold the plates together, and form a secondary surface for heat
 732 transfer. We leave aside here the question of heat transfer, and assess the ability of our numerical
 733 framework to minimize the total pressure drop in a design domain representative of the refrigerant
 734 distributor of a heat exchanger comprising multiple outlet orifices. Only the distributor section is
 735 modeled numerically. The design domain is the cuboid of height 0.5 and aspect ratio 2.7:0.5:0.3
 736 shown in Fig. 13, that has one cylindrical inlet at the top, and 18 rectangular outlets at the
 737 bottom, each of which is the entry section into one of the 18 plates and has 1/18 of the fluid flow
 738 entering through the inlet. In return, there is no need to model either the fins or the hot fluid,
 739 since the latter flows orthogonally between the plates. Parabolic velocity profiles are prescribed

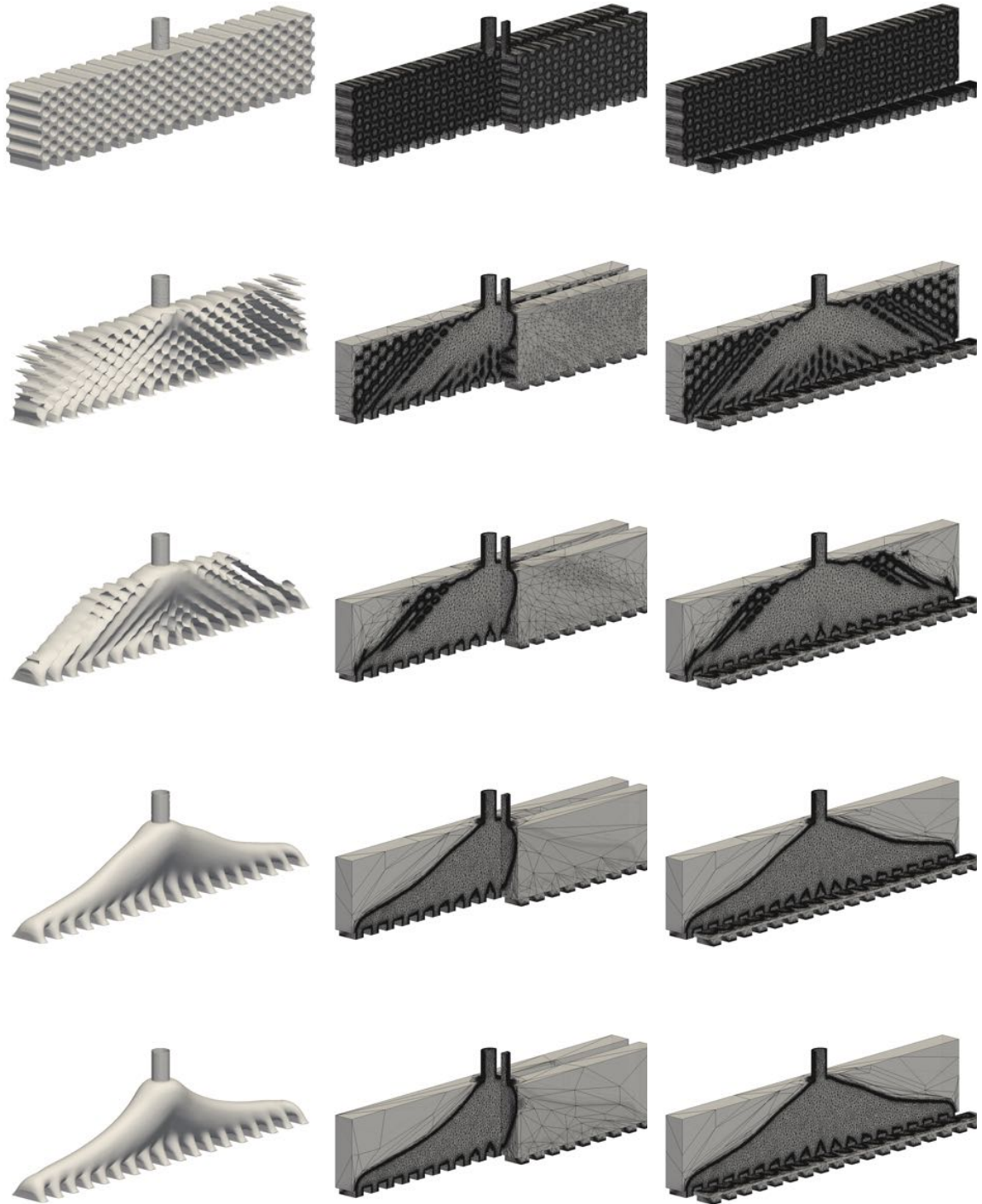


Figure 14: Optimization of the plate fin heat exchanger distributor presented in Fig. 13. The zero iso-value of the level set function and associated anisotropic adapted meshes are sampled at intermediate iterations 1, 80, 120, 200 and 400 (from top to bottom) using the parameters given in Tab. 2. The associated volumes of fluid are 50%, 39.2%, 34.6%, 25.4% and 24.9%, respectively.

740 at the inlets and the outlets, with the outflow distribution in each outlet section defined as the
741 Cartesian product of parabolic variations along the two lines of symmetry, which stands as a
742 first approximation of the series-based, theoretical velocity profile of flow through a channel of

743 rectangular cross-section [61]. The entire domain is meshed with 5M elements, with the remaining
 744 parameters given in Table 5. Due to the cavity low aspect ratio in the third dimension, the initial
 745 design is initialized with cylindrical solid occlusions occupying about 65% of the cavity, and the
 746 volume of fluid is progressively decreased until it reaches the target within the desired tolerance.
 747 This yields the optimal duct shown in Fig. 14, that delivers most of the fluid in the center area
 748 of the cavity before distributing it to the plates via the comb-like structure at the bottom, and
