



HAL
open science

A stabilized finite element framework for anisotropic adaptive topology optimization of incompressible fluid flows

Wassim Abdel Nour, Joseph Jabbour, Damien Serret, Philippe Meliga, Elie Hachem

► To cite this version:

Wassim Abdel Nour, Joseph Jabbour, Damien Serret, Philippe Meliga, Elie Hachem. A stabilized finite element framework for anisotropic adaptive topology optimization of incompressible fluid flows. *Fluids*, 2023, 8 (8), pp.232. 10.3390/fluids8080232 . hal-04245158

HAL Id: hal-04245158

<https://hal.science/hal-04245158>

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.

A stabilized finite element framework for anisotropic adaptive topology optimization of incompressible fluid flows

Wassim Abdel Nour ^{1,2}, Joseph Jabbour ², Damien Serret ², Philippe Meliga ¹ and Elie Hachem ^{1,*}

¹ Mines Paris, PSL University, Centre for material forming (CEMEF), UMR CNRS, 06904 Sophia Antipolis, France

² TEMISTh SAS, Technocentre des Florides, 13700 Marignane, France

* Correspondence: elie.hachem@minesparis.psl.eu; Tel.: +33 4 93 95 74 58

Abstract: This paper assesses the feasibility of performing topology optimization of laminar incompressible flows governed by the steady state Navier–Stokes equations using anisotropic mesh adaptation to achieve a high-fidelity description of all fluid-solid interfaces. The present implementation combines an immersed volume method solving stabilized finite element formulations cast in the Variational Multiscale (VMS) framework, and level set representations of the fluid-solid interfaces, used as a posteriori anisotropic error estimator to minimize the interpolation error under the constraint of a prescribed number of nodes in the mesh. Numerical results provided for several two-dimensional problems of power dissipation minimization show that the optimal designs are mesh-independent (although the convergence rate does decrease as the number of nodes increases), agree well with reference results from the literature, and provide superior accuracy over prior studies solved on isotropic meshes (fixed or adaptively refined).

Keywords: Topology Optimization; Fluid mechanics; Level Set Method; Anisotropic mesh adaptation

1. Introduction

Fluid flow topology optimization is the process of finding the best path for a fluid to flow in a prescribed design domain to maximize a measure of performance under a set of design constraints, for instance, to minimize dissipation subject to a constant volume of fluid. Such an approach originates from solid mechanics [1,2], where it has matured into a powerful, reliable and increasingly available tool for engineers in the early stages of complex structural design processes at the component level [3,4]. It has since spread to a variety of other physics modeled after partial differential equations; see Refs. [5,6] for surveys of the evolving methods and applications, and Ref. [7] for a recent literature review within the context of fluid flow problems. Topology optimization has mathematical foundation built on iterative analysis and design update steps, often steered by gradient evaluations. What stands out (compared to the size and shape optimization methods it has emerged from) is the great design freedom, that allows generating non-intuitive designs from arbitrary initial guesses, possibly meeting conflicting requirements and complex interdependencies between design parameters and system response.

We leave aside here explicit boundary methods, that represent the fluid-solid interface by edges or faces of a body-fitted mesh, and have limited flexibility to handle complicated topological changes. The prevalent classes of methods for topology optimization are the density and level set methods. Density methods rely on a Brinkman penalization of the solid domain, where the flow is modeled as a fictitious porous material with very low permeability [1,8,9]. They manage drastic topological changes, as the gradient information (or sensitivity) is distributed over a large part of the domain, but can lead to spurious or leaking flows if the penalization factor is not well-calibrated, since the velocity and pressure fields are computed in the entire domain (both the solid and fluid regions). Level set

Citation: Adel Nour, W.; Jabbour, J., Serret, D.; Meliga, P.; Hachem, E. A stabilized finite element framework for anisotropic adaptive topology optimization of incompressible fluid flows. *Fluids* **2023**, *1*, 0. <https://doi.org/>

Received:

Revised:

Accepted:

Published:

Copyright: © 2023 by the authors. Submitted to *Fluids* for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

methods conversely model the solid boundaries by iso-contours of a level set function [10–12]. They lack a nucleation mechanism to create new holes, due to the sensitivities being located only at the solid-fluid interface, which is often relieved using initial designs with many holes. Meanwhile, they easily handle complicated topological changes (e.g., merging or cancellation of holes), and allow for well defined, crisp interface representations while avoiding the intermediate material phases (grayscale) and mesh-dependent spatial oscillations of the interface geometry (staircasing) often encountered in density methods [13].

The norm in topology optimization is to employ fixed finite element meshes with close-to-uniform element size, small enough that all relevant physical phenomena are reliably captured, but not so small that the cost of performing the optimization becomes unaffordable. A recent trend has been to use adaptive remeshing techniques to maintain a competitive computational cost. Such an approach consists in generating a coarse base grid, then in adding recursively finer and finer subgrids in the regions requiring higher resolution. This repeats either until a maximum level of refinement is reached, or until the local truncation error drops below a certain tolerance, for more sophisticated implementations endowed with error estimation routines. Within the context of fluid flow problems, particular emphasis has been put on (but not limited to) adaptive meshing refinement (AMR) schemes, using both density [14,15] and level set methods [16,17]; see also [18] for an application to phase field methods¹ and [19–21] for recent efforts applying a different remeshing scheme to a combination of level set functions and adaptive body-conforming meshes.

There is still ample room for progress, though, as almost all adaptive algorithms applied so far to fluid flow topology optimization support only isotropic size maps. Fluid dynamics conversely involves convection dominated phenomena for which anisotropic meshes are highly desirable [22], especially in the vicinity of the solid boundaries, where the fluid velocity exhibits steep gradients in the wall-normal direction and skin-friction plays a defining role. The premise of this study is that the ability to generate highly stretched elements in boundary layer regions can substantially increase the accuracy of the geometric representation, compared to what is often seen in topology optimization of flow problems, and naturally convey said accuracy to the numerical solution without sophisticated interpolation or discretization techniques. We note that this is all perfectly in line with the recommendations made in [7] to improve upon the current state of the art. Nonetheless, our literature review did not reveal any other study combining anisotropic mesh adaptation and fluid flow topology optimization, besides the density-based optimisation of Stokes flow in Ref. [15], possibly because the notorious difficulty of finding spatial discretization schemes that meet the level of robustness required by automatic anisotropic mesh adaptation.

This research intends to fill the gap by introducing a novel numerical framework for topology optimization of Navier–Stokes flows. The latter combines level set methods and anisotropic mesh adaptation to handle arbitrary geometries immersed in an unstructured mesh. The Navier–Stokes system is solved by a variational multiscale (VMS) stabilized finite element method supporting elements of aspect ratio up to the order of 1000:1 [23]. The same numerical method is used to solve the adjoint Navier–Stokes system underlying the sensitivity analysis needed to evolve the level set function. The metric map providing both the size and the stretching of mesh elements in a very condensed information data is derived from the level set. A posteriori anisotropic error estimator is then used to minimize the interpolation error under the constraint of a prescribed number of nodes in the mesh. The latter can be adjusted over the course of optimization, meaning that the base grid can be either refined or coarsened on demand. This contrasts with AMR, whose total number of mesh elements cannot be controlled, and whose mesh cannot be coarsened further than its base configuration. Since it reduces the cost of modelling the solid material away from the interface, such an approach is expected to achieve further speed-ups while also helping improve manufacturability of the optimal design, which remains an issue as most classical

¹ Another class of interface capturing schemes that remain less popular due to the larger computational cost and the difficulty of numerically discretizing the biharmonic phase-field equation.

topology optimization methods render organic designs that can be difficult to translate into computer-aided design models. 89
90

The paper is organized as follows: the governing equations are formulated in Sec. 2. The anisotropic mesh adaptation algorithm and the immersed, stabilized finite element numerical framework used to perform the design update step are described in Secs. 3 and 4, respectively. The details of the implemented topology optimization algorithm are provided in Sec. 5. Finally, numerical experiments showcasing the potential of the approach on two-dimensional power dissipation minimization problems are presented in section 6, with particular attention paid to highlighting the improved accuracy and mesh-independence of the obtained solutions. 91
92
93
94
95
96
97
98

2. Immersed model for fluid flow topology optimization 99

In the following, we denote by Ω a fixed, open bounded domain in \mathbb{R}^d (with d the space dimension), with boundary $\partial\Omega$ oriented with inward-pointing normal vector \mathbf{n} . Throughout this study, $\Omega = \Omega_f \cup \Omega_s$ is the disjoint reunion of two domains Ω_f and Ω_s (for simplicity, we refer to Ω_f as the fluid domain, and to Ω_s as the solid domain, although we also fill Ω_s with a fluid for numerical convenience, as further explained in the following). The two domains are separated by an interface $\Gamma = \Omega_f \cap \Omega_s$, whose position we seek to optimize with respect to a certain measure of performance, here a cost function J to minimize. 100
101
102
103
104
105
106

2.1. State equations 107

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

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

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

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 , i.e., the combined boundary of all surfaces where fluid enters the domain, and outlet Γ_o , i.e., 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 pressure and viscous stress conditions at the outlet, and zero velocity at the interface, hence

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

$$p \mathbf{n} = \mu \boldsymbol{\varepsilon}(\mathbf{u}) \cdot \mathbf{n} = \mathbf{0} \quad \text{on } \Gamma_o, \quad (4)$$

$$\mathbf{u} = \mathbf{0} \quad \text{on } \Gamma. \quad (5)$$

2.2. Adjoint-based sensitivity analysis 108

We assume in the following that the cost function (i) can be formulated as a surface integral over the domain boundary (rather than its interior), and (ii) does not depend on the flow quantities on the wall, which is most often true in topology optimization. It is thus expressed as integrals over all or any part of inlet and/or outlet, i.e., 109
110
111
112

$$J_s = \int_{\Gamma_i \cup \Gamma_o} J ds. \quad (6)$$

The problem of minimizing the cost function subject to Navier–Stokes as state equations is tackled using the continuous adjoint method. The reader interested in the technicalities of the method is referred to [24]. One first forms the Lagrangian

$$\mathcal{L} = \int_{\Gamma_i \cup \Gamma_o} J \, ds - \int_{\Omega_f} \tilde{p} \nabla \cdot \mathbf{u} \, dv - \int_{\Omega_f} \tilde{\mathbf{u}} \cdot (\rho \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p - \nabla \cdot (2\mu \boldsymbol{\varepsilon}(\mathbf{u}))) \, dv, \quad (7)$$

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

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

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

$$\begin{aligned} \delta_{(\mathbf{u}, p)} \mathcal{L} &= \int_{\Omega_f} (\nabla \cdot \tilde{\mathbf{u}}) \delta p \, dv + \int_{\Omega_f} (-\rho \mathbf{u} \cdot \nabla \tilde{\mathbf{u}} + \rho \nabla \mathbf{u}^T \cdot \tilde{\mathbf{u}} - \nabla \tilde{p} - \nabla \cdot (2\mu \boldsymbol{\varepsilon}(\tilde{\mathbf{u}}))) \cdot \delta \mathbf{u} \, dv \\ &+ \int_{\Gamma_i \cup \Gamma_o} \partial_{\mathbf{u}} J \cdot \delta \mathbf{u} \, ds + \int_{\partial \Omega_f} (\tilde{p} \mathbf{n} + 2\mu \boldsymbol{\varepsilon}(\tilde{\mathbf{u}}) \cdot \mathbf{n} + \rho(\mathbf{u} \cdot \mathbf{n}) \tilde{\mathbf{u}}) \cdot \delta \mathbf{u} \, ds \\ &- \int_{\Gamma_i \cup \Gamma_o} \partial_p J \mathbf{n} \cdot (-\delta p \mathbf{n} + 2\mu \boldsymbol{\varepsilon}(\delta \mathbf{u}) \cdot \mathbf{n}) \, ds - \int_{\partial \Omega_f} \tilde{\mathbf{u}} \cdot (-\delta p \mathbf{n} + 2\mu \boldsymbol{\varepsilon}(\delta \mathbf{u}) \cdot \mathbf{n}) \, ds, \end{aligned} \quad (9)$$

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 (9). 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

$$\nabla \cdot \tilde{\mathbf{u}} = 0 \quad \text{in } \Omega_f, \quad (10)$$

$$-\rho \mathbf{u} \cdot \nabla \tilde{\mathbf{u}} + \rho \nabla \mathbf{u}^T \cdot \tilde{\mathbf{u}} = \nabla \tilde{p} + \nabla \cdot (2\mu \boldsymbol{\varepsilon}(\tilde{\mathbf{u}})) \quad \text{in } \Omega_f, \quad (11)$$

driven by the non-homogeneous boundary conditions

$$\tilde{\mathbf{u}} = -\partial_p J \mathbf{n} \quad \text{on } \Gamma_i, \quad (12)$$

$$\tilde{p} \mathbf{n} + 2\mu \boldsymbol{\varepsilon}(\tilde{\mathbf{u}}) \cdot \mathbf{n} + \rho(\mathbf{u} \cdot \mathbf{n}) \tilde{\mathbf{u}} = -\partial_{\mathbf{u}} J \quad \text{on } \Gamma_o, \quad (13)$$

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

associated to (3)-(5). Note, the minus sign ahead of the first term of the adjoint momentum equation (11) reflects the reversal in directionality due to the non-normality of the linearized evolution operator [25]. Expressing the interface normal deformation after [26] as

$$\delta \mathbf{u} = \beta \nabla \mathbf{u} \cdot \mathbf{n}, \quad (15)$$

the variation with respect to the design variable (now encompassing the domain deformation) is ultimately computed as

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

where the last equality stems from the incompressibility of the state and adjoint solutions [24]. This enables efficient design update schemes via first-order gradient descent methods, as the second term in the integrand is the desired sensitivity to a displacement β at some specific

point of the interface. For instance the simplest steepest-descent algorithm implemented herein moves down the cost function, in the direction of the steepest slope using

$$\beta = -\mu(\nabla\tilde{\mathbf{u}} \cdot \mathbf{n}) \cdot (\nabla\mathbf{u} \cdot \mathbf{n}), \quad (16)$$

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

2.3. Level set representation of the interface

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

$$\varphi(\mathbf{x}) = \begin{cases} -\text{dist}(\mathbf{x}, \Gamma) & \text{if } \mathbf{x} \in \Omega_f, \\ 0 & \text{if } \mathbf{x} \in \Gamma, \\ \text{dist}(\mathbf{x}, \Gamma) & \text{if } \mathbf{x} \in \Omega_s, \end{cases} \quad (17)$$

with the convention that $\varphi < 0$ in the fluid domain. Once the sensitivity analysis has output a displacement β in the direction of the steepest slope, the position of the level set is updated solving a transport equation with normal velocity $\beta\mathbf{n}/\Delta\tau$, where $\Delta\tau$ is a pseudo-time step to convert from displacement to velocity, that has no physical relevance since we are not concerned by the absolute displacement of a given point on the interface, only by its relative displacement with respect to its neighbors. This equation is posed in the whole domain Ω , which is because the normal vector recovered at the interface as $\mathbf{n} = \nabla\varphi/\|\nabla\varphi\|$ is easily extended to Ω using (17). The main problem with this approach is that the level set after transport is generally no longer a distance function, which is especially problematic when a specific remeshing strategy depending on the distance property is used at the interface (as is the case in this study). As a result, the distance function needs to be reinitialized, which is done here using a coupled convection-reinitialization method wherein the level set function is automatically reinitialized during the resolution of the transport equation. In practice, the signed distance function is cut off using a hyperbolic tangent filter, as defined by

$$\phi = E \tanh\left(\frac{\varphi}{E}\right), \quad (18)$$

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

$$\partial_\tau\phi + \mathbf{a}_\tau \cdot \nabla\phi = S, \quad (19)$$

where we note

$$\mathbf{a}_\tau = \frac{\beta}{\Delta\tau}\mathbf{n} + \frac{\lambda}{\Delta\tau} \text{sgn}(\phi) \frac{\nabla\phi}{\|\nabla\phi\|}, \quad S = \frac{\lambda}{\Delta\tau} \text{sgn}(\phi) \left(1 - \left(\frac{\phi}{E}\right)^2\right), \quad (20)$$

and λ is a parameter homogeneous to a length, set to the mesh size h_\perp in the direction normal to the interface. Such an approach is shown in [27–29] to reduce the computational cost and to ensure a better mass conservation compared to the classical Hamilton–Jacobi method in which both steps are performed in succession. Moreover, since the filtered level set defined in (18) is bounded, Dirichlet boundary conditions $\phi = \pm E$ are easily appended to Eq. (19) to explicitly design fluid and solid sub-regions of $\partial\Omega$.

3. Anisotropic mesh adaptation

A primitive pseudo-code of the procedure for solving the topology optimization problem posed in Sec. 2 is provided in the following algorithm:

Algorithm 1 Simplified update scheme

Require: Anisotropic mesh adapted to initial level set function

- 1: **loop**
 - 2: Compute state
 - 3: Compute adjoint
 - 4: Compute cost function sensitivity
 - 5: Set displacement in the direction of steepest slope
 - 6: Update level set
 - 7: Generate anisotropic mesh adapted to new level set
-

to repeat until a maximum number of iterations or a convergence threshold has been reached. In a nutshell, this is done here using a finite element immersed numerical framework combining implicit representation of the different domains, level set description of the interface, and anisotropic remeshing capabilities. For the sake of readability, the mesh adaptation algorithm, whose implementation in the context of fluid flow topology optimization makes for the main novelty of this study, is presented here as a stand-alone section. We then walk through each of the other steps in Sec. 4 and review the various problems involved and the numerical methods for solving them.

3.1. Construction of an anisotropic mesh

The main idea of anisotropic, metric-based mesh adaptation is to generate a uniform mesh (with unit length edges and regular elements) in a prescribed Riemannian metric space, but anisotropic and well adapted (with highly stretched elements) in the Euclidean space. Assuming that, in the context of metric-based adaptation methods, controlling the interpolation error suffices to master the global approximation error, the objective can be formulated as finding the mesh, made up of at most N_n nodes, that minimizes the linear interpolation error in the L^1 norm. Following the lines of [30,31], an edge-based error estimator combined to a gradient recovery procedure is used to compute, for each node, a metric tensor that prescribes a set of anisotropic directions and stretching factors along these directions, without any direct information from the elements, nor any underlying interpolation. The optimal stretching factor field is obtained by solving an optimization problem using the equi-distribution principle under the constraint of a fixed number of nodes in the mesh, after which a new mesh is generated using the procedure described in [32], based on a topological representation of the computational domain.

