3D topology optimization of conjugate heat transfer considering a mean compliance constraint: advancing toward graphical user interface and prototyping

Hao Li, Simon Garnotel, Pierre Jolivet, Hiroshi Ogawa, Tsuguo Kondoh, Kozo Furuta, Joe Alexandersen, Shinji Nishiwaki

▶ To cite this version:

Hao Li, Simon Garnotel, Pierre Jolivet, Hiroshi Ogawa, Tsuguo Kondoh, et al.. 3D topology optimization of conjugate heat transfer considering a mean compliance constraint: advancing toward graphical user interface and prototyping. 2024. hal-04726444

HAL Id: hal-04726444
https://hal.science/hal-04726444v1

Preprint submitted on 8 Oct 2024

3D topology optimization of conjugate heat transfer considering a mean compliance constraint: advancing toward graphical user interface and prototyping

Hao Li[a,b,*], Simon Garnotel[c], Pierre Jolivet[d], Hiroshi Ogawa[e], Tsuguo Kondoh[a], Kozo Furuta[a], Joe Alexandersen[b] and Shinji Nishiwaki[a]

[a] *Department of Mechanical Engineering and Science, Kyoto University, 615-8540, Kyoto, Japan*

[b] *Institute of Mechanical and Electrical Engineering, University of Southern Denmark, Odense, Denmark*

[c] *R&D Division, Airthium SAS, 21 Av. de Norvége, 91140 Villebon-sur-Yvette, France*

[d] *Sorbonne Université, CNRS, LIP6, Paris, France*

[e] *Heat Exchanger R&D Division, DENSO CORPORATION, 448-8661, Aichi, Japan*

ARTICLE INFO

Keywords:
topology optimization
heat exchanger
mean compliance constraint
density-based approach
distributed mesh adaptation
additive manufacturing
3D reconstruction
graphical user interface (GUI)

ABSTRACT

Conjugate heat transfer in heat exchangers is at the heart of numerous industrial applications. Topology optimization (TO) is a promising numerical method that allows for the design of high-performance thermo-hydraulic systems from scratch. However, full-scale three-dimensional thermofluidic TO remains largely within the academic sphere and has yet to be easily explored by thermal engineers. To bridge this gap, this paper presents an integrated design workflow tailored for three-dimensional, high-resolution topology optimization of conjugate heat transfer systems, incorporating a mean compliance constraint to ensure structural integrity and load-bearing capability. This is achieved using a dual-mesh approach within the density-based TO framework. We also introduce Tanatloc, a user-friendly graphical user interface developed in JavaScript, which provides versatile functionalities and an interactive experience for thermal engineers. Finally, a 3D printed metal-based prototype is fabricated, and reverse engineering is conducted to reconstruct a CAD model using CT-scan images, paving the way for future experimental investigations.

1. Introduction

Heat exchangers are devices designed to transfer thermal energy between a fluid and a solid (heat sink, also known as a passive heat exchanger) or between two or more fluids (known as a bi-fluid heat exchanger). Conjugate heat transfer of heat exchangers are at the heart of numerous industrial applications, ranging from combustion engine cooling, air conditioning, power production, and refrigeration, to the recent advancements in microelectronic packaging and micro-reactors [1]. Despite their widespread use in industry and coverage in basic heat and mass transfer courses, heat exchangers are typically designed using empirical methods based on classical heat transfer theory, where geometric layouts are predefined intuitively. The ability to predict thermohydraulic behavior can significantly benefit from advanced design and optimization techniques.

---

[*] Corresponding author

✉ `hli@sdu.dk` (H. Li)

ORCID(s): 0000-0003-4316-1253 (H. Li); 0000-0001-5138-4321 (T. Kondoh); 0000-0002-6198-0466 (K. Furuta); 0000-0001-7872-6272 (J. Alexandersen); 0000-0003-1263-7263 (S. Nishiwaki)

Topology optimization (TO) is a promising numerical method that combines simulations with mathematical optimization algorithms to deliver conceptual designs with a high degree of design freedom. It allows designers to start from scratch and maximize functional performance, facilitating an optimization process. TO originated in structural optimization, with the seminal paper by Bendsøe and Kikuchi [2] introducing the homogenization-based design approach. However, this method was considered complex and often resulted in solutions that could not be manufactured. The advent of simpler approaches such as the density-based approach and the level set-based method provided researchers with accessible TO techniques, enabling the exploration of more complicated multiphysics problems, including thermofluidic challenges. We refer readers to three review articles for a more comprehensive overview of this field [3–5]. Since the early works [6–9], thermofluidic topology optimization has advanced significantly in various directions, such as multi-objective trade-offs [10, 11] and pseudo-3D modeling [12–15], and recently multi-scale cooling [16–18] although many studies remain limited to two-dimensional settings. Given that real-world applications are inherently three-dimensional, full-scale 3D thermofluidic TO has become increasingly attractive in both academia and industry.

In the literature on full-scale three-dimensional topology optimization of thermofluidic problems, we summarize selected representative works in Table 1, highlighting different design strategies and numerical techniques. The categorization of these works is visualized in the Sankey diagram shown in Fig. 1, with a brief discussion of these categories below.

Fluid flow can be steady or unsteady state, and further classified into laminar, turbulent, or Darcy flow. Steady-state flows are simpler and less computationally intensive [19–38], whereas unsteady flows capture transient phenomena but require more computational resources [39–41]. Laminar flows are easier to model but less representative of practical applications [19, 20, 22, 25–33, 35–41], while turbulent flows, though more complex and computationally demanding, are more realistic [21, 34]. Darcy flow is specific to porous media and provides insights into fluid movement through such materials [23, 24].

Topology optimization methods include density-based approaches, the level-set method, moving morphable components, and the ersatz level-set method. Density-based approaches are widely used due to their simplicity and robustness but may suffer from misinterpretation caused by the existence of grey-scale elements (or cells) [20, 25, 26, 28, 30, 34, 41]. The level set method offers precise boundary descriptions but is more computationally intensive, often requiring the reinitialization of the level-set function and the computation of the signed distance function [24, 27, 29, 37, 39]. Moving Morphable components [33] and the ersatz level-set method [19, 31, 38] combine aspects of both density-based and level set methods, aiming to leverage their advantages while mitigating their disadvantages.

Sensitivity analysis can be performed using continuous adjoint, discrete adjoint, or automatic differentiation. Continuous adjoint methods provide analytical sensitivities but can be complex to derive. Discrete adjoint methods are more straightforward to implement numerically but require storage of the Jacobian matrix from the governing equations, which can be memory-intensive. Automatic differentiation offers high accuracy and ease of implementation, but comes with increased computational expense.

Various optimization algorithms are employed, including the method of moving asymptotes (MMA), globally convergent MMA (GCMMA), steepest-descent, reaction-diffusion equation, Hamilton-Jacobi equation, and null space optimizer. MMA and GCMMA are well-known for their robustness and efficiency; they work by approximating the objective function with a convex approximation, making them particularly effective for large-scale problems with complex constraints. The steepest-descent method, on the other hand, is simple to implement and easy to understand, involving iterative steps in the direction of the negative gradient to minimize the objective function, but it can be slow to converge, especially for problems with ill-conditioned Hessians. Reaction-diffusion and Hamilton-Jacobi equations offer more sophisticated approaches for handling complex design spaces. These methods often use the Augmented Lagrangian method for updating Lagrange multipliers, which heavily relies on numerical implementation experience. The null space optimizer is effective in high-dimensional problems but seems to be less straightforward to implement. The interested readers are referred to in [42, 43] for a tutorial and a comparison among some of these widely used optimizers.

Different numerical methods such as finite element analysis (FEA), finite volume method (FVM), lattice Boltzmann method (LBM), and extended finite element method (XFEM) are used. FEA is versatile and widely used in topology optimization community [19, 20, 23, 24, 26–31, 35–37], but FVM is better suited for fluid flow problems because it inherently ensures local and global conservation properties [21, 22, 25, 32–34]. LBM, as an explicit numerical method, offers advantages in handling time-dependent problems but can be less accurate [38, 40, 41]. XFEM allows for modeling discontinuities within elements, adding complexity, but it is the most accurate approach for implementing the level-set method without requiring body-fitted mesh adaptation [39]. Mesh types include fixed hexahedral mesh, body-fitted mesh, isotropic adaptive mesh, anisotropic adaptive mesh, and fixed tetrahedral mesh. Fixed hexahedral meshes are simple and efficient but less flexible. Body-fitted meshes provide accurate boundary representation but are harder to generate [44]. Adaptive meshes (both isotropic and anisotropic) adjust to the solution's features, improving accuracy and efficiency but adding complexity to the mesh generation process. Fixed tetrahedral meshes offer flexibility in handling complex geometries but can be less efficient compared to the adaptive mesh and less accurate than a good hexahedral mesh.

The tools used range from commercial software such as COMSOL Multiphysics, to in-house codes and open-source software such as FeniCS, OpenFOAM, and FreeFEM. Commercial software provides robust and user-friendly environments but can be expensive and may not be able to tackle large-scale problem. Open-source software offers flexibility and cost savings but may require more effort to set up and use effectively. In-house codes can be highly tailored to specific needs but require significant development time and expertise.

The studies focus on various applications, including forced convection cooling devices, natural convection heat sinks, and bi-fluid heat exchangers. Forced convection devices utilize external means, such as fans or pumps, to enhance the heat transfer process by actively moving the fluid over the heat exchange surface [19, 21, 22, 24, 25, 27, 28, 32, 33, 35–37, 40]. In contrast, natural convection devices rely on the buoyancy-driven flow of fluid caused by temperature differences, without any external mechanical aid, to facilitate heat transfer [20, 23, 31, 38, 39, 41]. Bi-fluid heat exchangers are devices designed to transfer heat between two different fluid flows, typically separated by solid walls to prevent mixing while allowing efficient thermal exchange [26, 29, 30, 34]. By understanding these categories and their associated methodologies, one can appreciate the diverse approaches and trade-offs in the field of thermofluidic topology optimization, guiding future research and application development.

As discussed above, although thermofluidic TO has seen tremendous development, it remains largely within the academic sphere and has yet to be easily explored by engineers. This can be attributed to various challenges, including but not limited to:

1. Accurate modeling of fluid flow motions requires solving the full Navier–Stokes equations rather than relying on simplified models such as Newton's law of cooling or Darcy's law. Moreover, handling high-resolution 3D cases necessitates constructing a distributed framework to handle large-scale problems. This typically involves significant algorithmic efforts, including scalable domain decomposition, matrix assembly, parallel interpolation, and linear solvers, which are not readily available in general-purpose libraries. Many state-of-the-art works as shown in Table 1 rely on multi-node clusters, posing a challenge in terms of affordability and accessibility for general users.
2. The lack of interpretation of TO results, including boundary descriptions, poses a challenge in transforming these outcomes into models appropriate for post-processing, simulation, and manufacturing applications. Some works presented CAD-oriented structural topology optimization method [47], while most of the other works highlights optimized results using characteristic functions and illustrates the cost function obtained from the TO solver, ensuring a high-quality interpretation of these optimized structures is equally crucial for real-world product development.

Table 1: Computational techniques used in the state-of-the-art works on the topology optimization of full-scale three-dimensional thermofluidic problems, listed in chronological order. (LSM: Level-Set Method; MMC: Moving Morphable Component; A.D.: Automatic Differentiation; RDE: Reaction-Diffusion Equation; MMA: Method of Moving Asymptotes; GCMMA: Globally Convergent Method of Moving Asymptotes; H-J: Hamilton-Jacobi Equation; NLSPACE: Null-Space Optimizer; FEA: Finite Element Analysis; FVM: Finite Volume Method; LBM: Lattice Boltzmann Method; XFEM: Extended Finite Element Method; Fixed hex.: Fixed hexahedrons; Fixed tetra.: Fixed tetrahedrons; adapt.: adaptive mesh; FS HS: Forced Convection Heat Sink; NC HS: Natural Convection Heat Sink; SF HX: Single Fluid Heat Exchanger; Bi-F: Bi-Fluid Heat Exchanger).

Reference	Flow	Specific const.	TO	Adjoint	Optimizer	Numerical method	Cell number	Process num.	Mesh	Numerical tool	Application
Yaji et al. [19]	steady laminar	-	Ersatz LSM	continuous	RDE	FEA	n/a	n/a	fixed hex.	COMSOL	FC HS
Alexandersen et al. [20, 45]	steady laminar	-	Density	discrete	MMA	FEA	$3.3 \cdot 10^8$	2560	fixed hex.	PETSc	NC HS
Coffin & Maute [39]	unsteady laminar	-	LSM	discrete	GCMMA	XFEM	n/a	n/a	fixed tetra.	Inhouse	NC HS
Yaji et al. [40]	unsteady laminar	-	Density	continuous	MMA	LBM	$6.4 \cdot 10^6$	128	fixed hex.	Fortran	FC HS
Dilgen et al. [21]	steady turbulent	-	Density	A.D.	MMA	FVM	$4 \cdot 10^5$	120	fixed hex.	Inhouse	FC HS
Pietropaoli et al. [22]	steady laminar	-	Density	continuous	Steepest-descent	FVM	$7.5 \cdot 10^5$	n/a	fixed hex.	Inhouse	SF HX
Pollini et al. [23]	steady Darcy	-	Density	discrete	MMA	FEA	$8.19 \cdot 10^6$	500	fixed hex.	PETSc	NC HS
Kambampati & Kim [24]	steady Darcy	max. length-scale	LSM	discrete	H-J	FEA	$1.02 \cdot 10^6$	n/a	fixed hex.	PETSc	FC Channel
Yu et al. [25]	steady laminar	Structural const.	Density	continuous	MMA	FVM	$5.16 \cdot 10^5$	20	fixed hex.	OpenFOAM	FC HS
Høghøj et al. [26]	steady laminar	-	Density	discrete	MMA	FEA	$2.88 \cdot 10^6$	320	fixed hex.	PETSc	Bi-F HX
Feppon et al. [27]	steady laminar	-	LSM	continuous	NLSPACE	FEA	$6.16 \cdot 10^5$	24	body-fitted	FreeFEM	SF HX
Sun et al. [28]	steady laminar	-	Density	continuous	MMA	FEA	$6 \cdot 10^6$	96	fixed tetra.	FeniCS	FC HS
Kobayashi et al. [30]	steady laminar	-	Density	discrete	MMA	FEA	$1.3 \cdot 10^6$	n/a	fixed hex.	COMSOL	Bi-F HX
Feppon et al. [29]	steady laminar	non-mixing const.	LSM	continuous	NLSPACE	FEA	$1.7 \cdot 10^6$	30	body-fitted	FreeFEM	Bi-F HX
Li et al. [31]	steady laminar	-	Ersatz LSM	continuous	RDE	FEA	$9.16 \cdot 10^5$	32	body-fitted	FreeFEM	NC HS
Pan et al. [33]	steady laminar	-	MMC-Density	continuous	MMA	FVM	$1.76 \cdot 10^5$	20	isotropic adapt.	OpenFOAM	FC Channel
Yu et al. [32]	steady laminar	structural const.	Density	continuous	MMA	FVM	$8.29 \cdot 10^5$	20	fixed hex.	OpenFOAM	FC HS
Galanos et al. [34]	steady turbulent	-	Density	continuous	MMA	FVM	$1.1 \cdot 10^6$	128	fixed hex.	OpenFOAM	Bi-F HX
Tanabe et al. [41]	unsteady laminar	-	Density	Semi-continuous	MMA	LBM	$1 \cdot 10^6$	128	fixed hex.	C++	NC HS
Rogié and Andreasen [35]	steady laminar	milling const.	Density	discrete	MMA	FEA	$4.2 \cdot 10^6$	72	fixed hex.	PETSc	FC HS
Wang et al. [36]	steady laminar	overhang angle	Density	continuous	MMA	FEA	$1.05 \cdot 10^6$	n/a	fixed tetra.	FeniCS	FC HS
Meliga et al. [37]	steady laminar	-	LSM	continuous	Steepest-descent	FEA	$5 \cdot 10^6$	64	anisotropic adapt.	Inhouse	SF HX
Luo et al. [38]	steady laminar	-	Ersatz LSM	continuous	RDE	LBM	$1.02 \cdot 10^6$	128	fixed hex.	Fortran	NC HS
Lamarche-Gagnon et al. [46]	steady laminar	-	Density	discrete	MMA	FEA	$9.93 \cdot 10^5$	48	fixed tetra.	C++	Conformal cooling
This work	steady laminar	structural const.	Density	continuous	MMA	FEA	$3 \cdot 10^6$	60	anisotropic adapt.	FreeFEM	SF HX

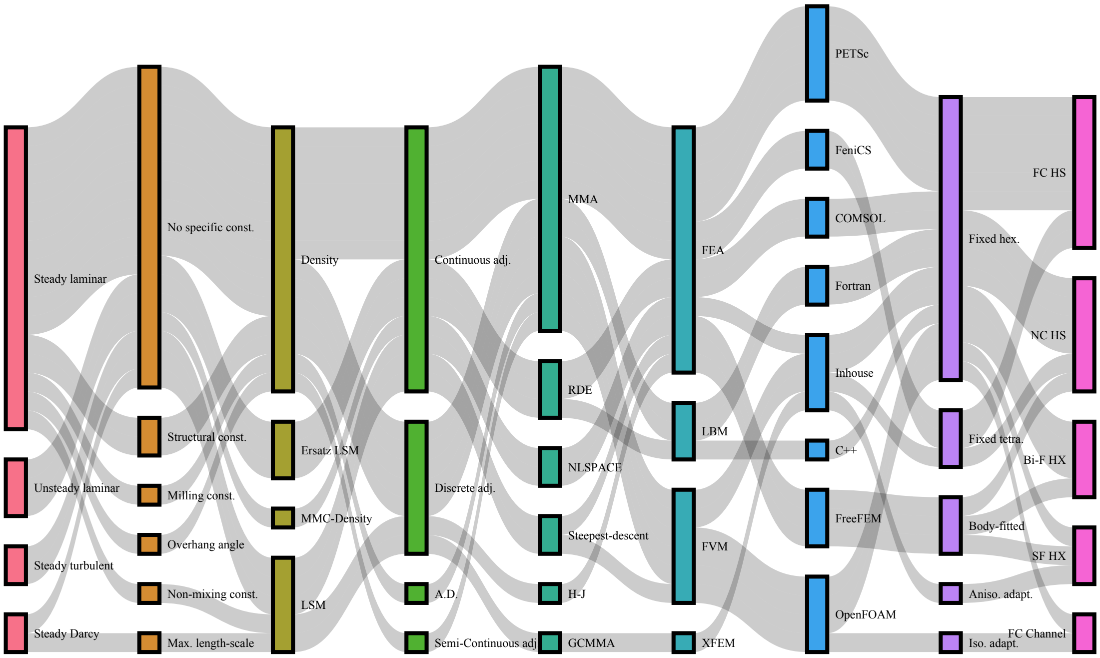


Fig. 1: Categorization of the 25 representative existing works summarized in Table 1 and their correlations. (LSM: Level-Set Method; MMC: Moving Morphable Component; A.D.: Automatic Differentiation; RDE: Reaction-Diffusion Equation; MMA: Method of Moving Asymptotes; GCMMA: Globally Convergent Method of Moving Asymptotes; H-J: Hamilton-Jacobi Equation; NLSPACE: Null-Space Optimizer; FEA: Finite Element Analysis; FVM: Finite Volume Method; LBM: Lattice Boltzmann Method; XFEM: Extended Finite Element Method; Fixed hex.: Fixed hexahedrons; Fixed tetra.: Fixed tetrahedrons; Iso. adapt.: Isotropic adaptive mesh; Aniso. adapt.: Anisotropic adaptive mesh; FS HS: Forced Convection Heat Sink; NC HS: Natural Convection Heat Sink; SF HX: Single Fluid Heat Exchanger; Bi-F: Bi-Fluid Heat Exchanger).

3. The lack of user-oriented graphical user interface limits its accessibility and functionality. To date, only a few studies demonstrate Graphical User Interfaces (GUIs) for topology optimization. Aage developed the TopOpt App [48], available for iOS, Android, Windows, and OSX, later extended it to TopOpt 3D [49], . These applications enable real-time interactive control over loading conditions and support positions for minimum compliance problems, allowing users to visualize design changes in real-time. Similarly, liteITD [50], implemented in MATLAB, focuses on the topology optimization of 2D continuum structures using von Mises stress isolines. Designed primarily for educational purposes, it supports multiple materials and loading conditions, offering a free, accessible tool for students, designers, and engineers. Another notable application, Toptimiz3D [51], goes beyond educational purposes by handling more complicated real-world geometries and unstructured meshes. It addresses five different optimal design problems, including the minimum compliance problem, multiload compliance problem, compliant mechanism problem, and stress constraint problem, and integrates three optimizers for users to choose from, including MMA, OC, and IPOPT. These GUIs represent significant strides in making topology optimization more interactive, educational, and applicable to real-world design problems in the context of compliant mechanical systems. Furthermore, Xie and his co-workers have successfully integrated their bi-directional evolutionary structural optimization (BESO) technique into the commercial topology optimization software package, Ameba [52, 53]. This tool provides a robust platform for a variety of design applications, particularly in architecture [54] enabling designers to leverage advanced topology optimization methods in real-world products. However, to the best of our knowledge,

very few GUI tools have been specifically dedicated to multiphysics topology optimization, such as thermofluidic problems. Notable exceptions include Diabatix [55] and ToffeeX [56], though the lack of detailed documentation regarding their algorithms makes it less straightforward for users to understand the theoretical background and engage with these tools.

This paper aims to narrow the gap between recent advancements in thermofluidic TO techniques and their practical applications by taking a stride in this direction. We present a comprehensive workflow tailored for three-dimensional, high-resolution design optimization for conjugate heat transfer systems. A mean compliance constraint is introduced into the optimization formulation, ensuring both structural integrity and load-bearing capability. We conduct numerous numerical test cases to illustrate distinct trade-offs between thermal-hydraulic-structural performances. A highlight of our algorithm is the use of a dual-mesh approach within the density-based TO framework, allowing us to achieve feature-rich designs on a high-resolution fixed mesh of approximately 4.5 million tetrahedral elements while performing forward and adjoint analyses on a locally refined mesh to enhance computational efficiency. Additionally, we showcase Tanatloc, a user-friendly graphical user interface developed in JavaScript, which offers versatile functionalities and an interactive user experience for thermal engineers. Eventually, we fabricate a 3D printed metal-based prototype and reconstruct the CAD model using CT-scan images, paving the way for further experimental investigations.

The remainder of this paper is structured as follows: in Section 2, we present the mathematical framework for the thermal-fluid structure system, including the material interpolation, filtering, projection schemes, optimization mathematical model, and sensitivity analysis. In Section 3, we delve into the implementation details of the optimization algorithms. In Section 4, we showcase the proposed TO framework through various numerical examples. Following that, we introduce a user interface in Section 5. Finally, we present the fabrication of a metal-based prototype in Section 6, and conclude by discussing the future work in Section 7.

2. Formulation

In this section, we start by presenting the governing equations for fluid flow, conjugate heat transfer, and linear elasticity in Section 2.1. Then, we illustrate the material interpolation functions, PDE filtering and projection scheme in Section 2.2. Finally, we formulate the optimal design problem for a weakly coupled thermal-fluid structure system in Section 2.3, followed by the sensitivity analysis.