 749 showcases the potential of the method for smooth, lightweight heat exchanger solutions.

750 7.5. Discussion

751 This last section is dedicated to discussing the numerical cost of the presented approach. Fig-
 752 ure 15(a) presents detailed timing results obtained by averaging (and normalizing to achieve unit
 753 average time per iteration) dedicated update steps performed on 64 cores (150 steps in 2-D, 50
 754 steps in 3-D). In 2-D, the cost of an iteration is dominated by that of computing the state solution
 755 (about 10 Navier–Stokes iterations representing 50% of the total cost, which can be scaled down
 756 substantially in the context of steady-state problems using an iterative Newton-like method), and
 757 otherwise by that of adapting the mesh (about 30% of the total cost). Using the same number of
 758 processors, the cost of a 3-D iteration is larger than its 2-D counterpart by roughly three orders
 759 of magnitude, the cost of which is essentially that of the two passes of mesh adaptation (about a
 760 cumulative 75% of the total cost, although the cost of the first pass is twice as large as that of the
 761 second pass, since (i) the volume constraint is not applied at each design step, only when the dif-
 762 ference between the actual and target volumes exceeds the 5% tolerance, and (ii) less elements and
 763 nodes need to be moved and migrated across processors. Meanwhile, the cost of both geometrically
 764 reinitializing the signed distance function set and of optimizing the volume constraint offset is
 765 very affordable, as it represents less than 4% in total, with 4-5 dichotomy iterations needed to reach
 766 the desired accuracy. The timing results reported in Figs. 15(c-f) show that the same conclusions
 767 carry over when applying the method to the other multiple inlet/outlet duct flows tackled herein,
 768 including the more complicated, practical PFHE case in Sec. 7.4. The only difference is in the
 769 cost of the volume constraint step, as the frequency at which consecutive corrections are applied
 770 indirectly depends on the number of design steps taken to reach the target volume. This gives hope
 771 that the observed trends may carry over to any other problem of same dimensionality, tackled with
 772 comparable parameters.