3.2. Edge error estimate

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 $\mathbf{x}^j \in \Sigma(i)$, where $\Sigma(i)$ is the set of nodes connected to \mathbf{x}^i , and the number of such nodes is noted as $|\Sigma(i)|$. Also, given a regular analytical (scalar) function ψ defined on Ω , and its P1 finite element approximation ψ_h computed on Ω_h , we follow [30] and estimate the interpolation error along the edge \mathbf{x}^{ij} as the projection along the edge of the second derivative of ψ . This is obtained projecting along the edge a Taylor expansion of the gradient of ψ at \mathbf{x}^j to give

$$\varepsilon_{ij} = |\mathbf{g}^{ij} \cdot \mathbf{x}^{ij}|, \quad (21)$$

where the i and j superscripts indicate nodal values at nodes \mathbf{x}^i and \mathbf{x}^j , respectively, $\mathbf{g}^i = \nabla\psi(\mathbf{x}^i)$ 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 the edge. Although Eq. (21) involves only values of the gradient at the edge extremities and can thus be evaluated without resorting to resource expensive Hessian reconstruction methods, this however requires the gradient of ψ to be known and continuous at the nodes, which in turn requires full knowledge of ψ . Meanwhile, only the linear interpolate ψ_h is known in practice, whose gradient is piecewise constant and

discontinuous from element to element, although its projection along the edges *is* continuous since it depends only on the nodal values of the field. 201

A recovery procedure is thus used to build a continuous gradient estimator defined directly at the nodes. It is shown in [30] that a suitable error estimate preserving second-order accuracy is obtained substituting the reconstructed gradient for the exact gradient in (21), to give 202 203 204 205 206

$$\varepsilon_{ij} = |\bar{\mathbf{g}}^{ij} \cdot \mathbf{x}^{ij}|, \quad (22)$$

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 defined in a least-square sense as 207 208

$$\bar{\mathbf{g}}^i = \operatorname{argmin}_{\mathbf{g} \in \mathbb{R}^d} \sum_{j \in \Sigma(i)} |(\mathbf{g} - \nabla \psi_h) \cdot \mathbf{x}^{ij}|^2, \quad (23)$$

for which an approximate solution using the nodal values as sole input is shown in [30] to be 209

$$\bar{\mathbf{g}}^i = (\mathbf{X}^i)^{-1} \cdot \sum_{j \in \Sigma(i)} (\psi_h(\mathbf{x}^j) - \psi_h(\mathbf{x}^i)) \mathbf{x}^{ij}, \quad (24)$$

where \mathbf{X}^i is the length distribution tensor defined as 210

$$\mathbf{X}^i = \frac{1}{|\Sigma(i)|} \sum_{j \in \Sigma(i)} \mathbf{x}^{ij} \otimes \mathbf{x}^{ij}, \quad (25)$$

that gives an average representation of the distribution of the edges sharing an extremity. 211

3.3. Metric construction 212

In order to relate the error indicator ε_{ij} defined in (22) to a metric suitable for mesh adaptation purposes, we introduce the stretching factor s_{ij} as the ratio between the length of the edge \mathbf{x}^{ij} after and before the adaptation. The metric at node \mathbf{x}^i is sought to generate unit stretched edge length in the metric space, that is, 213 214 215 216

$$(s_{ij} \mathbf{x}^{ij})^T \cdot \mathbf{M}^i \cdot (s_{ij} \mathbf{x}^{ij}) = 1, \quad \forall j \in \Sigma(i), \quad (26)$$

for which an approximate least-square solution is shown in [30] to be 217

$$\mathbf{M}^i = \left(\frac{d}{|\Sigma(i)|} \sum_{j \in \Sigma(i)} s_{ij}^2 \mathbf{x}^{ij} \otimes \mathbf{x}^{ij} \right)^{-1}, \quad (27)$$

provided the nodes in $\Sigma(i)$ form at least d non co-linear edges with \mathbf{x}^i (which is the case if the mesh is valid). The metric solution of (27) is ultimately computed setting a target total number of nodes N_n . Assuming a total error equi-distributed among all edges, the stretching factor is shown in [31] to be 218 219 220 221

$$s_{ij} = \left(\frac{\sum_i N_i(1)}{N_n} \right)^{\frac{2}{d}} \varepsilon_{ij}^{-1/2}, \quad (28)$$

where $N_i(1)$ is the number of nodes generated in the vicinity of node \mathbf{x}^i for a unit error, given by 222 223

$$N_i(1) = \left(\det \left(\frac{d}{|\Sigma(i)|} \sum_{j \in \Sigma(i)} \varepsilon_{ij}^{1/2} \frac{\mathbf{x}^{ij}}{|\mathbf{x}^{ij}|} \otimes \frac{\mathbf{x}^{ij}}{|\mathbf{x}^{ij}|} \right) \right)^{-1/2}. \quad (29)$$

3.4. Summary 224

In order to simplify and clarify the presentation, the main steps needed for metric construction at the nodes is summarized in the following algorithm: 225

Algorithm 2 Anisotropic mesh adaptation algorithm 226

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 (25)
 - 5: Compute nodal recovered gradient $\bar{\mathbf{g}}^i$ using (24)
 - 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 (22)
 - 9: Compute stretching factor s_{ij} using (28)
 - 10: Compute metric \mathbf{M}^i using (27)
 - 11: Generate new mesh by local improvement in the neighborhood of the nodes and edges [32]
 - 12: Interpolate ψ_h on new mesh
-

where classical linear interpolation from one mesh to another is applied. 227

3.5. Level set-based adaptation criteria 228

In practice, the variable used for error estimation purpose is the filtered level set defined in (18), as it satisfies the metric property in a thin layer around the interface (in particular it preserves the zero iso-value of φ , which is the only relevant information for mesh adaptation purposes), but avoids unnecessary adaption of the mesh further away from the interface (where the interpolation error is close-to-zero, due to $\|\nabla\phi\| \sim 0$). This means that the criterion for mesh adaptation is purely geometric, i.e., the same mesh is pre-adapted around the fluid-solid interface, then used to compute all quantities needed to perform the next design update step. The flexibility of the proposed mesh adaptation technique is illustrated in Figs. 1 and 2, where a solid circle, square and regular pentagram defined by level set functions have been immersed close to the boundary of a square cavity filled with fluid, to assess the capability to handle different features (angles, singular points, curvatures) even under drastic conditions. Four meshes made up of 500, 1000, 2500 and 5000 nodes are considered, each of which comes in two flavors, one structured and the other anisotropic, adapted to the level set. On the one hand, the adapted meshes exhibit the expected orientation and deformation of the mesh elements, whose longest edges are parallel to the solid boundaries. On the other hand, they are naturally and automatically coarsened in smooth regions where the filtered level set is constant, while extremely refined near the interface. Also, the transition is finer with an anisotropic adaptive mesh, which allows maintaining a very good accuracy even for a low number of nodes, as evidenced in Fig. 2 by the zero iso-value of the level sets. More quantitative results are available in [33], where it is shown that at least ten times more elements are required in a structured mesh to achieve the same accuracy, as measured computing the total perimeter and area of the three immersed objects. 229

Nonetheless, it is worth mentioning that the approach also supports more complex adaptation criteria featuring physical quantities, thus providing the ability to dynamically adapt the mesh during the simulations. The common method to adapt a mesh to several variables is to combine the metrics corresponding to each individual variable using metric intersection algorithms, which is known to incur a relatively high computational cost and to have potentially non-unique, suboptimal outcome. Conversely, the present approach allows building directly a unique metric from a multi-component error vector combining level set and any relevant flow quantity of interest, as definition (22) is easily extended to account for several sources of error [34]. Indeed, if we consider $\boldsymbol{\psi} = (\psi_1, \psi_2, \dots, \psi_p)$ a 230

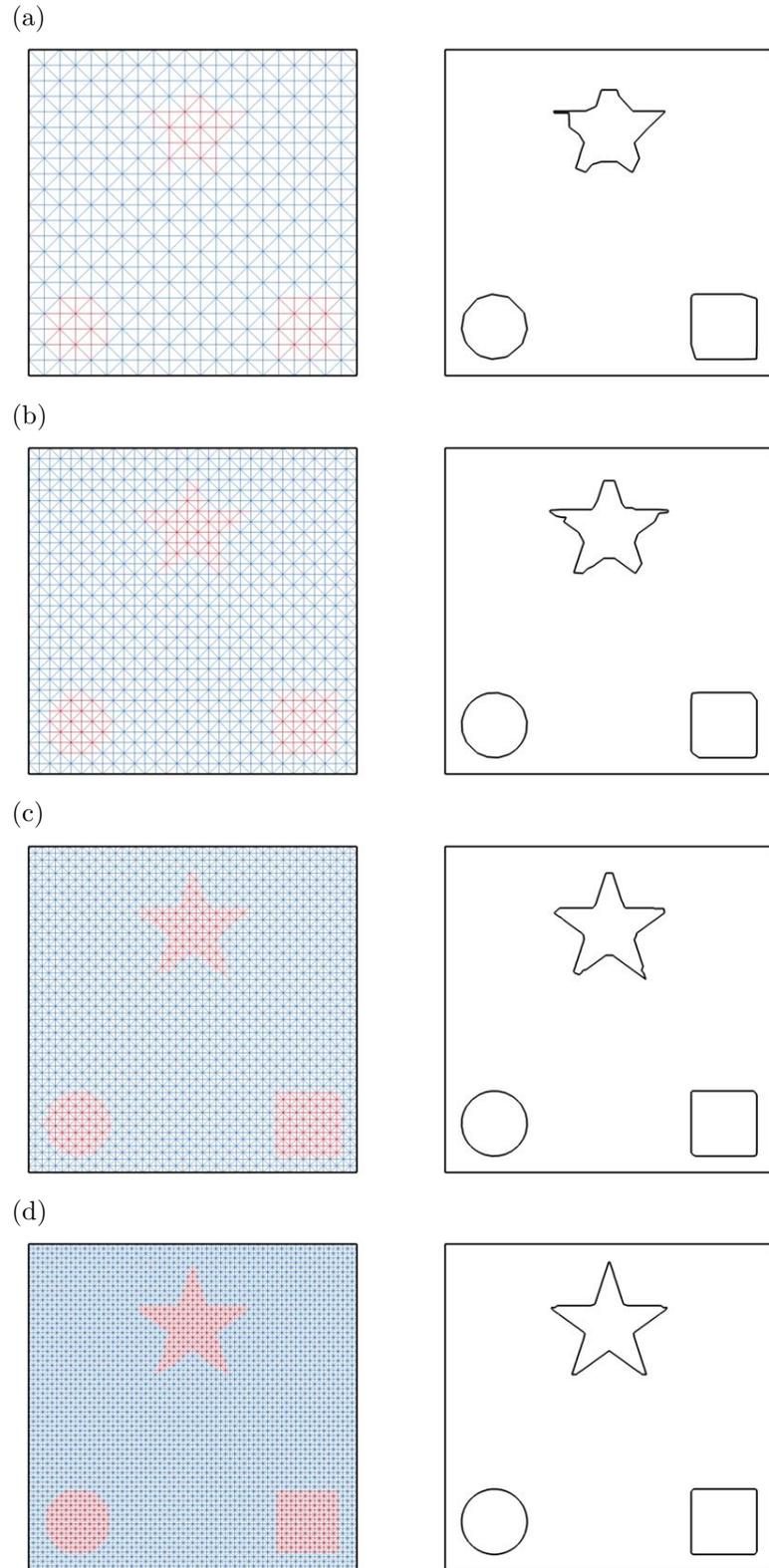


Figure 1. Three immersed solid objects inside a squared cavity filled with fluid using structured meshes. (a) Mesh and zero iso-value of the level set function for a structured mesh with 500 nodes. (b-d) Same as (a) for a structured mesh with (b) 1000 nodes, (c) 2500 nodes and (d) 5000 nodes. The red and blue hues correspond to the solid and fluid domains, respectively.

vector consisting of p scalar variables, it comes out straightforwardly that 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 for the

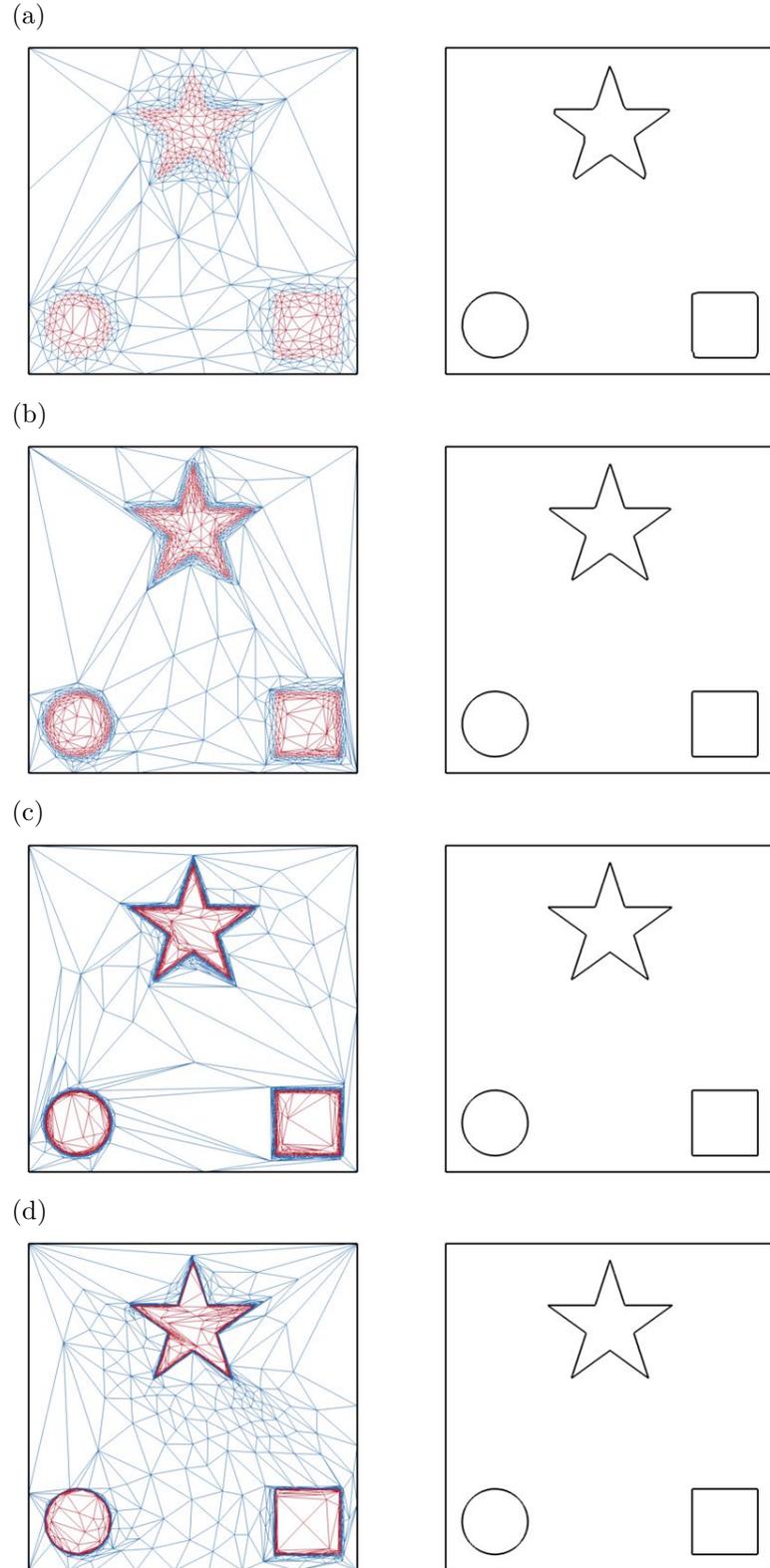


Figure 2. Same as Fig. 1 using anisotropic adapted meshes. (a) Mesh and zero iso-value of the level set function for an anisotropic mesh with 500 nodes, adapted using the level set filtered with $E = 2 \times 10^{-3}$. (b-d) Same as (a) for an anisotropic mesh with (b) 1000 nodes with $E = 10^{-3}$, (c) 2500 nodes with $E = 5 \times 10^{-4}$ and (d) 5000 nodes with $E = 10^{-4}$.

edge from which to compute the stretching factor (28) and ultimately, the metric solution of (27). For instance, the $2d + 3$ sized nodal vector field defined as

$$\psi_h(\mathbf{x}^i) = \left(\frac{\phi_h^i}{\max_{j \in \Sigma(i)} \phi_h^j}, \frac{u_{h_{k \in \{1 \dots d\}}}^i}{\|\mathbf{u}_h^i\|}, \frac{\|\mathbf{u}_h^i\|}{\max_{j \in \Sigma(i)} \|\mathbf{u}_h^j\|}, \frac{\tilde{u}_{h_{k \in \{1 \dots d\}}}^i}{\|\tilde{\mathbf{u}}_h^i\|}, \frac{\|\tilde{\mathbf{u}}_h^i\|}{\max_{j \in \Sigma(i)} \|\tilde{\mathbf{u}}_h^j\|} \right), \quad (30)$$

can be used to combine adaptivity with respect to the norm and direction of the state and adjoint velocity vectors, in addition to the level set. Because all fields are normalized by their respective global maximum, a field much larger in magnitude cannot dominate the error estimator, meaning that the variations of all variables are fairly taken into account. This benefits problems involving more complex physics (e.g., turbulence, heat transfer, fluid-structure interaction, multiple phases, possibly in interaction with one another), all the more so in the context of topology optimization, as the difference in the spatial supports of the state and adjoint quantities resulting from the non-normality of the linearized evolution operator [35] may otherwise yield conflicting requirements in terms of the regions of the computational domain most in need of refinement.

4. Computational methods

This section is devoted to the stabilized finite element numerical framework used to compute all solutions of interest on anisotropic adapted meshes and to perform the design update steps. For the sake of simplicity in the notations, and as long as it does not lead to ambiguity, we omit in what follows the distinction between all continuous variables (e.g., domains, solutions, operators) and their discrete finite element counterparts, as well as the dependency of all variables on the iteration of the optimization process.