2.1. Governing equations

2.1.1. Fluid flow

The fluid motion within a fixed and bounded domain Ω in $\mathbb{R}^d$, where $d = 2$ or 3, is described by the velocity field $\boldsymbol{v} : \Omega \to \mathbb{R}^d$ and pressure $p : \Omega \to \mathbb{R}$. This motion, characterized by the Reynolds number $\mathrm{Re} \equiv \rho U L/\mu$, where ρ is the fluid density, U is the characteristic velocity, and μ is the dynamic viscosity of the fluid. The fluid flow is governed by the incompressible, steady-state Navier-Stokes equations under laminar conditions, as follows:

$$\begin{cases} -\operatorname{div}\left(\boldsymbol{\sigma}_{\mathrm{f}}(\boldsymbol{v},p)\right) + (\boldsymbol{v}\cdot\nabla)\boldsymbol{v} = \boldsymbol{f} & \text{in } \Omega, \\ -\operatorname{div}(\boldsymbol{v}) = 0 & \text{in } \Omega, \\ \boldsymbol{v} = \boldsymbol{v}_0 & \text{on } \Gamma_{\mathrm{in}}, \\ \boldsymbol{\sigma}_{\mathrm{f}}(\boldsymbol{v},p)\cdot\boldsymbol{n} = \boldsymbol{0} & \text{on } \partial\Omega^N_{(\boldsymbol{v},p)}, \\ \boldsymbol{v} = \boldsymbol{0} & \text{on } \Gamma_{\mathrm{wall}}. \end{cases} \tag{1}$$

In the above system of equations, the internal stress of the fluid is modeled with the fluid stress tensor $\boldsymbol{\sigma}_{\mathrm{f}}(\boldsymbol{v}, p)$, which is defined as follows:

$$\boldsymbol{\sigma}_{\mathrm{f}}(\boldsymbol{v}, p) \coloneqq \frac{2}{\mathrm{Re}} e(\boldsymbol{v}) - p\mathbf{I}, \tag{2}$$

where the rate of strain tensor is defined as $e(\boldsymbol{v}) \coloneqq \frac{1}{2}\left(\nabla \boldsymbol{v} + \nabla \boldsymbol{v}^{\mathrm{T}}\right)$. A fictitious body force $\boldsymbol{f} = -\alpha(\boldsymbol{x})\boldsymbol{v}$, representing the fluid flows through a porous medium is used to penalize flow in solid regions. The inverse permeability $\alpha(\boldsymbol{x})$ varies with spatial position $\boldsymbol{x}$ and is defined as follows:

$$\alpha\left(\boldsymbol{x}\right) = \begin{cases} 0 & \text{if } \boldsymbol{x} \in \Omega_{\mathrm{f}}, \\ \alpha_{\max} & \text{if } \boldsymbol{x} \in \Omega \backslash \Omega_{\mathrm{f}}. \end{cases} \tag{3}$$

The unit normal vector $\boldsymbol{n}$ is directed from Ω_{f} towards Ω_{s}.

The boundary $\partial\Omega_{(\boldsymbol{v},p)}$ is composed of a Dirichlet boundary $\partial\Omega^{D}_{(\boldsymbol{v},p)}$ and a Neumann boundary $\partial\Omega^{N}_{(\boldsymbol{v},p)}$. On $\partial\Omega^{D}_{(\boldsymbol{v},p)} = \Gamma_{\mathrm{in}} \cup \Gamma_{\mathrm{wall}}$, fluid enters Ω via the inlet Γ_{in} with a given velocity profile $\boldsymbol{v}_0$, and a no-slip boundary condition is imposed on the wall Γ_{wall}. On $\partial\Omega^{N}_{(\boldsymbol{v},p)}$ (or outlet part), the fluid exits Ω with a zero normal stress vector.

For the Navier-Stokes equations outlined in Eq. (1), using the finite element analysis (FEA), we derive the weak forms of the Jacobian and the residual at a given linearization point $(\boldsymbol{v}, p)$ for an increment $(\delta\boldsymbol{v}, \delta p)$, with the Pressure-Stabilizing Petrov–Galerkin (PSPG), as follows:

$$\begin{aligned} F(\boldsymbol{v}, p) = & \int_{\Omega} ((\boldsymbol{v} \cdot \nabla)\boldsymbol{v}) \cdot \tilde{\boldsymbol{v}} \,\mathrm{d}\Omega + \frac{2}{\mathrm{Re}} \int_{\Omega} e(\boldsymbol{v}) : \nabla \tilde{\boldsymbol{v}} \,\mathrm{d}\Omega \\ & - \int_{\Omega} p \nabla \cdot \tilde{\boldsymbol{v}} \,\mathrm{d}\Omega - \int_{\Omega} \tilde{p} \nabla \cdot \boldsymbol{v} \,\mathrm{d}\Omega + \int_{\Omega} \alpha(\boldsymbol{x})\boldsymbol{v} \cdot \tilde{\boldsymbol{v}} \,\mathrm{d}\Omega - \int_{\partial\Omega} \left(\boldsymbol{\sigma}_{\mathrm{f}}(\boldsymbol{v}, p) \cdot \boldsymbol{n}\right) \cdot \tilde{\boldsymbol{v}} \,\mathrm{d}\Gamma \\ & - \sum_{K \in \mathcal{T}} \int_{\Omega_e} \tau_{\mathrm{PSPG}} \nabla \tilde{p} \cdot ((\boldsymbol{v} \cdot \nabla)\boldsymbol{v} + \nabla p + \alpha(\boldsymbol{x})\boldsymbol{v}) \,\mathrm{d}\Omega_e \quad \forall (\tilde{\boldsymbol{v}}, \tilde{p}) \in \mathscr{U}_{\boldsymbol{v},p}, (\boldsymbol{v}, p) \in \mathscr{U}_{\boldsymbol{v},p}, \end{aligned} \tag{4a}$$

$$\begin{aligned} DF(\boldsymbol{v}, p)(\delta\boldsymbol{v}, \delta p) = & \int_{\Omega} ((\delta\boldsymbol{v} \cdot \nabla)\boldsymbol{v}) \cdot \tilde{\boldsymbol{v}} \,\mathrm{d}\Omega + \int_{\Omega} ((\boldsymbol{v} \cdot \nabla)\delta\boldsymbol{v}) \cdot \tilde{\boldsymbol{v}} \,\mathrm{d}\Omega + \frac{2}{\mathrm{Re}} \int_{\Omega} e(\delta\boldsymbol{v}) : \nabla \tilde{\boldsymbol{v}} \,\mathrm{d}\Omega \\ & - \int_{\Omega} \delta p \nabla \cdot \tilde{\boldsymbol{v}} \,\mathrm{d}\Omega - \int_{\Omega} \tilde{p} \nabla \cdot \delta\boldsymbol{v} \,\mathrm{d}\Omega + \int_{\Omega} \alpha(\boldsymbol{x})\delta\boldsymbol{v} \cdot \tilde{\boldsymbol{v}} \,\mathrm{d}\Omega - \int_{\partial\Omega} \left(\boldsymbol{\sigma}_{\mathrm{f}}(\delta\boldsymbol{v}, \delta p) \cdot \boldsymbol{n}\right) \cdot \tilde{\boldsymbol{v}} \,\mathrm{d}\Gamma \\ & - \sum_{K \in \mathcal{T}} \int_{\Omega_e} \tau_{\mathrm{PSPG}} \nabla \tilde{p} \cdot ((\delta\boldsymbol{v} \cdot \nabla)\boldsymbol{v} + (\boldsymbol{v} \cdot \nabla)\delta\boldsymbol{v} + \nabla \delta p + \alpha(\boldsymbol{x})\delta\boldsymbol{v}) \,\mathrm{d}\Omega \\ & - \sum_{K \in \mathcal{T}} \int_{\Omega_e} \left(\delta_{(\boldsymbol{v},p)} \tau_{\mathrm{PSPG}}\right) \nabla \tilde{p} \cdot ((\boldsymbol{v} \cdot \nabla)\boldsymbol{v} + \nabla p + \alpha(\boldsymbol{x})\boldsymbol{v}) \,\mathrm{d}\Omega \\ & \forall (\tilde{\boldsymbol{v}}, \tilde{p}) \in \mathscr{U}_{\boldsymbol{v},p}, (\boldsymbol{v}, p) \in \mathscr{U}_{\boldsymbol{v},p}, \end{aligned} \tag{4b}$$

where $\tilde{\boldsymbol{v}}$ and $\tilde{p}$ are the test functions, with the Hilbert space $\mathscr{U}_{\boldsymbol{v},p}$ defined as follows:

$$\mathscr{U}_{\boldsymbol{v},p} \coloneqq \left\{ (\tilde{\boldsymbol{v}}, \tilde{p}) \in H^1\left(\Omega_{\mathrm{f}}, \mathbb{R}^d\right) \times L^2\left(\Omega_{\mathrm{f}}\right) \mid \tilde{\boldsymbol{v}} = 0 \text{ on } \partial\Omega^{D}_{\mathrm{f}} \right\}. \tag{5}$$

The stabilization parameter τ_{PSPG} is defined as follows [57]:

$$\tau_{\text{PSPG}} = \left(\tau_1^{-2} + \tau_3^{-2} + \tau_4^{-2}\right)^{-1/2}, \tag{6}$$

where τ_1, τ_3, and τ_4 are defined as follows:

$$\tau_1 = \frac{h}{2\sqrt{\boldsymbol{v}\cdot\boldsymbol{v}}}, \quad \tau_3 = \frac{h^2}{12/\operatorname{Re}}, \quad \tau_4 = \frac{1}{\alpha(\boldsymbol{x})}, \tag{7}$$

where h denotes the mesh element size. The variation of the stabilization term $\delta_{(\boldsymbol{v},p)}\tau_{\text{PSPG}}$ in Eq. (4b) is derived in Eq. (A.5).

Subsequently, the nonlinear problem is solved using a Newton method with a tolerance value set to 10^{-6}, wherein the following linearized equations are solved in sequence: Find $(\delta\boldsymbol{v}, \delta p) \in \mathscr{U}_{\boldsymbol{v},p}$ with $\delta\boldsymbol{v} = 0$ on $\partial\Omega^D_{(\boldsymbol{v},p)}$ such that:

$$DF(\boldsymbol{v}, p)(\delta\boldsymbol{v}, \delta p) = F(\boldsymbol{v}, p). \tag{8}$$

2.1.2. Conjugate heat transfer

Once the Navier-Stokes equations are solved, they are then weakly coupled with the non-dimensional steady-state convection-diffusion equation governing the conjugate heat transfer within the system, characterized by the temperature $T : \Omega \to \mathbb{R}$, as follows:

$$\begin{cases} \operatorname{Re}\operatorname{Pr}(\boldsymbol{v}\cdot\nabla T) - \operatorname{div}(\kappa(\boldsymbol{x})\nabla T) = 0 & \text{in } \Omega, \\ T = T_0 & \text{on } \partial\Omega^D_T, \\ \kappa(\boldsymbol{x})\nabla T \cdot \boldsymbol{n} = 0 & \text{on } \partial\Omega^N_T, \end{cases} \tag{9}$$

where $\operatorname{Pr} \equiv c_p\mu/\kappa$ denotes the Prandtl number, where c_p is the specific heat capacity. The thermal conductivity $\kappa(\boldsymbol{x})$ varies with spatial position $\boldsymbol{x}$ and is defined as follows:

$$\kappa(\boldsymbol{x}) = \begin{cases} \kappa_{\text{f}} & \text{if } \boldsymbol{x} \in \Omega_{\text{f}}, \\ \kappa_{\text{s}} & \text{if } \boldsymbol{x} \in \Omega\backslash\Omega_{\text{f}}. \end{cases} \tag{10}$$

The boundary $\partial\Omega_T$ consists of a Dirichlet boundary $\partial\Omega^D_T$ and a Neumann boundary $\partial\Omega^N_T$. Specifically, $\partial\Omega^D_T = \Gamma_{\text{in}} \cup \Gamma_{\text{hot}}$, where the inlet Γ_{in} admits cold fluid with a lower temperature, while the hot wall Γ_{hot} has a high temperature imposed on it. The remaining boundaries are subject to an adiabatic boundary condition.

The convection-diffusion equation defined in Eq. (9) can be written in weak form as: find $T \in \mathscr{P}$ with $T = T_0$ on $\partial\Omega^D_T$ such that:

$$\int_\Omega \operatorname{Re}\operatorname{Pr}(\boldsymbol{v}\cdot\nabla T)\tilde{T}\,\mathrm{d}\Omega + \int_\Omega \kappa(\boldsymbol{x})\nabla T\cdot\nabla\tilde{T}\,\mathrm{d}\Omega = 0 \quad \forall\tilde{T} \in \mathscr{P}, T \in \mathscr{P}, \tag{11}$$

with the Hilbert space $\mathscr{P}$ defined as follows:

$$\mathscr{P} := \left\{\tilde{p} \in H^1(\Omega) \mid \tilde{p} = 0 \text{ on } \partial\Omega^D_T\right\}. \tag{12}$$

2.1.3. Linear elasticity

Finally, given the velocity $\boldsymbol{v}$, pressure p, and temperature T within the computational domain Ω, we can calculate the displacement $\boldsymbol{u} : \Omega \to \mathbb{R}^d$ as the solution to a linear elasticity system. In this paper, we make the following assumptions: (1) Small displacements and deformations are observed, implying that the changes in the fluid-structure boundaries $\Gamma_{\mathrm{s,f}}$ are negligible when the solid interacts with the viscous fluid flow load. (2) The effects of body forces and gravity are neglected. (3) The linear thermoelasticity system comprises materials with negligible thermal expansion coefficients over the temperature range of interest, meaning that the structural response is primarily influenced by mechanical loading rather than thermal effects or the viscous fluid flow load. Note that for completeness, the thermal expansion term is included in the formulation.

$$\begin{cases} -\operatorname{div}\left(\boldsymbol{\sigma}_{\mathrm{s}}(\boldsymbol{u},T)\right)=0 & \text{in } \Omega, \\ \boldsymbol{u}=\boldsymbol{u}_0 & \text{on } \partial\Omega_{\boldsymbol{u}}^D, \\ \boldsymbol{\sigma}_{\mathrm{s}}(\boldsymbol{u},T)\cdot\boldsymbol{n}=\boldsymbol{t} & \text{on } \partial\Omega_{\boldsymbol{u}}^N, \\ \boldsymbol{\sigma}_{\mathrm{s}}(\boldsymbol{u},T)\cdot\boldsymbol{n}=\boldsymbol{\sigma}_{\mathrm{f}}(\boldsymbol{v},p)\boldsymbol{n} & \text{on } \Gamma_{\mathrm{s,f}}, \end{cases} \tag{13}$$

With the isotropic thermoelastic material characterized by Lamé coefficients λ and μ, thermal expansion coefficient α_T, and reference temperature T_{ref}, the solid stress tensor is defined as:

$$\boldsymbol{\sigma}_{\mathrm{s}}(\boldsymbol{u},T)=\mathbb{C}(\boldsymbol{x}):e(\boldsymbol{u})-\alpha_T\left(T-T_{\mathrm{ref}}\right)\mathbf{I}. \tag{14}$$

The fourth-order elasticity tensor $\mathbb{C}$ in Eq. (14) is defined as follows:

$$\mathbb{C}:=\lambda\delta_{ij}\delta_{kl}+\mu\left(\delta_{ik}\delta_{jl}+\delta_{il}\delta_{jk}\right), \tag{15}$$

where δ_{ij} denotes the Kronecker delta function. The elasticity tensor varies depending on spatial position $\boldsymbol{x}$ as follows:

$$\mathbb{C}(\boldsymbol{x})=\begin{cases}\mathbb{C}_{\mathrm{v}} & \text{if } \boldsymbol{x}\in\Omega_{\mathrm{f}}, \\ \mathbb{C}_{\mathrm{s}} & \text{if } \boldsymbol{x}\in\Omega\backslash\Omega_{\mathrm{f}}.\end{cases} \tag{16}$$

Then, the Lamé equation defined in Eq. (9) can be written in weak form as: find $\boldsymbol{u}\in\mathscr{U}$ with $\boldsymbol{u}=\boldsymbol{u}_0$ on $\partial\Omega_{\boldsymbol{u}}^D$ such that:

$$\int_{\Omega}\left(e(\boldsymbol{u}):\mathbb{C}(\boldsymbol{x})\right):e(\tilde{\boldsymbol{u}})\,\mathrm{d}\Omega=\int_{\Omega}\alpha_T\left(T-T_{\mathrm{ref}}\right)\mathbf{I}:\nabla\tilde{\boldsymbol{u}}\,\mathrm{d}\Omega+\int_{\partial\Omega_{\boldsymbol{u}}^N}\boldsymbol{t}\cdot\tilde{\boldsymbol{u}}\,\mathrm{d}\Gamma\quad\forall\tilde{\boldsymbol{u}}\in\mathscr{U},\boldsymbol{u}\in\mathscr{U}, \tag{17}$$

with the Hilbert space $\mathscr{U}$ defined as follows:

$$\mathscr{U}:=\left\{\tilde{\boldsymbol{u}}\in H^1\left(\Omega,\mathbb{R}^d\right)\mid\tilde{\boldsymbol{u}}=0\text{ on }\partial\Omega_s^D\right\}. \tag{18}$$

2.2. Material interpolation functions

To perform topology optimization, a pseudo-density field $\gamma(\boldsymbol{x})$ continuously varying between 0 and 1, is introduced. The solid phase is represented by $\gamma(\boldsymbol{x})=0$ and the fluid phase by $\gamma(\boldsymbol{x})=1$, i.e.:

$$\gamma(\boldsymbol{x})=\begin{cases}0 & \text{if } \boldsymbol{x}\in\Omega_{\mathrm{s}}, \\ 1 & \text{if } \boldsymbol{x}\in\Omega_{\mathrm{f}}.\end{cases}$$

To mitigate the well-known checkerboard phenomenon, filtering techniques are employed [58, 59]. The design variables γ is smoothened by solving the following PDE with r the regularization parameter:

$$\begin{cases} -r^2\Delta\tilde{\gamma} + \tilde{\gamma} = \gamma & \text{in } \Omega, \\ \nabla\tilde{\gamma}\cdot\boldsymbol{n} = 0 & \text{on } \partial\Omega. \end{cases} \tag{19}$$

The weak form for Eq. (19) is as follows: find $\tilde{\gamma} \in \mathscr{P}$ such that:

$$\int_D r^2\nabla\tilde{\gamma}\cdot\nabla v + \tilde{\gamma}v\mathrm{d}\Omega = \int_D \gamma v\mathrm{d}\Omega \quad \forall\delta v \in \mathscr{P}, \tilde{\gamma} \in \mathscr{P}. \tag{20}$$

A smoothed Heaviside function is used to obtained the projected density field $\hat{\gamma}$ with the threshold η and sharpness β:

$$\hat{\gamma}(\tilde{\gamma}, \beta, \eta) = \frac{\tanh(\beta\eta) + \tanh(\beta(\tilde{\gamma} - \eta))}{\tanh(\beta\eta) + \tanh(\beta(1 - \eta))}. \tag{21}$$

Then, the inverse permeability $\alpha\left(\hat{\gamma}\right)$, thermal conductivity $\kappa\left(\hat{\gamma}\right)$, and elasticity tensor $\mathbb{C}\left(\hat{\gamma}\right)$ can be interpolated as:

$$\begin{cases} \alpha\left(\hat{\gamma}\right) = \alpha_{\max}\dfrac{q_\alpha\left(1-\hat{\gamma}\right)}{q_\alpha + \hat{\gamma}}, \\ \kappa\left(\hat{\gamma}\right) = 1 + \left(\dfrac{\kappa_{\mathrm{s}}}{\kappa_{\mathrm{f}}} - 1\right)\dfrac{q_\kappa\left(1-\hat{\gamma}\right)}{q_\kappa + \hat{\gamma}}, \\ \mathbb{C}\left(\hat{\gamma}\right) = \mathbb{C}_{\mathrm{v}} + (1-\hat{\gamma})^{q_{\mathbb{C}}}\left(\mathbb{C}_{\mathrm{s}} - \mathbb{C}_{\mathrm{v}}\right). \end{cases} \tag{22}$$

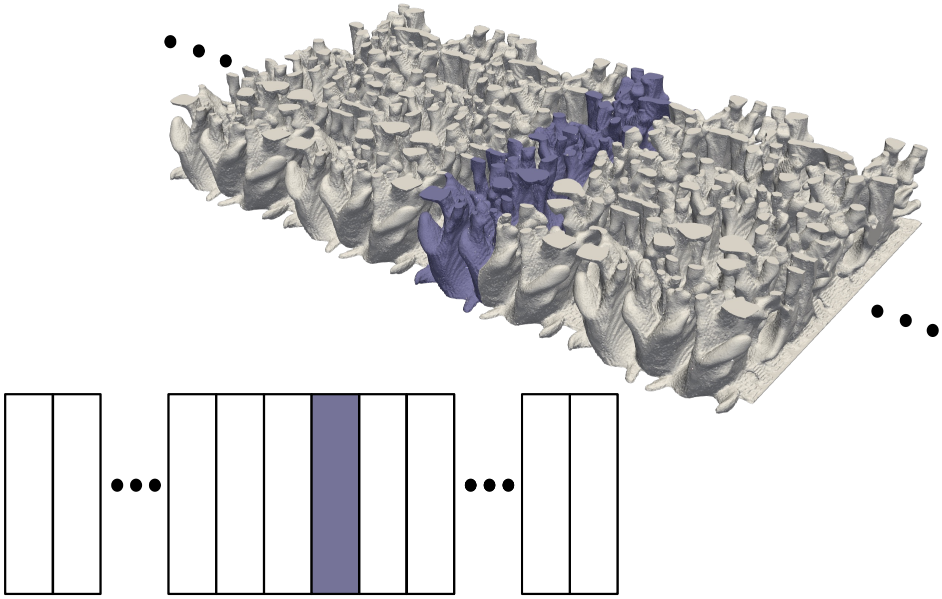
where q_α, q_κ and $q_{\mathbb{C}}$ are the penalty factors for α, κ, and $\mathbb{C}$ respectively. The projection threshold η is set to 0.5. For the projection sharpness β, a continuation scheme is employed as follows:

$$\beta_{\mathrm{it}+1} = \min\left(60, \beta_{\mathrm{it}} + 0.2\right), \quad \beta_1 = 1.0, \tag{23}$$

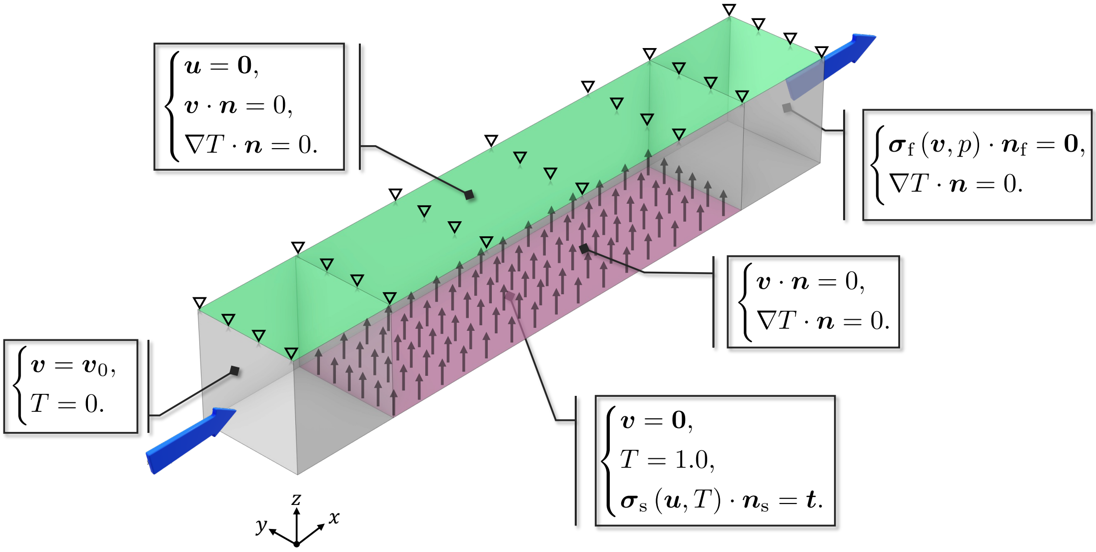
where the subscript denotes the optimization iteration number. Note that in comparison to the stepwise continuation approach, where β is held constant for a set number of iterations before being increased, the incremental continuation scheme employed in this work offers a smoother transition by gradually increasing β at each iteration. While the stepwise approach provides more stable sensitivity calculations within each block of iterations, it can lead to abrupt changes when β is increased. The incremental approach, on the other hand, promotes smoother transitions, enhancing optimization speed by avoiding disruptive changes. Although this may slightly reduce sensitivity accuracy in the early stages, this is not an issue during the conceptual design phase, where the focus is on broad exploration rather than fine-tuning.