773 The associated absolute run times per iteration shown in Fig 16 are seen to be very consistent,
 774 in the sense that they change little from case to case. In return, the total run times reported in
 775 Tab. 6 are entirely driven by the number of design steps needed to converge. Here, the reported cost
 776 is essentially that of recovering the proper volume of fluid, as fulfilling the proper volume constraint
 777 from the outset requires a larger number of smaller solid inclusions, which would either dramatically
 778 increase the surface of the interfaces that needs be captured (and thus the number of mesh elements
 779 needed to maintain the numerical accuracy), or risk clogging the fluid path due to insufficient mesh
 780 refinement. Because the single inlet/multiple outlet duct flow presented in Fig. 4(d) and the plate
 781 fin heat exchanger distributor presented in Fig. 13 all rely on cylindrical, not spherical inclusions,
 782 their run time benefits from a lower initial volume of fluid (in the range between 35 and 50%, while
 783 all other case start above 75%), which speeds up the process of meeting the desired target volume.
 784 Once this has been done, all cases converge within about 200 iterations (about 60h of resolution
 785 time, a similar run time being achieved when initializing the single inlet/single outlet test case
 786 with a quarter torus fitting exactly to the inlet and outlet). The only exception is the multiple
 787 inlet/multiple outlet duct flow presented in Fig. 4(d), as an extra 300 iterations are needed to
 788 bypass the basin of attraction of the symmetric local minimizer and reach the asymmetric global
 789 minimizer. The reported run times, while large in a vacuum, are actually much lower than those
 790 that required to converge on a fixed uniform grid with similar mesh refinement. To give a taste,
 791 discretizing the single inlet/single outlet case with a uniform element size of 5×10^{-3} would require
 792 about 70M elements, even though the interface value achieved here is one order of magnitude
 793 smaller. It is also worth emphasizing in this regards that we did not seek to optimize efficiency,
 794 neither by adjusting the initial design (we actually used numerous inclusions on purpose to showcase
 795 the ability of the method to support complex topological changes), nor by fine tuning the descent
 796 factor (the only requirement being that the displacement achieved at each step must be below the
 797 cut-off thickness of the level set for the evolved interface to remain accurately tracked).