4.1. Immersed volume method

The immersed volume method (IVM) [36,37] is used to combine the fluid and solid phases of the problem into a single fluid with variable material properties (density and viscosity). This amounts to solving state and adjoint equations identical to those introduced in Sec. 2, but formulated not just the fluid domain Ω_f , but in the whole domain Ω , with phase-dependent density and viscosity fields adequately interpolated over a small layer around the interface, and otherwise equal to their fluid and solid values. Note, the thickness of the interpolation layer is user-defined and does not increase in size during the optimization, unlike the homogenization method or any other generalized material method. Using the level set function (17) as criterion for anisotropic mesh adaptation ensures that individual material properties can be distributed accurately and smoothly as possible over the smallest possible thickness around the interface. This is classically done by linear interpolation between the fluid and solid values, using a smooth Heaviside function computed from the level set to avoid discontinuities by creating an interface transition with a thickness of a few elements. Such an approach is simpler than the Ersatz material approach [38], that adds a Brinkman penalization term to the Navier–Stokes equations, and has clear connections to density-based methods through the material distribution [17]. It is especially relevant to thermal coupling problems, as having composite conductivity and specific heat means that the amount of heat exchanged at the interface then proceeds solely from the individual material properties on either side of it, and removes the need for a heat transfer coefficient. For the pure flow problems tackled here, though, it suffices to use constant density and viscosity (equal to the fluid values) and to set the velocity to zero at all grid nodes located inside the solid domain Ω_s . Compared to using a very high solid to fluid viscosity ratio to ensure that the velocity is zero in the solid domain, this can be seen as a hard penalty preventing the fluid from leaking across the immersed interface. The latter holds numerically because anisotropic mesh adaptation precisely aligns the mesh element edges along the interface. It thus ensures that the latter does not intersect arbitrarily the mesh elements, which would otherwise compromise the accuracy of the finite element approach.

4.2. Variational multiscale modeling

The convective terms in the incompressible Navier-Stokes and level set transport equations may cause spurious node-to-node velocity oscillations. Furthermore, the equal order linear/linear approximations used for the velocity and pressure variables, albeit very desirable due to its simplicity of implementation and affordable computing cost (especially for 3D applications), may give rise to spurious pressure oscillations. To prevent these numerical

instabilities, we solve here stabilized formulations cast in the Variational Multiscale (VMS) framework, that enhance the stability of the Galerkin method via a series of additional integrals over element interior. The basic idea is to split all quantities into coarse and fine scale components, corresponding to different levels of resolution, and to approximate the effect of the fine scale (that cannot be resolved by the finite element mesh) onto the coarse scale via consistently derived residual based terms.

4.2.1. Navier–Stokes equations

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

$$\int_{\Omega} (\rho \partial_t \mathbf{u} + \rho \mathbf{u} \cdot \nabla \mathbf{u}) \cdot \mathbf{w} \, dv + \int_{\Omega} 2\mu \boldsymbol{\varepsilon}(\mathbf{u}) : \boldsymbol{\varepsilon}(\mathbf{w}) \, dv - \int_{\Omega} p(\nabla \cdot \mathbf{w}) \, dv + \int_{\Omega} (\nabla \cdot \mathbf{u}) q \, dv - \sum_{k=1}^{N_e} \int_{\Omega_k} \tau_1 \mathbf{r}_1 \cdot (\rho \mathbf{u} \cdot \nabla \mathbf{w}) \, dv - \sum_{k=1}^{N_e} \int_{\Omega_k} \tau_1 \mathbf{r}_1 \cdot \nabla q \, dv - \sum_{k=1}^{N_e} \int_{\Omega_k} \tau_2 r_2 (\nabla \cdot \mathbf{w}) \, dv = 0, \quad (31)$$

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

$$-\mathbf{r}_1 = \rho \partial_t \mathbf{u} + \rho \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p, \quad -r_2 = \nabla \cdot \mathbf{u}, \quad (32)$$

whose second derivatives vanish since we use linear interpolation functions. Finally, τ_1 and τ_2 are ad-hoc stabilization coefficients, computed on each element after [37,40] as

$$\tau_1 = \frac{1}{\rho(\tau_t^2(u) + \tau_d^2)^{1/2}}, \quad \tau_2 = \frac{h^2}{\tau_1}, \quad (33)$$

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

$$\tau_t(u) = c_t \frac{u}{h}, \quad \tau_d = c_d \frac{\mu}{\rho h^2}. \quad (34)$$

Here, u is a characteristic norm of the velocity on the element, computed as the average L^2 norm of the nodal element velocities. Also, h is the element size, computed as its diameter in the direction of the velocity to support using anisotropic meshes with highly stretched elements [41], and $c_{t,d}$ are algorithmic constants taken as $c_t = 2$ and $c_d = 4$ for linear elements [40]. Equation (31) is discretized with a first-order-accurate time-integration scheme combining semi-implicit treatment of the convection term, implicit treatment of the viscous, pressure and divergence terms, and explicit treatment of the stabilization coefficients. All linear systems are preconditioned with a block Jacobi method supplemented by an incomplete LU factorization, and solved with the GMRES iterative algorithm, with tolerance threshold set to 10^{-6} .

4.2.2. Adjoint Navier–Stokes equations

Application of the stabilized formulation, as described above, to the adjoint Navier–Stokes equations yields the following coarse scale variational problem

$$\begin{aligned} & \int_{\Omega} (-\rho \mathbf{u} \cdot \nabla \tilde{\mathbf{u}} + \rho \nabla \mathbf{u}^T \cdot \tilde{\mathbf{u}}) \cdot \mathbf{w} \, dv + \int_{\Omega} 2\mu \boldsymbol{\varepsilon}(\tilde{\mathbf{u}}) : \boldsymbol{\varepsilon}(\mathbf{w}) \, dv + \int_{\Omega} \tilde{p}(\nabla \cdot \mathbf{w}) \, dv + \int_{\Omega} (\nabla \cdot \tilde{\mathbf{u}}) q \, dv \\ & - \sum_{k=1}^{N_e} \int_{\Omega_k} \tilde{\tau}_1 \tilde{\mathbf{r}}_1 \cdot (-\rho \mathbf{u} \cdot \nabla \mathbf{w}) \, dv - \sum_{k=1}^{N_e} \int_{\Omega_k} \tilde{\tau}_1 \tilde{\mathbf{r}}_1 \cdot \nabla q \, dv - \sum_{k=1}^{N_e} \int_{\Omega_k} \tilde{\tau}_2 \tilde{r}_2 (\nabla \cdot \mathbf{w}) \, dv \\ & - \int_{\Gamma_o} \rho(\mathbf{u} \cdot \mathbf{n})(\tilde{\mathbf{u}} \cdot \mathbf{w}) \, ds = \int_{\Gamma_o} \partial_{\mathbf{u}} J \cdot \mathbf{w} \, ds. \end{aligned} \quad (35)$$

The associated momentum and continuity residuals read

$$-\tilde{\mathbf{r}}_1 = -\rho \mathbf{u} \cdot \nabla \tilde{\mathbf{u}} + \rho \nabla \mathbf{u}^T \cdot \tilde{\mathbf{u}} - \nabla \tilde{p}, \quad -\tilde{r}_2 = \nabla \cdot \tilde{\mathbf{u}}, \quad (36)$$

and the stabilization coefficients are computed on each element after [42] as

$$\tilde{\tau}_1 = \frac{1}{(\tau_t^2(u) + \tau_d^2 + \tau_r^2)^{1/2}}, \quad \tilde{\tau}_2 = \tau_2, \quad (37)$$

Note, τ_r is an additional component corresponding to the reaction-dominated limit, which stems from the $\rho \nabla \mathbf{u}^T \cdot \tilde{\mathbf{u}}$ term describing the production of adjoint perturbations. It is defined as

$$\tau_r = \rho \nabla u, \quad (38)$$

where ∇u is a characteristic norm of $\nabla \mathbf{u}$ on the element, computed as the average L^2 norm of the nodal velocity gradients. It is important to note that the adjoint stabilization coefficients depend solely on \mathbf{u} , not $\tilde{\mathbf{u}}$, which is because the adjoint flow field is transported at (minus) the state velocity. Note also, Eq. (35) features boundary terms evaluated at the outlet, which is because the integration by part of the pressure and viscous terms unveils a boundary term

$$\int_{\partial\Omega} (\tilde{p} \mathbf{n} + 2\mu \boldsymbol{\varepsilon}(\tilde{\mathbf{u}}) \cdot \mathbf{n}) \cdot \mathbf{w} \, ds = - \int_{\Gamma_o} (\rho(\mathbf{u} \cdot \mathbf{n}) \tilde{\mathbf{u}} + \partial_{\mathbf{u}} J) \cdot \mathbf{w} \, ds, \quad (39)$$

due to the adjoint boundary condition (13). Equation (35) is fully implicitly integrated, except the outflow boundary term that needs be treated explicitly for implementation convenience. Even though the last computed adjoint solution (hence pertaining to the previous design) is used to evaluate the boundary term, this simple scheme has been found to converge to identical shapes and cost function minimum, compared to solving iteratively with relaxed sub-iterations. Due to the linearity of Eqs. (10)-(11), this in turn cuts down the numerical effort, as only one single linear system needs be solved at each update step, for which we use a BCGS iterative algorithm with tolerance threshold set to 10^{-12} and LU factorization as preconditioner.

4.2.3. Interface update scheme using the convective level set method

The auto-reinitialization level set problem (19) is solved with an SUPG method [43,44], whose stabilization proceeds from that of the ubiquitous convection-diffusion-reaction equation [45,46]. The associated variational problem is formulated as

$$\int_{\Omega} (\partial_{\tau} \phi + \mathbf{a}_{\tau} \cdot \nabla \phi) \xi \, dv - \int_{\Omega_k} \tau_3 r_3 \mathbf{a}_{\tau} \cdot \nabla \xi \, dv = \int_{\Omega} S \xi \, dv, \quad (40)$$

with residual

$$-r_3 = \partial_{\tau} \phi + \mathbf{a}_{\tau} \cdot \nabla \phi - S, \quad (41)$$

and stabilization coefficient

$$\tau_3 = \frac{1}{\tau_t(a_\tau)}. \quad (42)$$

It is easily checked that all terms scale as $1/\Delta\tau$, so we can set $\Delta\tau = 1$ without any loss of generality because the solution is ultimately independent on the pseudo-time step value. As the convection velocity \mathbf{a}_τ depends on main unknown ϕ , Eq. (40) is solved with semi-implicit treatment of the convection term and explicit treatment of the source term and stabilization coefficients. All linear systems are solved using the GMRES algorithm with incomplete LU factorization as preconditioner, and tolerance threshold set to 10^{-8} .

5. Numerical implementation

5.1. Geometrical constraints

Fluid flow topology optimization is generally performed under geometrical constraints, typically, constant or upper bounded surfaces and/or volumes to avoid the two extreme cases of the solid domain clogging the entire design domain (as in pressure drop minimization problems), or disappearing altogether (as in drag minimization problems). This is usually done adding penalty terms to the Lagrangian, each of which consists of an empirical penalty parameter multiplied by a measure of violation of the constraint, and whose variations with respect to the state and design variables snowballs into the derivation of the adjoint problem and of the cost function sensitivity. Here, the constraint of a constant volume of fluid V_{target} is applied a posteriori, i.e., we solve the unconstrained problem presented in Sec. 2, in the sense that no penalty term is added to the Lagrangian, although the optimization remains subject to Navier–Stokes as state equations. Once the convective level set method presented in Sec. 4.2.3 has updated the interface position, a first pass of anisotropic mesh adaptation is performed, after which the volume of the fluid domain is computed as

$$V_\varphi = \int_{\Omega} H_\epsilon(\varphi) \, dv, \quad (43)$$

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

$$H_\epsilon(\varphi) = \begin{cases} 1 & \text{if } \varphi < -\epsilon, \\ \frac{1}{2} \left(1 - \frac{\varphi}{\epsilon} - \frac{1}{\pi} \sin\left(\pi \frac{\varphi}{\epsilon}\right) \right) & \text{if } |\varphi| \leq \epsilon, \\ 0 & \text{if } \varphi > \epsilon, \end{cases} \quad (44)$$

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

5.2. Steepest descent update rule

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

$$\beta = -\theta \frac{\mu(\nabla \tilde{\mathbf{u}} \cdot \mathbf{n}) \cdot (\nabla \mathbf{u} \cdot \mathbf{n}) \chi_{\Gamma}(\mathbf{x})}{\max_{\Omega} \mu(\nabla \tilde{\mathbf{u}} \cdot \mathbf{n}) \cdot (\nabla \mathbf{u} \cdot \mathbf{n}) \chi_{\Gamma}(\mathbf{x}) \prod_l \zeta(\|\mathbf{x} - \mathbf{x}_s^l\|)}, \quad (45)$$

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

- χ_{Γ} is a binary filter returning a value of 1 only at nodes within a distance E of the interface. 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 (for instance the centerline of a channel).
- ζ is a smooth filter assigning 0 value to a position $\mathbf{x}_s \in \partial\Omega$ 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

$$\zeta(r) = \frac{1}{2} + \frac{1}{2} \tanh\left(\alpha_s \tan\left(-\frac{\pi}{2} + \frac{\pi}{2} \frac{r}{r_s + \epsilon_{s1}} + \epsilon_{s2}\right)\right), \quad (46)$$

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

$$4r_s < \min_{l,m} \|\mathbf{x}_s^l - \mathbf{x}_s^m\|, \quad (47)$$

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 spheres of radius r_s centered on the singularities.

5.3. Descent factor

It follows from Eq. (45) 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 in the solid domain. Hence, the displacement, being driven by the velocity gradients, is 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, but 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

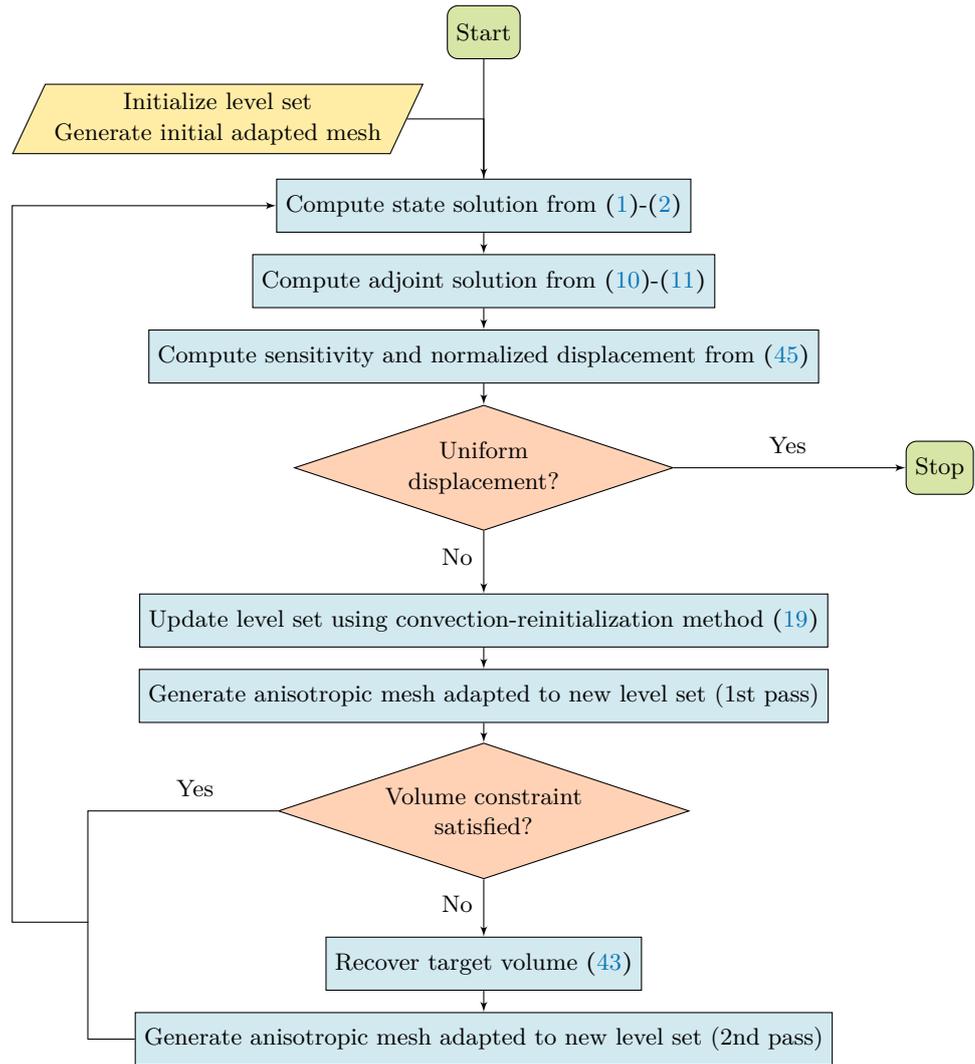


Figure 3. Flowchart of performance topology optimization procedure.

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 very 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.

5.4. 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. Note, as a consequence of the level set-based technique used to enforce the volume of fluid constraint, convergence is achieved not when the displacement is identically zero (as would be the case using a penalized Lagrangian approach), but when the displacement is uniform along the interface. This is not easily done on the fly, though, so we rather iterate until a maximum number of iterations has been reached and evaluate convergence a posteriori; see Sec. 7.

6. Numerical benchmarks

This section assesses the accuracy and efficiency of the numerical framework through three examples of two-dimensional ($d = 2$) topology optimization problems recently consid-

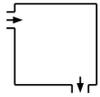
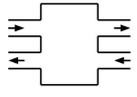
			
$\Omega = [0; 1] \times [0; 1]$	$[0; 0.7] \times [0; 1]$	$[0; 1.5] \times [0; 1]$	Design domain
$V_{target} = 0.25$	0.28	0.5	Target volume of fluid
$Re = 2$	»	»	Reynolds number
$q_i = 0.0266$	»	0.0222	Injected volumetric flow rate
$e_i = 0.2$	»	0.166	Inlet width
$e_o = 0.2$	»	0.166	Outlet width
$l_c = 0.1$	0.4	0.1	Conveying pipes length
$N_n = 30000$	»	40000	Nb. mesh nodes
$N_{el} = 60000$	»	80000	Nb. mesh elements
$h_{\perp} = 0.0001$	»	»	Min. interface normal mesh size
$\Delta t = 0.1$	»	»	CFD Numerical time step
$E = 0.005$	»	»	Level set cut off thickness
$ \delta\varphi = 0.001$	»	»	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. Numerical parameters for the pipe bend, four terminal device and double pipe topology optimization problems.

ered in the fluid mechanics literature. It is thus worth insisting that the novelty lies not in the associated optimal designs themselves, but in the accuracy to which the optimal interfaces are captured in the simulation model.