2.3. Optimization problem

The focus of this study is to develop pin-fin structures that strike a balance between maximizing heat transfer by the fluid phase and minimizing viscous energy dissipation, while also meeting a maximum allowed mean compliance to ensure the load-carrying capacity of the optimized fin structures. The schematic diagram illustrating the design settings is presented in Fig. 2. As depicted in Fig. 2a, we assume that the fin structures are arranged periodically in the y-direction, with the design domain highlighted in purple. Further details of our design model are outlined in Fig. 2b. The computational domain measures $3.0 \times 0.5 \times 0.5$. A cold fluid flow, characterized by the Reynolds number Re, enters the computational domain through an inlet boundary denoted as Γ_{in} (left end of the domain), with a prescribed velocity $\boldsymbol{v} = \boldsymbol{v}_0$ exhibiting a parabolic profile $\boldsymbol{v} = \boldsymbol{v}(z)$ and a prescribed temperature $T = 0$. The fluid exits the domain through the outlet boundary Γ_{out} (right end of the domain) with a zero normal-stress boundary condition. The top and side walls are subject to a free-slip boundary condition, expressed as $\boldsymbol{v}\cdot\boldsymbol{n} = 0$, while the bottom wall has a no-slip boundary condition, represented as $\boldsymbol{v} = \mathbf{0}$. The middle portion of the bottom wall is imposed to have a prescribed high temperature of $T = 1$, and an adiabatic boundary condition, $\nabla T \cdot \boldsymbol{n} = 0$, is imposed on the side walls and the top wall. Additionally, a surface traction t is applied to the middle portion of the bottom wall, while the top wall is held fixed with $\boldsymbol{u} = \mathbf{0}$.



(a) Highlighting of the design domain within the periodically arranged fin structures is depicted in purple.



(b) Optimization design model.

Fig. 2: Schematic diagram of the design setting.

The objective function considered is to maximize the heat flux out of the system, and the corresponding optimization mathematical model is formulated as follows:

$$\min_{\gamma \in D} J(\Omega) = -\int_{\Gamma_{\text{out}}} (\boldsymbol{v} \cdot \boldsymbol{n}) T \, \mathrm{d}\Gamma, \tag{24a}$$

$$\text{s.t.} \begin{cases} \text{Eqs. (1), (9) and (13),} \\ G_1 = \dfrac{\int_D 1 - \hat{\gamma} \, \mathrm{d}\Omega}{\int_D \, \mathrm{d}\Omega} \leq V_{\max}, \\ G_2 = \Phi(\Gamma, \boldsymbol{v}(\Gamma), p(\Gamma)) \leq \Phi_{\max}, \\ G_3 = \Psi(\Gamma, \boldsymbol{u}(\Gamma)) \leq C_{\text{Comp}} \Psi_0, \\ 0 \leq \gamma(\boldsymbol{x}) \leq 1 \quad \forall \boldsymbol{x} \in D, \end{cases} \tag{24b}$$

where G_1, G_2, and G_3 represent the volume constraint, energy dissipation constraint, and mean compliance constraint, respectively. $V_{\max}$ denotes the maximum allowable volume fraction for the solid phase. $\Phi_{\max}$ indicates the maximum allowed viscous energy dissipation by the fluid. Ψ_0 denotes the mean compliance at the initial iteration, and C_{Comp} indicates the maximum permitted mean compliance fraction with respect to the initial value. The energy dissipation, Φ, and mean compliance, Ψ, are formulated as follows:

$$\Phi(\Gamma, \boldsymbol{v}(\Gamma), p(\Gamma)) = \int_{\Gamma_{\text{in}} \cup \Gamma_{\text{out}}} \left(p + \frac{1}{2}|\boldsymbol{v}|^2\right) \left(-\boldsymbol{v} \cdot \boldsymbol{n}_{\text{f}}\right) \, \mathrm{d}\Gamma, \tag{25a}$$

$$\Psi(\Gamma, \boldsymbol{u}(\Gamma)) = \int_{\partial\Omega_{\boldsymbol{u}}^N} \boldsymbol{t} \cdot \boldsymbol{u} \, \mathrm{d}\Gamma. \tag{25b}$$

Sensitivity analysis can be conducted using the adjoint method, which can be further classified into continuous and discrete adjoint methods. To ensure this paper is self-contained, we briefly discuss both methods. The choice between them depends on user preference and software requirements. This methodology is well-documented in various textbooks, such as [60]. Interested readers are encouraged to refer to these textbooks for a more mathematical background.

For the optimum design problem defined in Eq. (24), evaluating the sensitivities w.r.t. the objective J and the two constraint functions (G_2 and G_3) requires solving three systems of adjoint equations. Each adjoint system involves solving the adjoint elasticity equation (only if the displacement depends on thermal expansion), adjoint convection-diffusion equation, and adjoint Navier-Stokes equations in a weakly coupled manner. Note that for a given system, two Gâteaux derivatives are required for both the residual $\mathbf{r}$ and cost functions J and G_i: one w.r.t. the design variable γ and the other w.r.t. the state variables $\mathbf{s}$. These Gâteaux derivatives can be obtained either analytically or using symbolic differentiation tools. In this paper, we derived them manually.

To this end, the continuous adjoint method involves deriving the adjoint equations from the continuous form of the governing equations before discretizing them. This method provides analytical expressions for the adjoint systems, which can be highly accurate in the continuous sense but mathematically intensive and complex. The derivation of the continuous adjoint equations can be found in Appendix A.1.

Alternatively, the discrete adjoint method involves first discretizing the governing equations and then deriving the adjoint equations from this discrete form. For instance, in the case of the adjoint Navier-

Stokes equations, the Jacobian matrix built for the final non-linear Newton iteration is reused, provided that the Newton tolerance is sufficiently small [57]. The transposed Jacobian is then used to solve the adjoint problems with the corresponding right-hand sides. The derivation of the discrete adjoint equations is detailed in Appendix A.2.

The design variable γ is updated through the method of moving asymptotes (MMA) [61]. During each optimization iteration, the sensitivities serve as input parameters. Subsequently, a convex approximating subproblem is formulated and solved.

3. Implementation details

As previously mentioned, thermofluidic TO cases demand substantial computational resources due to the repeated solving of weakly coupled systems of equations (governing and adjoint equations). Moreover, achieving a high-resolution TO design in a three-dimensional context requires a fine mesh, especially around the fluid–solid interface, intensifying the challenge. To address these challenging issues, this paper proposes a distributed density-based TO framework that integrates anisotropic mesh adaptation. More specifically, we use the h-adaptation technique [62, 63] which is recently popularized in the level set-based TO framework [37, 64, 65]. In this paper, ParMmg [66] is used to perform the h-adaptation in a distributed fashion, and Mmg [67] to generate body-fitted meshes with a single-core code.

FreeFEM [68, 69] is used to discretize the PDEs presented in Section 2, while PETSc [70] serves as the distributed linear algebra backend. FreeFEM is a domain-specific language (DSL) designed to perform finite element analyses with a high level of abstraction. Once the weak (or variational) form of a partial differential equation is established, users are relieved from the cumbersome tasks inherent to the finite element method, such as vector or matrix assembly and mesh adaptation, as these can be efficiently executed within the FreeFEM DSL using a syntax that closely resembles the mathematical formulation. PETSc, on the other hand, is an algebraic backend widely used in scientific libraries globally and has been deployed on a range of architectures, from laptops to large exascale systems. Together, this ecosystem offers a flexible infrastructure to handle coupled and high-dimensional systems of equations using Message Passing Interface (MPI) for distributed-memory parallelism.

The proposed TO algorithm utilizes two different meshes. The first, denoted as $\mathcal{T}_{\text{analysis}}$, is used for solving the governing and adjoint equations. Specifically, at each optimization iteration, a metric tensor $\mathcal{M}$, constructed based on error estimations involving the Hessian of the target output (i.e., $\hat{\gamma}$), regulates mesh sizes and edge directions. The second mesh, $\mathcal{T}_{\text{design}}$, remains fixed throughout the optimization process and is used for updating design variables and conducting PDE-filter operations. The use of a fixed-design mesh simplifies the implementation of the MMA algorithm, as it mandates the number of design variables to remain consistent across iterations, see similar "mesh-independent" approaches in the context of structural optimization [65, 71]. Note that the mesh size must be smaller than or equal to the filter radius. An efficient finite element interpolator is used to conduct interpolations between these two meshes [68].

Now, we provide details in Algorithm 1 of the optimization algorithm. First, we partition $\mathcal{T}_{\text{analysis}}$ and $\mathcal{T}_{\text{design}}$ using METIS [72]. Subsequently, we build PETSc matrices to enable the interaction between FreeFEM and PETSc. We define the finite element spaces, declare the unknowns, and define the weak forms for the PDEs. We summarize in Table 2 the physics-tailored preconditioners used for solving the PDEs. Within the optimization loop, the governing equations are solved to compute velocity, pressure, temperature, and displacement fields, and the objective and constraint values are computed. Next, the adjoint equations are solved to obtain adjoint variables, and sensitivities are computed. These sensitivities are then interpolated from $\mathcal{T}_{\text{analysis}}$ to $\mathcal{T}_{\text{design}}$. The design variables are updated using MMA, and the updated variables are interpolated from $\mathcal{T}_{\text{design}}$ to $\mathcal{T}_{\text{analysis}}$. Afterward, the analysis mesh $\mathcal{T}_{\text{analysis}}$ is remeshed for the subsequent optimization iteration simulation, while the density, velocity, and pressure fields are interpolated from the current iteration. The optimization algorithm continues to iterate until convergence or until reaching a maximum number of iterations.

Table 2
The preconditioners being used to solve the PDEs.

PDEs	Preconditioner
Navier-Stokes equations, cf. Eq. (8)	Modified Augmented Lagrangian preconditioner [73]
Convection-diffusion equation, cf. Eq. (11)	Additive Schwarz Method (ASM) [74]
Lamé equation, cf. Eq. (17)	Geometric Algebraic MultiGrid (GAMG) [75]
Adjoint Lamé equation, cf. Eq. (A.4a)	Geometric Algebraic MultiGrid (GAMG) [75]
Adjoint convection-diffusion equation, cf. Eq. (A.4b)	Additive Schwarz Method (ASM) [74]
Adjoint Navier-Stokes equations, cf. Eq. (A.4c)	Additive Schwarz Method (ASM) [74]
Reaction-diffusion equation, cf. Eq. (20)	High Performance reconditioner (hypre) [76]

Finally, it should be noted that when employing the density-based approach and relying on the Brinkman term to represent the fluid/solid phases, it is essentially needed to validate the performance using a body-fitted mesh. With the body-fitted meshes, it is possible to disjoint reunion of a fluid subdomain Ω_f and a solid subdomain Ω_s, separated by the fluid–solid interface $\Gamma_{\mathrm{s,f}}$. Hereinafter, we outline the governing equations used for the body-fitted simulation.

For the fluid flow, the Navier-Stokes equations without any body-force term are solved within Ω_f, with the no-slip boundary condition imposed on $\Gamma_{\mathrm{s,f}}$, as follows:

$$\begin{cases} -\operatorname{div}\left(\boldsymbol{\sigma}_\mathrm{f}(\boldsymbol{v},p)\right) + (\boldsymbol{v}\cdot\nabla)\boldsymbol{v} = 0 & \text{in } \Omega_\mathrm{f}, \\ -\operatorname{div}(\boldsymbol{v}) = 0 & \text{in } \Omega_\mathrm{f}, \\ \boldsymbol{v} = \boldsymbol{v}_0 & \text{on } \partial\Omega^D_{(\boldsymbol{v},p)}, \\ \boldsymbol{\sigma}_\mathrm{f}(\boldsymbol{v},p)\cdot\boldsymbol{n} = 0 & \text{on } \partial\Omega^N_{(\boldsymbol{v},p)}, \\ \boldsymbol{v} = 0 & \text{on } \Gamma_\mathrm{wall}\cup\Gamma_{\mathrm{s,f}}. \end{cases} \tag{26}$$

With the fluid velocity $\boldsymbol{v}$, we can then compute the temperature field T across the entire domain. This involves solving convection-diffusion equation within Ω_f, and pure diffusion equation within Ω_s. Both temperature and normal heat flux remain continuous across the interface $\Gamma_{\mathrm{s,f}}$ separating Ω_f and Ω_s. As a result, the conjugate heat transfer is governing by the following equations:

$$\begin{cases} \mathrm{Re}\,\mathrm{Pr}(\boldsymbol{v}\cdot\nabla T) - \operatorname{div}(\kappa_\mathrm{f}\nabla T) = 0 & \text{in } \Omega_\mathrm{f}, \\ -\operatorname{div}(\kappa_\mathrm{s}\nabla T) = 0 & \text{in } \Omega_\mathrm{s}, \\ T = T_0 & \text{on } \partial\Omega^D_T, \\ \kappa_\mathrm{f}\nabla T\cdot\boldsymbol{n}_\mathrm{f} = 0 & \text{on } \partial\Omega^N_T\backslash\partial\Omega_\mathrm{s}, \\ \kappa_\mathrm{s}\nabla T\cdot\boldsymbol{n}_\mathrm{s} = 0 & \text{on } \partial\Omega^N_T\backslash\partial\Omega_\mathrm{f}, \\ T_\mathrm{s} = T_\mathrm{f} & \text{on } \Gamma_{\mathrm{s,f}}, \\ -\kappa_\mathrm{f}\nabla T\cdot\boldsymbol{n}_\mathrm{f} = -\kappa_\mathrm{s}\nabla T\cdot\boldsymbol{n}_\mathrm{s} & \text{on } \Gamma_{\mathrm{s,f}}. \end{cases} \tag{27}$$

Lastly, the displacement $\boldsymbol{u}$ can be computed only inside the Ω_s using the following linear thermoelastic equation:

$$\begin{cases} -\operatorname{div}\left(\boldsymbol{\sigma}_\mathrm{s}(\boldsymbol{u},T)\right) = 0 & \text{in } \Omega_\mathrm{s}, \\ \boldsymbol{u} = \boldsymbol{u}_0 & \text{on } \partial\Omega^D_{\boldsymbol{u}}, \\ \boldsymbol{\sigma}_\mathrm{s}(\boldsymbol{u},T)\cdot\boldsymbol{n}_\mathrm{s} = \boldsymbol{t} & \text{on } \partial\Omega^N_{\boldsymbol{u}}. \end{cases} \tag{28}$$

Algorithm 1: Topology optimization of heat exchangers.

```
input: $\mathcal{T}_{\text{analysis}}$, $\mathcal{T}_{\text{design}}$
initialization
domain decomposition, create PETSc matrices
create finite element spaces, declare unknowns, define variational formulations
initialize the density fields $\gamma$
begin
    for it = 0; it < Maxloop; it = it + 1 do
        solve Navier-Stokes equations, cf. Eq. (8) to get $(\boldsymbol{v}, p)$
        solve convection-diffusion equation, cf. Eq. (11) to get $T$
        solve Lamé equation, cf. Eq. (17) to get $\boldsymbol{u}$
        compute objective and constraint values, cf. Eq. (24)
        if $\left\|J_{\text{it}+1} - J_{\text{it}}\right\| < \varepsilon$, $\left\|G_{\text{it}+1} - G_{\text{it}}\right\| < \varepsilon$ then
            break
        else
            solve adjoint Lamé equations, cf. Eq. (A.4a) (or Eq. (A.17a)) to get $\boldsymbol{u}_a$
            solve adjoint convection-diffusion equations, cf. Eq. (A.4b) (or Eq. (A.17b)) to get $T_a$
            solve adjoint Navier-Stokes equations, cf. Eq. (A.4c) (or Eq. (A.17c)) to get $(\boldsymbol{v}_a, p_a)$
            compute $\delta_{\hat{\gamma}}\mathcal{L}$, cf. Eq. (A.9)
            interpolate $\delta_{\hat{\gamma}}\mathcal{L}$ from $\mathcal{T}_{\text{analysis}}$ to $\mathcal{T}_{\text{design}}$
            compute sensitivity $\delta_{\gamma}\mathcal{L}$ on $\mathcal{T}_{\text{design}}$, cf. Eq. (20)
            update design variable $\gamma$ on $\mathcal{T}_{\text{design}}$ using MMA
            solve PDE-filter to get $\tilde{\gamma}$, cf. Eq. (20); Heaviside projection to get $\hat{\gamma}$, cf. Eq. (21)
            interpolate $\hat{\gamma}$ from $\mathcal{T}_{\text{design}}$ to $\mathcal{T}_{\text{analysis}}$
            if Remesh then
                centralize $\hat{\gamma}$ to a single MPI process
                call ParMmg to remesh $\mathcal{T}_{\text{analysis}_{\text{it}}}$ to $\mathcal{T}_{\text{analysis}_{\text{it}+1}}$ based on the metric
                domain decomposition, create PETSc matrices
                interpolate $\hat{\gamma}_{\text{it}}$, $\left(\boldsymbol{v}_{\text{it}}, p_{\text{it}}\right)$ from $\mathcal{T}_{\text{analysis}_{\text{it}}}$ to $\mathcal{T}_{\text{analysis}_{\text{it}+1}}$
            end
        end
    end
end
```

All the numerical experiments including topology optimization and performance validation were conducted on Rescale[1], using 1 node (60 MPI processes) equipped with AMD EPYC 7742 (Rome), operating at a clock frequency of 2.5 GHz, and with 456 GB of memory allocated.

4. Numerical investigations

In this section, we present various numerical examples, as depicted by the design model in Fig. 2b. We begin in Section 4.1 by outlining the optimization parameters utilized for these test cases, along with the rationale behind their selection. Following this, in Section 4.2, we provide a detailed demonstration of the optimization algorithm using one of these test cases, accompanied by performance validation using body-fitted meshes. Finally, in Section 4.3, we conduct a parameter study by showcasing the optimized results for all the test cases.

4.1. Optimization parameters

For all these test cases, the density field is initialized with a uniform value of $\gamma = 0.3$. We specify the Reynolds number as $\text{Re} = 100$, the Prandtl number as $\text{Pr} = 1.0$, and the ratio of thermal conductivity

---

[1] Rescale: A cloud-based platform for computational engineering and R&D

between solid and fluid phases as $\kappa_s/\kappa_f = 100$. The Young's modulus for the solid domain is set to $E_s = 1.0$, while for the void domain, it is $E_v = 10^{-6}$. The Poisson's ratio is set to 0.3. Furthermore, the maximum inverse permeability is set to $\alpha_{\max} = 10^4$. Lastly, the RAMP interpolation coefficients are set to $q_\alpha = 0.01$ and $q_\kappa = 1/30$, while the SIMP penalty factor is set to $q_{\mathbb{C}} = 3$. The selection of these coefficients and the penalty factor is based on finding a balance between reducing the stiffness of the optimization problem and maintaining accuracy w.r.t. the body-fitted mesh model. Our experience suggests that smaller values of q_α and q_κ, and a larger value of $q_{\mathbb{C}}$, tend to stiffen the optimization problem. Additionally, the moving limit of the MMA algorithm is set to 0.2.

For the design mesh $\mathcal{T}_{\text{design}}$, we choose a mesh size of 0.01, resulting in approximately 4.5×10^6 tetrahedral elements. For the analysis mesh $\mathcal{T}_{\text{analysis}}$, the initial mesh size is set to 0.025, resulting in a mesh with approximately 2.88×10^5 tetrahedral elements and approximately $2.13 \cdot 10^5$ unknowns for the flow equations. Mesh adaptation is triggered after 200 iterations and is activated every 5 iterations thereafter, and the optimization process is terminated after 500 iterations. The maximum mesh size of $\mathcal{T}_{\text{analysis}}$ is limited to 0.1, while the minimum mesh size is determined dynamically as $10^{-2} \times \left(1 - n_{\text{it}}/5000\right)$. The PDE-filter radius is set to $r \approx 2.4h$ where h is the edge size of $\mathcal{T}_{\text{design}}$.

Our primary focus lies in examining the effects of varying maximum allowed energy dissipation and mean compliance constraint. Therefore, across all test cases, we maintain a consistent maximum allowed volume fraction of the solid phase, set to $V_{\max} = 37\%$. Detailed values of $\Phi_{\max}$ and C_{DP} are summarized in Table 3. It is important to note that the design outcomes and observations are heavily influenced by the problem setup and parameters. Our goal is not to derive generalized conclusions; rather, we aim to demonstrate the effectiveness of the proposed algorithm. And given that some level of hyper-parameter tuning is necessary especially for the nonlinear problems, we do not aim to conduct seemingly exhausted parameter study.

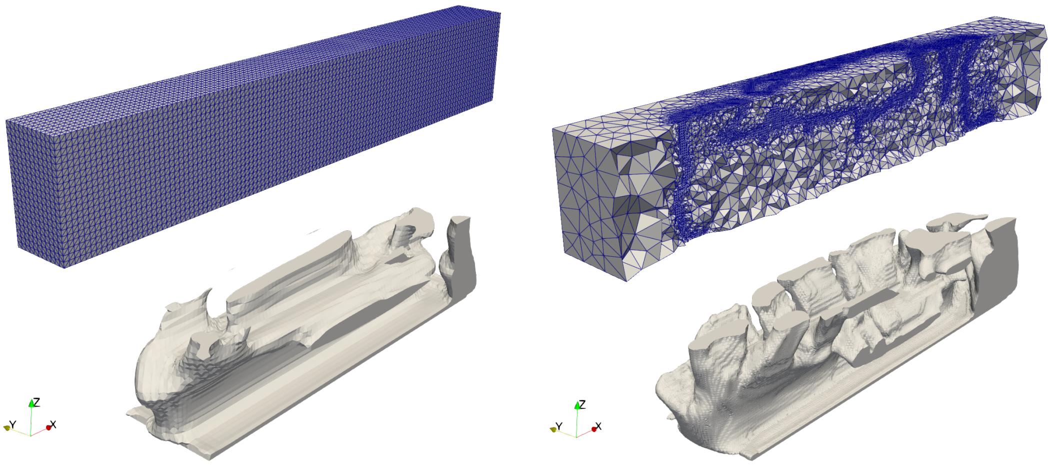
Table 3
The values of Re, $V_{\max}$, $\Phi_{\max}$, and C_{Comp} used for each test case.