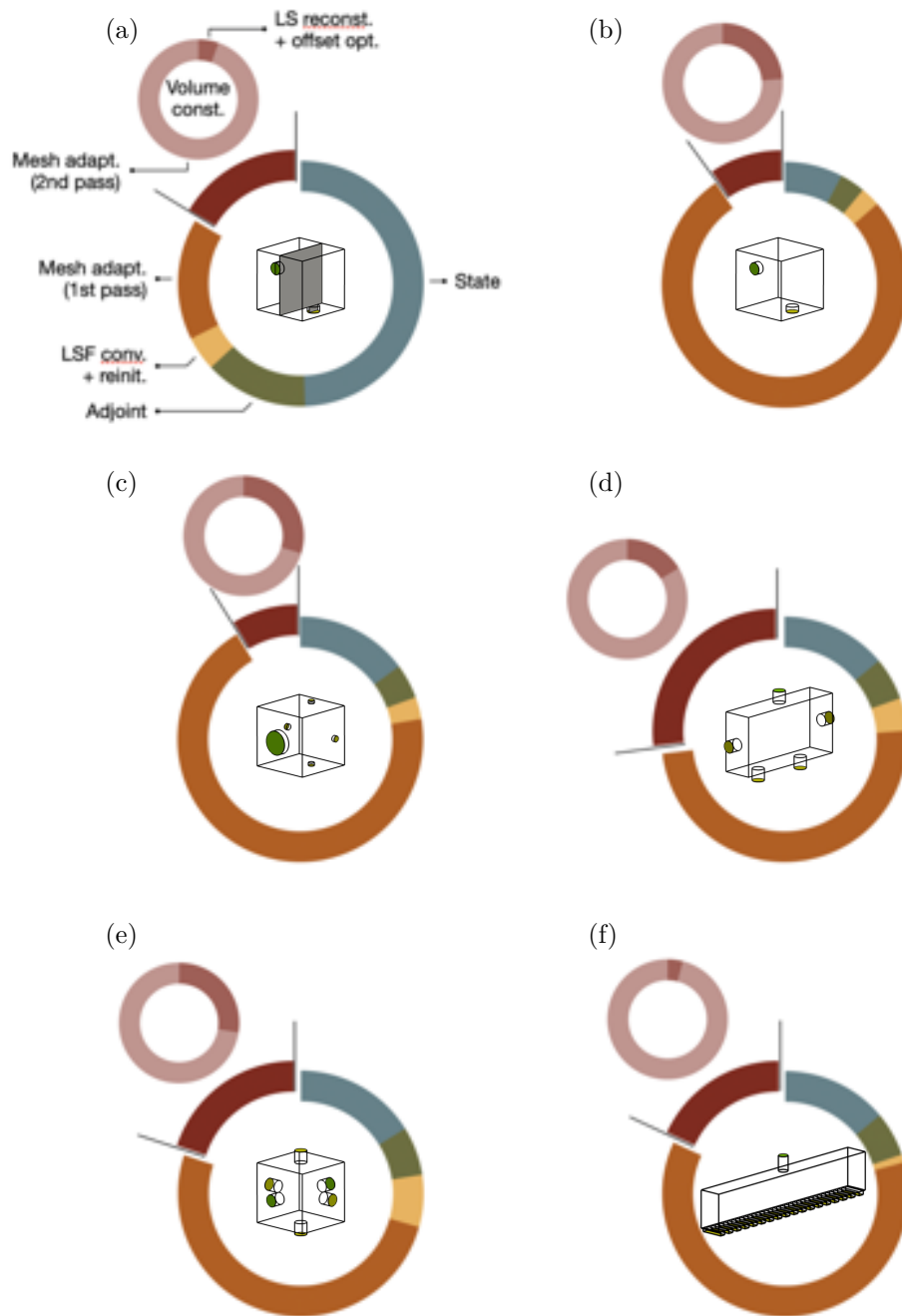


Figure 15: (a) Computational cost of the implemented algorithm, as obtained averaging 150 update steps of the 2-D single inlet/single duct flow presented in Fig. 4(a). (b) Same as (a) for 50 update steps of the 3-D single inlet/single duct flow presented in Fig. 4(a). (c-f) Same as (b) for the (c) single inlet/multiple outlet duct flow presented in Fig. 4(b), (d) single inlet/multiple outlet duct flow presented in Fig. 4(c), (e) multiple inlet/multiple outlet duct flow presented in Fig. 4(d), and (f) plate fin heat exchanger distributor presented in Fig. 13. All simulation parameters are those provided in Tabs. 15. The LS and LSF labels stand for level set (LS) and filtered level set (LSF), respectively.

798 8. Conclusion

799 The present study performs topology optimization of large-scale, three-dimensional Navier–
 800 Stokes flows using anisotropic meshes adapted under the constraint of a fixed number of nodes.
 801 The proposed approach combines a level set method to represent the boundary of the fluid domain
 802 by the zero iso-value of a signed distance function, and stabilized formulations of the state, adjoint,
 803 and level set transport equations cast in the Variational Multiscale (VMS) framework. The method

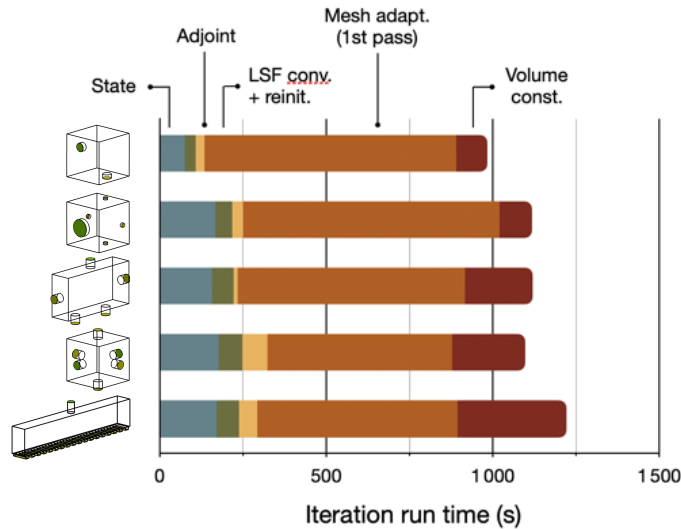


Figure 16: Average run time per iteration for the various cases documented in Fig. 15

Case 1	Case 2	Case 3	Case 4	Case 5	
64	>>	>>	>>	>>	Nb. cores
290h	340h	150h	330h	170h	Run time
1000	1000	500	1000	400	Nb. design steps
800	800	200	500	250	Nb. steps to target volume

