Taylor & Francis
Taylor & Francis Group

Engineering Optimization

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/geno20

Flow–heat topology optimization of internally cooled high temperature applications using a voxelization approach for domain initialization

Jonas Lundgren, Jan-Erik Lundgren, Hossein Nadali Najafabadi & Carl-Johan Thore

ENGINEERING OPTIMIZATION
https://doi.org/10.1080/0305215X.2023.2196420

Taylor & Francis
Taylor & Francis Group

OPEN ACCESS Check for updates

Flow–heat topology optimization of internally cooled high temperature applications using a voxelization approach for domain initialization

Jonas Lundgren [a], Jan-Erik Lundgren [b], Hossein Nadali Najafabadi [c] and Carl-Johan Thore [a]

[a]Division of Solid Mechanics, Linköping University, Linköping, Sweden; [b]Siemens Energy AB, Finspång, Sweden; [c]Applied Thermodynamics and Fluid Mechanics, Linköping University, Linköping, Sweden

ABSTRACT

A method is presented for obtaining topology optimized designs for internally cooled high temperature applications, using a flexible geometry description, by means of a voxelization methodology and a novel boundary detection algorithm. A conjugate heat transfer approach is taken; the physics is described by a Stokes–Brinkman model for the flow, weakly coupled with a convection–diffusion model for the heat transfer. A practically relevant optimization formulation, consisting of a maximum temperature objective with a mass flow constraint, is used, and applied to an industrial-relevant non-trivial geometry resembling a guide vane in a gas turbine. Temperatures and velocities from the optimized design are compared with the response from a Stokes flow model with body-fitted mesh and a high-fidelity Reynolds-averaged Navier–Stokes model. A comparison of the performance from a mixed and a penalty approach for solving the flow problem is included. The voxelization approach shows good promise for handling complex design domains.

ARTICLE HISTORY
Received 12 September 2022
Accepted 22 March 2023

KEYWORDS
Topology optimization; conjugate heat transfer; mass flow constraint; voxelization; high performance computing

1. Introduction

Topology optimization (TO) (Bendsøe and Sigmund 2004) is a potent design tool for engineers in the early design process owing to its generous design freedom, parametrizing every point in the design space. Realization of non-trivial TO designs in 3D was for long not feasible, but lately it has become possible by means of additive manufacturing (AM) (Gibson *et al.* 2021), a technique by which parts are made by adding material in a layer-by-layer fashion.

Lately, AM has gained interest in the gas turbine industry, where AM components can be found in the hot sections of turbines, *e.g.* around the combustion chamber and turbine blades (Andersson *et al.* 2017). The gas turbine could act as a grid stabilizer in a future, sustainable but more volatile, energy system, using hydrogen as fuel (Öberg, Odenberger, and Johnsson 2022). However, the higher flame temperature of hydrogen (Glassman, Yetter, and Glumac 2015) increases the rate of wear and tear in turbine components owing to, for example, increased thermomechanical loads and deteriorated material properties, causing fatigue, creep and large plastic deformations.

A gas turbine component that is a promising candidate to be designed using TO and manufactured by AM is the guide vane. Guide vanes are found in the hot part of gas turbines, statically mounted in a

CONTACT Jonas Lundgren jonas.lundgren@liu.se

Supplemental data for this article can be accessed here. https://doi.org/10.1080/0305215X.2023.2196420