Case #	Re	$V_{\max}$	$\Phi_{\max}$	C_{Comp}
#1, cf. Fig. 8a	100	37%	10.0	0.2
#2, cf. Fig. 8b	100	37%	10.0	0.7
#3, cf. Fig. 8c	100	37%	10.0	-
#4, cf. Fig. 8d	100	37%	5.0	0.2
#5, cf. Fig. 8e	100	37%	5.0	0.7
#6, cf. Fig. 8f	100	37%	5.0	-
#7, cf. Fig. 8g	100	37%	1.0	0.2
#8, cf. Fig. 8h	100	37%	1.0	0.7
#9, cf. Fig. 8i	100	37%	1.0	-

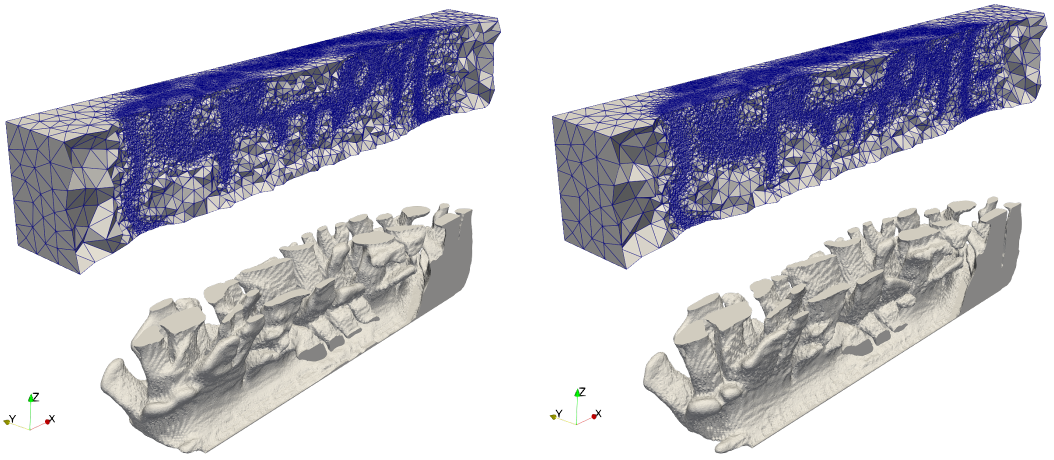
4.2. Demonstration of the TO algorithm

Before presenting the optimized results for all the test cases, we first provide details using one of them, Case #4. Several snapshots of the topological shape represented by the isovolume of $\hat{\gamma} = 0.5$ and the cross section view of the locally refined mesh are depicted in Fig. 3. Additionally, the convergence histories of the objective value and constraint values are plotted in Fig. 4.

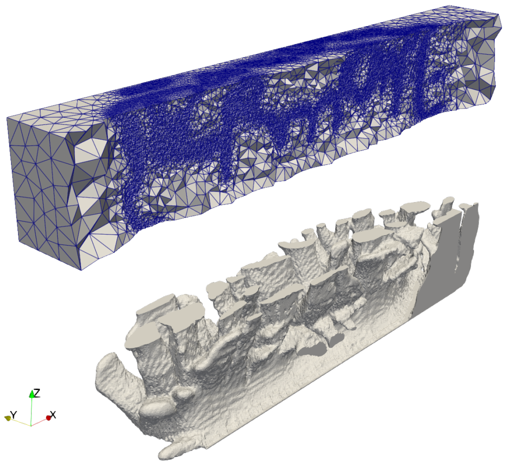
Significant observations can be made throughout the optimization process. Initially, within the first 30 iterations, there is a notable decrease in the volume of the solid phase, primarily aimed at fulfilling the fluid energy dissipation constraint by eliminating solid material from the bottom half section. Subsequently, around the 50th iteration mark, a prominent pin-fin structure begins to form in the middle, accompanied by the emergence of several major branches on the top half section. Concurrently, to adhere to the mean compliance constraint, supportive structures start to appear, characterized by plate-like extensions



(a) $n_{\text{it}} = 100$. (b) $n_{\text{it}} = 200$.

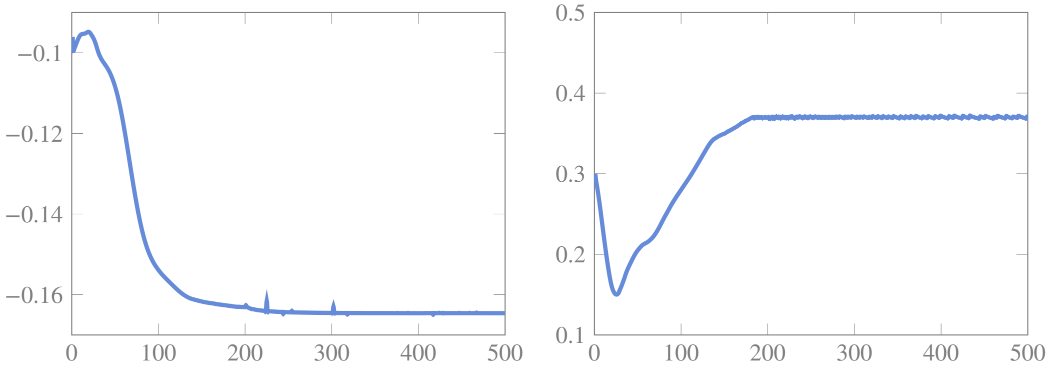


(c) $n_{\text{it}} = 300$. (d) $n_{\text{it}} = 400$.



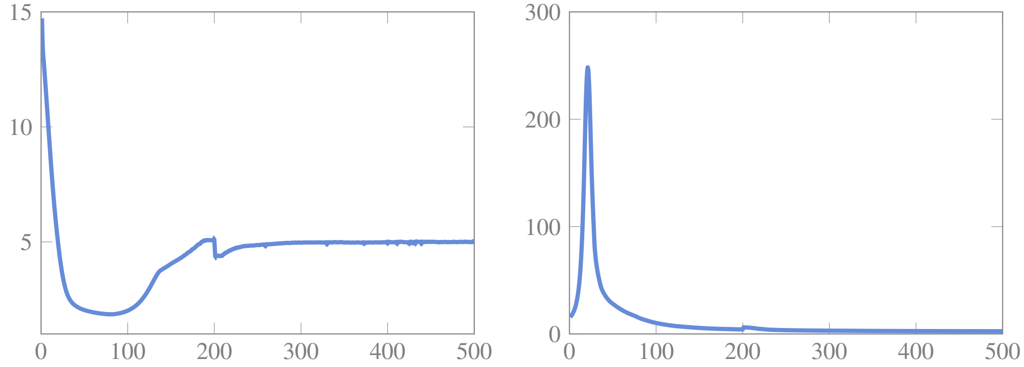
(e) $n_{\text{it}} = 500$.

Fig. 3: Evolutionary history of topological shape and mesh for Case #4. In each subfigure, the top figure displays a cross-sectional view of the mesh $\mathcal{T}_{\text{analysis}}$ at the xz-plane $y = 0.25$. Note that local mesh refinement is initiated at the 200th iteration. The bottom figure illustrates the isovolume of $\hat{\gamma} \leq 0.5$.



(a) Objective value, J.

(b) Volume fraction of solid phase, V_s.



(c) Energy dissipation, Φ.

(d) Mean compliance, Ψ.

Fig. 4: Convergence histories of (a) objective value, (b) volume fraction of the solid phase, (c) viscous energy dissipation, and (d) mean compliance for Case #4.

touching the top surface. Between iterations 100 and 200, the energy dissipation value gradually rises to its upper limit, while the mean compliance decreases to meet its maximum allowed value, along with an increase in the volume fraction reaching its upper limit. This phase also witnesses the generation of additional branches to augment the contacting area, thereby enhancing heat exchange as evidenced by the decrease in the objective value. Beyond the 200th iteration, most constraints are satisfactorily met, and the locally refined mesh facilitates the capture of finer features. However, small fluctuations persist in the convergence history, most likely attributed to the inherent high nonlinearity of the problem and the cost functions is rather sensitive to minor mesh variations due to the use of mesh adaptation. Despite these fluctuations, they appear to stabilize within a certain range, thereby we terminate the optimizer after 500 iterations.

Next, we assess the computational efficiency of the proposed TO framework by analyzing the runtime for Case #4. It is important to recall that the algorithm utilizes a fixed design mesh $\mathcal{T}_{\text{design}}$ for updating design variables and a dynamically refined mesh for physics computations $\mathcal{T}_{\text{analysis}}$, with mesh adaptation activated after 200 iterations. Before iteration 200, the degree of freedom (DOF) count for solving the fluid flow governing equation is approximately $2.13 \cdot 10^5$, increasing to $2.02 \cdot 10^6$ by iteration 500. To evaluate overall computational efficiency, we present in Table 4 a runtime breakdown for each finite element action. Additionally, Fig. 5 provides a pie chart showing the percentage distribution for a clearer understanding.

Table 4
Time breakdown of the finite element actions for Case #4 performed on Rescale HPC platform using AMD EPYC 7742 @ 2.5 GHz.

Action	Runtime
Solve Navier-Stokes equations	2 h34 min60 s
Solve convection-diffusion equation	10 min16 s
Solve linear elasticity equation	44 min31 s
Solve adjoint convection-diffusion equations	4 min26 s
Solve adjoint Navier-Stokes equations	2 h18 min41 s
Assemble sensitivities	2 min60 s
Update design variable using MMA	5 min2 s
PDE filtering and projection	9 min58 s
Centralize solution to one MPI process	2 min20 s
Distributed mesh adaptation (60 times)	13 min50 s
Mesh partitioning, create PETSc matrices	6 min45 s
Interpolate solution to the updated mesh	2 min43 s
Total	6 h36 min35 s

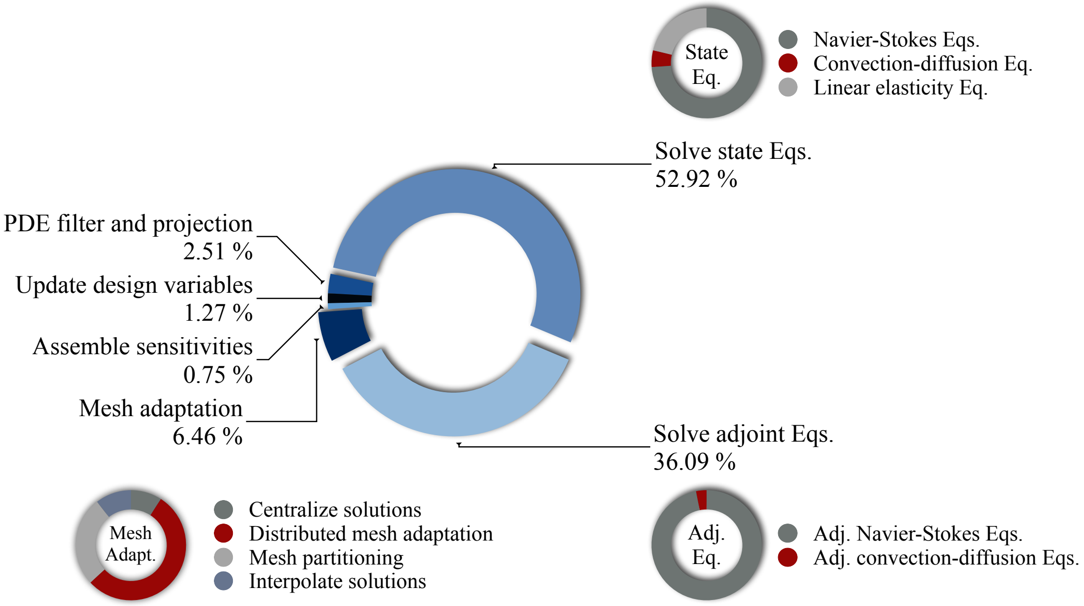


Fig. 5: Pie chart illustrating the percentage distribution of runtime breakdown allocated to the finite element actions.

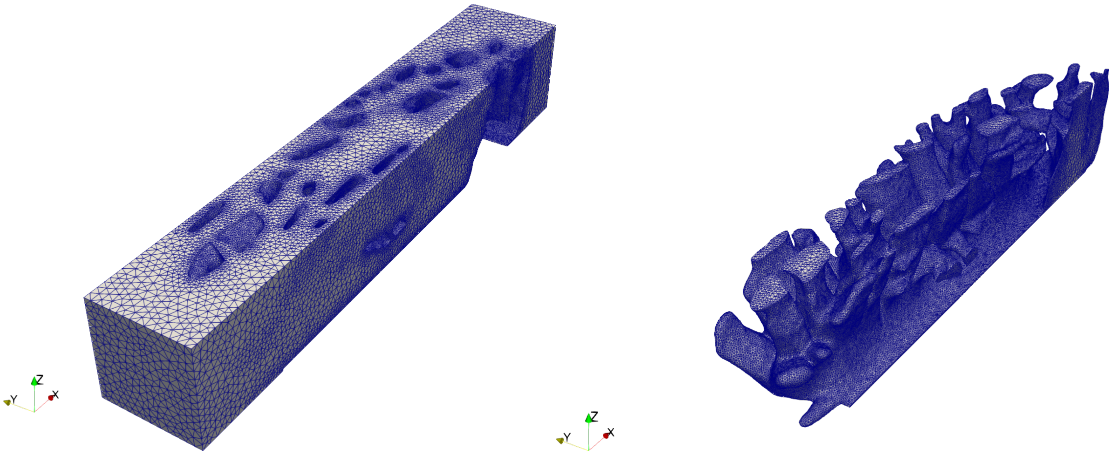
Examining the percentage of FE actions against total runtime reveals that solving the governing equations and their adjoint equations are the most computationally demanding tasks. Specifically, solving the Navier-Stokes equations and their adjoints equations for 39.1% and 35.0%, respectively, of the total runtime. It is worth noting that mesh adaptation significantly reduces this computational load; solving these PDEs on a fixed fine mesh would be considerably more resource-intensive, likely beyond the capacity of 60 MPI processes to achieve a similar level of high-resolution design in reasonable time.

Tasks performed on the fixed fine mesh, including updating design variables using MMA, PDE filtering, and projection operations, are notably efficient, collectively consuming 2.5% of the total runtime. However, mesh adaptation poses a significant challenge. While it is executed in a distributed manner using ParMmg, the global solution must first be centralized on a single MPI process for mesh adaptation. Once the new mesh is generated, it must be gathered again on a single MPI process before being partitioned for the next iteration's computations. This process could become a bottleneck when scaling the problem to larger sizes, potentially leading to memory leakage when attempting to centralize an ultra-large global mesh on a single core.

Overall, the scaling achieved is satisfactory, striking a good balance between executing complex multi-physics topology optimization and reasonable CPU time.

As discussed earlier, when employing the density-based approach and integrating the Brinkman term to delineate fluid/solid phases, it is essential to validate the performance using a body-fitted mesh. This goal of such validation is to examine that the density field has been evolved into a clear binary structure, and that the fluid flow does not intrude into the solid phase. Achieving these conditions confirms the accuracy of the topology optimization (TO) method, instilling confidence in its outcomes.

With body-fitted mesh, it allows the disjoint of a fluid subdomain Ω_f (as depicted in Fig. 6a) and a solid subdomain Ω_s (as depicted in Fig. 6b), with the fluid-solid interface $\Gamma_{\mathrm{s,f}}$ represented by the isosurface where $\tilde{\gamma} = 0.5$. Subsequently, we solve the governing equations outlined in Eqs. (26) to (28). The velocity, temperature, and displacement fields obtained from both the TO solver (left column) and the body-fitted solver (right column) are plotted in Fig. 7. It is noticeable that these distributions exhibit a good alignment with minor discrepancies in magnitude that fall within a reasonable relative error range, as demonstrated in Table 5.



(a) $\mathcal{T}_{\mathrm{fluid}}$.

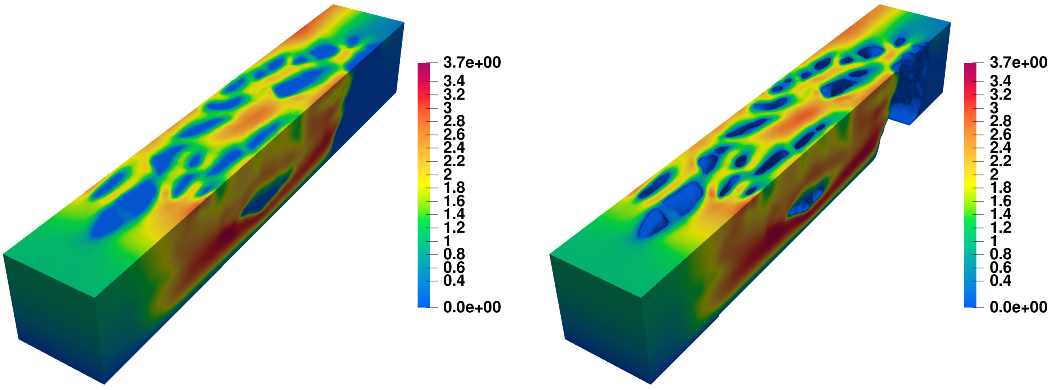
(b) $\mathcal{T}_{\mathrm{solid}}$.

Fig. 6: Body-fitted mesh obtained based on the isosurface of $\tilde{\gamma} = 0.5$ for Case #4.

Table 5

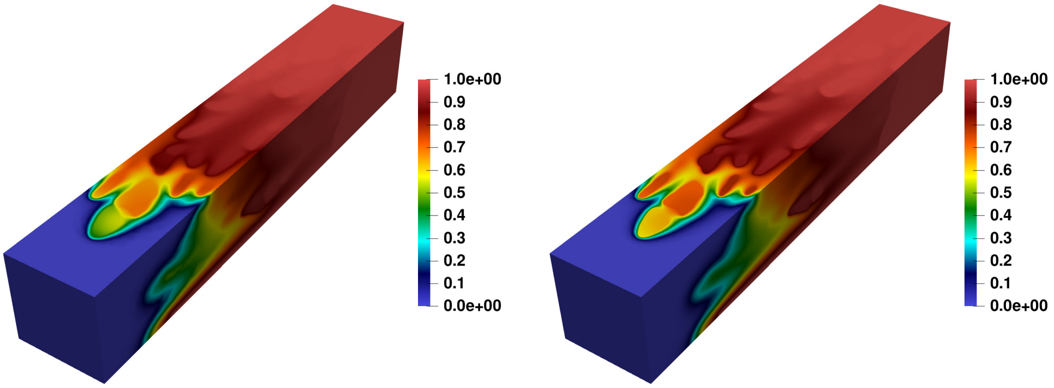
Performance validation comparing the TO solution with the body-fitted simulation using Case #4, cf. Fig. 7

Cost function	TO solver	Body-fitted	Relative error
Objective function J	-0.1646	-0.1652	0.36%
Energy dissipation Φ	5.01	5.097	1.7%
Mean compliance Ψ	2.218	2.104	5.1%



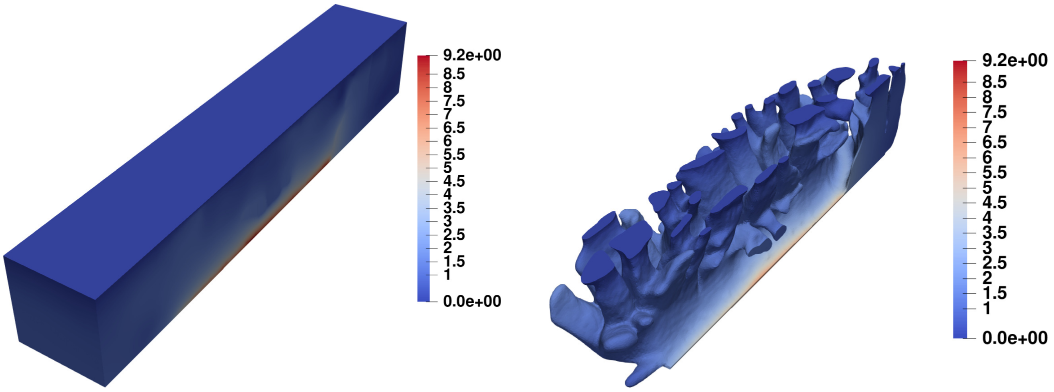
(a) Velocity field on h–adapted mesh $\mathcal{T}_{\text{analysis}}$.

(b) Velocity field on body-fitted mesh $\mathcal{T}_{\text{fluid}}$.



(c) Temperature field on h–adapted mesh $\mathcal{T}_{\text{analysis}}$.

(d) Temperature field on body-fitted mesh $\mathcal{T}_{\text{total}}$.



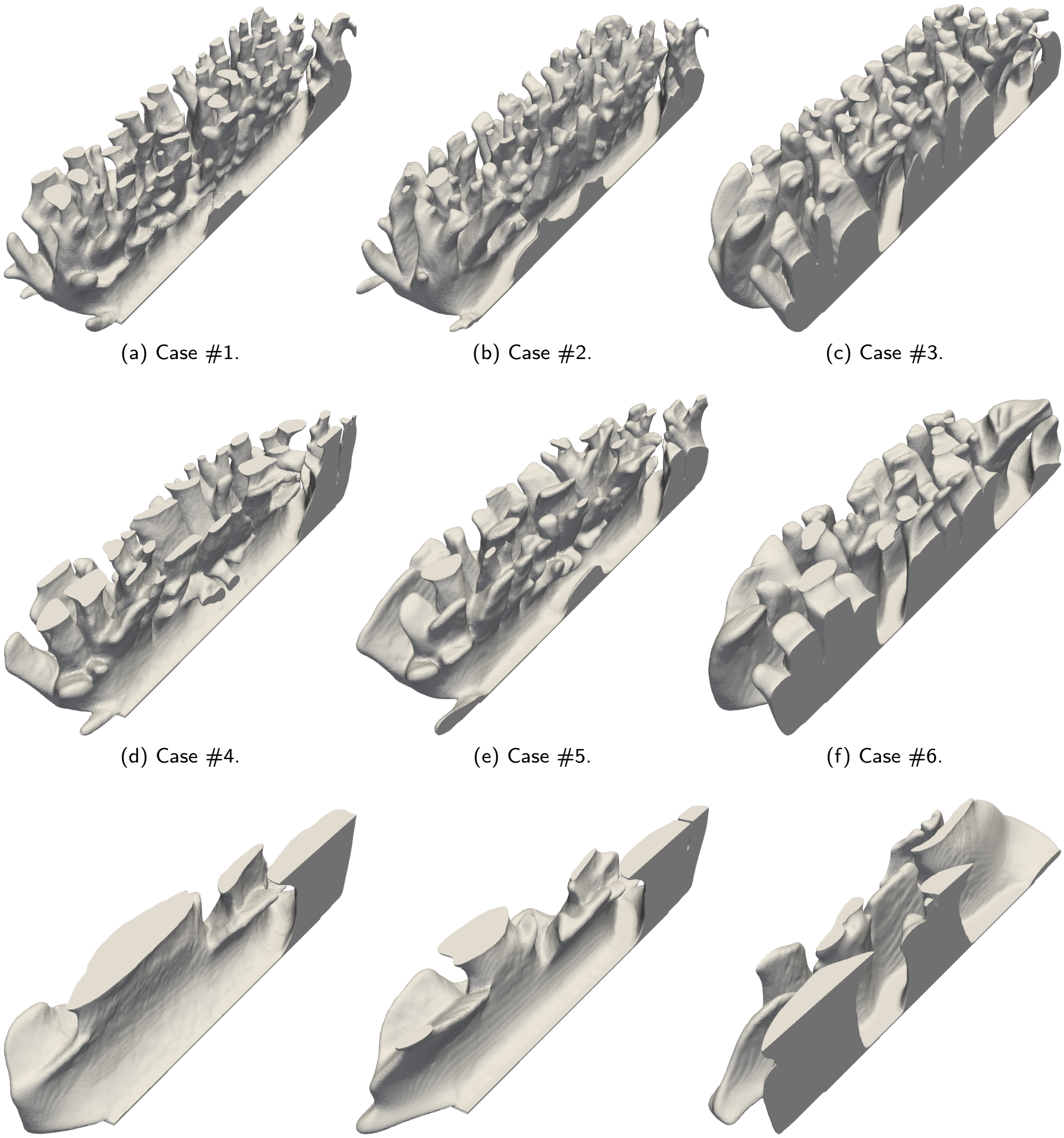
(e) Displacement field on h–adapted mesh $\mathcal{T}_{\text{analysis}}$.