Table 6: Run times for the various cases documented in Fig. 15

804 has been shown to allow for drastic topology changes during the optimization process. Nonetheless,
 805 the main advantage over existing methods is the ability to capture all interfaces to a very high
 806 degree of accuracy using adapted meshes whose anisotropy matches that of the numerical solutions.
 807 Also, the retained approach considerably decreases the cost of improving the numerical precision,
 808 as the number of nodes needs be increased only in the anisotropy direction, hence only 2 times as
 809 many nodes are required to improve the resolution by a factor of 2, while 8 times as many nodes
 810 are required in classical isotropic 3-D calculations. This gives hope that the method can ease the
 811 transition to manufacturable CAD models that closely resemble the optimal topology.

812 The method has been tested on several large-scale examples of power dissipation minimization
 813 involving several dozen million state degrees of freedom. The obtained optimal designs agree well
 814 with the existing literature, which assesses the relevance of the present implementation for designing
 815 complex fluidic devices. This is further illustrated by a simplified (in terms of flow regime) industrial
 816 case aimed at optimizing the distributor section of a fin plate heat exchanger. All cases considered
 817 converge within a few hundred iterations, with detailed computational efficiency data showing that
 818 the computational time in large-scale 3-D problems is dominated by the meshing/remeshing steps.
 819 It is worth emphasizing in this regards that we did not seek to optimize efficiency, neither by
 820 adjusting the initial design, nor by fine tuning the descent factor, the only requirement being that
 821 the displacement achieved at each step must be below the cut-off thickness of the level set for
 822 the evolved interface to remain accurately tracked. Here, the reported cost is essentially that of
 823 recovering the proper volume of fluid, as fulfilling the proper volume constraint from the outset
 824 requires a larger number of smaller solid inclusions, which would either dramatically increase the
 825 surface of the interfaces that needs be captured (and thus the number of mesh elements needed to
 826 maintain the numerical accuracy), or risk clogging the fluid path due to insufficient mesh refinement.
 827 Future work should include application to real life, multiphysics problems (e.g., heat transfer of

828 multiphase flows), as well as the extension to transient and turbulent problems. Evaluating multi-
829 component adaptation criteria taking into account the difference in the spatial supports of the state
830 and adjoint solutions is also of primary interest to further improve the accuracy of the gradient
831 evaluations.

832 Acknowledgement

833 This work is part of the PANTTHER project, that has received funding from the Clean Sky2
834 Joint Undertaking (JU) under grant agreement No 886698. The JU receives support from the
835 European Union’s Horizon 2020 research and innovation program and the Clean Sky 2 JU members
836 other than the Union. It reflects only the authors’ view and the JU is not responsible for any use
837 that may be made of the information it contains.