6.1. Preliminaries

All examples feature either a single inlet or multiple identical inlets of width e_i , and either a single outlet, or multiple identical outlets of width e_o . Parabolic flow profiles normal to the boundary are prescribed at all inlets, as defined by

$$\mathbf{u}_i = \frac{3q_i}{2e_i} \left(1 - \left(\frac{2r}{e_i} \right)^2 \right) \mathbf{n}, \quad (48)$$

where q_i is the injected volumetric flow rate (the same for all inlets), and r is the distance from the inlet centerline. For each case, the sole control parameter is the Reynolds number defined as $Re = \rho q_i / \mu$, which amounts to using the inlet width and mean inlet velocity as reference length and velocity scales. The cost function to minimize is the net inward flux of total pressure through the boundaries, taken as a measure of the total power dissipated by a fluid dynamic device. Since the 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 of (6) using

$$J = p_{tot}(\mathbf{u} \cdot \mathbf{n}) = \left(p + \frac{1}{2} \rho (\mathbf{u} \cdot \mathbf{u}) \right) (\mathbf{u} \cdot \mathbf{n}), \quad (49)$$

from which the derivatives needed to complete the derivation of the adjoint boundary conditions deduce as

$$\partial_p J = \mathbf{u} \cdot \mathbf{n}, \quad \partial_{\mathbf{u}} J = p_{tot} \mathbf{n} + \rho (\mathbf{u} \cdot \mathbf{n}) \mathbf{u}. \quad (50)$$

The remainder of the practical implementation details are as follows:

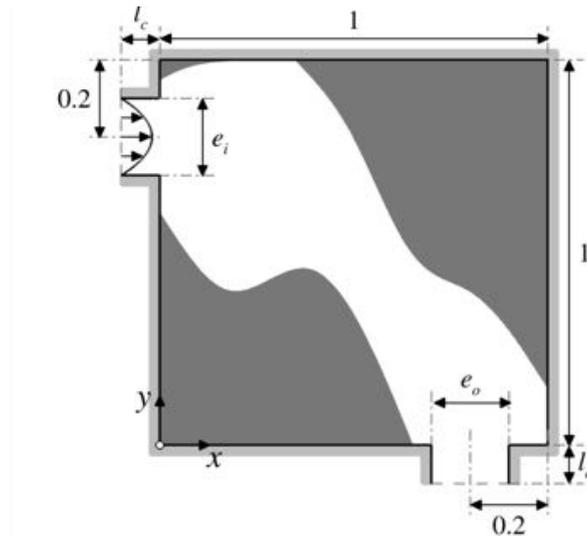


Figure 4. Set-up of the pipe bend problem. The light gray shade denotes parts of the boundary where solid boundary conditions are appended to level set auto-reinitialization equation.

- All design domains are initialized with spherical solid inclusions coming in various sizes, adjusted for the initial volume² of fluid to match the target within the desired tolerance. This essentially removes the need to create new holes by a dedicated nucleation mechanism. The admissible error on the target volume is set to 1%. 489-492
- Leads of length l_c appended normal to the boundary are used to systematically convey the fluid into and out of the design domain. This is for numerical consistence, as the exact problem formulation may vary depending on the case, the reference and the problem dimensionality, and it is not always clear whether such leads are included in the design domain (which they are here, although they are not considered in the volume constraint, neither in definition of the target volume nor in the computation of the volume of fluid). 493-499
- Since the reference design domains (without the leads) consist of square and rectangular cavities, the singular points excluded from the displacement normalization step are the sharp intersections between the leads and the boundary of the cavities (without it being a consequence of explicitly representing the leads, as the exact same procedure has been found suitable without such appendage). 500-504
- The leads are excluded from the displacement normalization step, for which we simply add to the max argument of (45) 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). 505-510
- Boundary conditions are appended to the auto-reinitialization level set equation, under the form of fluid at the inlet and outlet, and solid everywhere else. 511-512
- All meshes have been checked to have an element-to-node ratio close to 2 (as should be for dense meshes made up of triangular elements). The mesh information is thus documented in the following in terms of its equivalent number of elements $N_{el} = 2N_n$ to ease the comparison with the available literature. 513-516

6.2. Design of a pipe bend 517

We consider first the design of a pipe bend, a standard example for topology optimization in fluid dynamics [14,17,47–50] used to provide a first verification and characterization of 518-519

² Actually cross-sectional area or volume per unit length in the third dimension since $d = 2$, but we choose to keep the volume terminology for the sake of generality

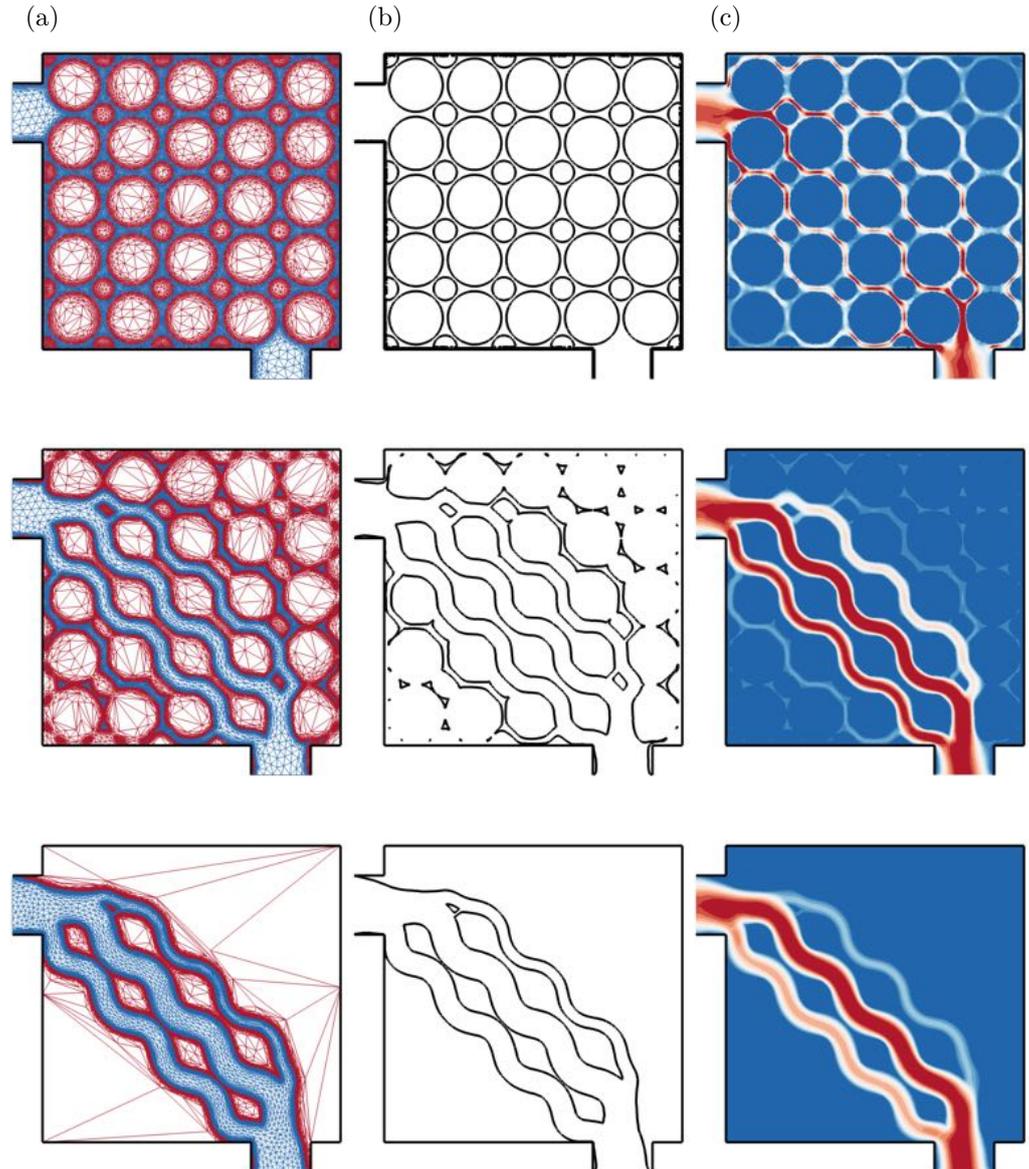


Figure 5. Designs of a pipe bend sampled over the course of optimization using the parameters given in Tab. 1. (a) Anisotropic adapted mesh. (b) Zero iso-value of the level set function. (c) Norm of the velocity vector.

the method. All relevant problem parameters are given in Tab. 1. The design domain is a square cavity of unit length, that has one inlet (left side) and one outlet (bottom side); see Fig. 4. 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 25% of the cavity, which is the same volume as a quarter torus fitting exactly to the inlet and outlet.

A total of 400 iterations has been run with 60000 mesh elements, as illustrated in Fig. 5 by the anisotropic adapted mesh, zero iso-value of the level set function and velocity norm of a selected sample. The method is found to easily handle the multiple topological changes (e.g., merging or cancellation of holes) occurring over the course of optimization. Also, consistently with the results in Sec. 3, all meshes exhibit the expected refinement and deformation, with coarse and regular elements away from the interface between solid and fluid (all the more so in the solid domain, where only a few ten elements are used), but fine, extremely stretched elements on either side of the interface, to allow the velocity to smoothly transition to zero across the boundary layer; see the close-up in Fig. 6. In

520
521
522
523
524
525
526
527
528
529
530
531
532
533
534

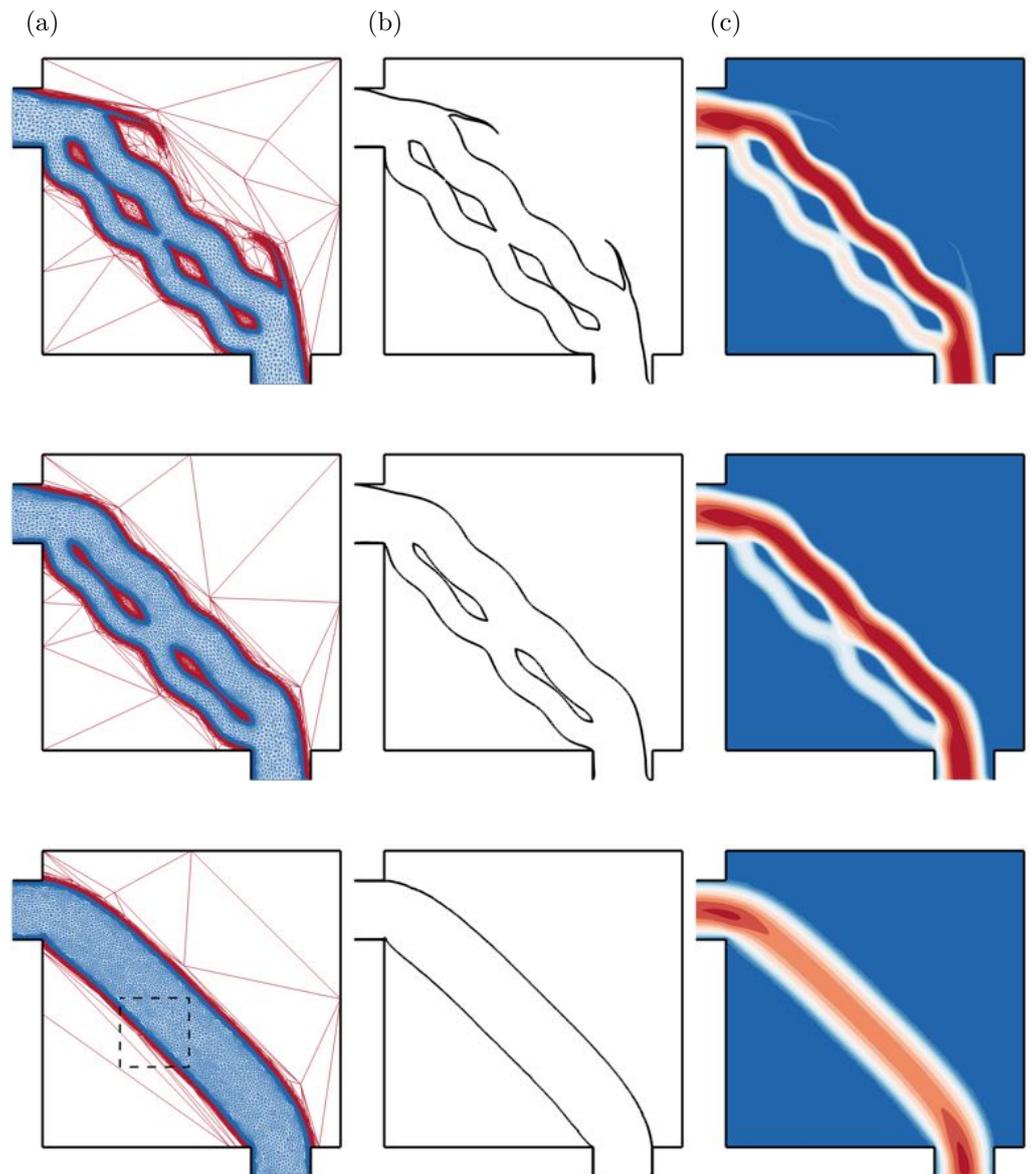


Figure 5. (cont.) Designs of a pipe bend sampled over the course of optimization using the parameters given in Tab. 1. (a) Anisotropic adapted mesh. (b) Zero iso-value of the level set function. (c) Norm of the velocity vector. The optimal pipe bend is shown at the bottom.

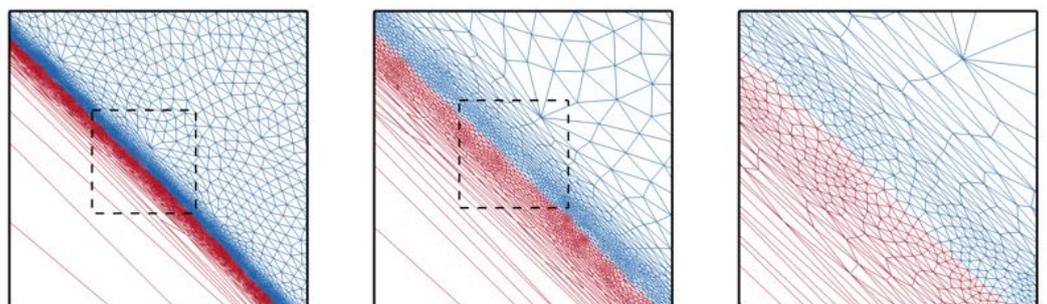


Figure 6. Successive close-ups on the anisotropic adapted mesh of the optimal pipe bend, starting from the dashed box shown in Fig. 5.

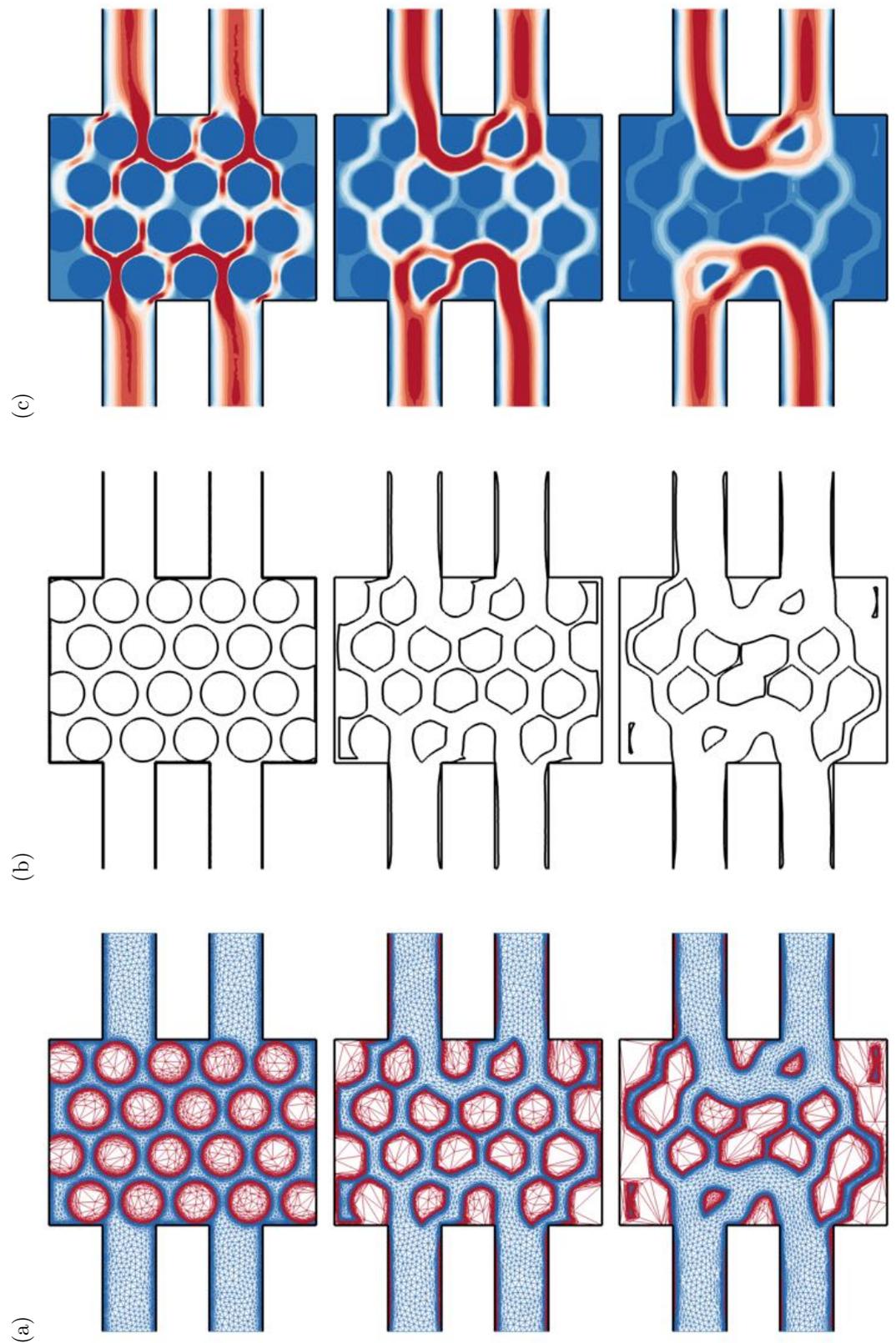


Figure 7. Designs of a four terminal device sampled over the course of optimization using the parameters given in Tab. 1. (a) Anisotropic adapted mesh. (b) Zero iso-value of the level set function. (c) Norm of the velocity vector.