(f) Displacement field on body-fitted mesh $\mathcal{T}_{\text{solid}}$.

Fig. 7: Velocity, temperature, and displacement fields computed on the h-adapted mesh $\mathcal{T}_{\text{analysis}}$ utilized for topology optimization at the 500th iteration, cf. Fig. 3e, alongside those obtained on the body-fitted mesh for performance validation, cf. Fig. 6.

4.3. Test cases

Now, we execute the TO solver followed by the body-fitted validation solver to tackle the test cases outlined in Table 3. The optimized outcomes for Cases #1–#9 are depicted in Fig. 8. In each row, the maximum permitted energy dissipation decreases from top to bottom, while in each column, the maximum allowed mean compliance increases from left to right (the rightmost column displays results obtained without imposing a mean compliance constraint). The calculated objective value, energy dissipation, and mean compliance obtained using the body-fitted validation solver are summarized in Table 6. The streamlines and temperature fields for the optimized results obtained with different maximum allowed energy dissipation values are shown in Figs. 9 and 11, respectively.


(a) Case #1. (b) Case #2. (c) Case #3.

(d) Case #4. (e) Case #5. (f) Case #6.

(g) Case #7. (h) Case #8. (i) Case #9.

Fig. 8: Optimized results for Cases #1–#9, represented by body-fitted meshes based on the isovolume of $\tilde{\gamma} \leq 0.5$. In each row, the maximum permitted energy dissipation decreases from top to bottom, while in each column, the maximum allowed mean compliance increases from left to right (the rightmost column displays results obtained without imposing a mean compliance constraint).

As illustrated in Fig. 8, the optimized outcomes showcase a remarkable diversity of forms, ranging from

Table 6
The calculated objective value, energy dissipation, and mean compliance obtained using the body-fitted validation solver for each test case.

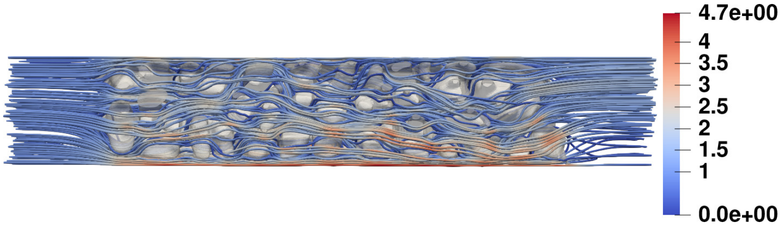
Case #	J	Φ	Ψ
#1, cf. Fig. 8a	-0.1658	10.154	2.181
#2, cf. Fig. 8b	-0.1659	10.118	9.721
#3, cf. Fig. 8c	-0.1665	10.147	-
#4, cf. Fig. 8d	-0.1652	5.097	2.104
#5, cf. Fig. 8e	-0.1653	5.068	7.187
#6, cf. Fig. 8f	-0.1657	5.084	-
#7, cf. Fig. 8g	-0.1486	0.993	2.988
#8, cf. Fig. 8h	-0.1517	1.001	7.507
#9, cf. Fig. 8i	-0.1547	1.048	-

simplistic, solitary shapes to intricate, branching structures. Notably, many of these configurations bear a striking resemblance to features found in natural ecosystems, such as coral reefs. Delving into the evolutionary mechanisms behind this resemblance reveals intriguing insights. The shapes and topologies of coral colonies are finely adapted to their unique growth environments, which including factors like fluid motion, nutrition in their surroundings, allowing them to behave very differently from one another and highlighting the remarkable diversity and resilience of these ecosystems [77]. For instance, corals situated in regions with strong water currents, like near reef edges or in channels, tend to adopt simpler, more compact forms to minimize drag [78]. This adaptation is evident in Cases #7–#9, where lower limits on viscous energy dissipation are imposed, cf. Fig. 9c. Conversely, in regions with calmer waters such as sheltered lagoons or shallow bays, corals have the opportunity to develop intricately branched and elaborate structures. This is because reduced water flow allows sediment particles to accumulate, providing a substrate for coral growth and fostering the formation of complex colonies [79]. This pattern is mirrored in Cases #1–#3, where higher energy dissipation encourages the development of irregular-shaped tentacles, facilitating the efficient dissipation of heat energy from the bottom hot wall to the surrounding coolants through these tentacles, see Fig. 9a. Moreover, the variability in coral morphology can be attributed to the unique challenges and opportunities presented by their specific growth environments. In some instances, corals must contend with limited access to light exposure, necessitating specialized shapes and structures to thrive [80]. In the context of our test cases, the distinct requirements arise from the need for load-bearing capability. Consequently, with a mean compliance constraint, loading paths emerge to distribute surface traction, often manifesting as flattened extensions in contact with the fixed wall.

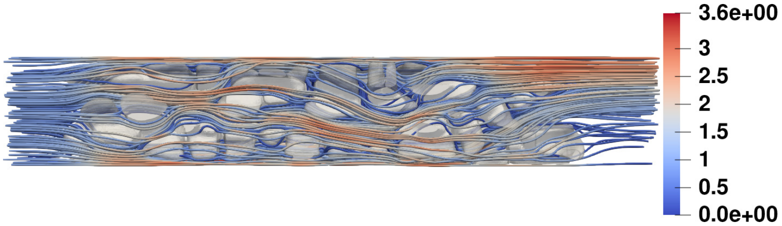
Upon closer examination of the computed cost function values presented in Table 6, we further normalize them to serve as performance indices $\mathrm{Idx}_i \in [0.5, 1]$, $i = 1, 2, \ldots, 9$, evaluating the trade-offs of thermal, hydraulic, and structural performance. This normalization is achieved using the following linear rescaling:

$$\mathrm{Idx}_i = \frac{|J_i| - \min_{i=1}^{n} |J_i|}{2\left(\max_{i=1}^{n} |J_i| - \min_{i=1}^{n} |J_i|\right)} + 0.5 \quad i = 1, 2, \ldots, 9, \tag{29}$$

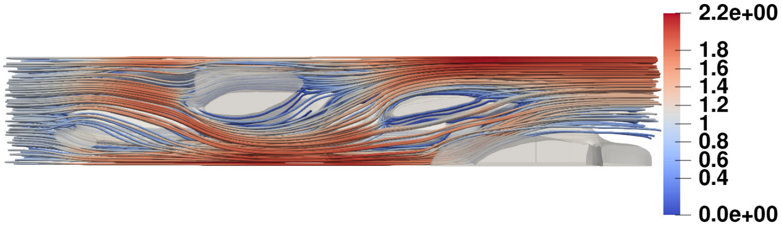
where the subscript denotes the case number. Fig. 10 showcases different trade-offs obtained across the nine test case. It is notable that the objective value achieved in the case without a mean compliance constraint slightly outperforms those with such a constraint. These insignificant differences suggest that the influence of the mean compliance constraint is quite trivial, although this observation may vary depending on changes in the problem setting. Another observation arises when considering cases with relatively lower energy dissipation, the upstream parts appear to have limited heat exchange between fluid and solid phases, as depicted in Figs. 11c and f. With an increase in the allowable energy dissipation, the heat transfer by fluid approaches its upper limit, and further increases in energy dissipation do not contribute significantly to the enhancement of heat transfer.



(a) Case #2.



(b) Case #5.



(c) Case #8.

Fig. 9: Streamlines for the optimized results obtained with different maximum allowed energy dissipation values but under the same mean compliance constraint.

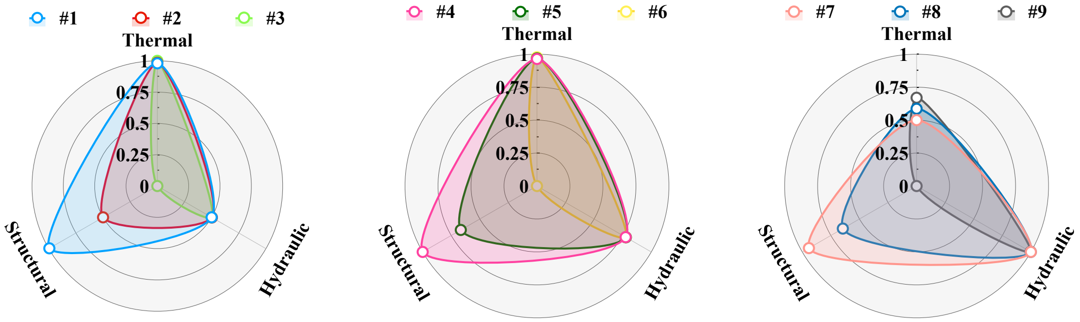
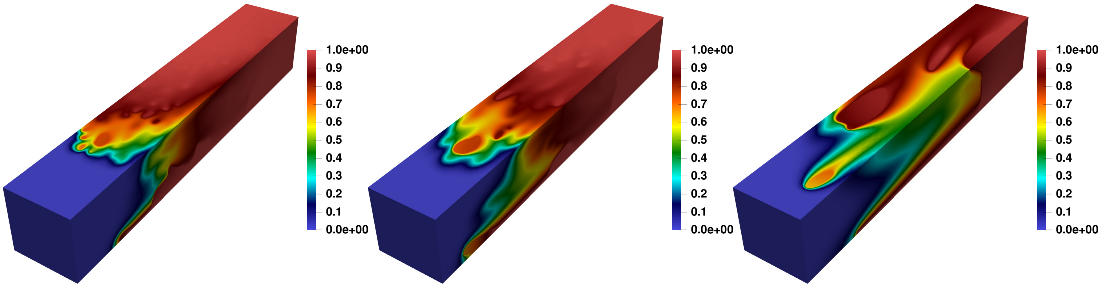
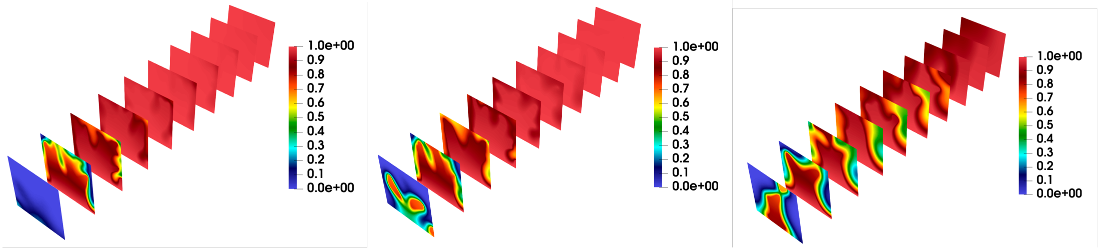


Fig. 10: Radar chart showcasing different trade-offs obtained by Cases #1–#9.



(a) Case #2. (b) Case #5. (c) Case #8.



(d) Case #2. (e) Case #5. (f) Case #8.

Fig. 11: Temperature fields for the optimized results obtained with different maximum allowed energy dissipation values.

5. A graphical user interface (GUI)

Tanatloc[2] is a multiphysics FEA software developed by our industrial partner Airthium[3]. It provides built-in models for both common physical problems, such as Poisson's problem and linear elasticity, and specialized ones, such as magnetostatics [81] and mechanical contact [82]. Users can create their own models or collaborate with experts to develop custom solutions tailored to specific needs, such as topology optimization. It features a user-friendly Graphical User Interface (GUI) that enables users to execute the TO algorithm presented in Algorithm 1. Users can input all relevant data, including geometry files, optimization parameters, boundary conditions and more, through the GUI. An overview of the developed Tanatloc window is presented in Fig. 12.

The user interface consists of severals panels and a menu bar where the user can load and visualize geometries, set up the TO problem parameters, execute the optimization case, visualize the simulation results in real time and postprocess the results, as shown in Figs. 13–14. Geometries are dropped through the "Geometry" panel and automatically processed to get the labels, as shown in Fig. 13a. Mesh refinement parameters can be defined for each geometry. Various input parameters can be introduced and adjusted through "Parameters" panel, as shown in Fig. 13b. These parameters include Reynolds number, Prandtl number, the maximum allowed volume fraction of the solid phase, viscous energy dissipation, etc. Boundary conditions are set up through "Boundary conditions" panel, as shown in Fig. 13c. Boundaries can be selected directly in the GUI for applying corresponding boundary conditions, and labels are automatically mapped to the meshes labels. Computing server can be selected through the "Run" panel. The simulation can be run directly on the host computer, on Rescale, or on a specific cluster. For example, as depicted in Fig. 13d, the users are allowed to choose their preferred core type offered by Rescale.

Once the problem is configured as described above, Tanatloc generate the Gmsh [83] and the FreeFEM [68]

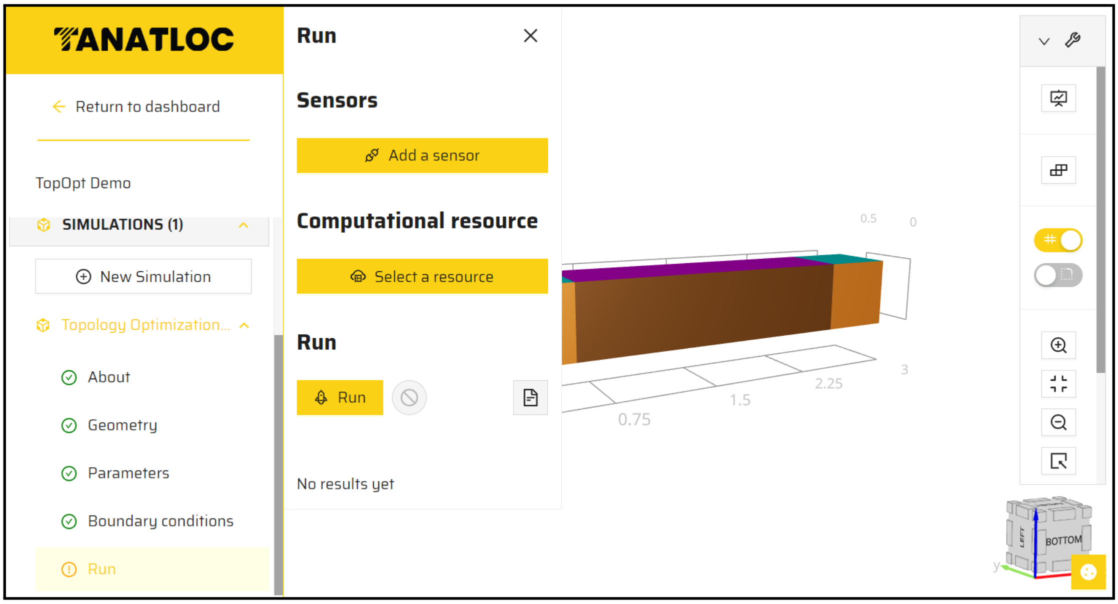
---

[2] Tanatloc: An easy-to-use FEA software that allows you to implement your own physics models.

[3] Airthium: A heat pump to decarbonize the planet.

scripts using a template engine, then run the mesh build and the FreeFEM simulation on the selected computing server. Once the computation is terminated, the obtained design is shown in the Graphic Panel, as shown in Fig. 14. The velocity, temperature, iterative history, and isosurface can be viewed by selecting the corresponding field in the GUI. The meshes and results can be visualized and manipulated in the graphic panel, allowing rotation, movement, and other operations like section view. Post-processing operations are available in the GUI, Tanatloc uses internally Paraview python script to obtain the post-processed result. Tanatloc can export the results in VTK format for post-processing using ParaView [84], and in the Inria Medit format, which is natively parsed by FreeFEM and Mmg, for subsequent simulations such as performance validation using a body-fitted mesh.

The initial meshes need to be generated using Gmsh [83] and exported to the MSH2 format, which can be read by FreeFEM. Note that the current version of FreeFEM only supports 2D triangular and 3D tetrahedral meshes.



No results yet

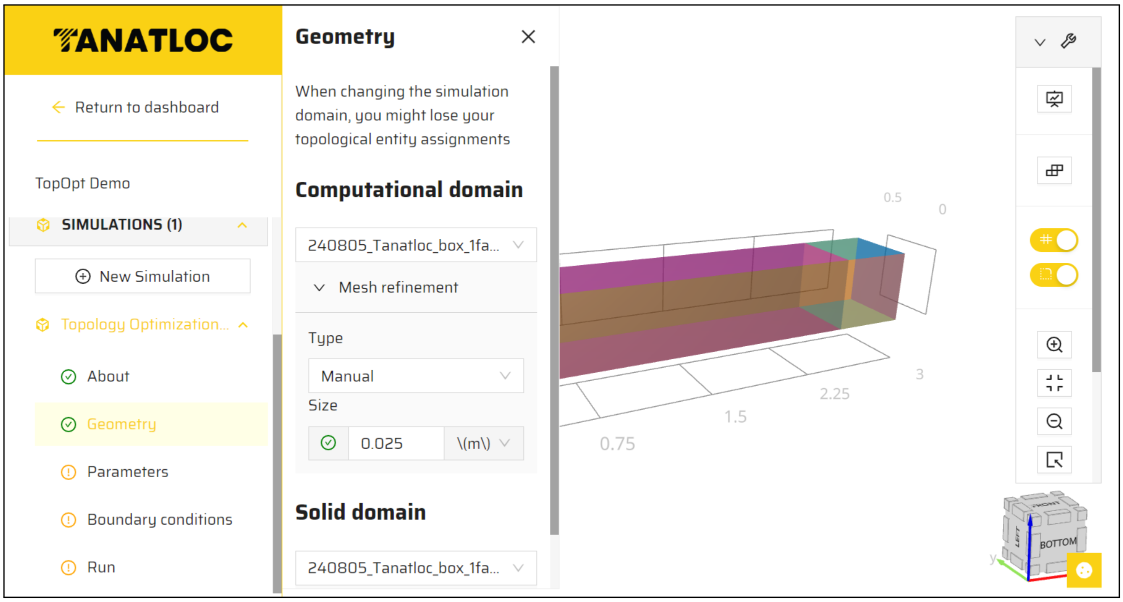
Fig. 12: Tanatloc window.

6. Fabrication of metal-based prototypes

With the rapid development of topology optimization approaches and the growing industry trends surrounding "3D printing" technologies, the gap between innovative design and advanced manufacturing has significantly narrowed [85]. In this section, we leverage additive manufacturing techniques to bring our optimized designs into prototypes, demonstrating the practical application of these advanced manufacturing methods.

6.1. Additive manufacturing of the prototype

The optimized structures obtained from our numerical simulations were converted into stereolithography (STL) format as shown in Fig. 15a using ParaView [84] and they were further processed using Simpleware ScanIP (Synopsis Inc., UK) before they were ready for additive manufacturing. To showcase the process, we specifically fabricated the design from Case #4. This was realized using Selective Laser Melting (SLM) method with a powder material comprising AlSi10Mg (Toyo Aluminium K.K., Japan), with particle diameters ranging between 10 to 45 µm. The chemical composition of this powder is listed in Table 7.



Geometry

When changing the simulation domain, you might lose your topological entity assignments

Computational domain

240805_Tanatloc_box_1fa...

Mesh refinement

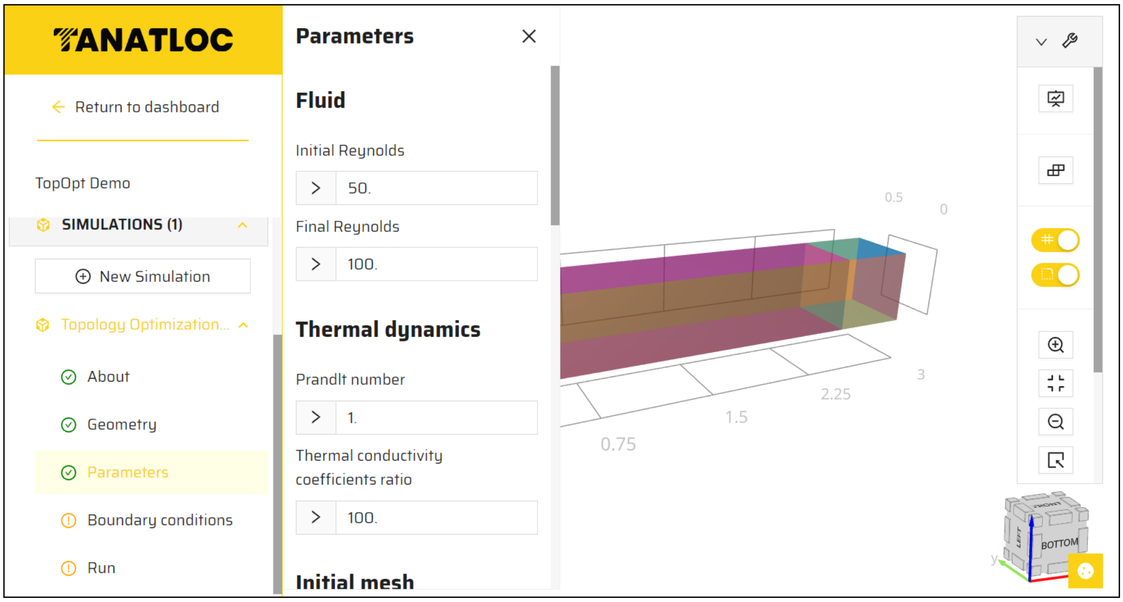
Type: Manual

Size: 0.025 \(m\)

Solid domain

240805_Tanatloc_box_1fa...

(a) Geometry and meshing.



Parameters

Fluid

Initial Reynolds: 50.

Final Reynolds: 100.

Thermal dynamics

Prandlt number: 1.

Thermal conductivity coefficients ratio: 100.

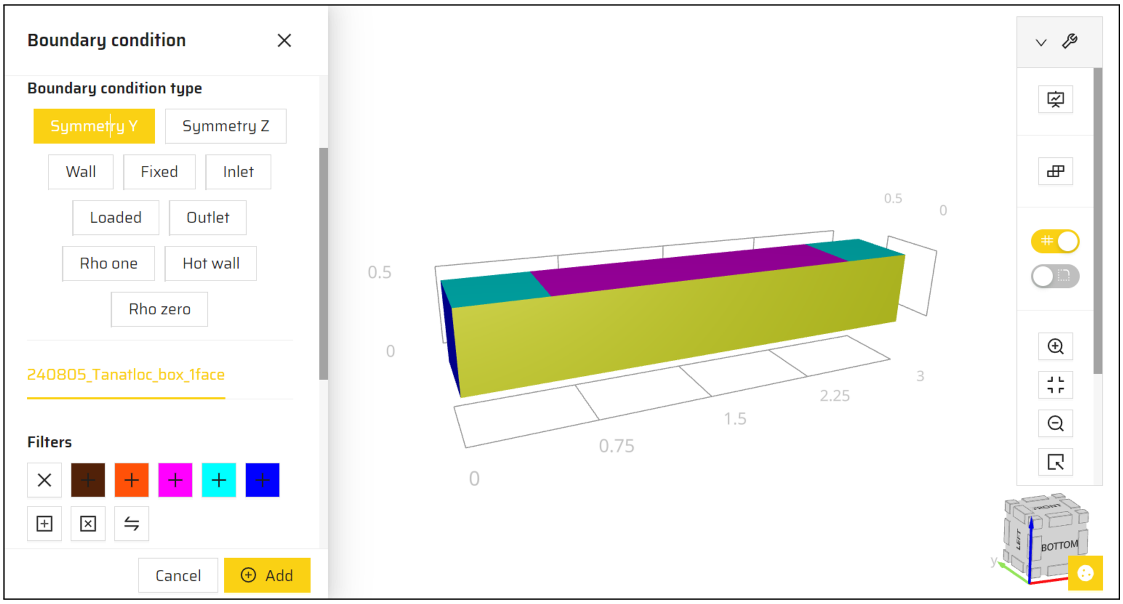
Initial mesh

(b) Input parameter.