838 References

- 839 [1] M. P. Bendsøe, N. Kikuchi, Generating optimal topologies in structural design using a homog-
840 enization method, *Comput. Methods Appl. Mech. Engrg.* 71 (1988) 197–224.
- 841 [2] M. P. Bendsøe, O. Sigmund, *Topology optimization: theory, methods, and applications*,
842 Springer Science & Business Media, 2003.
- 843 [3] J. Alexandersen, C. S. Andreasen, A review of topology optimisation for fluid-based problems,
844 *Fluids* 5 (2020) 29.
- 845 [4] O. Sigmund, K. Maute, Topology optimization approaches, *Struct. Multidiscipl. Optim.* 48
846 (2013) 1031–1055.
- 847 [5] K. Suzuki, N. Kikuchi, A homogenization method for shape and topology optimization, *Com-
848 put. Methods Appl. Mech. Engrg.* 93 (1991) 291–318.
- 849 [6] G. Allaire, E. Bonnetier, G. Francfort, F. Jouve, Shape optimization by the homogenization
850 method, *Numer. Math.* 76 (1997) 27–68.
- 851 [7] J. A. Sethian, A. Wiegmann, Structural boundary design via level set and immersed interface
852 methods, *J. Comput. Phys.* 163 (2000) 489–528.
- 853 [8] M. Y. Wang, X. Wang, D. Guo, A level set method for structural topology optimization,
854 *Comput. Methods Appl. Mech. Engrg.* 192 (2003) 227–246.
- 855 [9] G. Allaire, F. Jouve, A.-M. Toader, Structural optimization using sensitivity analysis and a
856 level-set method, *J. Comput. Phys.* 194 (2004) 363–393.
- 857 [10] N. P. Van Dijk, K. Maute, M. Langelaar, F. Van Keulen, Level-set methods for structural
858 topology optimization: a review, *Struct. Multidiscipl. Optim.* 48 (2013) 437–472.
- 859 [11] C. S. Andreasen, M. O. Elingaard, N. Aage, Level set topology and shape optimization by
860 density methods using cut elements with length scale control, *Struct. Multidiscipl. Optim.* 62
861 (2020) 685–707.
- 862 [12] X.-B. Duan, F.-F. Li, X.-Q. Qin, Adaptive mesh method for topology optimization of fluid
863 flow, *Appl. Math. Lett.* 44 (2015) 40–44.
- 864 [13] K. E. Jensen, Topology optimization of stokes flow on dynamic meshes using simple optimizers,
865 *Comp. Fluids* 174 (2018) 66–77.
- 866 [14] X.-B. Duan, X.-Q. Qin, Optimality criteria coupled adaptive mesh method for optimal shape
867 design of stokes flow, *Math. Methods Appl. Sci.* 39 (2016) 3910–3920.
- 868 [15] X.-B. Duan, F.-F. Li, X.-Q. Qin, Topology optimization of incompressible Navier–Stokes prob-
869 lem by level set based adaptive mesh method, *Comput. Math. Appl.* 72 (2016) 1131–1141.

- 870 [16] H. Garcke, C. Hecht, M. Hinze, C. Kahle, Numerical approximation of phase field based shape
871 and topology optimization for fluids, *SIAM J. Sci. Comput.* 37 (2015) A1846–A1871.
- 872 [17] F. Feppon, G. Allaire, F. Bordeu, J. Cortial, C. Dapogny, Shape optimization of a coupled
873 thermal fluid–structure problem in a level set mesh evolution framework, *SeMA* 76 (2019)
874 413–458.
- 875 [18] F. Feppon, G. Allaire, C. Dapogny, P. Jolivet, Topology optimization of thermal fluid–
876 structure systems using body-fitted meshes and parallel computing, *J. Comput. Phys.* 417
877 (2020) 109574.
- 878 [19] F. Feppon, G. Allaire, C. Dapogny, P. Jolivet, Body-fitted topology optimization of 2d and
879 3d fluid-to-fluid heat exchangers, *Comput. Methods Appl. Mech. Engrg.* 376 (2021) 113638.
- 880 [20] C. Talischi, A. Pereira, G. H. Paulino, I. F. M. Menezes, M. S. Carvalho, Polygonal finite
881 elements for incompressible fluid flow, *Int. J. Numer. Meth. Fl.* 74 (2014) 134–151.
- 882 [21] P. F. Antonietti, M. Bruggi, S. Scacchi, M. Verani, On the virtual element method for topology
883 optimization on polygonal meshes: a numerical study, *Comput. Math. Appl.* 74 (2017) 1091–
884 1109.
- 885 [22] M. A. A. Suárez, J. S. Romero, A. Pereira, I. F. M. Menezes, On the virtual element method
886 for topology optimization of non-Newtonian fluid-flow problems, *Eng. Comput.* 38 (2022)
887 5445–5466.
- 888 [23] A. L. Gain, G. H. Paulino, L. S. Duarte, I. F. M. Menezes, Topology optimization using
889 polytopes, *Comput. Methods Appl. Mech. Engrg.* 293 (2015) 411–430.
- 890 [24] C. H. Villanueva, K. Maute, CutFEM topology optimization of 3D laminar incompressible
891 flow problems, *Comput. Methods Appl. Mech. Engrg.* 320 (2017) 444–473.
- 892 [25] K. E. Jensen, Solving stress and compliance constrained volume minimization using anisotropic
893 mesh adaptation, the method of moving asymptotes and a global p-norm, *Struct. Multidiscipl.
894 Optim.* 54 (2016) 831–841.
- 895 [26] S. Micheletti, S. Perotto, L. Soli, Topology optimization driven by anisotropic mesh adapta-
896 tion: Towards a free-form design, *Comput. Struct.* 214 (2019) 60–72.
- 897 [27] T. Borrvall, J. Petersson, Large-scale topology optimization in 3d using parallel computing,
898 *Comput. Methods Appl. Mech. Engrg.* 190 (2001) 6201–6229.
- 899 [28] A. Evgrafov, C. J. Rupp, K. Maute, M. L. Dunn, Large-scale parallel topology optimization
900 using a dual-primal substructuring solver, *Struct. Multidiscipl. Optim.* 36 (2008) 329–345.
- 901 [29] N. Aage, B. S. Lazarov, Parallel framework for topology optimization using the method of
902 moving asymptotes, *Struct. Multidiscipl. Optim.* 47 (2013) 493–505.
- 903 [30] N. Aage, B. S. Lazarov, Topology optimization using PETSc: An easy-to-use, fully parallel,
904 open source topology optimization framework, *Struct. Multidiscipl. Optim.* 51 (2015) 565–572.
- 905 [31] J. Alexandersen, O. Sigmund, N. Aage, Large scale three-dimensional topology optimisation
906 of heat sinks cooled by natural convection, *Int. J. Heat Mass Transfer* 100 (2016) 876–891.
- 907 [32] C. Othmer, A continuous adjoint formulation for the computation of topological and surface
908 sensitivities of ducted flows, *Int. J. Numer. Meth. Fl.* 58 (2008) 861–877.
- 909 [33] O. Soto, R. Löhner, On the computation of flow sensitivities from boundary integrals, *AIAA-
910 2004-0112* (2004).
- 911 [34] L. Ville, L. Silva, T. Coupez, Convected level set method for the numerical simulation of fluid
912 buckling, *Int. J. Numer. Meth. Fl.* 66 (2011) 324–344.