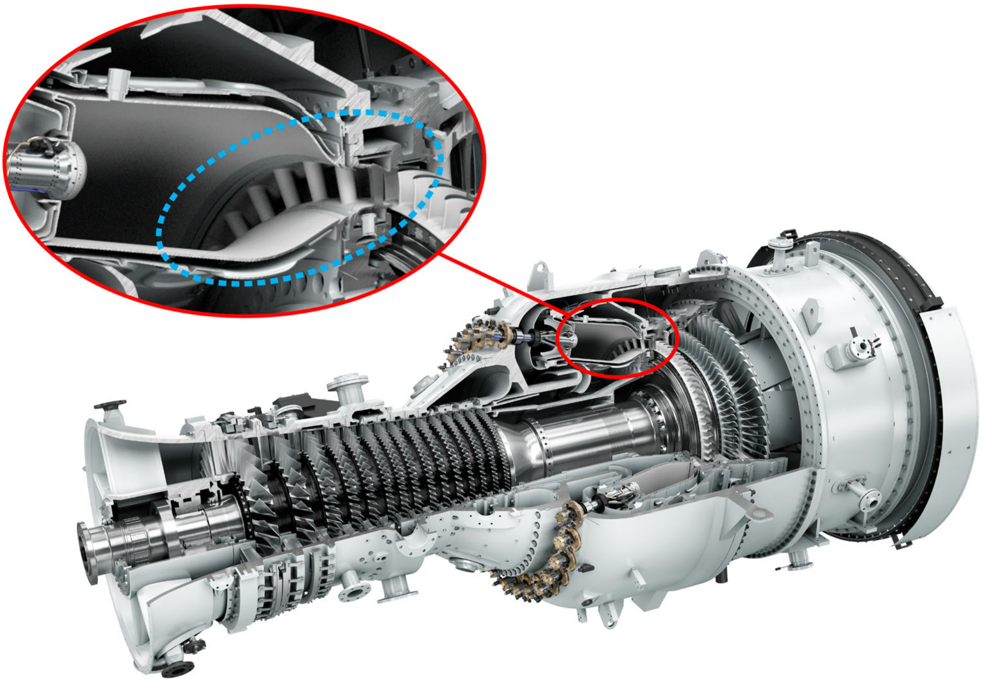


Figure 1. The SGT-800 gas turbine, with a detailed picture of the guide vanes (dashed) at the exit of the combustion chamber. Courtesy of Siemens Energy AB.

radial configuration with fixed inner and outer radii, directing the hot gas flow from the combustors to the turbine blades, see Figure 1. Owing to the hostile environment, the guide vanes need to be cooled from within to keep the material at permissible temperatures. Coolant enters the interior of the guide vane radially from the outward facing short end and is released back into the hot gas stream through the trailing edge of the guide vane (and to some extent also through so-called shower heads, which create a protective film of coolant along the leading edge, a configuration that is not considered herein). The coolant consists of compressed air taken from the latter stages of the compressor and is therefore very expensive to use, as it would otherwise have been used to power the turbine. However, the coolant needs to be highly pressurized to avoid back-flow of the hot gas stream through the guide vane.

Since guide vanes essentially act like compact heat exchangers when cooled from within, it is possible to get an idea of how a good interior design might look by investigating the design of heat exchangers in general. A main concept is to have a large surface-to-volume ratio to enhance the heat transfer between solid and fluid. Furthermore, fins, tubes and lattice structures are common design features of heat exchangers (Holman 2010). This corresponds to the guide vane's having a large and complex interior interface relative to the space occupied by the coolant.

A desirable design of the guide vane is one where the outer surface temperature is kept as low as possible, but without spending too much cooling air, and thus decreasing the efficiency of the turbine. This constitutes a conflict of interest that is addressed by analysing a multiphysics TO problem of conjugate heat transfer (CHT), using the maximum temperature on the surface of the component as objective, and the coolant mass flow as constraint.

Herein, numerical examples using a 3D geometry resembling a gas turbine guide vane are presented. To facilitate the computations, the PETSc[1] suite (Smith 2011) is utilized, which uses the message passing interface (MPI) standard for parallelization to efficiently solve partial differential equations numerically. Aage, Andreassen, and Lazarov (2015) demonstrated the capability for doing high performance computing (HPC) TO in PETSc, and the implementation presented herein is inspired by that work.

A motivation of this work is to allow for 3D design domains with non-trivial shapes. To that end, a voxelization method (Zhang *et al.* 2021) is employed to analyse general geometries on unfitted structured meshes, somewhat similarly to CutFEM (Burman *et al.* 2015) and other approaches for unfitted meshes (Badia, Martorell, and Verdugo 2022). A boundary detection algorithm is implemented to refine the imposing of boundary conditions. It is believed that the voxelization, together with the new boundary detection algorithm, contributes to making TO a viable option in the industrial design process by achieving a higher degree of automation. The voxelization approach has the potential to produce high quality meshes with extremely fine resolution, something that is beneficial for simulations of complex geometries, such as flow through narrow channels.