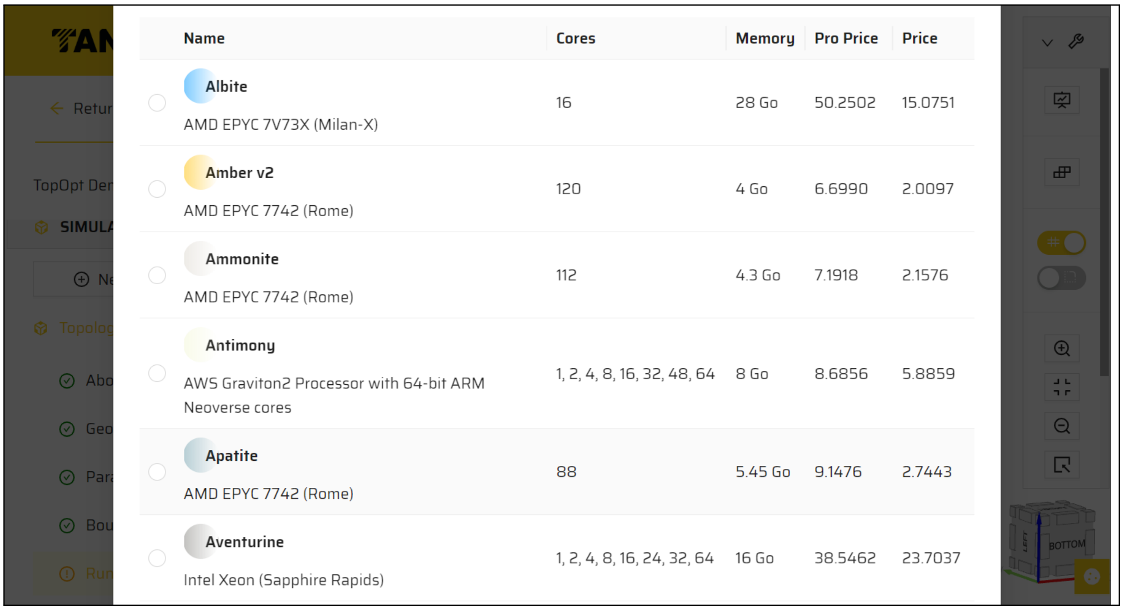
Fig. 13: Panels for input optimization data (Part 1).

Table 7
Chemical composition of powder.

Element	Si	Mg	Fe	Mn	Cu	Zn	Ti	Al
Contents	$9.0 - 11.0$	$0.2 - 0.45$	< 0.55	< 0.45	< 0.05	< 0.1	< 0.15	Bal.



(c) Allocate boundary conditions.



	Name	Cores	Memory	Pro Price	Price
○	**Albite** AMD EPYC 7V73X (Milan-X)	16	28 Go	50.2502	15.0751
○	**Amber v2** AMD EPYC 7742 (Rome)	120	4 Go	6.6990	2.0097
○	**Ammonite** AMD EPYC 7742 (Rome)	112	4.3 Go	7.1918	2.1576
○	**Antimony** AWS Graviton2 Processor with 64-bit ARM Neoverse cores	1, 2, 4, 8, 16, 32, 48, 64	8 Go	8.6856	5.8859
○	**Apatite** AMD EPYC 7742 (Rome)	88	5.45 Go	9.1476	2.7443
○	**Aventurine** Intel Xeon (Sapphire Rapids)	1, 2, 4, 8, 16, 24, 32, 64	16 Go	38.5462	23.7037

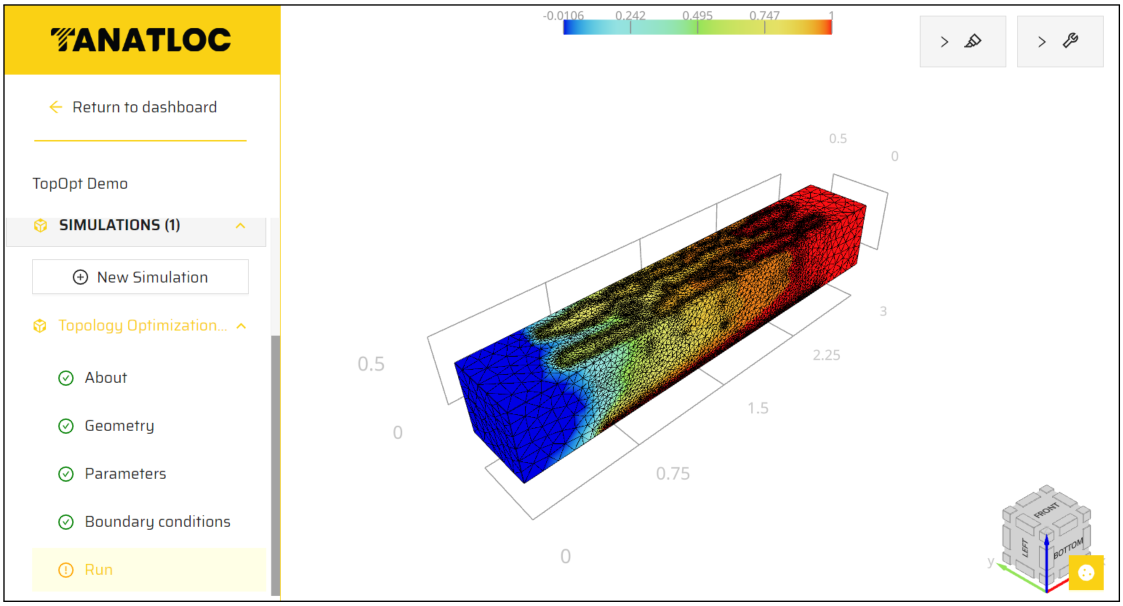
(d) Select computing resource.

Fig. 13: Panels for input optimization data (Part 2).

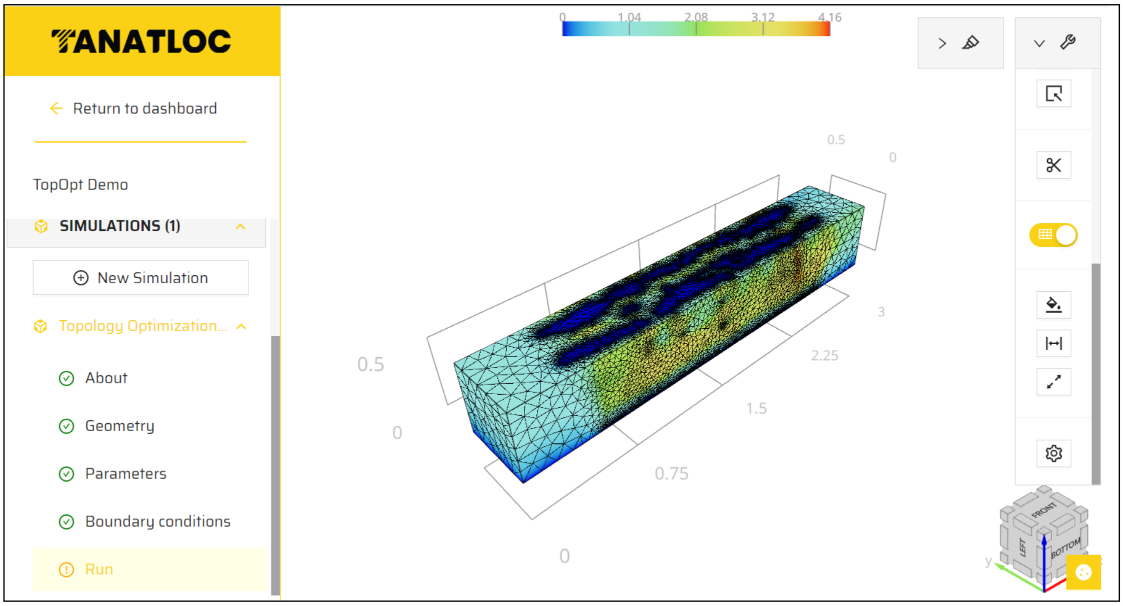
Our industrial collaborator, DENSO Corporation[4], utilized SLM methods to produce prototypes shown in Figs. 15b and c, depicting the prototypes before and after blasting treatment, respectively. Specialized manufacturing parameters tailored for this material were implemented. These parameters were optimized to yield components with relative densities approaching 100%, ensuring comparable mechanical properties to those obtained through conventional methods such as molding or machining. A Laser Powder Bed

---

[4] DENSO Corporation: A global automotive components manufacturer.



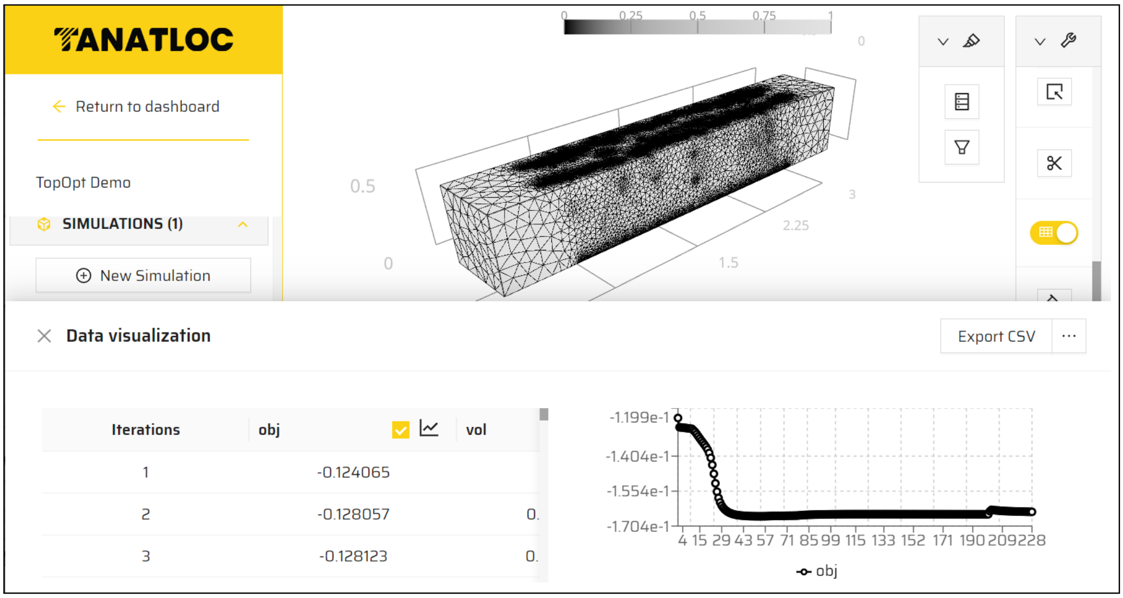
(a) Temperature field.



(b) Velocity field.

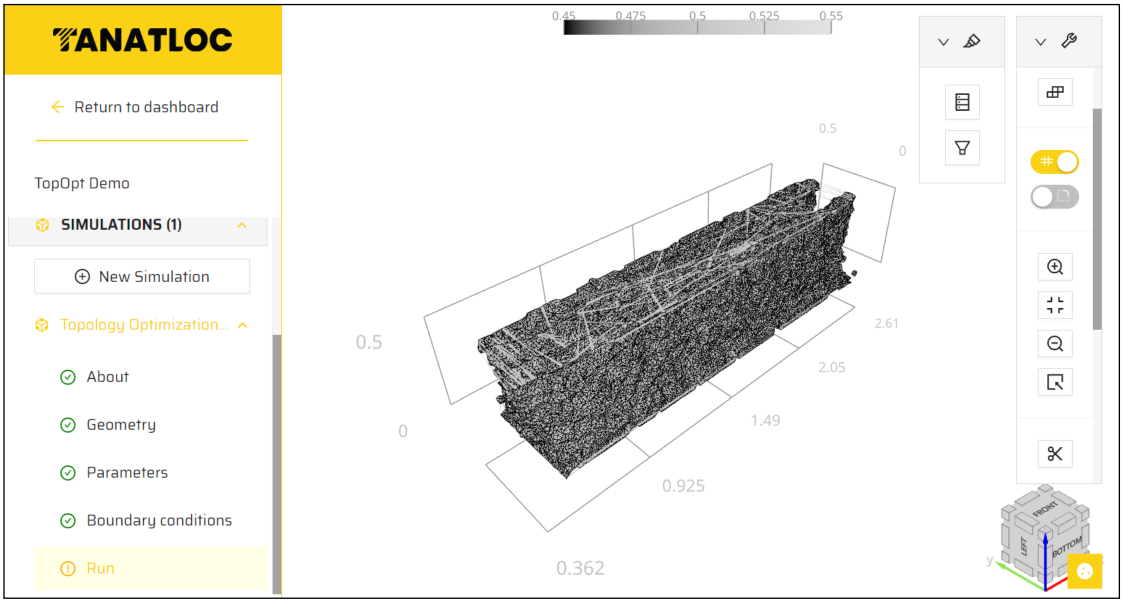
Fig. 14: Panels for output optimization data (Part 1).

Fusion (LPBF) machine (EOSM400-4, EOS) with an energy density of $40-60\ \mathrm{J\,m^{-3}}$ was employed for the manufacturing process. The production chamber was maintained at a temperature of 165 °C, with Nitrogen used as a protective gas to prevent oxidation and explosions. The material was deposited layer by layer, following the, with each layer measuring 0.03 µm in thickness (resulting in 738 layers per sample). Temporary supports generated by Materialise Magics e-Stage (Materialise N.V., Belgium) were utilized to secure the test-pieces during production. The total fabrication time was 6 hours and 53 seconds. Residual stress relief annealing was not performed, but shot peening was conducted to remove the



Iterations	obj	vol
1	-0.124065	
2	-0.128057	0.
3	-0.128123	0.

(c) Iterative history.



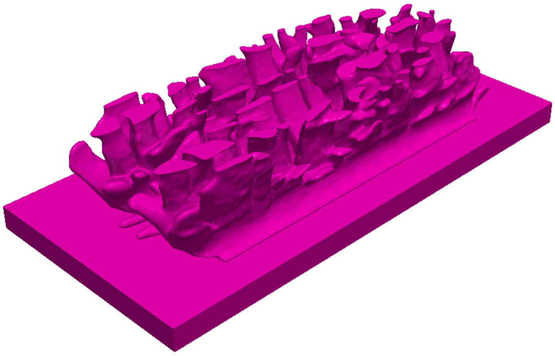
(d) Isosurface of $\hat{\gamma} \leq 0.5$.

Fig. 14: Panels for output optimization data (Part 2).

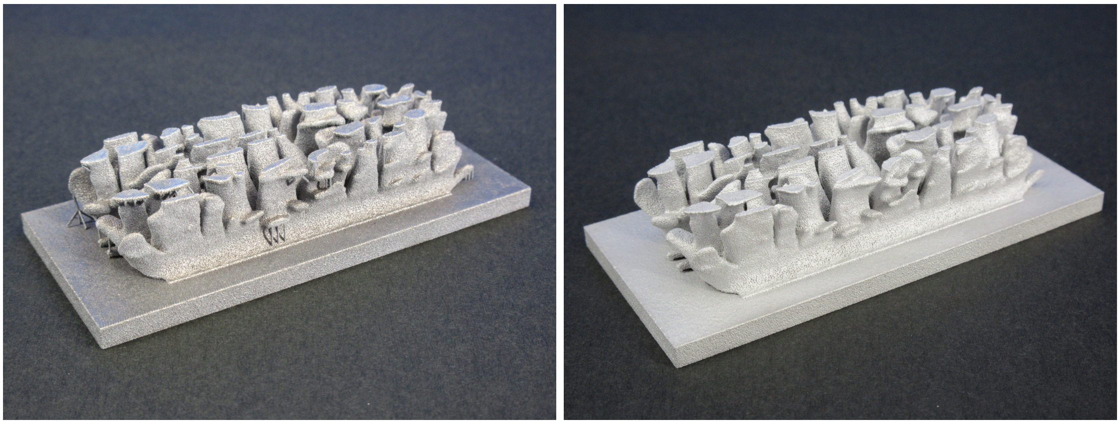
supports.

6.2. CT scans and reconstruction of CAD model

Once the prototype was printed, we conducted a reverse engineering [86] for the reconstruction of the CAD model based on the prototype, facilitating a thorough experimental and numerical validation using commercial CFD software, which is targeted in our future work. It should be pointed out that there is a notable lack of experimental validation of thermal-fluid topology optimization (TO) in existing works,



(a) STL data.



(b) Prototype before blasting treatment.

(c) Prototype after blasting treatment.

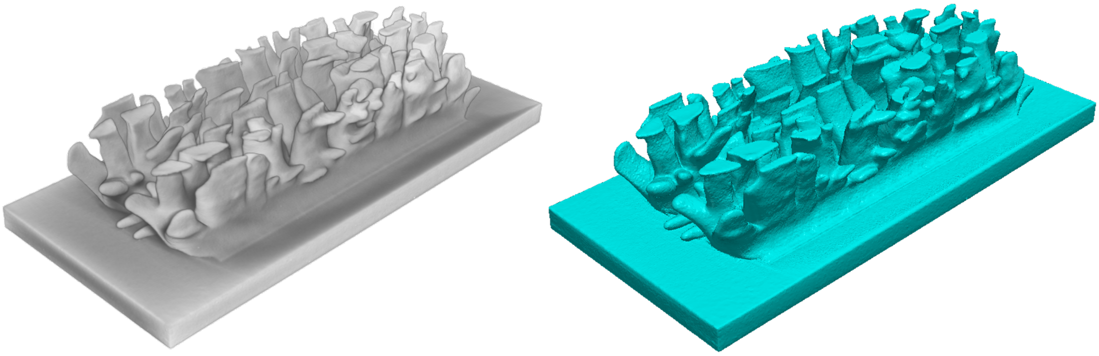
Fig. 15: Photographs of metal-based prototype for the Case #4, cf.Fig. 8d.

which are often limited to 2D designs [12, 87], with even fewer examples in full 3D [46, 88]. We used a Computed Tomography (CT) scanning machine (TXScanner 33000, Toshiba) to capture precise 3D images of the printed prototype, as depicted in Fig. 16a. Subsequently, the CT data was exported in DICOM format and further processed using Simpleware ScanIP. This processing encompassed segmentation of the CT-scan images, mask development, contour smoothing for each slice, and surface reconstruction. The resulting CAD data from the CT-scan images is illustrated in Fig. 16b. Following the reconstruction, we conducted an accuracy analysis of this 3D reconstruction. We registered the two datasets (from the original printing data and the reconstructed CAD model) and performed superimposition and surface deviation analyses using Simpleware ScanIP, as depicted in Fig. 17. The root mean square error (RMSE) was $9.799 \cdot 10^{-2}$ mm and the mean deviation was $-2.388 \cdot 10^{-2}$ mm, indicating a high level of accuracy.

7. Conclusions

To conclude, this paper presents an integrated workflow tailored for the three-dimensional topology optimization of conjugate heat transfer problems. Through leveraging advanced algorithms, optimization techniques, and computational tools, this framework facilitates the generation of high-resolution designs. Additionally, we provide a graphical user interface, enabling seamless prototyping and performance validation. The primary findings are summarized as follows:

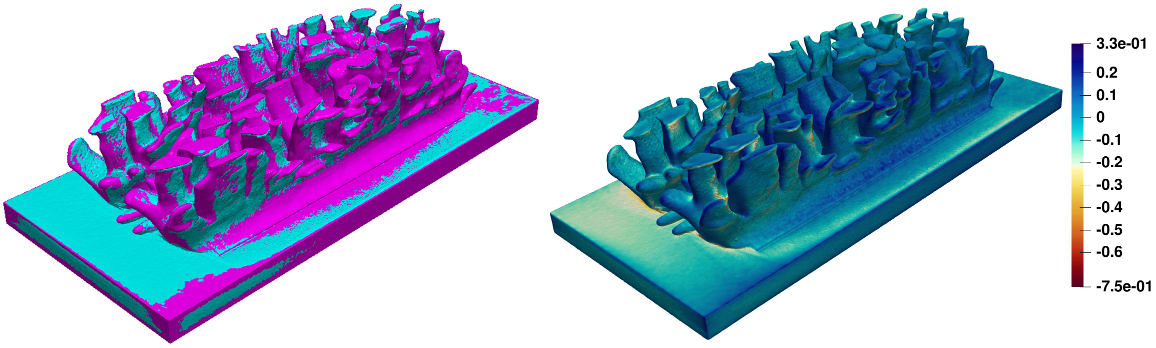
1. By incorporating a mean compliance constraint into the thermofluidic design problem formulation, we ensure both the integrity and load-bearing capability of the optimized structures. Through



(a) CT-scan images.

(b) Reconstructed CAD model.

Fig. 16: Reverse engineering for the printed prototype.



(a) Superimposition (overlay).

(b) Surface deviation analysis result.

Fig. 17: Analysis of accuracy of reconstructed CAD model w.r.t. the data used for 3D printing.

numerous numerical test cases with varied constraint values, we illustrate distinct trade-offs, reminiscent of the growth mechanisms observed in natural ecosystems, such as coral. These trade-offs are effectively visualized using radar diagrams to emphasize their performance metrics. Our case studies suggest that, in these specific scenarios, the influence of the mean compliance constraint seems to be rather insignificant.

2. A dual-mesh approach is used in the density-based TO framework. This allows us to achieve feature-rich designs represented by the pseudo-density fields on a high-resolution fixed mesh comprising approximately 4.5 million tetrahedral elements. Concurrently, forward and adjoint analyses are conducted on a locally refined mesh, notably reducing the main computational burden. Runtime analysis confirms the good computational efficiency of the constructed framework.
3. A user-friendly and easy-to-navigate graphical user interface, Tanatloc, is developed in JavaScript. It offers versatile functionalities and an interactive user experience for thermal engineers to explore and engage with advanced topology optimization algorithms.
4. A 3D printed metal-based prototype is fabricated, followed by the reconstruction of the CAD model through CT-scan images. Through the surface deviation analyses, a high level of accuracy of the fabricated prototype has been confirmed. This process paves the way for practical applications and further experimental investigations.

Below are the primary limitations of the current work, along with prospects targeted in future works:

1. The current implementation necessitates centralizing the solution on a single core before employing ParMmg for distributed mesh adaptation. Subsequently, the updated mesh is partitioned and redistributed to all MPI processes. This part of the solver limits the feasibility of ultra-high-resolution topology optimization involving billions of DOFs. Future efforts will focus on developing a fully-distributed framework.
2. Incorporating manufacturability constraints, such as overhang constraints, is vital for addressing more practical and industrial cases. Furthermore, the experimental investigations are planned.
3. Improving the graphical user interface to cover a broader spectrum of conjugate heat transfer problems, such as designing active/passive heat sinks and achieving flow uniformity distribution, will significantly boost the interface's versatility and applicability.

Supplementary Information

No supplementary information is used in this paper.