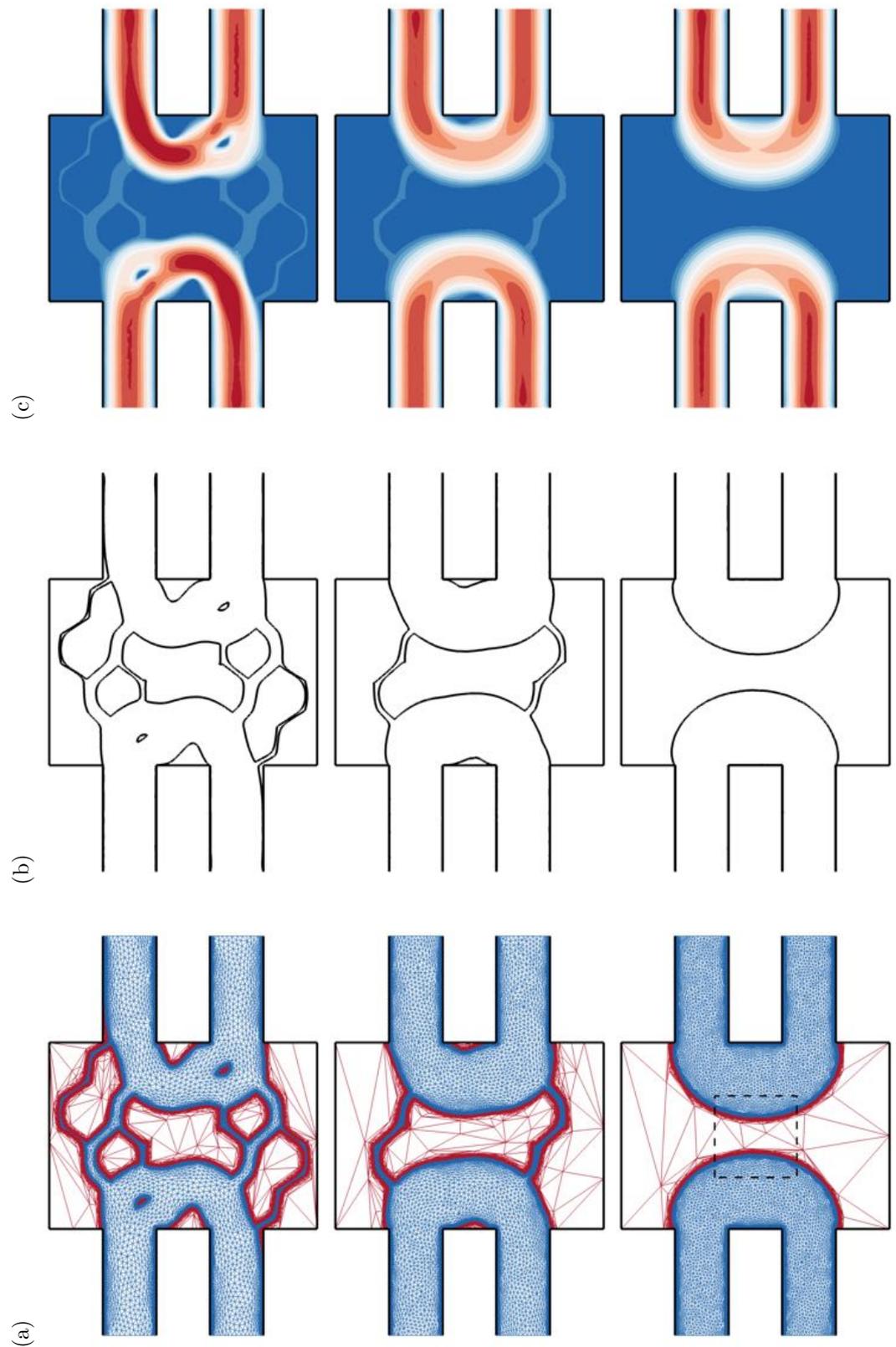


Figure 7. (cont.) Designs of a four terminal device sampled over the course of optimization using the parameters given in Tab. 1. (a) Anisotropic adapted mesh. (b) Zero iso-value of the level set function. (c) Norm of the velocity vector. The optimal four terminal device is shown at the bottom.

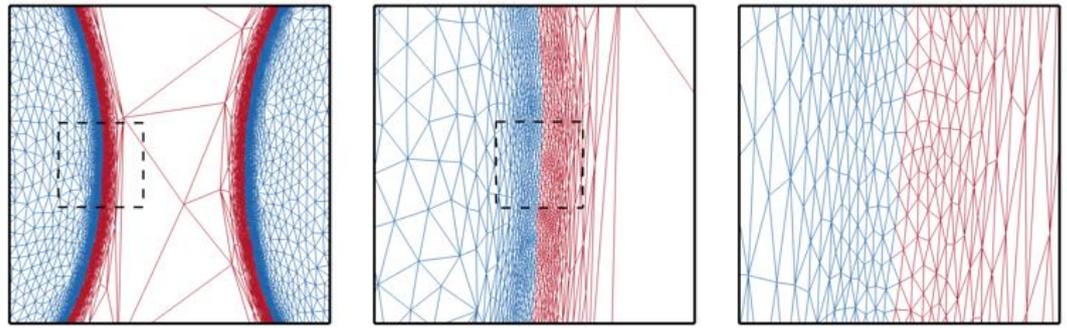


Figure 8. Successive close-ups on the anisotropic adapted mesh of the optimal four terminal device, starting from the dashed box shown in Fig. 7.

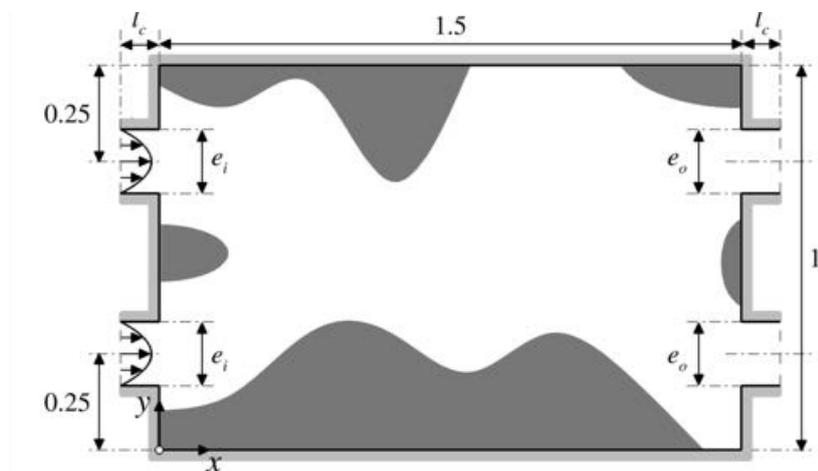


Figure 9. Set-up of the double pipe problem. The light gray shade denotes parts of the boundary where solid boundary conditions are appended to level set auto-reinitialization equation.

6.4. Design of a double pipe

In the third numerical example, we consider the double pipe problem, another benchmark for fluid topology optimization [47,50,52,53], whose parameters are provided in Tab. 1. The design domain is a rectangular cavity of unit height and aspect ratio 3:2, that has two inlets (left side) and two outlets (right side); see Fig. 9. The aim is to determine the optimal design of the double pipe that connects the inlets to the outlets and minimizes the dissipated power subject to the constraint that the fluid must occupy 33.3% of the cavity, which is the same volume as two straight parallel pipes fitting the upper and lower pairs of inlet/outlet.

A total of 3000 iterations has been run with 80000 mesh elements (due to the larger design domain), during which the design goes through several complex stages all accurately represented on anisotropic adapted meshes, as evidenced by the selected sample shown in Figs. 10-11. Ultimately, the optimal design resembles a single-ended wrench, with the two inlet pipes connecting to a wider pipe in the center of the domain, that itself connects to a single outlet (either the upper or the lower outlet since the setup has horizontal reflectional symmetry). Since the optimal flow pipe at low Reynolds numbers is preferably short and wide, this represents the better trade-off between transporting fluid the shortest way, and transporting it in the widest possible pipe. Note, the obtained solution differs from the double-ended wrench documented in [47,50,52], in which the center pipe ultimately connects to the two outlet. This is because the authors prescribe parabolic flow profiles at both the inlets and the outlets. The flow is thus forced to exit via both outlets, while it can exit via a single outlet under the more physical zero pressure/zero viscous stress condition

567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588

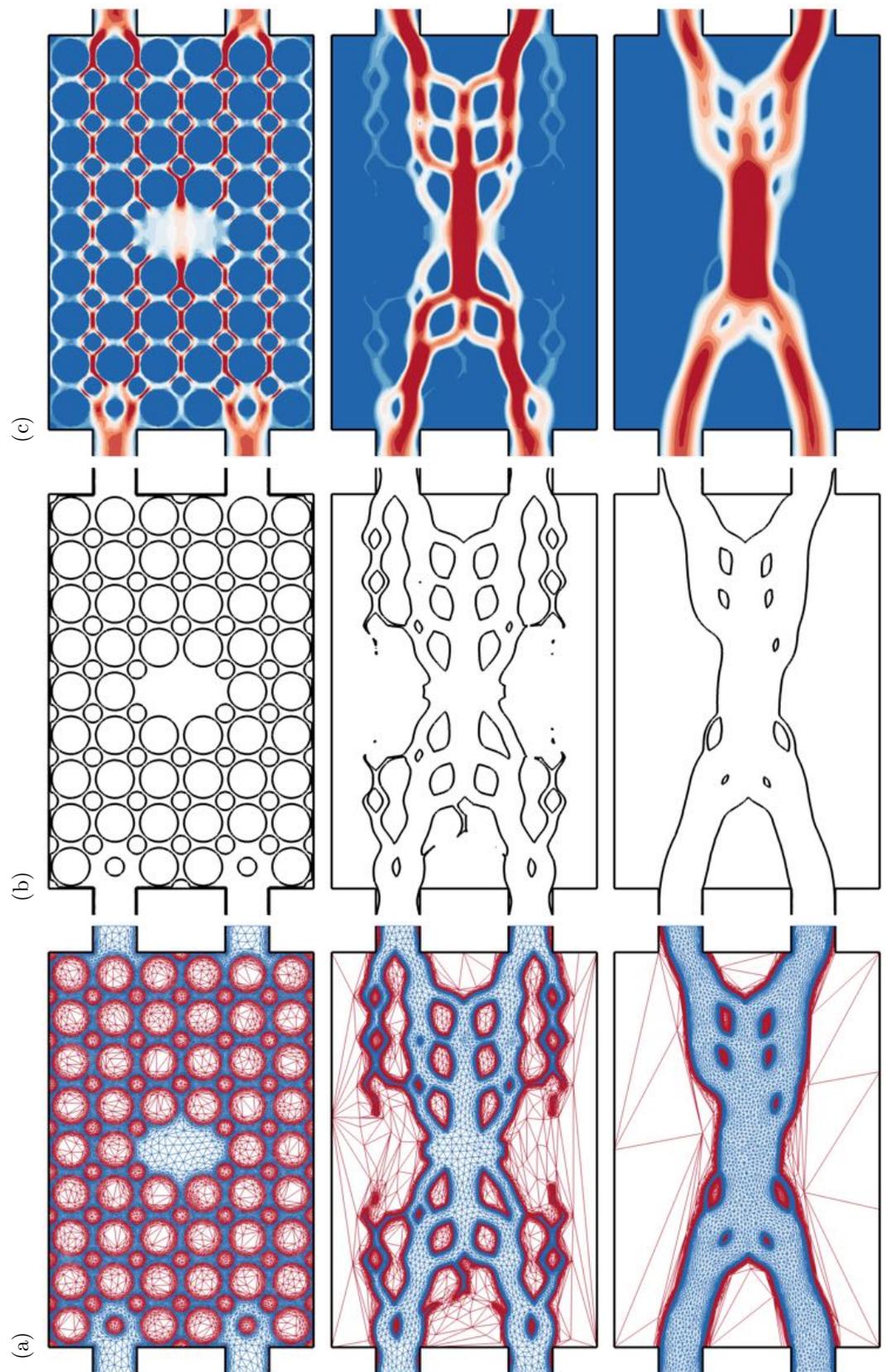


Figure 10. Designs of a double pipe sampled over the course of optimization using the parameters given in Tab. 1. (a) Anisotropic adapted mesh. (b) Zero iso-value of the level set function. (c) Norm of the velocity vector. The optimal double pipe is shown at the bottom.

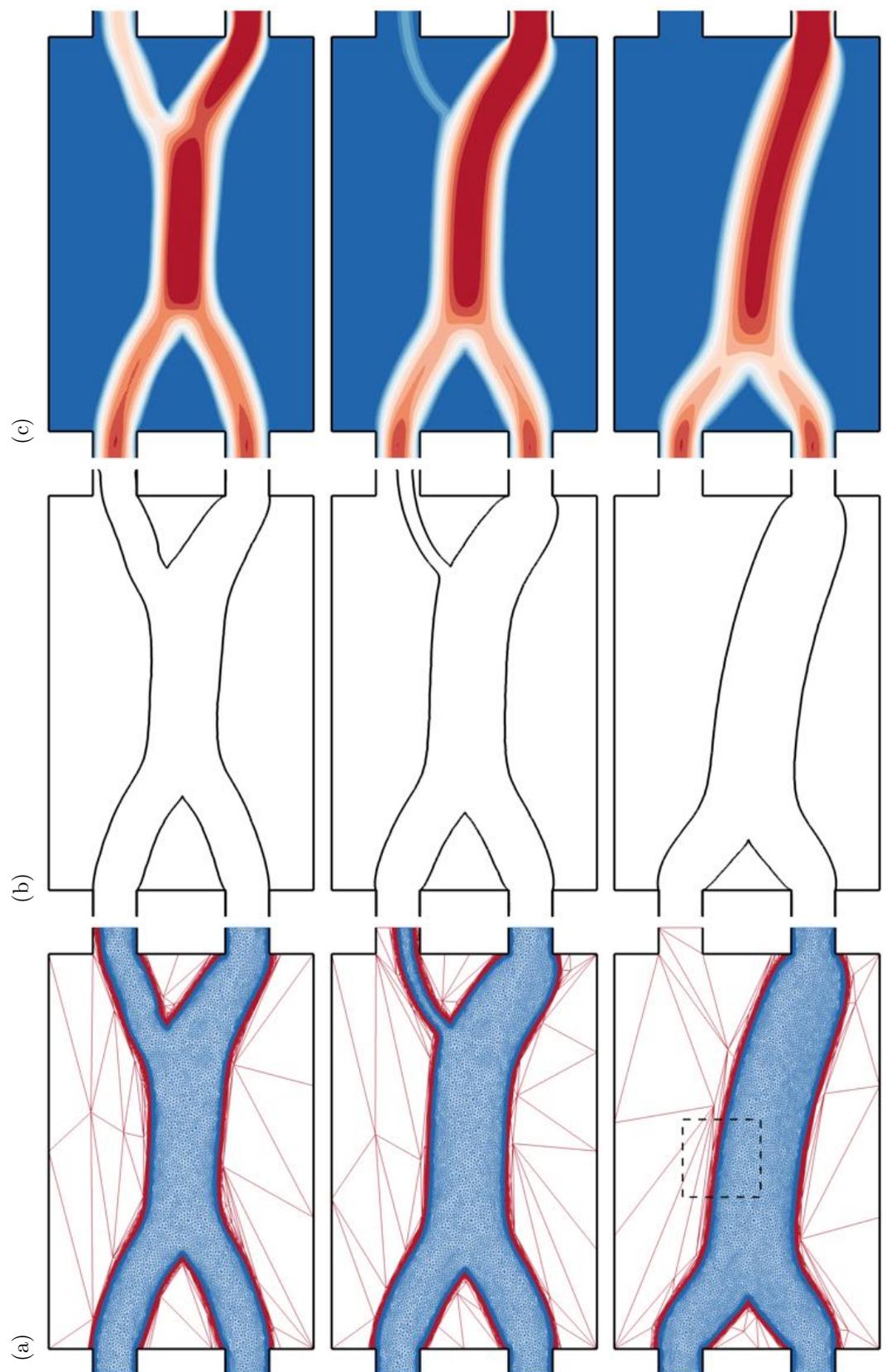


Figure 10. (cont.) Designs of a double pipe sampled over the course of optimization using the parameters given in Tab. 1. (a) Anisotropic adapted mesh. (b) Zero iso-value of the level set function. (c) Norm of the velocity vector. The optimal double pipe is shown at the bottom.

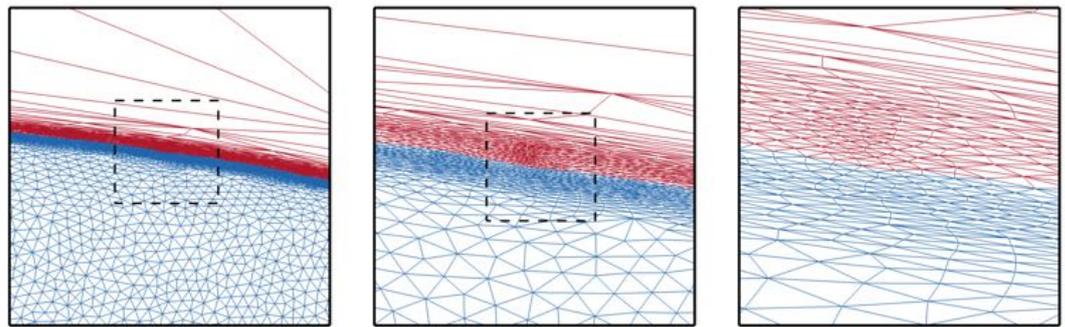


Figure 11. Successive close-ups on the anisotropic adapted mesh of the optimal four terminal device, starting from the dashed box shown in Fig. 10.

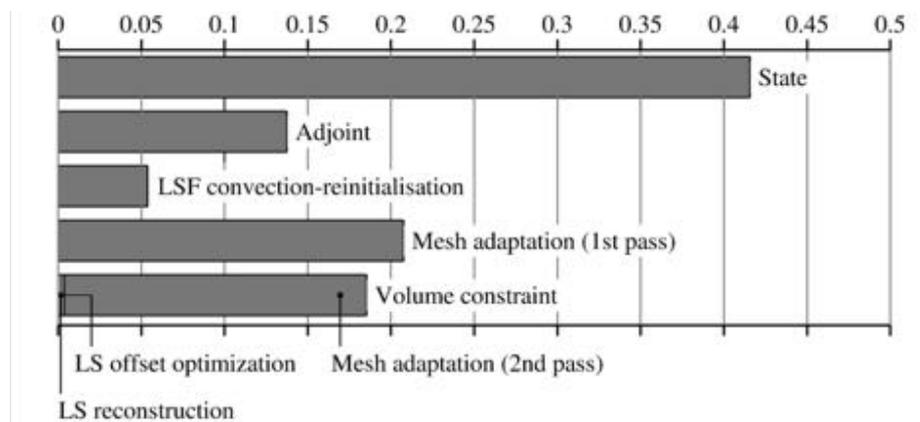


Figure 12. Computational cost of the implemented algorithm, as obtained averaging 300 update steps of the pipe bend, double pipe, and four terminal device problems (100 steps for each) using the simulation parameters provided in Tab. 1. All results normalized to achieve unit average time per iteration. The LS and LSF labels stand for level set (LS) and filtered level set (LSF), respectively.

used here, which allows saving the cost of pipe splitting [53]. The number of iterations for this case is larger by one order of magnitude compared to the pipe bend and four terminal problems, which is easily explained by the fact that the optimization must bypass the basin of attraction of the double-ended wrench, that keeps being a local minimizer. This is all the more difficult because the cost function of both minimizers differs by only 10%, but we show in Sec 7 that this particular feature is ultimately very sensitive to the number of nodes used to perform the mesh adaptation.