- 913 [35] T. Coupez, L. Silva, E. Hachem, Implicit boundary and adaptive anisotropic meshing, in:
914 S. Perotto, L. Formaggia (Eds.), *New challenges in grid generation and adaptivity for scientific*
915 *computing*, Springer, 2015, pp. 1–18.
- 916 [36] A. Bonito, J.-L. Guermond, S. Lee, Numerical simulations of bouncing jets, *Int. J. Numer.*
917 *Meth. Fl.* 80 (2016) 53–75.
- 918 [37] E. Hachem, H. Dignonnet, E. Massoni, T. Coupez, Immersed volume method for solving natural
919 convection, conduction and radiation of a hat-shaped disk inside a 3d enclosure, *Int. J. Numer.*
920 *Method H.* 22 (2012) 718–741.
- 921 [38] E. Hachem, S. Feghali, R. Codina, T. Coupez, Immersed stress method for fluid-structure
922 interaction using anisotropic mesh adaptation, *Int. J. Numer. Meth. Eng.* 94 (2013) 805–825.
- 923 [39] E. Hachem, B. Rivaux, T. Kloczko, H. Dignonnet, T. Coupez, Stabilized finite element method
924 for incompressible flows with high Reynolds number, *J. Comput. Phys.* 229 (23) (2010) 8643–
925 8665.
- 926 [40] R. Codina, Stabilized finite element approximation of transient incompressible flows using
927 orthogonal subscales, *Comput. Methods Appl. Mech. Engrg.* 191 (2002) 4295–4321.
- 928 [41] T. Tezduyar, Y. Osawa, Finite element stabilization parameters computed from element ma-
929 trices and vectors, *Comput. Methods Appl. Mech. Engrg.* 190 (3–4) (2000) 411–430.
- 930 [42] R. Codina, Stabilization of incompressibility and convection through orthogonal sub-scales in
931 finite element methods, *Comput. Methods Appl. Mech. Engrg.* 190 (2000) 1579–1599.
- 932 [43] R. Codina, Comparison of some finite element methods for solving the diffusion-convection-
933 reaction equation, *Comput. Methods Appl. Mech. Engrg.* 156 (1998) 185–210.
- 934 [44] S. Badia, R. Codina, Analysis of a stabilized finite element approximation of the transient
935 convection-diffusion equation using an ALE framework, *SIAM J. Numer. Anal.* 44 (2006)
936 2159–2197.
- 937 [45] T. Coupez, Metric construction by length distribution tensor and edge based error for
938 anisotropic adaptive meshing, *J. Comput. Phys.* 230 (2011) 2391–2405.
- 939 [46] G. Jannoun, E. Hachem, J. Veysset, T. Coupez, Anisotropic meshing with time-stepping
940 control for unsteady convection-dominated problems, *Appl. Math. Model.* 39 (7) (2015) 1899–
941 1916.
- 942 [47] T. Coupez, Génération de maillage et adaptation de maillage par optimisation locale, *Rev.*
943 *Eur. Elem. Finis* 9 (4) (2000) 403–423.
- 944 [48] T. Coupez, G. Jannoun, J. Veysset, E. Hachem, Edge-based anisotropic mesh adaptation for
945 CFD applications, in: X. Jiao, J.-C. Weill (Eds.), *Procs. of the 21st International Meshing*
946 *Roundtable*, Springer, 2013, pp. 567–583.
- 947 [49] P. Meliga, J.-M. Chomaz, D. Sipp, Unsteadiness in the wake of disks and spheres: Instability,
948 receptivity and control using direct and adjoint global stability analyses, *J. Fluid Struct.* 25
949 (2009) 601–616.
- 950 [50] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin,
951 A. Dener, V. Eijkhout, W. Gropp, et al., *PETSc users manual (rev. 3.13)*, Technical report
952 ANL-95/11-Rev. 3.13, Argonne National Lab. (2020).
- 953 [51] Y. Mesri, H. Dignonnet, H. Guillard, Mesh partitioning for parallel computational fluid dynam-
954 ics applications on a grid, in: *Procs. of the 4th International Symposium on. Finite Volumes*
955 *for Complex Applications*, 2005, pp. 631–642.
- 956 [52] H. Dignonnet, L. Silva, T. Coupez, Cimlib: a fully parallel application for numerical simulations
957 based on components assembly, in: *Procs. of the 9th International Conference on Numerical*
958 *Methods in Industrial Forming Processes*, 2007, pp. 269–274.