Acknowledgement

This work is supported by the Heat Exchanger R&D Division, DENSO CORPORATION. Furthermore, Hao Li and Joe Alexandersen acknowledge financial support from the European Union through the Marie Skłodowska-Curie Actions Postdoc Fellowship (no. 101106842). They also extend their gratitude to the Danish Center for Applied Mathematics and Mechanics (DCAMM) for financially supporting a short visit from Pierre Jolivet.

The first author also sincerely thanks Minghao Yu for his contribution to integrating the MMA algorithm into the workflow and for engaging in fruitful discussions. The authors are also grateful to Professor Atsushi Suzuki from Osaka University for his valuable insights on formulation and source code compilation on Rescale Japan HPC platform. Finally, the authors express their appreciation to Shiya Zhao from Kyoto University for her inputs regarding data analysis and visualization.

Declarations

Conflict of interest

The authors declare that they have no competing financial interests for this paper.

Replication of results

A detailed procedure and flowchart of the proposed method have been presented in Section 3 and one can follow them and reproduce the results. In case of further queries, please contact the corresponding author(s).

Author contributions

Hao Li (`hli@sdu.dk`): Conceptualization, Methodology, Software (FreeFEM-PETSc-ParMmg framework), Validation, Writing – original draft. Simon Garnotel (`simon.garnotel@airthium.com`): Software (Tanatloc user interface), Validation, Writing – Review & Editing. Pierre Jolivet (`pierre@joliv.et`): Software (FreeFEM-PETSc interface), Validation, Writing – Review & Editing. Hiroshi Ogawa (`hiroshi.ogawa.j3j@jp.denso.com`): Validation, prototyping, reverse engineering, Funding acquisition, Writing – Review & Editing. Tsuguo Kondoh: Methodology (optimization formulation), Supervision, Writing-Review. Kozo Furuta (`furuta.kozo.5n@kyoto-u.ac.jp`): Methodology, Validation, Writing-Review. Joe Alexandersen (`joal@sdu.dk`): Methodology (adjoint sensitivity analysis), Supervision, Writing – Review & Editing. Shinji Nishiwaki (`shinji@prec.kyoto-u.ac.jp`): Conceptualization, Supervision, Writing – Review, Funding acquisition.

References

[1] Edreis Edreis and A Petrov. Types of heat exchangers in industry, their advantages and disadvantages, and the study of their parameters. In *IOP Conference Series: Materials Science and Engineering*, volume 963, page 012027. IOP Publishing, 2020.

[2] Martin Philip Bendsøe and Noboru Kikuchi. Generating optimal topologies in structural design using a homogenization method. *Computer Methods in Applied Mechanics and Engineering*, 71(2):197–224, 1988.

[3] Talib Dbouk. A review about the engineering design of optimal heat transfer systems using topology optimization. *Applied Thermal Engineering*, 112:841–854, 2017.

[4] Joe Alexandersen and Casper Schousboe Andreasen. A review of topology optimisation for fluid-based problems. *Fluids*, 5(1):29, 2020.

[5] Ahmad Fawaz, Yuchao Hua, Steven Le Corre, Yilin Fan, and Lingai Luo. Topology optimization of heat exchangers: A review. *Energy*, page 124053, 2022.

[6] Ercan M Dede. Multiphysics topology optimization of heat transfer and fluid flow systems. In *proceedings of the COMSOL Users Conference*, 2009.

[7] Gil Ho Yoon. Topological design of heat dissipating structure with forced convective heat transfer. *Journal of Mechanical Science and Technology*, 24(6):1225–1233, 2010.

[8] Adriano A Koga, Edson Comini C Lopes, Helcio F Villa Nova, Cícero R De Lima, and Emílio Carlos Nelli Silva. Development of heat sink device by using topology optimization. *International Journal of Heat and Mass Transfer*, 64:759–772, 2013.

[9] Tadayoshi Matsumori, Tsuguo Kondoh, Atsushi Kawamoto, and Tsuyoshi Nomura. Topology optimization for fluid–thermal interaction problems under constant input power. *Structural and Multidisciplinary Optimization*, 47(4):571–581, 2013.

[10] Yuki Sato, Kentaro Yaji, Kazuhiro Izui, Takayuki Yamada, and Shinji Nishiwaki. An optimum design method for a thermal-fluid device incorporating multiobjective topology optimization with an adaptive weighting scheme. *Journal of Mechanical Design*, 140(3), 2018.

[11] Hao Li, Xiaohong Ding, Fanzhen Meng, Dalei Jing, and Min Xiong. Optimal design and thermal modelling for liquid-cooled heat sink based on multi-objective topology optimization: An experimental and numerical study. *International Journal of Heat and Mass Transfer*, 144:118638, 2019.

[12] Shi Zeng, Bugra Kanargi, and Poh Seng Lee. Experimental and numerical investigation of a mini channel forced air heat sink designed by topology optimization. *International Journal of Heat and Mass Transfer*, 121:663–679, 2018.

[13] Jan HK Haertel, Kurt Engelbrecht, Boyan S Lazarov, and Ole Sigmund. Topology optimization of a pseudo 3d thermofluid heat sink model. *International Journal of Heat and Mass Transfer*, 121:1073–1088, 2018.

[14] Suna Yan, Fengwen Wang, Jun Hong, and Ole Sigmund. Topology optimization of microchannel heat sinks using a two-layer model. *International Journal of Heat and Mass Transfer*, 143:118462, 2019.

[15] Tao Zeng, Hu Wang, Mengzhu Yang, and Joe Alexandersen. Topology optimization of heat sinks for instantaneous chip cooling using a transient pseudo-3D thermofluid model. *International Journal of Heat and Mass Transfer*, 154:119681, 2020.

[16] Akihiro Takezawa, Xiaopeng Zhang, Masaki Kato, and Mitsuru Kitamura. Method to optimize an additively-manufactured functionally-graded lattice structure for effective liquid cooling. *Additive Manufacturing*, 28:285–298, 2019.

[17] Akihiro Takezawa, Kenjiro Matsui, Shomu Murakoshi, Kentaro Taniguchi, Ryota Moritoyo, and Mitsuru Kitamura. Validity of the quasi-2d optimal variable density lattice for effective liquid cooling based on darcy–forchheimer theory. *Thermal Science and Engineering Progress*, page 102898, 2024.

[18] Hao Li, Pierre Jolivet, and Joe Alexandersen. Multi-scale topology optimisation of microchannel cooling using a homogenisation-based method. 2024.

[19] Kentaro Yaji, Takayuki Yamada, Seiji Kubo, Kazuhiro Izui, and Shinji Nishiwaki. A topology optimization method for a coupled thermal–fluid problem using level set boundary expressions. *International Journal of Heat and Mass Transfer*, 81:878–888, 2015.

[20] Joe Alexandersen, Ole Sigmund, and Niels Aage. Large scale three-dimensional topology optimisation of heat sinks cooled by natural convection. *International Journal of Heat and Mass Transfer*, 100:876–891, 2016.

[21] Sumer B Dilgen, Cetin B Dilgen, David R Fuhrman, Ole Sigmund, and Boyan S Lazarov. Density based topology optimization of turbulent flow heat transfer systems. *Structural and Multidisciplinary Optimization*, 57(5):1905–1918, 2018.

[22] M Pietropaoli, F Montomoli, and A Gaymann. Three-dimensional fluid topology optimization for heat transfer. *Structural and Multidisciplinary Optimization*, 59:801–812, 2019.

[23] Nicolò Pollini, Ole Sigmund, Casper Schousboe Andreasen, and Joe Alexandersen. A “poor man’s” approach for high-resolution three-dimensional topology design for natural convection problems. *Advances in Engineering Software*, 140:102736, 2020.

[24] Sandilya Kambampati and H Alicia Kim. Level set topology optimization of cooling channels using the darcy flow model. *Structural and Multidisciplinary Optimization*, pages 1–17, 2020.

[25] Minghao Yu, Shilun Ruan, Junfeng Gu, Mengke Ren, Zheng Li, Xinyu Wang, and Changyu Shen. Three-dimensional topology optimization of thermal-fluid-structural problems for cooling system design. *Structural and Multidisciplinary Optimization*, 62(6):3347–3366, 2020.

[26] Lukas Christian Høghøj, Daniel Ruberg Nørhave, Joe Alexandersen, Ole Sigmund, and Casper Schousboe Andreasen. Topology optimization of two fluid heat exchangers. *International Journal of Heat and Mass Transfer*, 163:120543, 2020.

[27] Florian Feppon, Grégoire Allaire, Charles Dapogny, and Pierre Jolivet. Topology optimization of thermal fluid–structure systems using body-fitted meshes and parallel computing. *Journal of Computational Physics*, 417:109574, 2020.

[28] Sicheng Sun, Piotr Liebersbach, and Xiaoping Qian. 3D topology optimization of heat sinks for liquid cooling. *Applied Thermal Engineering*, 178:115540, 2020.

[29] Florian Feppon, Grégoire Allaire, Charles Dapogny, and Pierre Jolivet. Body-fitted topology optimization of 2D and 3D

fluid-to-fluid heat exchangers. *Computer Methods in Applied Mechanics and Engineering*, 376:113638, 2021.

[30] Hiroki Kobayashi, Kentaro Yaji, Shintaro Yamasaki, and Kikuo Fujita. Topology design of two-fluid heat exchange. *Structural and Multidisciplinary Optimization*, 63:821–834, 2021.

[31] Hao Li, Tsuguo Kondoh, Pierre Jolivet, Kozo Furuta, Takayuki Yamada, Benliang Zhu, Heng Zhang, Kazuhiro Izui, and Shinji Nishiwaki. Optimum design and thermal modeling for 2d and 3d natural convection problems incorporating level set-based topology optimization with body-fitted mesh. *International Journal for Numerical Methods in Engineering*, 123(9):1954–1990, 2022.

[32] Minghao Yu, Xinyu Wang, Junfeng Gu, Shilun Ruan, Zheng Li, Sihao Qian, Jingjing Zhang, and Changyu Shen. A synergic topology optimization approach on distribution of cooling channels and diverse-intensity heat sources for liquid-cooled heat sink. *Structural and Multidisciplinary Optimization*, 65(2):48, 2022.

[33] Sheng Pan, Minghao Yu, Hao Li, Zheng Li, Mengke Ren, Junfeng Gu, and Changyu Shen. An integrated two-step strategy for an optimal design of liquid-cooled channel layout based on the mmc–density approach. *Structural and Multidisciplinary Optimization*, 65(8):221, 2022.

[34] Nikolaos Galanos, Evangelos M Papoutsis-Kiachagias, Kyriakos C Giannakoglou, Yoshiyuki Kondo, and Koichi Tanimoto. Synergistic use of adjoint-based topology and shape optimization for the design of bi-fluid heat exchangers. *Structural and Multidisciplinary Optimization*, 65(9):245, 2022.

[35] Brice Rogié and Casper Schousboe Andreasen. Design complexity tradeoffs in topology optimization of forced convection laminar flow heat sinks. *Structural and Multidisciplinary Optimization*, 66(1):6, 2023.

[36] Tianye Wang, Joshua Gasick, Sicheng Sun, and Xiaoping Qian. A comparison of manufacturing constraints in 3d topologically optimized heat sinks for forced air cooling. *Engineering with Computers*, 39(3):1711–1733, 2023.

[37] Philippe Meliga, Wassim Abdel Nour, Delphine Laboureur, Damien Serret, and Elie Hachem. Multi-objective topology optimization of conjugate heat transfer using level sets and anisotropic mesh adaptation. *Fluids*, 9(5):105, 2024.

[38] Ji-Wang Luo, Li Chen, Hanbing Ke, Chuangde Zhang, Yang Xia, and Wen-Quan Tao. Three-dimensional topology optimization of natural convection using double multiple-relaxation-time lattice boltzmann method. *Applied Thermal Engineering*, 236:121732, 2024.

[39] Peter Coffin and Kurt Maute. A level-set method for steady-state and transient natural convection problems. *Structural and Multidisciplinary Optimization*, 53(5):1047–1067, 2016.

[40] Kentaro Yaji, Masao Ogino, Cong Chen, and Kikuo Fujita. Large-scale topology optimization incorporating local-in-time adjoint-based method for unsteady thermal-fluid problem. *Structural and Multidisciplinary Optimization*, 58(2):817–822, 2018.

[41] Yuta Tanabe, Kentaro Yaji, and Kuniharu Ushijima. Topology optimization using the lattice boltzmann method for unsteady natural convection problems. *Structural and Multidisciplinary Optimization*, 66(5):103, 2023.

[42] Susana Rojas-Labanda and Mathias Stolpe. Benchmarking optimization solvers for structural topology optimization. *Structural and Multidisciplinary Optimization*, 52(3):527–547, 2015.

[43] Florian Feppon. Density-based topology optimization with the null space optimizer: a tutorial and a comparison. *Structural and Multidisciplinary Optimization*, 67(1):4, 2024.

[44] Hao Li, Tsuguo Kondoh, Pierre Jolivet, Kozo Furuta, Takayuki Yamada, Benliang Zhu, Kazuhiro Izui, and Shinji Nishiwaki. Three-dimensional topology optimization of a fluid–structure system using body-fitted mesh adaption based on the level-set method. *Applied Mathematical Modelling*, 101:276–308, 2022.

[45] Joe Alexandersen, Ole Sigmund, Knud Erik Meyer, and Boyan Stefanov Lazarov. Design of passive coolers for light-emitting diode lamps using topology optimisation. *International Journal of Heat and Mass Transfer*, 122:138–149, 2018.

[46] Marc-Étienne Lamarche-Gagnon, Marjan Molavi-Zarandi, Vincent Raymond, and Florin Ilinca. Additively manufactured conformal cooling channels through topology optimization. *Structural and Multidisciplinary Optimization*, 67(8):1–19, 2024.

[47] Lipeng Jiu, Weihong Zhang, Liang Meng, Ying Zhou, and Liang Chen. A cad-oriented structural topology optimization method. *Computers & Structures*, 239:106324, 2020.

[48] Niels Aage, Morten Nobel-Jørgensen, Casper Schousboe Andreasen, and Ole Sigmund. Interactive topology optimization on hand-held devices. *Structural and Multidisciplinary Optimization*, 47:1–6, 2013.

[49] Morten Nobel-Jørgensen, Niels Aage, Asger Nyman Christiansen, Takeo Igarashi, J Andreas Bærentzen, and Ole Sigmund. 3d interactive topology optimization on hand-held devices. *Structural and Multidisciplinary Optimization*, 51:1385–1391, 2015.

[50] Mariano Victoria, Osvaldo M Querin, Concepción Díaz, and Pascual Martí. liteitd a matlab graphical user interface (gui) program for topology design of continuum structures. *Advances in Engineering Software*, 100:126–147, 2016.

[51] Ernesto Aranda, José Carlos Bellido, and Alberto Donoso. Toptimiz3d: A topology optimization software using unstructured meshes. *Advances in Engineering Software*, 148:102875, 2020.

[52] XIE-Engineering-Technologies. Ameba - Bi-directional evolutionary structural optimization (BESO) software. `https://ameba.xieym.com/`, 2024. Accessed: 2024-10-08.

[53] Qiang Zhou, Wei Shen, Jin Wang, Yi Yi Zhou, and Yi Min Xie. Ameba: A new topology optimization tool for architectural design. In *Proceedings of IASS Annual Symposia*, volume 2018, pages 1–8. International Association for Shell and Spatial Structures (IASS), 2018.

[54] Ding Wen Bao, Xin Yan, Roland Snooks, and Yi Min Xie. Bioinspired generative architectural design form-finding and advanced robotic fabrication based on structural performance. In *Architectural Intelligence: Selected Papers from the 1st*

International Conference on Computational Design and Robotic Fabrication (CDRF 2019), pages 147–170. Springer, 2020.

[55] Diabatix. Diabatix - ai-driven generative design for thermal management. https://www.diabatix.com/, 2024. Accessed: 2024-09-20.

[56] Toffee Express. Toffee express - cloud-based topology optimization tool. https://toffeex.com/, 2024. Accessed: 2024-09-20.

[57] Joe Alexandersen. A detailed introduction to density-based topology optimisation of fluid flow problems with implementation in matlab. *Structural and Multidisciplinary Optimization*, 66(1):12, 2023.

[58] Boyan Stefanov Lazarov and Ole Sigmund. Filters in topology optimization based on helmholtz-type differential equations. *International Journal for Numerical Methods in Engineering*, 86(6):765–781, 2011.

[59] Atsushi Kawamoto, Tadayoshi Matsumori, Shintaro Yamasaki, Tsuyoshi Nomura, Tsuguo Kondoh, and Shinji Nishiwaki. Heaviside projection based topology optimization by a pde-filtered scalar function. *Structural and Multidisciplinary Optimization*, 44(1):19–24, 2011.

[60] Kyung K Choi and Nam-Ho Kim. *Structural sensitivity analysis and optimization 1: linear systems*. Springer Science & Business Media, 2006.

[61] Krister Svanberg. The method of moving asymptotes—a new method for structural optimization. *International journal for numerical methods in engineering*, 24(2):359–373, 1987.

[62] Luca Arpaia, Héloïse Beaugendre, Luca Cirrottola, Algiane Froehly, Marco Lorini, Léo Nouveau, and Mario Ricchiuto. $h-$ and $r-$adaptation on simplicial meshes using MMG tools. In *Mesh Generation and Adaptation*, pages 183–208. Springer, 2022.

[63] Pascal Frey and Frédéric Alauzet. Anisotropic mesh adaptation for CFD computations. *Computer Methods in Applied Mechanics and Engineering*, 194(48-49):5068–5082, 2005.

[64] Hao Li, Tsuguo Kondoh, Pierre Jolivet, Nari Nakayama, Kozo Furuta, Heng Zhang, Benliang Zhu, Kazuhiro Izui, and Shinji Nishiwaki. Topology optimization for lift–drag problems incorporated with distributed unstructured mesh adaptation. *Structural and Multidisciplinary Optimization*, 65(8):1–15, 2022.

[65] Hao Li, Minghao Yu, Pierre Jolivet, Joe Alexandersen, Tsuguo Kondoh, Tiannan Hu, Kozo Furuta, Kazuhiro Izui, and Shinji Nishiwaki. Reaction–diffusion equation driven topology optimization of high-resolution and feature-rich structures using unstructured meshes. *Advances in Engineering Software*, 180:103457, 2023.

[66] Luca Cirrottola and Algiane Froehly. Parallel unstructured mesh adaptation based on iterative remeshing and repartitioning. In *WCCM-Eccomas 2020-14th World Congress on Computational Mechanic*, 2021.

[67] Charles Dapogny, Cécile Dobrzynski, and Pascal Frey. Three-dimensional adaptive domain remeshing, implicit domain meshing, and applications to free and moving boundary problems. *Journal of Computational Physics*, 262:358–378, 2014.

[68] Frédéric Hecht. New development in FreeFem++. *Journal of Numerical Mathematics*, 20(3-4):251–266, 2012.

[69] Pierre Jolivet, Victorita Dolean, Frédéric Hecht, Frédéric Nataf, Christophe Prud'homme, and Nicole Spillane. High-Performance Domain Decomposition Methods on Massively Parallel Architectures with FreeFem++. *Journal of Numerical Mathematics*, 20(4):287–302, 2012.

[70] Satish Balay, Shrirang Abhyankar, Mark F. Adams, Steven Benson, Jed Brown, Peter Brune, Kris Buschelman, Emil Constantinescu, Lisandro Dalcin, Alp Dener, Victor Eijkhout, Jacob Faibussowitsch, William D. Gropp, Václav Hapla, Tobin Isaac, Pierre Jolivet, Dmitry Karpeev, Dinesh Kaushik, Matthew G. Knepley, Fande Kong, Scott Kruger, Dave A. May, Lois Curfman McInnes, Richard Tran Mills, Lawrence Mitchell, Todd Munson, Jose E. Roman, Karl Rupp, Patrick Sanan, Jason Sarich, Barry F. Smith, Stefano Zampini, Hong Zhang, Hong Zhang, and Junchao Zhang. PETSc/TAO users manual. Technical Report ANL-21/39 - Revision 3.19, Argonne National Laboratory, 2023.

[71] Miguel A Salazar de Troya and Daniel A Tortorelli. Three-dimensional adaptive mesh refinement in stress-constrained topology optimization. *Structural and Multidisciplinary Optimization*, 62(5):2467–2479, 2020.

[72] Paul-Louis George and Houman Borouchaki. Delaunay triangulation and meshing. *Hermes*, 1998.

[73] Johann Moulin, Pierre Jolivet, and Olivier Marquet. Augmented lagrangian preconditioner for large-scale hydrodynamic stability analysis. *Computer Methods in Applied Mechanics and Engineering*, 351:718–743, 2019.

[74] Olof Widlund and Maksymilian Dryja. An additive variant of the schwarz alternating method for the case of many subregions. 1987.

[75] Mark Adams, Harun H. Bayraktar, Tony M. Keaveny, and Panayiotis Papadopoulos. Ultrascalable Implicit Finite Element Analyses in Solid Mechanics with over a Half a Billion Degrees of Freedom. In *Proceedings of the 2004 ACM/IEEE Conference on Supercomputing*, SC04, pages 34:1–34:15. IEEE Computer Society, 2004.

[76] Robert Falgout and Ulrike Yang. *hypre*: a library of high performance preconditioners. *Computational Science—ICCS 2002*, pages 632–641, 2002.

[77] Jeremy BC Jackson. Adaptation and diversity of reef corals. *BioScience*, pages 475–482, 1991.

[78] Shai Asher, Stephan Niewerth, Katinka Koll, and Uri Shavit. Vertical variations of coral reef drag forces. *Journal of Geophysical Research: Oceans*, 121(5):3549–3563, 2016.

[79] Juan D Osorio-Cano, Andrés F Osorio, Juan C Alcérreca-Huerta, and Hocine Oumeraci. Drag and inertia forces on a branched coral colony of acropora palmata. *Journal of Fluids and Structures*, 88:31–47, 2019.

[80] JD Pakulski, P Aas, W Jeffrey, M Lyons, LG Van Waasbergen, D Mitchell, and R Coffin. Influence of light on bacterioplankton production and respiration in a subtropical coral reef. *Aquatic Microbial Ecology*, 14(2):137–148, 1998.

[81] Houssam Houssein. Symmetric formulation for non-conforming meshes in magnetostatic problems. *IEEE Transactions on*

Magnetics, 2024.

[82] Houssam Houssein, Simon Garnotel, and Frédéric Hecht. A symmetric algorithm for solving mechanical contact problems using freefem. In *Advances in Computational Methods and Technologies in Aeronautics and Industry*, pages 235–250. Springer, 2022.

[83] Christophe Geuzaine and Jean-François Remacle. Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009.

[84] Utkarsh Ayachit. *The paraview guide: a parallel visualization application*. Kitware, Inc., 2015.

[85] Liang Meng, Weihong Zhang, Dongliang Quan, Guanghui Shi, Lei Tang, Yuliang Hou, Piotr Breitkopf, Jihong Zhu, and Tong Gao. From topology optimization design to additive manufacturing: Today's success and tomorrow's roadmap. *Archives of Computational Methods in Engineering*, 27:805–830, 2020.

[86] Abir Dutta, Menaka Singh, Kathryn Kumar, Aida Ribera Navarro, Rodney Santiago, Ruchi Pathak Kaul, Sanganagouda Patil, and Deepak M Kalaskar. Accuracy of 3d printed spine models for pre-surgical planning of complex adolescent idiopathic scoliosis (ais) in spinal surgeries: a case series. *Annals of 3D Printed Medicine*, 11:100117, 2023.