It is not common practice in the TO literature to include numerical verifications of results from density-based TO models with a monolithic mesh (*i.e.* interpolating between solid and fluid using the Brinkman parameter), especially not for fluid applications. One exception is the recent work from Rogié and Andreasen (2023), in which a heat sink is optimized and later verified against a CFD model using a body-fitted mesh, and other examples include Zhao *et al.* (2018), Haertel *et al.* (2018) and Pietropaoli, Gaymann, and Montomoli (2020). A numerical verification is often a way to avoid the complicated procedure of manufacturing and experimental testing. Before AM was common, it was only reasonable to manufacture and conduct experimental testing of simple (often extruded 2D) geometries (Koga *et al.* 2013; Dede, Joshi, and Zhou 2015; Li *et al.* 2019), and few complex 3D designs have been considered (Lei *et al.* 2018; Lazarov *et al.* 2018). Details are included on the numerical comparison of responses from the TO model and two models using a segregated mesh: a Stokes flow model (a TO model twin), and a high-fidelity Reynolds-averaged Navier–Stokes (RANS) turbulence model.

Dede (2009) and Yoon (2010) published the first articles on multiphysics flow–heat TO. Since then, much has been published on laminar flow in 2D, mainly including examples of heat exchangers with trivial geometries, see *e.g.* Alexandersen and Andreasen (2020) and Fawaz *et al.* (2022) for comprehensive reviews.

Few articles include 3D examples of TO for CHT (Alexandersen and Andreasen 2020). Several approaches are taken to avoid modelling full 3D, such as using an extruded 2D design for planar flow in a 3D context (Koga *et al.* 2013; S. Qian *et al.* 2018). More advanced models include two-layer models (McConnell and Pingen 2012; Zeng, Kanargi, and Lee 2018; Haertel *et al.* 2018; Yan *et al.* 2019) and other 2D-design/3D-application models (Haertel and Nellis 2017). When full 3D examples are presented, low-fidelity flow models are often used, such as the Darcy flow model (Zhao *et al.* 2018; Kambampati and Kim 2020). Even examples without flow models exist, *e.g.* where the convection is simulated by interpolating the heat transfer coefficient based on the design density values in 2D (Bruns 2007), 3D (Zhou *et al.* 2016) and 3D with 2D design (Lundgren *et al.* 2019).

Steady, laminar Navier–Stokes (NS) flow models have been used recently for CHT problems in 3D (Yaji *et al.* 2015; Pietropaoli, Montomoli, and Gaymann 2019; Sun, Liebersbach, and Qian 2020). Regarding NS with turbulence modelling, Kontoleontos *et al.* (2013) presented 2D examples with a prescribed temperature distribution in the solid, and Dilgen *et al.* (2018) modelled turbulence in 3D CHT problems, something that is rare in a TO context (Alexandersen and Andreasen 2020; Fawaz *et al.* 2022). However, a few more recent articles on that topic exist (Pietropaoli, Gaymann, and Montomoli 2020; Holka *et al.* 2022).

Fawaz *et al.* (2022) showed that articles investigating the TO of heat exchangers mostly consider only the temperature as an objective (Yaji *et al.* 2015; Haertel and Nellis 2017; Haertel *et al.* 2018; Zhao *et al.* 2018), or a weighted sum multiphysics-objective (Marck, Nemer, and Harion 2013; Pietropaoli *et al.* 2017; Sato *et al.* 2018; Thore, Lundgren, and Lundgren 2023). One exception is the work by X. Qian and Dede (2016), where the weighted sum multiphysics-objective is accompanied by a tangential thermal gradient constraint. Herein, a restriction is used for the mass flow instead of the temperature.

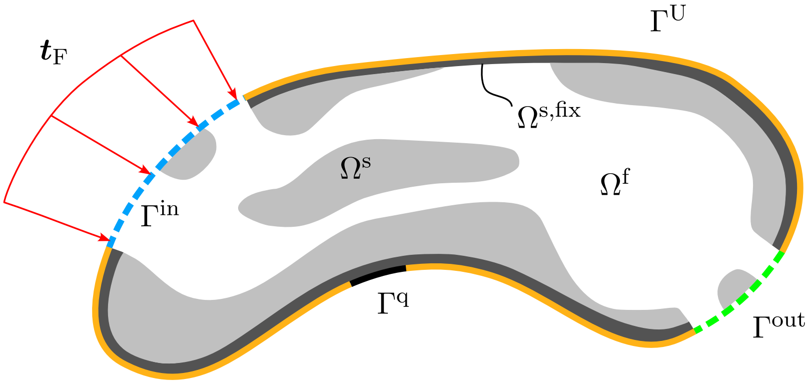


Figure 2. The computational domain Ω.

By doing this, it is not possible to prescribe the inlet mass flow or flow velocity, as is commonly seen in the literature. Very few publications treat CHT with unspecified inlet mass flow. However, some examples exist, *e.g.* Yaji *et al.* (2015) in a transient context, and Haertel and Nellis (2017) and Haertel *et al.* (2018) for simplified (2D-)3D models.