7. Discussion

7.1. Computational efficiency

Figure 12 presents detailed timing results obtained by averaging 300 dedicated update steps performed with the parameters compiled in Tab. 1; 100 steps for each case presented in Secs. 6.2-6.4. As could have been expected, the cost of an iteration is dominated by that of computing the state solution. This takes about 10 Navier–Stokes iterations representing 40% of the total cost, which can be scaled down substantially in the context of steady-state problems using an iterative Newton-like method. Otherwise, the cost of performing the two passes of mesh adaptation represents about a cumulative 40% of the total cost. Meanwhile, the cost of both geometrically reinitializing the signed distance function level set and of optimizing the volume constraint offset is very affordable (less than 1% in total, with 4-5 dichotomy iterations needed to reach the desired accuracy of 1%). Such conclusions presumably carry over to any other problem of same dimensionality, tackled with comparable parameters.

Convergence iter.	Cost function	Nb. mesh elements	
295	33.1	80000	
306	32.7	60000	
212	32.9	40000	
148	32.1	20000	
153	68.9	80000	
129	69.1	60000	
104	69.0	40000	
68	68.6	20000	
2460	68.6	105000	
1750	67.6	80000	
2130	68.2	55000	
1594	67.0	25000	

Table 2. Convergence data for the pipe bend, four terminal device and double pipe topology optimization problems. All cost function values made non dimensional using the inlet width and mean inlet velocity (equivalently, using $\rho q_i^3 / e_i^2$ as reference cost functional value).

7.2. Convergence and mesh dependency

Since we perform here a fixed number of iterations, convergence is assumed here when the sliding average over the 10 latest cost functional values is less than a prescribed error set to 2% of the cost functional average over the 50 final iterations. The reason is twofold: first, the cost function keeps varying even after convergence because the mesh slightly changes between consecutive iterations, and so does the volume of fluid as long as the deviation from the target does not exceed the admissible error. Second, assuming convergence simply when the relative difference between two successive cost functional values is less than a prescribed error has been found to yield premature convergence to the double-ended wrench local minimizer of the double pipe problem. Note, all data discussed in the following pertain to a single optimization run. Rigorously speaking, convergence is best assessed by averaging results over multiple independent runs, as mesh adaptation is not a deterministic process (the outcome depends on the processors and number of processors used, and any initial difference propagates over the course of optimization because the meshes keep being adapted at each iteration), but we have found very little variability by doing so.

Exhaustive convergence data are provided in Tab. 2 for all three cases reported above. Putting the obtained results in a broader context is uneasy because convergence is rarely documented in the literature, and even when it is, the key factors explicitly affecting convergence (e.g., initial shape, convergence criterion and threshold) are not. In practice, our literature review did not reveal any other study putting all these levels of information together. Here, the bend pipe problem converges within 306 iterations, which is well above the convergence iteration reported in the seminal paper by Borrvall & Peterson [47] that lies in a range from 64 (using 2500 mesh elements) to 85 (using 10000 mesh elements). A first explanation is that all designs in the aforementioned reference are evaluated on the same isotropic mesh, hence the descent factor is not constrained by the thickness of the level set, and larger values can be used to speed up convergence. Another possibility further discussed below is that most studies in the literature rely on a limited number of elements in a range from 5000 to 20000. We conversely use a much larger value, which is on purposes to equally assess all steps of the optimization, but ultimately slows down the convergence rate.

A first important point is that such a large number of nodes is mostly useful during the early stage of optimization, where the many solid inclusions dramatically increase the surface of the interfaces that needs be captured. In practice, the latter has been found to

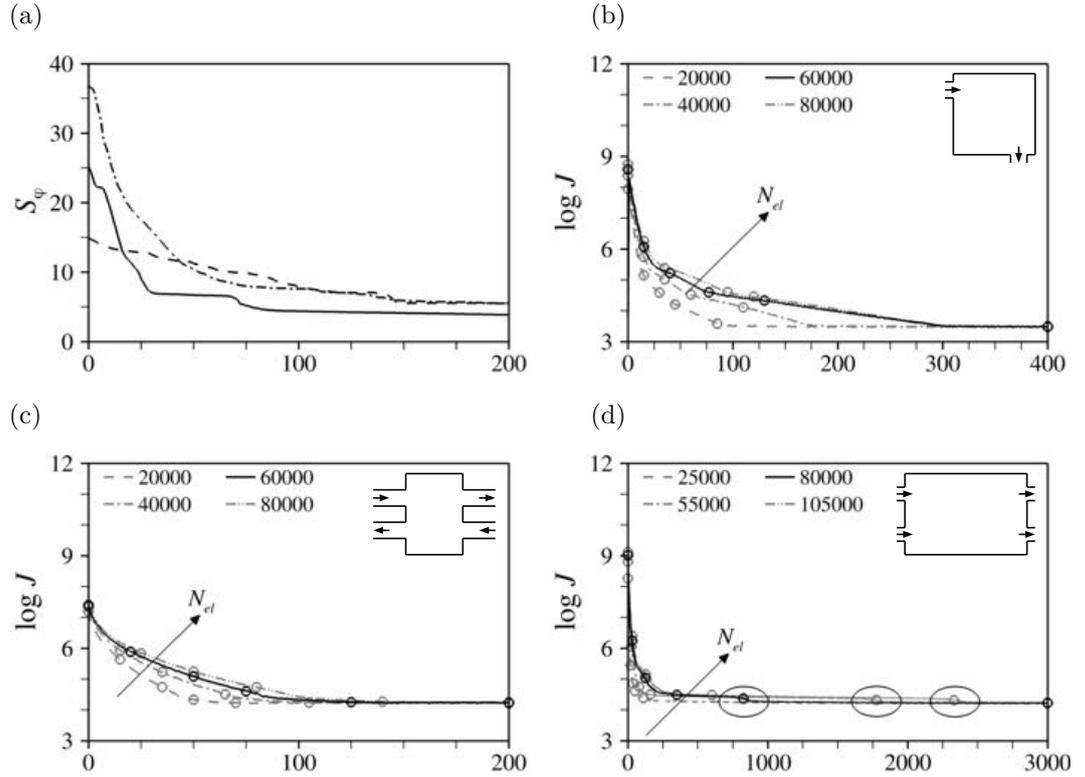


Figure 13. (a) Evolution per design step of the Interface surface area for the bend pipe (—), four terminal device (---) and double pipe (-.-.) problems. (b-d) Convergence against number of mesh elements for (b) the bend pipe, (c) four terminal device and (d) double pipe problems. The circle symbols mark the iterations sampled in (b) Figs. 5-14, (c) Figs. 7-15, and (d) Figs. 10-16. All cost function values made non dimensional using the inlet width and mean inlet velocity (equivalently, using $\rho q_i^3 / e_i^2$ as reference cost functional value). The ellipses in (d) indicate the transition from the double to the single-ended wrench minimizer.

decrease significantly after the first dozens of iterations (by a factor of 3-10 depending on the case); see Fig. 13(a) showing the surface area computed over the first 200 iterations as

$$S_\varphi = \int_{\Omega} \delta_\epsilon(\varphi) dv, \quad (51)$$

where δ_ϵ is the Dirac function

$$\delta_\epsilon(\varphi) = \begin{cases} \frac{1}{2\epsilon} \left(1 + \cos\left(\pi \frac{\varphi}{\epsilon}\right) \right) & \text{if } |\varphi| \leq \epsilon, \\ 0 & \text{if } |\varphi| > \epsilon, \end{cases} \quad (52)$$

smoothed with the same regularization parameter ϵ as the Heaviside function (44). A second important point is that the anisotropic mesh adaptation algorithm refines the mesh in hierarchical importance of the level set gradient. If new geometrical features appear in the solution (associated with high gradients), the mesh is automatically coarsened in regions with lower gradient and refined near the newly emerging features. If the number of nodes is large, as has been the case so far, then the decrease in the interface surface area allows resolving finer, more complex patterns without degrading the accuracy in other parts of the design domain, because the coarsened regions are actually over-resolved. This shows through the progressive mesh refinement in the fluid domain in Figs. 5-10, as more and more elements become available to improve the mesh in other regions of the domain. If the number of nodes is small, all essential features of the solution will remain well captured (albeit to a slightly lower accuracy), but the finest, most intricate topologies will be smoothed out,

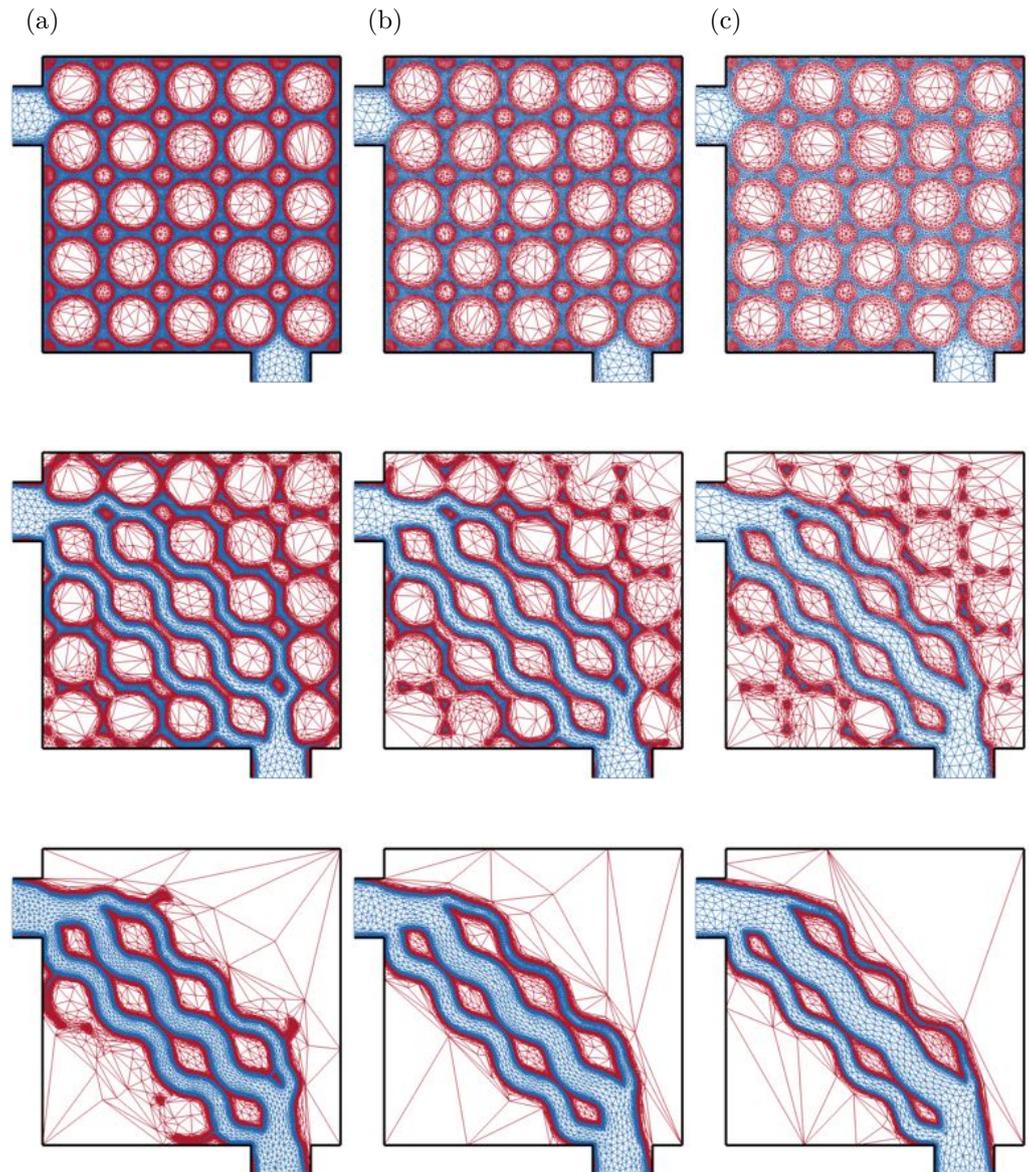


Figure 14. Anisotropic meshes of a pipe bend sampled over the course of optimization, using (a) 80000, (b) 40000 and (c) 20000 mesh elements.

which is expected to yield faster convergence because the sensitivity will have less overshoots and the displacement will be more homogeneously distributed over the interface.

Confirmation comes from additional runs performed on both denser and coarser meshes. The look-alike design samples documented in Figs. 14-16 indicate that all runs follow the same optimization path, with smaller details being captured as the number of elements increases. Just as important is the fact all optimal solutions are independent of the mesh size. This means that the ability of the method to represent smaller and smaller features does not result in smaller and smaller features being represented in the optimal designs, as can occur in stiffness optimization of mechanical structures [54]; see also [47] for proof that total power dissipation minimization is well posed in this respect. For the bend pipe and four terminal devices, the expected behavior is observed, as coarser-mesh runs converge substantially faster, for instance the bend pipe with 20000 elements converges within 165 iterations, which is lower by about 45% compared to using 60000 elements. If a less restrictive convergence threshold of 5%, is enforced, this drops to 102, which is only a tad above the 85 iterations of [47]. The improvement carries over to the four terminal device problem, whose run with 20000 elements converges within 68 iterations, which is lower by about 50% compared to

657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672

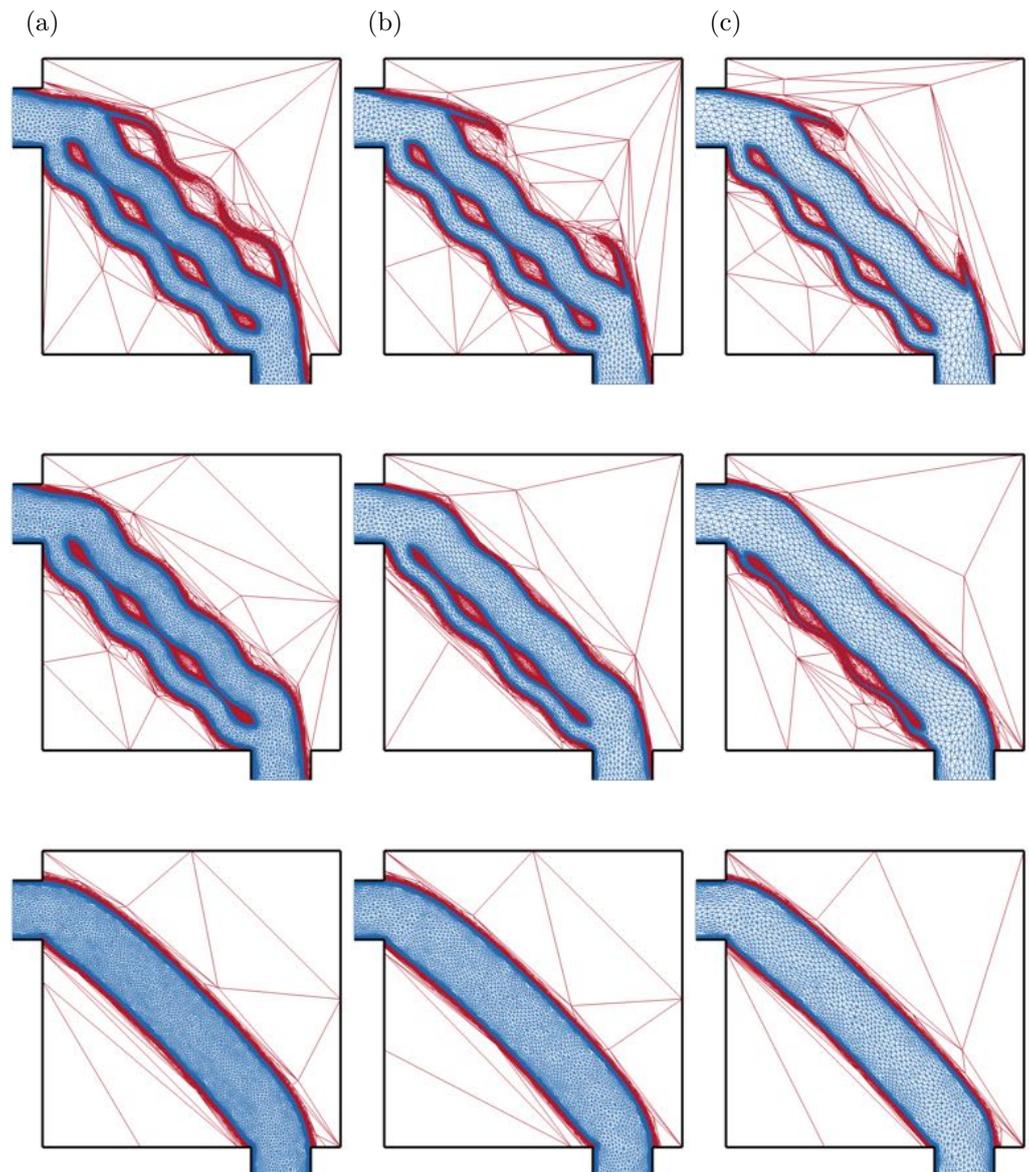


Figure 14. (cont.) Anisotropic meshes of a pipe bend sampled over the course of optimization, using (a) 80000, (b) 40000 and (c) 20000 mesh elements.

using 60000 elements (this further drops to 61 using a convergence threshold of 5%). Note, in both cases, coarser does not equate coarse, as the convergence information compiled in Tab. 2 shows that the coarsest meshes actually resolves the optimal interface to an excellent accuracy.