[87] Hao Li, Xiaohong Ding, Dalei Jing, Min Xiong, and Fanzhen Meng. Experimental and numerical investigation of liquid-cooled heat sinks designed by topology optimization. *International Journal of Thermal Sciences*, 146:106065, 2019.

[88] Sicheng Sun, Behzad Rankouhi, Dan J Thoma, Michael J Cheadle, Gunnar D Maples, Mark H Anderson, Gregory Nellis, and Xiaoping Qian. Topology optimization, additive manufacturing and thermohydraulic testing of heat sinks. *International Journal of Heat and Mass Transfer*, 224:125281, 2024.

A. Sensitivity analysis

A.1. Continuous adjoint method

First, the Lagrange function $\boldsymbol{\mathcal{L}}$ is constructed as follows:

$$\begin{aligned}\boldsymbol{\mathcal{L}} := \boldsymbol{J} &+ \left\langle \boldsymbol{v}_a, -\operatorname{div}\left(\boldsymbol{\sigma}_{\mathrm{f}}(\boldsymbol{v},p)\right) + (\boldsymbol{v}\cdot\nabla)\boldsymbol{v} + \alpha\left(\boldsymbol{x}\right)\boldsymbol{v}\right\rangle + \langle p_a, -\operatorname{div}(\boldsymbol{v})\rangle \\ &+ \langle T_a, \mathrm{Re}\,\mathrm{Pr}(\boldsymbol{v}\cdot\nabla T) - \nabla\cdot(\kappa(\boldsymbol{x})\nabla T)\rangle + \left\langle \boldsymbol{u}_a, -\operatorname{div}\left(\boldsymbol{\sigma}_{\mathrm{s}}\left(\boldsymbol{u},T\right)\right)\right\rangle,\end{aligned} \tag{A.1}$$

where $\boldsymbol{v}_a$, p_a, T_a, and $\boldsymbol{u}_a$ are the adjoint velocity, adjoint pressure, adjoint temperature, and adjoint displacement, respectively. The variation of the Lagrange function is expressed as follows:

$$\delta\boldsymbol{\mathcal{L}} = \delta_{\hat{\gamma}}\boldsymbol{\mathcal{L}} + \delta_{(\boldsymbol{v},p)}\boldsymbol{\mathcal{L}} + \delta_T\boldsymbol{\mathcal{L}} + \delta_{\boldsymbol{u}}\boldsymbol{\mathcal{L}}. \tag{A.2}$$

In accordance with the Karush–Kuhn–Tucker (KKT) conditions applicable to PDE constrained optimization problems,

$$\begin{cases} \delta_{\boldsymbol{u}}\boldsymbol{\mathcal{L}} = 0, \\ \delta_T\boldsymbol{\mathcal{L}} = 0, \\ \delta_{(\boldsymbol{v},p)}\boldsymbol{\mathcal{L}} = 0. \end{cases} \tag{A.3}$$

The variation of the Lagrange function w.r.t. the state variables $\boldsymbol{u}$, T, and $(\boldsymbol{v}, p)$ are derived as follows:

$$\delta_{\boldsymbol{u}}\boldsymbol{\mathcal{L}} = \delta_{\boldsymbol{u}}\boldsymbol{J} + \int_\Omega (e(\delta\boldsymbol{u}) : \mathbb{C}(\hat{\gamma})) : e(\boldsymbol{u}_a)\,\mathrm{d}\Omega = 0 \quad \forall\delta\boldsymbol{u}\in\mathscr{U}, \boldsymbol{u}_a\in\mathscr{U}, \tag{A.4a}$$

$$\begin{aligned}\delta_T\boldsymbol{\mathcal{L}} = \delta_T\boldsymbol{J} &+ \int_\Omega \mathrm{Re}\,\mathrm{Pr}(\boldsymbol{v}\cdot\nabla\delta T)T_a\,\mathrm{d}\Omega + \int_\Omega \kappa\left(\hat{\gamma}\right)\nabla\delta T\cdot\nabla T_a\,\mathrm{d}\Omega \\ &- \int_\Omega \alpha_T\delta T\mathbf{I} : \nabla\boldsymbol{u}_a = 0 \quad \forall\delta T\in\mathscr{P}, T_a\in\mathscr{P},\end{aligned} \tag{A.4b}$$

$$
\begin{aligned}
\delta_{(\boldsymbol{v},p)}\mathcal{L} = \delta_{(\boldsymbol{v},p)}J
&+ \int_{\Omega} ((\boldsymbol{v}\cdot\nabla)\delta\boldsymbol{v})\cdot\boldsymbol{v}_a \,\mathrm{d}\Omega + \int_{\Omega} ((\delta\boldsymbol{v}\cdot\nabla)\boldsymbol{v})\cdot\boldsymbol{v}_a \,\mathrm{d}\Omega + \frac{2}{\mathrm{Re}}\int_{\Omega} e(\delta\boldsymbol{v}) : \nabla\boldsymbol{v}_a \,\mathrm{d}\Omega \\
&- \int_{\Omega} \delta p \nabla\cdot\boldsymbol{v}_a \,\mathrm{d}\Omega - \int_{\Omega} p_a \nabla\cdot\delta\boldsymbol{v} \,\mathrm{d}\Omega + \int_{\Omega} \alpha\left(\hat{\gamma}\right)\boldsymbol{v}_a\cdot\delta\boldsymbol{v}\,\mathrm{d}\Omega \\
&+ \int_{\Omega} \mathrm{Re}\,\mathrm{Pr}\,T_a \nabla T\cdot\delta\boldsymbol{v}\,\mathrm{d}\Omega \\
&- \sum_{K\in\mathcal{T}} \int_{\Omega_e} \tau_{\mathrm{PSPG}} \nabla p_a \cdot \left((\delta\boldsymbol{v}\cdot\nabla)\boldsymbol{v} + (\boldsymbol{v}\cdot\nabla)\delta\boldsymbol{v} + \nabla\delta p + \alpha(\boldsymbol{x})\delta\boldsymbol{v}\right)\,\mathrm{d}\Omega \\
&- \sum_{K\in\mathcal{T}} \int_{\Omega_e} \left(\delta_{(\boldsymbol{v},p)}\tau_{\mathrm{PSPG}}\right)\nabla p_a \cdot \left((\boldsymbol{v}\cdot\nabla)\boldsymbol{v} + \nabla p + \alpha(\boldsymbol{x})\boldsymbol{v}\right)\,\mathrm{d}\Omega = 0 \\
&\forall(\delta\boldsymbol{v},\delta p)\in\mathscr{U}_{\boldsymbol{v},p},\, (\boldsymbol{v}_a, p_a)\in\mathscr{U}_{\boldsymbol{v},p}.
\end{aligned}
\tag{A.4c}
$$

Hereinafter, we use τ for brevity. $\delta_{(\boldsymbol{v},p)}\tau$ is derived as follows:

$$
\begin{aligned}
\delta_{(\boldsymbol{v},p)}\tau &= \frac{\partial\tau}{\partial\tau_1}\frac{\partial\tau_1}{\partial\boldsymbol{v}} + \frac{\partial\tau}{\partial\tau_3}\frac{\partial\tau_3}{\partial\boldsymbol{v}} + \frac{\partial\tau}{\partial\tau_4}\frac{\partial\tau_4}{\partial\boldsymbol{v}} \\
&= \tau_1^{-3}\tau^3\left(-\frac{h}{2}\left(\boldsymbol{v}\cdot\boldsymbol{v}\right)^{-3/2}\left(\delta\boldsymbol{v}\cdot\boldsymbol{v}\right)\right).
\end{aligned}
\tag{A.5}
$$

Regarding the optimal design problem presented in Eq. (24), we derive the variations of the cost functions w.r.t. the state variables as follows:

$$
\begin{cases}
\delta_{\boldsymbol{u}} J = 0, \\
\delta_T J = -\displaystyle\int_{\Gamma_{\mathrm{out}}} \left(\boldsymbol{v}\cdot\boldsymbol{n}\right)\delta T\,\mathrm{d}\Gamma, \\
\delta_{(\boldsymbol{v},p)} J = -\displaystyle\int_{\Gamma_{\mathrm{out}}} \left(\delta\boldsymbol{v}\cdot\boldsymbol{n}\right) T\,\mathrm{d}\Gamma.
\end{cases}
\tag{A.6a}
$$

$$
\begin{cases}
\delta_{\boldsymbol{u}} G_2 = 0, \\
\delta_T G_2 = 0, \\
\delta_{(\boldsymbol{v},p)} G_2 = -\displaystyle\int_{\Gamma_{\mathrm{in}}\cup\Gamma_{\mathrm{out}}} \left(p + \frac{1}{2}|\boldsymbol{v}|^2\right)\left(\delta\boldsymbol{v}\cdot\boldsymbol{n}_{\mathrm{f}}\right) + \left(\delta\boldsymbol{v}\cdot\boldsymbol{v}\right)\left(\boldsymbol{v}\cdot\boldsymbol{n}_{\mathrm{f}}\right)\mathrm{d}\Gamma.
\end{cases}
\tag{A.6b}
$$

$$
\begin{cases}
\delta_{\boldsymbol{u}} G_3 = \displaystyle\int_{\partial\Omega_{\boldsymbol{u}}^N} \boldsymbol{t}\cdot\delta\boldsymbol{u}\,\mathrm{d}\Gamma, \\
\delta_T G_3 = 0, \\
\delta_{(\boldsymbol{v},p)} G_3 = 0.
\end{cases}
\tag{A.6c}
$$

Finally, the functional derivative of the Lagrange function w.r.t. design variables, $\delta_{\gamma}\mathcal{L}$, can be derived using the chain rule as follows:

$$\frac{\delta \mathcal{L}}{\delta \gamma} = \frac{\delta \mathcal{L}}{\delta \hat{\gamma}} \frac{\partial \hat{\gamma}}{\partial \tilde{\gamma}} \frac{\delta \tilde{\gamma}}{\delta \gamma}, \tag{A.7}$$

where the chain rule term $\frac{\partial \hat{\gamma}}{\partial \tilde{\gamma}}$ can be obtained by differentiation of the smoothed Heaviside function in Eq. (21) as

$$\frac{\partial \hat{\gamma}}{\partial \tilde{\gamma}} = \beta \frac{1 - \tanh^2(\beta(\tilde{\gamma} - \eta))}{\tanh(\beta\eta) + \tanh(\beta(1-\eta))}. \tag{A.8}$$

The functional derivative of the Lagrange function w.r.t. the projected density field $\delta_{\hat{\gamma}}\mathcal{L}$ can be derived as follows:

$$\begin{aligned} \delta_{\hat{\gamma}}\mathcal{L} = {} & \delta_{\hat{\gamma}}J + \frac{\partial \alpha\left(\hat{\gamma}\right)}{\partial \hat{\gamma}} \boldsymbol{v} \cdot \boldsymbol{v}_a + \frac{\partial \kappa\left(\hat{\gamma}\right)}{\partial \hat{\gamma}} \nabla T \cdot \nabla T_a + \left(e(\boldsymbol{u}) : \frac{\partial \mathbb{C}\left(\hat{\gamma}\right)}{\partial \hat{\gamma}} \right) : e(\boldsymbol{u}_a) \\ & - \delta_{\hat{\gamma}}\tau \nabla p_a \cdot ((\boldsymbol{v}\cdot\nabla)\boldsymbol{v} + \nabla p + \alpha\left(\hat{\gamma}\right)\boldsymbol{v}) - \tau \nabla p_a \cdot \frac{\partial \alpha\left(\hat{\gamma}\right)}{\partial \hat{\gamma}} \boldsymbol{v} \quad \text{in } \Omega, \end{aligned} \tag{A.9}$$

where the partial derivatives w.r.t. the projected density field can be derived as follows:

$$\begin{cases} \dfrac{\partial \alpha\left(\hat{\gamma}\right)}{\partial \hat{\gamma}} = -\dfrac{q_\alpha \alpha_{\max}\left(q_\alpha + 1\right)}{\left(q_\alpha + \hat{\gamma}\right)^2}, \\ \dfrac{\partial \kappa\left(\hat{\gamma}\right)}{\partial \hat{\gamma}} = -\dfrac{q_\kappa\left(c_k - 1\right)\left(q_\kappa + 1\right)}{\left(q_\kappa + \hat{\gamma}\right)^2}, \\ \dfrac{\partial \mathbb{C}\left(\hat{\gamma}\right)}{\partial \hat{\gamma}} = -q_{\mathbb{C}}\left(1 - \hat{\gamma}\right)^{q_{\mathbb{C}}-1}\left(\mathbb{C}_{\mathrm{s}} - \mathbb{C}_{\mathrm{v}}\right), \end{cases} \tag{A.10}$$

and as follows:

$$\begin{aligned} \delta_{\hat{\gamma}}\tau &= \frac{\partial \tau}{\partial \tau_4} \frac{\partial \tau_4}{\partial \alpha\left(\hat{\gamma}\right)} \frac{\partial \alpha\left(\hat{\gamma}\right)}{\partial \hat{\gamma}} \\ &= -\frac{\partial \alpha\left(\hat{\gamma}\right)}{\partial \hat{\gamma}} \frac{1}{\alpha\left(\hat{\gamma}\right)^2}\left(\tau_4^{-3}\tau^3\right). \end{aligned} \tag{A.11}$$

A.2. Discrete adjoint method

We denote the residual of the governing equations as $\boldsymbol{r}_{\mathrm{NS}}$, $\boldsymbol{r}_{\mathrm{Energy}}$, and $\boldsymbol{r}_{\mathrm{Lamé}}$ for the Navier-Stokes equations, convection diffusion equation, and the elasticity equation, respectively. The Lagrange function $\mathcal{L}$ is constructed as follows:

$$\mathcal{L} := J + \lambda_{(\boldsymbol{v},p)}^{\mathrm{T}} \boldsymbol{r}_{\mathrm{NS}} + \lambda_T^{\mathrm{T}} \boldsymbol{r}_{\mathrm{Energy}} + \lambda_{\boldsymbol{u}}^{\mathrm{T}} \boldsymbol{r}_{\mathrm{Lamé}}, \tag{A.12}$$

where $\lambda_{(\boldsymbol{v},p)}$, λ_T, and $\lambda_{\boldsymbol{u}}$ are the vectors of adjoint variables $\left(\boldsymbol{v}_a, p_a\right)$, T_a, and $\boldsymbol{u}_a$, respectively. The total derivative of the Lagrange function w.r.t. $\hat{\gamma}$ is then taken of the Lagrange function, as follows:

$$\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\hat{\gamma}} = \frac{\mathrm{d}J}{\mathrm{d}\hat{\gamma}} + \lambda_{(\boldsymbol{v},p)}^{\mathrm{T}} \frac{\mathrm{d}\boldsymbol{r}_{\mathrm{NS}}}{\mathrm{d}\hat{\gamma}} + \lambda_T^{\mathrm{T}} \frac{\mathrm{d}\boldsymbol{r}_{\mathrm{Energy}}}{\mathrm{d}\hat{\gamma}} + \lambda_{\boldsymbol{u}}^{\mathrm{T}} \frac{\mathrm{d}\boldsymbol{r}_{\mathrm{Lamé}}}{\mathrm{d}\hat{\gamma}}, \tag{A.13}$$

where the total derivative of the cost function is expressed as follows:

$$\frac{\mathrm{d}J}{\mathrm{d}\hat{\gamma}} = \frac{\partial J}{\partial \hat{\gamma}} + \frac{\partial J}{\partial \boldsymbol{s}_{(\boldsymbol{v},p)}} \frac{\partial \boldsymbol{s}_{(\boldsymbol{v},p)}}{\partial \hat{\gamma}} + \frac{\partial J}{\partial \boldsymbol{s}_T} \frac{\partial \boldsymbol{s}_T}{\partial \hat{\gamma}} + \frac{\partial J}{\partial \boldsymbol{s}_{\boldsymbol{u}}} \frac{\partial \boldsymbol{s}_{\boldsymbol{u}}}{\partial \hat{\gamma}}, \tag{A.14}$$

due to the implicit dependence of cost function J (and G_i) on the state variables $\boldsymbol{s}$. Thus, the total derivative of the Lagrange function can be expanded as follows:

$$\begin{aligned}\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\hat{\gamma}} = {} & \frac{\partial J}{\partial \hat{\gamma}} + \frac{\partial J}{\partial \boldsymbol{s}_{(\boldsymbol{v},p)}} \frac{\partial \boldsymbol{s}_{(\boldsymbol{v},p)}}{\partial \hat{\gamma}} + \frac{\partial J}{\partial \boldsymbol{s}_T} \frac{\partial \boldsymbol{s}_T}{\partial \hat{\gamma}} + \frac{\partial J}{\partial \boldsymbol{s}_{\boldsymbol{u}}} \frac{\partial \boldsymbol{s}_{\boldsymbol{u}}}{\partial \hat{\gamma}} \\ & + \boldsymbol{\lambda}_{(\boldsymbol{v},p)}^{\mathrm{T}} \left(\frac{\partial \boldsymbol{r}_{\mathrm{NS}}}{\partial \hat{\gamma}} + \frac{\partial \boldsymbol{r}_{\mathrm{NS}}}{\partial \boldsymbol{s}_{(\boldsymbol{v},p)}} \frac{\partial \boldsymbol{s}_{(\boldsymbol{v},p)}}{\partial \hat{\gamma}} \right) \\ & + \boldsymbol{\lambda}_T^{\mathrm{T}} \left(\frac{\partial \boldsymbol{r}_{\mathrm{Energy}}}{\partial \hat{\gamma}} + \frac{\partial \boldsymbol{r}_{\mathrm{Energy}}}{\partial \boldsymbol{s}_T} \frac{\partial \boldsymbol{s}_T}{\partial \hat{\gamma}} + \frac{\partial \boldsymbol{r}_{\mathrm{Energy}}}{\partial \boldsymbol{s}_{(\boldsymbol{v},p)}} \frac{\partial \boldsymbol{s}_{(\boldsymbol{v},p)}}{\partial \hat{\gamma}} \right) \\ & + \boldsymbol{\lambda}_{\boldsymbol{u}}^{\mathrm{T}} \left(\frac{\partial \boldsymbol{r}_{\mathrm{Lamé}}}{\partial \hat{\gamma}} + \frac{\partial \boldsymbol{r}_{\mathrm{Lamé}}}{\partial \boldsymbol{s}_{\boldsymbol{u}}} \frac{\partial \boldsymbol{s}_{\boldsymbol{u}}}{\partial \hat{\gamma}} + \frac{\partial \boldsymbol{r}_{\mathrm{Lamé}}}{\partial \boldsymbol{s}_T} \frac{\partial \boldsymbol{s}_T}{\partial \hat{\gamma}} \right),\end{aligned} \tag{A.15}$$

which can be rewritten as follows:

$$\begin{aligned}\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\hat{\gamma}} = {} & \frac{\partial J}{\partial \hat{\gamma}} + \boldsymbol{\lambda}_{(\boldsymbol{v},p)}^{\mathrm{T}} \frac{\partial \boldsymbol{r}_{\mathrm{NS}}}{\partial \hat{\gamma}} + \boldsymbol{\lambda}_T^{\mathrm{T}} \frac{\partial \boldsymbol{r}_{\mathrm{Energy}}}{\partial \hat{\gamma}} + \boldsymbol{\lambda}_{\boldsymbol{u}}^{\mathrm{T}} \frac{\partial \boldsymbol{r}_{\mathrm{Lamé}}}{\partial \hat{\gamma}} \\ & + \left(\frac{\partial J}{\partial \boldsymbol{s}_{(\boldsymbol{v},p)}} + \boldsymbol{\lambda}_{(\boldsymbol{v},p)}^{\mathrm{T}} \frac{\partial \boldsymbol{r}_{\mathrm{NS}}}{\partial \boldsymbol{s}_{(\boldsymbol{v},p)}} + \boldsymbol{\lambda}_T^{\mathrm{T}} \frac{\partial \boldsymbol{r}_{\mathrm{Energy}}}{\partial \boldsymbol{s}_{(\boldsymbol{v},p)}} \right) \frac{\partial \boldsymbol{s}_{(\boldsymbol{v},p)}}{\partial \hat{\gamma}} \\ & + \left(\frac{\partial J}{\partial \boldsymbol{s}_T} + \boldsymbol{\lambda}_T^{\mathrm{T}} \frac{\partial \boldsymbol{r}_{\mathrm{Energy}}}{\partial \boldsymbol{s}_T} + \boldsymbol{\lambda}_{\boldsymbol{u}}^{\mathrm{T}} \frac{\partial \boldsymbol{r}_{\mathrm{Lamé}}}{\partial \boldsymbol{s}_T} \right) \frac{\partial \boldsymbol{s}_T}{\partial \hat{\gamma}} \\ & + \left(\frac{\partial J}{\partial \boldsymbol{s}_{\boldsymbol{u}}} + \boldsymbol{\lambda}_{\boldsymbol{u}}^{\mathrm{T}} \frac{\partial \boldsymbol{r}_{\mathrm{Lamé}}}{\partial \boldsymbol{s}_{\boldsymbol{u}}} \right) \frac{\partial \boldsymbol{s}_{\boldsymbol{u}}}{\partial \hat{\gamma}}.\end{aligned} \tag{A.16}$$

The adjoint equations are then defined as what are inside the brackets, as follows:

$$\left(\frac{\partial \boldsymbol{r}_{\mathrm{Lamé}}}{\partial \boldsymbol{s}_{\boldsymbol{u}}} \right)^{\mathrm{T}} \boldsymbol{\lambda}_{\boldsymbol{u}} = -\frac{\partial J}{\partial \boldsymbol{s}_{\boldsymbol{u}}}, \tag{A.17a}$$

$$\left(\frac{\partial \boldsymbol{r}_{\mathrm{Energy}}}{\partial \boldsymbol{s}_T} \right)^{\mathrm{T}} \boldsymbol{\lambda}_T = -\frac{\partial J}{\partial \boldsymbol{s}_T} - \boldsymbol{\lambda}_{\boldsymbol{u}}^{\mathrm{T}} \frac{\partial \boldsymbol{r}_{\mathrm{Lamé}}}{\partial \boldsymbol{s}_T}, \tag{A.17b}$$

$$\left(\frac{\partial \boldsymbol{r}_{\mathrm{NS}}}{\partial \boldsymbol{s}_{(\boldsymbol{v},p)}} \right)^{\mathrm{T}} \boldsymbol{\lambda}_{(\boldsymbol{v},p)} = -\frac{\partial J}{\partial \boldsymbol{s}_{(\boldsymbol{v},p)}} - \boldsymbol{\lambda}_T^{\mathrm{T}} \frac{\partial \boldsymbol{r}_{\mathrm{Energy}}}{\partial \boldsymbol{s}_{(\boldsymbol{v},p)}}. \tag{A.17c}$$

Finally, the sensitivity w.r.t. $\hat{\gamma}$ can be derived as follows:

$$\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\hat{\gamma}} = \frac{\partial J}{\partial \hat{\gamma}} + \boldsymbol{\lambda}_{(\boldsymbol{v},p)}^{\mathrm{T}} \frac{\partial \boldsymbol{r}_{\mathrm{NS}}}{\partial \hat{\gamma}} + \boldsymbol{\lambda}_T^{\mathrm{T}} \frac{\partial \boldsymbol{r}_{\mathrm{Energy}}}{\partial \hat{\gamma}} + \boldsymbol{\lambda}_{\boldsymbol{u}}^{\mathrm{T}} \frac{\partial \boldsymbol{r}_{\mathrm{Lamé}}}{\partial \hat{\gamma}}. \tag{A.18}$$