- 959 [53] H. Dignonnet, L. Silva, T. Coupez, Massively parallel computation on anisotropic meshes, in:
960 Procs. of the 6th International Conference on Adaptive Modeling and Simulation, 2013, pp.
961 199–211.
- 962 [54] Y. Mesri, H. Dignonnet, T. Coupez, Advanced parallel computing in material forming with
963 CIMLib, *Eur. J. Comput. Mech.* 18 (2009) 669–694.
- 964 [55] T. Borrvall, J. Petersson, Topology optimization of fluids in Stokes flow, *Int. J. Numer. Meth.*
965 *Fl.* 41 (2003) 77–107.
- 966 [56] X.-B. Duan, Y.-C. Ma, R. Zhang, Shape-topology optimization for Navier–Stokes problem
967 using variational level set method, *J. Comput. Appl. Math.* 222 (2008) 487–499.
- 968 [57] A. Gersborg-Hansen, O. Sigmund, R. B. Haber, Topology optimization of channel flow prob-
969 lems, *Struct. Multidiscipl. Optim.* 30 (2005) 181–192.
- 970 [58] M. Abdelwahed, M. Hassine, M. Masmoudi, Optimal shape design for fluid flow using topo-
971 logical perturbation technique, *J. Math. Anal.* 356 (2009) 548–563.
- 972 [59] P. E. Papadopoulos, I. P. A. and Farrell, T. M. Surowiec, Computing multiple solutions of
973 topology optimization problems, *SIAM J. Sci. Comput.* 43 (2021) A1555–A1582.
- 974 [60] G. Pingen, M. Waidmann, A. Evgrafov, K. Maute, A parametric level-set approach for topol-
975 ogy optimization of flow domains, *Struct. Multidiscipl. Optim.* 41 (2010) 117–131.
- 976 [61] J. Boussinesq, Mémoire sur l’influence des frottements dans les mouvements réguliers des
977 fluids, *J. Math. Pures Appl.* 13 (1868) 377–424.