Meanwhile, convergence for the double pipe ends up being almost arbitrary and the algorithm has difficulties in finding the optimal topology due to the characteristics of the cost function landscape. The convergence history in Fig. 13(d) shows that the run with 55000 elements does indeed converge faster to the double-ended wrench solution minimizer but then needs more iterations to ultimately reach the single-ended wrench global minimizer, so convergence is ultimately slower than using 80000 elements. Interestingly, the run with 25000 elements successfully bypasses the local minimizer because the lack of elements does not allow representing the complexity prevailing in the early stage of the optimization. This ends up quickly breaking the horizontal reflectional symmetry, but the convergence rate ultimately remains comparable to that with 80000 elements, which raises the possibility that the wrench solutions are actually flat minimizers.

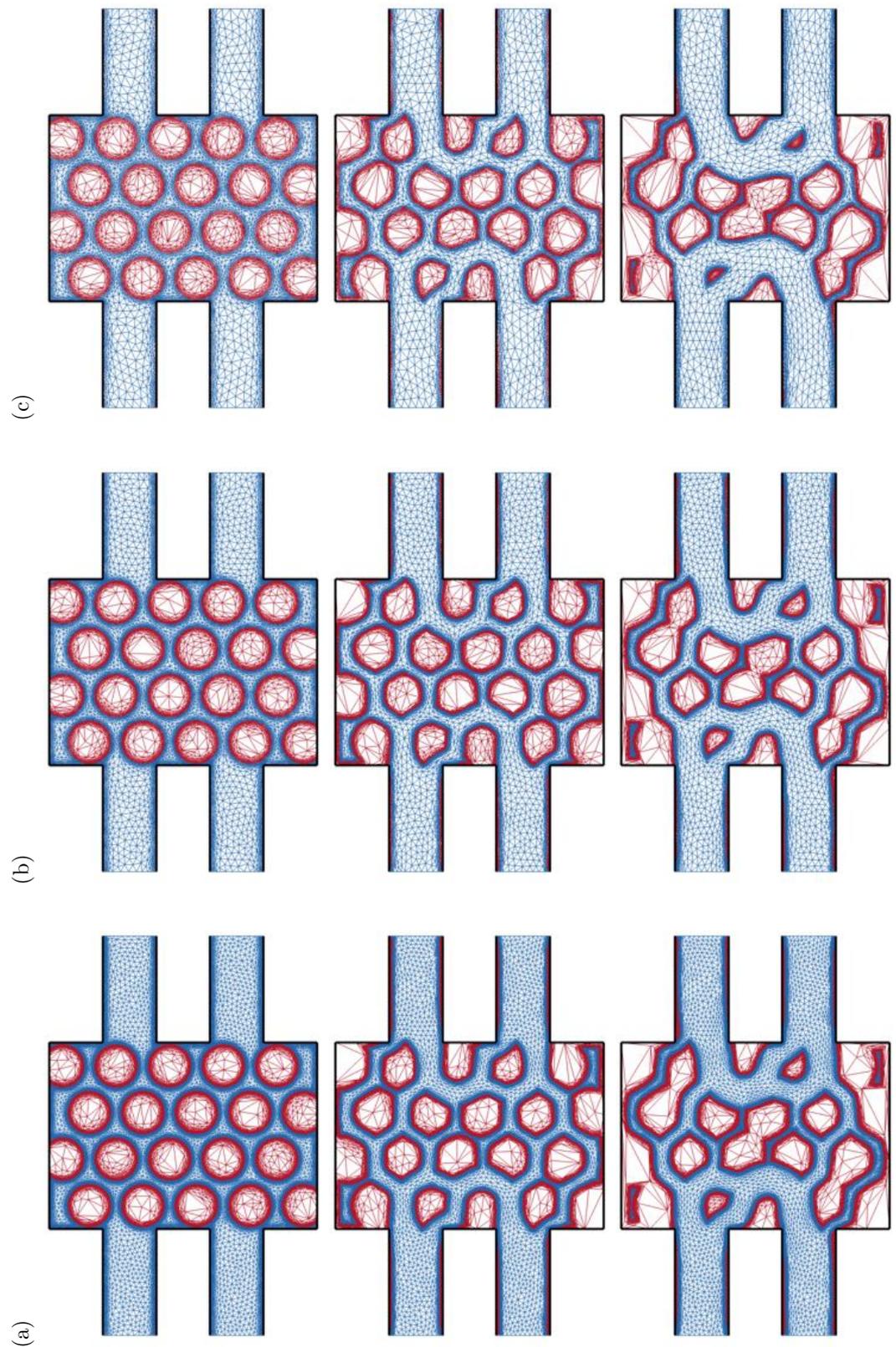


Figure 15. Anisotropic meshes of a four terminal device sampled over the course of optimization, using (a) 80000, (b) 40000 and (c) 20000 mesh elements.

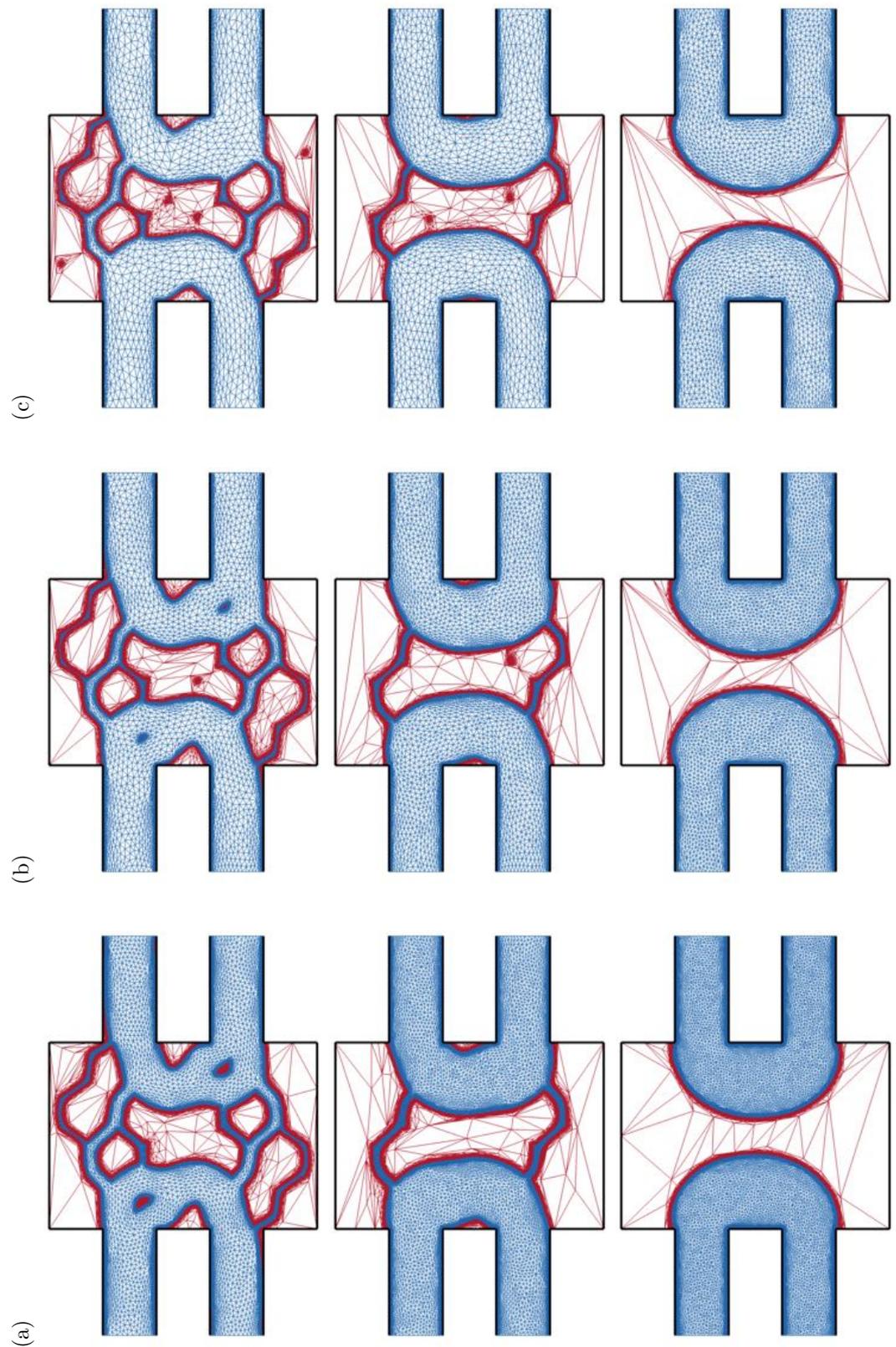


Figure 15. (cont.) Anisotropic meshes of a four terminal device sampled over the course of optimization, using (a) 80000, (b) 40000 and (c) 20000 mesh elements.

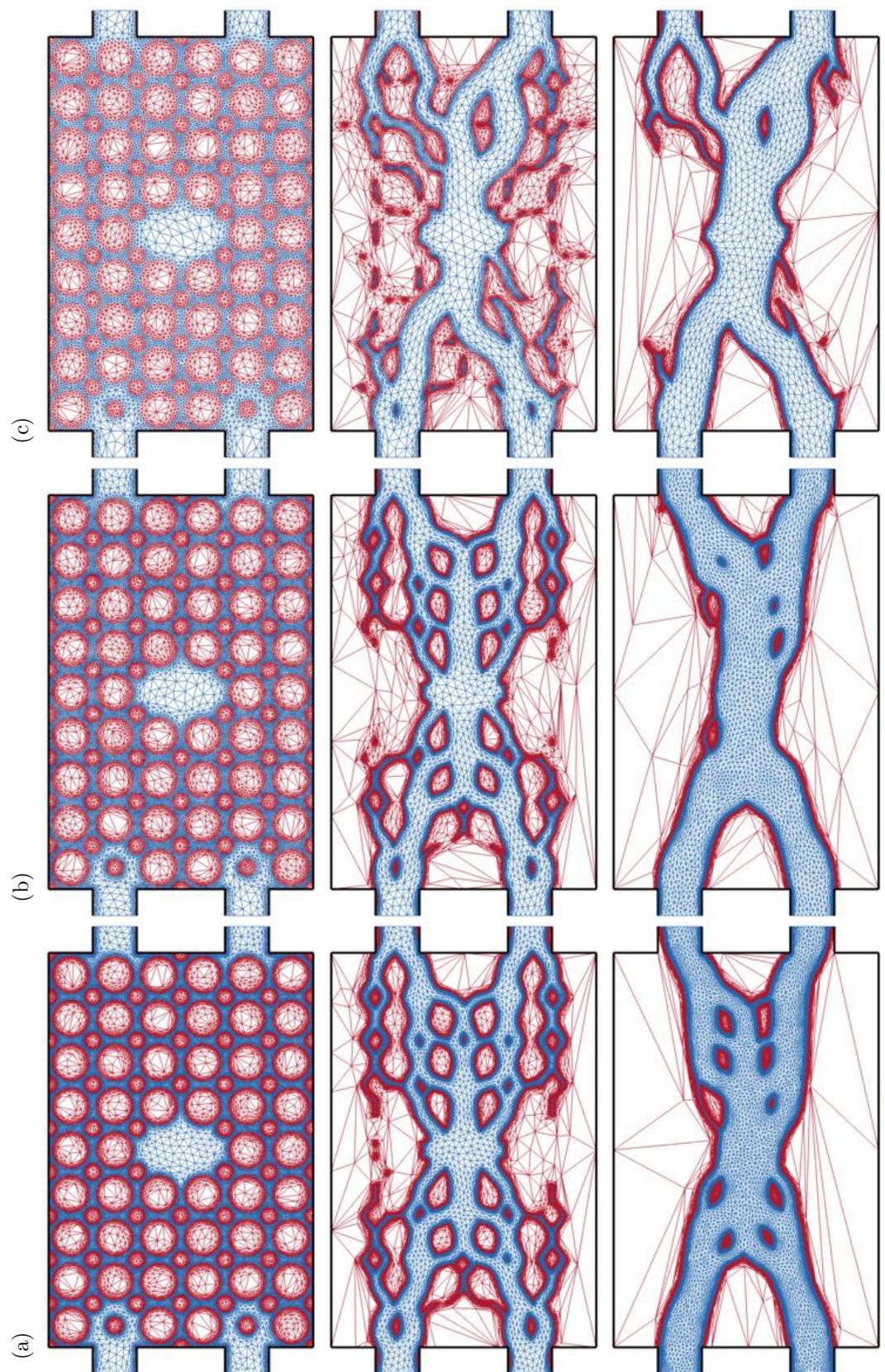


Figure 16. (cont.) Anisotropic meshes of a double pipe sampled over the course of optimization, using (a) 105000, (b) 55000 and (c) 25000 mesh elements.

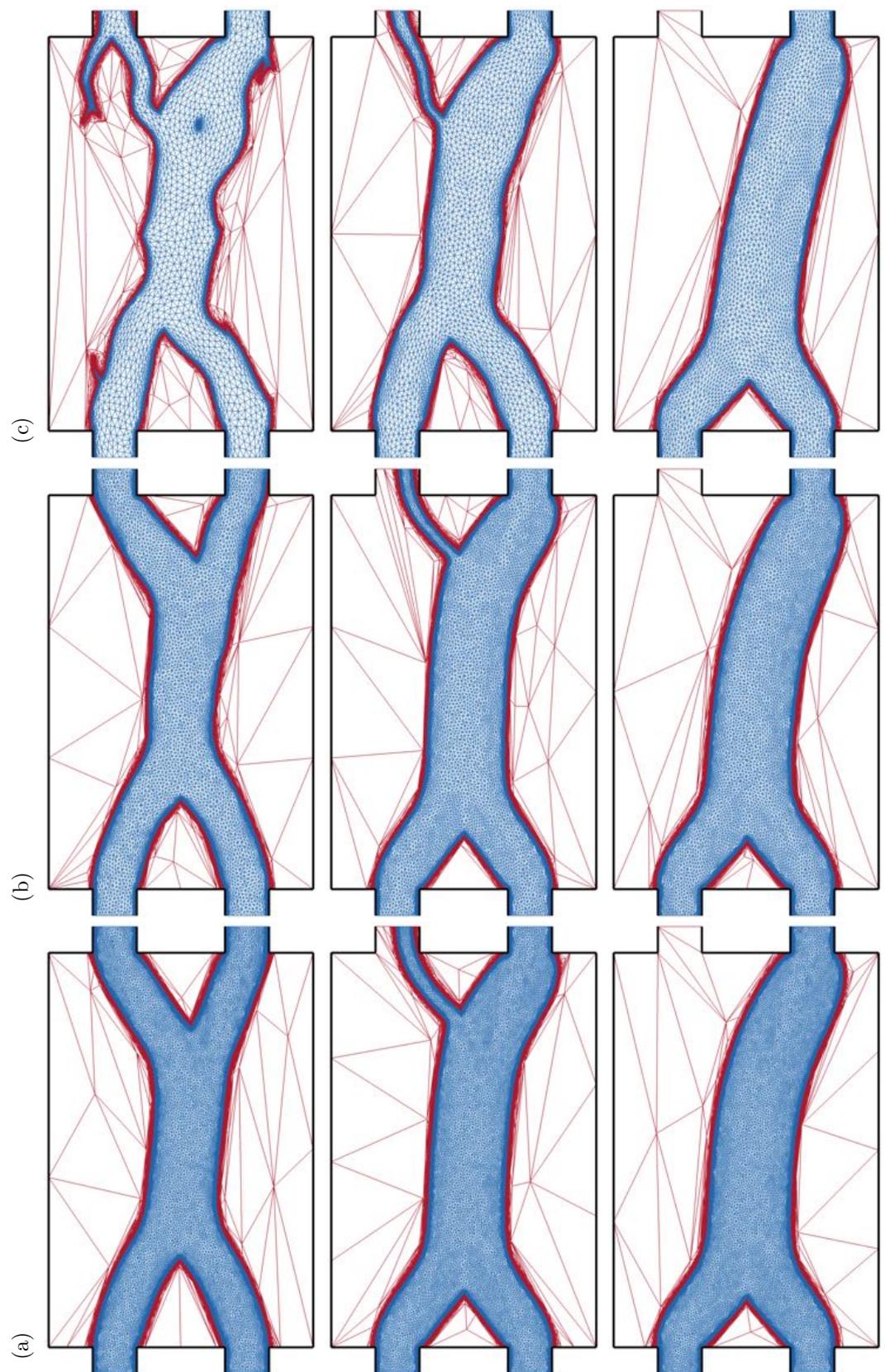


Figure 16. (cont.) Anisotropic meshes of a double pipe sampled over the course of optimization, using (a) 105000, (b) 55000 and (c) 25000 mesh elements.

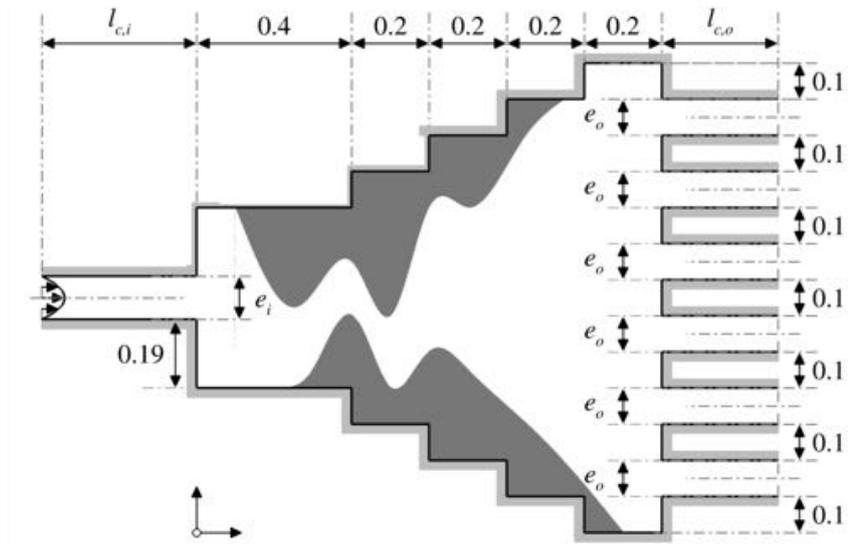


Figure 17. Set-up of the simplified flow distributor problem. The light gray shade denotes parts of the boundary where solid boundary conditions are appended to level set auto-reinitialization equation.

$\Omega = [0; 1.2] \times [0; 1.3]$	Design domain
$V_{target} = 0.4$	Target volume of fluid
$Re = 1$	Reynolds number
$q_i = 0.08$	Injected volumetric flow rate
$e_i = 0.12$	Inlet width
$e_o = 0.1$	Outlet width
$l_{c,i} = 0.4$	Inlet conveying pipes length
$l_{c,o} = 0.3$	Outlet conveying pipes length
$N_n = 25000$	Nb. mesh nodes
$N_{el} = 50000$	Nb. mesh elements

Table 3. Numerical parameters for the flow distributor problem.

7.3. Application to a simplified flow distributor problem

We finally consider a practical application of the developed framework with the simplified flow distributor problem shown in Fig. 17. The design domain for this case is a rectangular cavity of aspect ratio 0.4:0.5 widening through four consecutive steps of aspect ratio 0.2:0.1, hence a stair shape with overall aspect ratio 1.2:1.3. It features a single inlet on the left, and six identical outlets on the right. The aim is to determine the optimal design connecting the inlet to the outlets and minimizes the dissipated power subject to the constraint that the fluid must occupy 40% of the cavity, and the flow must be distributed evenly over the multiple outlet orifices for each outlet to have 1/6 of the fluid flow entering through the inlet. Since the zero pressure outflow condition does not force the inlet to connect to all the outlets (as has been assessed in Sec. 6.4 on the double pipe problem), we use the modified cost function

$$J = (1 - \omega)p_{tot}(\mathbf{u} \cdot \mathbf{n}) + \frac{\omega}{2} \|\mathbf{u} - \mathbf{u}_{target}\|^2, \quad (53)$$

689

690

691

692

693

694

695

696

697

698

699

700

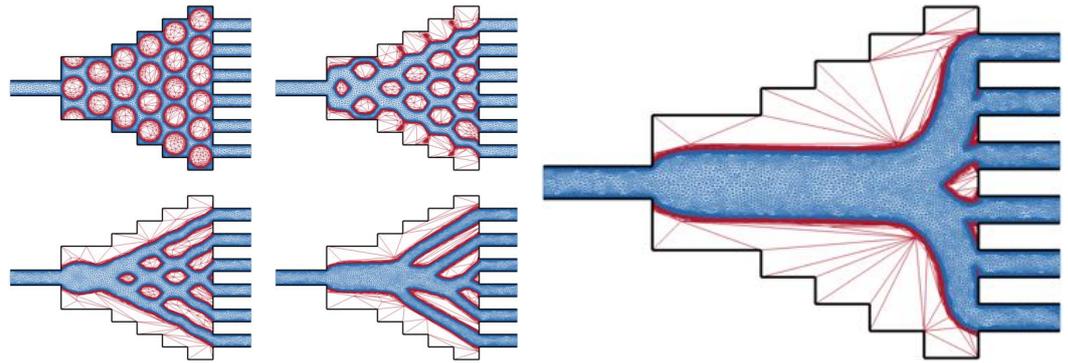


Figure 18. Intermediate (left) and optimal (right) distributor designs illustrated by their anisotropic adapted meshes.

where ω is a scalar-valued factor weighing the priority given to either power dissipation or uniformity of the outflow distribution, and \mathbf{u}_{target} is a target parabolic velocity distribution, whose outlet centerline velocity is adjusted for the mass flow to exit through the outlets to match exactly that entering through the inlet. In doing so, the theoretical framework developed in Sec. 2 carries over straightforwardly, provided the adjoint boundary conditions are updated accordingly using

$$\partial_p J = (1 - \omega) \mathbf{u} \cdot \mathbf{n}, \quad \partial_{\mathbf{u}} J = (1 - \omega) p_{tot} \mathbf{n} + (1 - \omega) \rho (\mathbf{u} \cdot \mathbf{n}) \mathbf{u} + \omega (\mathbf{u} - \mathbf{u}_{target}). \quad (54)$$

The entire domain for this case is meshed with 50000 elements, with the remaining parameters given in Tab. 3. All other parameters are identical to those in Tab. 1. Note a large weight $\omega = 0.999$ is used here to achieve comparable orders of magnitude for both the power dissipation and uniformity contributions to the cost function. This yields the optimal duct shown in Fig. 18, that delivers most of the fluid in the center area of the cavity before distributing it to the outlet channels (evenly to a 5% accuracy) via a fine comb-like structure. The obtained solution has all the attributes of a power dissipation optimal, an initial large straight pipe ultimately dividing into a near-perfect symmetrical network of six short pipes to minimize the cost of bending the fluid stream. It differs from that documented in [55] for a similar problem, which is because the authors do not consider power dissipation in their cost function, and impose only an upper bound on the volume of fluid. Even though, this showcases the potential of the method for smooth solutions to engineering problems of practical interest, for instance the design of compact and lightweight heat exchangers such those widely used in air conditioning (the design domain being representative of the refrigerant distributor section), or for microfluidics, where minimizing dissipation while maintaining an even fluid distribution in all branches of a network is of great interest to improve the performance of lab-on-a-chip devices.

8. Conclusion

The present study proves feasible to perform topology optimization of Navier–Stokes flows using anisotropic meshes adapted under the constraint of a fixed number of nodes. The proposed approach combines a level set method to represent the boundary of the fluid domain by the zero iso-value of a signed distance function, and stabilized formulations of the state, adjoint, and level set transport equations cast in the Variational Multiscale (VMS) framework. The method has been shown to allow for drastic topology changes during the optimization process. Nonetheless, the main advantage over existing methods is the ability to capture all interfaces to a very high degree of accuracy using adapted meshes whose anisotropy matches that of the numerical solutions. This gives hope that the method can ease the transition to manufacturable CAD models that closely resembles the optimal topology.

The method has been tested on several examples of power dissipation minimization in two dimensions. The obtained optimal designs are identical to reference literature results, which assesses the relevance of the present implementation for designing fluidic devices, as further illustrated by a simplified engineering case optimizing a flow distributor to minimize power dissipation while maintaining even flow distribution at multiple outlets. All optimal designs are shown to be mesh-independent, although the convergence rate does decrease as the number of nodes increases, despite the method being able to resolve smaller and smaller geometrical features. Exhaustive computational efficiency data are reported with the hope to foster future comparisons, but it is worth emphasizing that we did not seek to optimize said efficiency, for instance, using an iterative Newton-like method to compute all state solutions, which takes up the bulk of the computational time. The obtained results show the difficulty of determining the global minimum when two strong minima are competing, which simply reflect the fact that gradient-based algorithms are easily trapped in local optima, all the more so when applied to stiff nonlinear problems (gradient-free methods are better equipped in this regard, but can be more complex to implement and to use). Future work will be aimed at considering multi-component adaptation criterion to take into account the difference in the spatial supports of the state and adjoint solutions, and at extending the present method to more general two- and three-dimensional cases, including fluid-thermal coupling problems.

Author Contributions: Conceptualization, E.H.; methodology, E.H. and P.M.; software and validation, E.H., P.M. and W.A.N.; formal analysis, E.H. and P.M.; investigation, W.A.N. and J.J.; resources, E.H.; data curation, W.A.N. and J.J.; writing—original draft preparation, P.M.; writing—review and editing, E.H., P.M. and D.S.; visualization, E.H., P.M. and W.A.N.; supervision, E.H., P.M. and D.S.; project administration, E.H. and D.S.; funding acquisition, E.H. and D.S.. All authors have read and agreed to the published version of the manuscript.

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

Institutional Review Board Statement: Not applicable

Informed Consent Statement: Not applicable

Data Availability Statement: The data presented in this study are available on reasonable request from the corresponding author. The data are not publicly available due to confidential reasons.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bendsøe, M.P.; Kikuchi, N. Generating optimal topologies in structural design using a homogenization method. *Comput. Methods Appl. Mech. Engrg.* **1988**, *71*, 197–224.
2. Bendsøe, M.P.; Sigmund, O. *Topology optimization: theory, methods, and applications*; Springer Science & Business Media, 2003.
3. Yang, R.J.; Chahande, A.I. Automotive applications of topology optimization. *Struct. Opt.* **1995**, *9*, 245–249.
4. Zhu, J.H.; Zhang, W.H.; Xia, L. Topology optimization in aircraft and aerospace structures design. *Arch. Comput. Method. E.* **2016**, *23*, 595–622.
5. Sigmund, O.; Maute, K. Topology optimization approaches. *Struct. Multidiscipl. Optim.* **2013**, *48*, 1031–1055.
6. Deaton, J.D.; Grandhi, R.V. A survey of structural and multidisciplinary continuum topology optimization: post 2000. *Struct. Multidiscipl. Optim.* **2014**, *49*, 1–38.
7. Alexandersen, J.; Andreassen, C.S. A review of topology optimisation for fluid-based problems. *Fluids* **2020**, *5*, 29.
8. Suzuki, K.; Kikuchi, N. A homogenization method for shape and topology optimization. *Comput. Methods Appl. Mech. Engrg.* **1991**, *93*, 291–318.
9. Allaire, G.; Bonnetier, E.; Francfort, G.; Jouve, F. Shape optimization by the homogenization method. *Numer. Math.* **1997**, *76*, 27–68.

10. Sethian, J.A.; Wiegmann, A. Structural boundary design via level set and immersed interface methods. *J. Comput. Phys.* **2000**, *163*, 489–528. 790
11. Wang, M.Y.; Wang, X.; Guo, D. A level set method for structural topology optimization. *Comput. Methods Appl. Mech. Engrg.* **2003**, *192*, 227–246. 791
12. Allaire, G.; Jouve, F.; Toader, A.M. Structural optimization using sensitivity analysis and a level-set method. *J. Comput. Phys.* **2004**, *194*, 363–393. 792
13. Van Dijk, N.P.; Maute, K.; Langelaar, M.; Van Keulen, F. Level-set methods for structural topology optimization: a review. *Struct. Multidiscipl. Optim.* **2013**, *48*, 437–472. 793
14. Duan, X.B.; Li, F.F.; Qin, X.Q. Adaptive mesh method for topology optimization of fluid flow. *Appl. Math. Lett.* **2015**, *44*, 40–44. 794
15. Jensen, K.E. Topology optimization of Stokes flow on dynamic meshes using simple optimizers. *Comp. Fluids* **2018**, *174*, 66–77. 795
16. Duan, X.B.; Qin, X.Q. Optimality criteria coupled adaptive mesh method for optimal shape design of Stokes flow. *Math. Methods Appl. Sci.* **2016**, *39*, 3910–3920. 796
17. Duan, X.B.; Li, F.F.; Qin, X.Q. Topology optimization of incompressible Navier–Stokes problem by level set based adaptive mesh method. *Comput. Math. Appl.* **2016**, *72*, 1131–1141. 797
18. Garcke, H.; Hecht, C.; Hinze, M.; Kahle, C. Numerical approximation of phase field based shape and topology optimization for fluids. *SIAM J. Sci. Comput.* **2015**, *37*, A1846–A1871. 798
19. Feppon, F.; Allaire, G.; Bordeu, F.; Cortial, J.; Dapogny, C. Shape optimization of a coupled thermal fluid–structure problem in a level set mesh evolution framework. *SeMA* **2019**, *76*, 413–458. 799
20. Feppon, F.; Allaire, G.; Dapogny, C.; Jolivet, P. Topology optimization of thermal fluid–structure systems using body-fitted meshes and parallel computing. *J. Comput. Phys.* **2020**, *417*, 109574. 800
21. Feppon, F.; Allaire, G.; Dapogny, C.; Jolivet, P. Body-fitted topology optimization of 2D and 3D fluid-to-fluid heat exchangers. *Comput. Methods Appl. Mech. Engrg.* **2021**, *376*, 113638. 801
22. Alauzet, F.; Loseille, A. A decade of progress on anisotropic mesh adaptation for computational fluid dynamics. *Comput. Aided Des.* **2016**, *72*, 13–39. 802
23. Sari, J.; Cremonesi, F.; Khalloufi, M.; Cauneau, F.; Meliga, P.; Mesri, Y.; Hachem, E. Anisotropic adaptive stabilized finite element solver for RANS models. *Int. J. Numer. Meth. Fl.* **2018**, *86*, 717–736. 803
24. Othmer, C. A continuous adjoint formulation for the computation of topological and surface sensitivities of ducted flows. *Int. J. Numer. Meth. Fl.* **2008**, *58*, 861–877. 804
25. Chomaz, J.M. Global instabilities in spatially developing flows: Non-normality and nonlinearity. *Annu. Rev. Fluid Mech.* **2005**, *37*, 357–392. 805
26. Soto, O.; Löhner, R. On the computation of flow sensitivities from boundary integrals. *AIAA-2004-0112* **2004**. 806
27. Ville, L.; Silva, L.; Coupez, T. Convected level set method for the numerical simulation of fluid buckling. *Int. J. Numer. Meth. Fl.* **2011**, *66*, 324–344. 807
28. Coupez, T.; Silva, L.; Hachem, E. Implicit boundary and adaptive anisotropic meshing. In *New challenges in grid generation and adaptivity for scientific computing*; Perotto, S.; Formaggia, L., Eds.; Springer, 2015; pp. 1–18. 808
29. Bonito, A.; Guermond, J.L.; Lee, S. Numerical simulations of bouncing jets. *Int. J. Numer. Meth. Fl.* **2016**, *80*, 53–75. 809
30. Coupez, T. Metric construction by length distribution tensor and edge based error for anisotropic adaptive meshing. *J. Comput. Phys.* **2011**, *230*, 2391–2405. 810
31. Jannoun, G.; Hachem, E.; Veysset, J.; Coupez, T. Anisotropic meshing with time-stepping control for unsteady convection-dominated problems. *Appl. Math. Model.* **2015**, *39*, 1899–1916. 811
32. Coupez, T. Génération de maillage et adaptation de maillage par optimisation locale. *Rev. Eur. Elem. Finis* **2000**, *9*, 403–423. 812
33. Khalloufi, M.; Mesri, Y.; Valette, R.; Massoni, E.; Hachem, E. High fidelity anisotropic adaptive variational multiscale method for multiphase flows with surface tension. *Comput. Methods Appl. Mech. Engrg.* **2016**, *307*, 44–67. 813
34. Coupez, T.; Jannoun, G.; Veysset, J.; Hachem, E. Edge-based anisotropic mesh adaptation for CFD applications. In *Proceedings of the Procs. of the 21st International Meshing Roundtable*; Jiao, X.; Weill, J.C., Eds. Springer, 2013, pp. 567–583. 814
35. Meliga, P.; Chomaz, J.M.; Sipp, D. Unsteadiness in the wake of disks and spheres: Instability, receptivity and control using direct and adjoint global stability analyses. *J. Fluid Struct.* **2009**, *25*, 601–616. 815
36. Hachem, E.; Dignonnet, H.; Massoni, E.; Coupez, T. Immersed volume method for solving natural convection, conduction and radiation of a hat-shaped disk inside a 3D enclosure. *Int. J. Numer. Method H.* **2012**, *22*, 718–741. 816
37. Hachem, E.; Feghali, S.; Codina, R.; Coupez, T. Immersed stress method for fluid-structure interaction using anisotropic mesh adaptation. *Int. J. Numer. Meth. Eng.* **2013**, *94*, 805–825. 817
38. Dambrine, M.; Kateb, D. On the ersatz material approximation in level-set methods. *ESAIM Contr. Optim. Ca.* **2010**, *16*, 618–634. 818
39. Hachem, E.; Rivaux, B.; Kloczko, T.; Dignonnet, H.; Coupez, T. Stabilized finite element method for incompressible flows with high Reynolds number. *J. Comput. Phys.* **2010**, *229*, 8643–8665. 819

40. Codina, R. Stabilized finite element approximation of transient incompressible flows using orthogonal subscales. *Comput. Methods Appl. Mech. Engrg.* **2002**, *191*, 4295–4321. 848
41. Tezduyar, T.; Osawa, Y. Finite element stabilization parameters computed from element matrices and vectors. *Comput. Methods Appl. Mech. Engrg.* **2000**, *190*, 411–430. 849
42. Codina, R. Stabilization of incompressibility and convection through orthogonal sub-scales in finite element methods. *Comput. Methods Appl. Mech. Engrg.* **2000**, *190*, 1579–1599. 850
43. Hachem, E.; Kloczko, T.; Digonnet, H.; Coupez, T. Stabilized finite element solution to handle complex heat and fluid flows in industrial furnaces using the immersed volume method. *Int. J. Numer. Meth. Fl.* **2012**, *68*, 99–121. 851
44. Hachem, E.; Jannoun, G.; Veysset, J.; Coupez, T. On the stabilized finite element method for steady convection-dominated problems with anisotropic mesh adaptation. *Appl. Math. Comput.* **2014**, *232*, 581–594. 852
45. Codina, R. Comparison of some finite element methods for solving the diffusion-convection-reaction equation. *Comput. Methods Appl. Mech. Engrg.* **1998**, *156*, 185–210. 853
46. Badia, S.; Codina, R. Analysis of a stabilized finite element approximation of the transient convection-diffusion equation using an ALE framework. *SIAM J. Numer. Anal.* **2006**, *44*, 2159–2197. 854
47. Borrvall, T.; Petersson, J. Topology optimization of fluids in Stokes flow. *Int. J. Numer. Meth. Fl.* **2003**, *41*, 77–107. 855
48. Duan, X.B.; Ma, Y.C.; Zhang, R. Shape-topology optimization for Navier–Stokes problem using variational level set method. *J. Comput. Appl. Math.* **2008**, *222*, 487–499. 856
49. Gersborg-Hansen, A.; Sigmund, O.; Haber, R.B. Topology optimization of channel flow problems. *Struct. Multidiscipl. Optim.* **2005**, *30*, 181–192. 857
50. Abdelwahed, M.; Hassine, M.; Masmoudi, M. Optimal shape design for fluid flow using topological perturbation technique. *J. Math. Anal.* **2009**, *356*, 548–563. 858
51. Olesen, L.H.; Okkels, F.; Bruus, H. A high-level programming-language implementation of topology optimization applied to steady-state Navier–Stokes flow. *Int. J. Numer. Meth. Eng.* **2006**, *65*, 975–1001. 859
52. Guest, J.K.; Prévost, J.H. Topology optimization of creeping fluid flows using a Darcy–Stokes finite element. *Int. J. Numer. Meth. Eng.* **2006**, *66*, 461–484. 860
53. Papadopoulos, I. P. A. and Farrell, P.E.; Surowiec, T.M. Computing multiple solutions of topology optimization problems. *SIAM J. Sci. Comput.* **2021**, *43*, A1555–A1582. 861
54. Sigmund, O.; Petersson, K. Numerical instabilities in topology optimization: a survey on procedures dealing with checkerboards, mesh-dependencies and local minima. *Struct. Opt.* **1998**, *16*, 68–75. 862
55. Liu, Z.; Gao, Q.; Zhang, P.; Xuan, M.; Wu, Y. Topology optimization of fluid channels with flow rate equality constraints. *Struct. Multidiscipl. Optim.* **2011**, *44*, 31–37. 863

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content. 864