A mass flow constraint similar to the one used herein has been used before in pure flow models (Gersborg-Hansen, Berggren, and Dammann 2006; Aage *et al.* 2008; Behrou, Ranjan, and Guest 2019), but to the authors' best knowledge, the only example of application in flow–heat models is Thore *et al.* (2022), where it was combined with a thermal compliance objective.

2. The conjugate heat transfer problem

The computational domain (Figure 2) is denoted by $\Omega \subset \mathbb{R}^d$, $d = 2, 3$. Its boundary $\Gamma = \Gamma^{\text{in}} \cup \Gamma^{\text{out}} \cup \Gamma^{\text{U}} \cup \Gamma^{\text{q}}$ is made up of disjoint parts, with outward normal $\boldsymbol{n}$.

The domain Ω can be divided into disjoint parts Ω^{f} and Ω^{s}, containing a fluid and a solid, respectively. Ω^{s} represents a metal structure that almost entirely encloses Ω^{f}, by means of the outer non-designable solid shell $\Omega^{\text{s,fix}} \subset \Omega^{\text{s}}$. A coolant driven by a traction $\boldsymbol{t}_{\text{F}}$ occupies Ω_{f}, enters at the inlet Γ^{in}, and exits at the outlet Γ^{out}. The thermal driving force is a hot gas stream heating Γ^{U} by convection. The boundary Γ^{q} is adiabatic.

2.1. State problems

A standard density-based TO formulation is used wherein the design at a point $\mathbf{x} \in \Omega$ is described by $\rho(\mathbf{x}) \in [0, 1]$, such that $\rho(\mathbf{x}) = 1 \Leftrightarrow \mathbf{x} \in \Omega^{\text{s}}$. The design is regularized using a linear density filter (Bourdin 2001), with radius R, to avoid checkerboards and mesh dependency (Sigmund and Petersson 1998). The regularized design is denoted $\hat{\rho} = \hat{\rho}(\rho)$.

2.1.1. The flow problem

The strong formulation of the Stokes flow problem in Ω, augmented with a Brinkman term $-\alpha(\hat{\rho})\boldsymbol{u}$ (Borrvall and Petersson 2003) for the purpose of interpolating between Ω^{s} and Ω^{f}, is to find the velocity $\boldsymbol{u}$ and the hydrostatic pressure p satisfying

$$(\mathbb{F}) \begin{cases} \nabla \cdot \boldsymbol{\sigma} - \alpha(\hat{\rho})\boldsymbol{u} = \mathbf{0} & \text{in } \Omega, \\ \nabla \cdot \boldsymbol{u} = 0 & \text{in } \Omega, \\ \boldsymbol{u} = \mathbf{0} & \text{on } \Gamma \setminus (\Gamma^{\text{in}} \cup \Gamma^{\text{out}}), \\ \boldsymbol{u} = \mathbf{0} & \text{in } \Omega^{\text{s,fix}}, \\ \boldsymbol{\sigma}\boldsymbol{n} = \boldsymbol{t}_{\text{F}} & \text{on } \Gamma^{\text{in}}, \\ \boldsymbol{\sigma}\boldsymbol{n} = \mathbf{0} & \text{on } \Gamma^{\text{out}}. \end{cases}$$

The fluid is characterized by the constitutive relation

$$\boldsymbol{\sigma} = -p\boldsymbol{I} + \mu(\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^{\mathrm{T}}),$$

where $\mu > 0$ is the dynamic viscosity. The Brinkman parameter

$$\alpha(\hat{\rho}) = \overline{\alpha}\frac{\hat{\rho}}{1 + q(1 - \hat{\rho})} \tag{1}$$

describes the impermeability of the material as a function of the regularized design variable, where q is a penalty factor, and the impermeability parameter $\overline{\alpha} \gg 1$ represents the resistance to flow permeation in Ω^{s}.

In Section 3.1, the numerical solving of the flow problem is discussed, using both a mixed and a penalty approach.

2.1.2. The heat problem

The strong formulation of the convection–diffusion heat problem in Ω is to find the temperature T satisfying

$$(\mathbb{T})\begin{cases} -\nabla\cdot\big(k(\hat{\rho})\nabla T\big) + c\nabla T\cdot\boldsymbol{u} = 0 & \text{in } \Omega,\\ T = T_0 & \text{on } \Gamma^{\mathrm{in}},\\ -k(\hat{\rho})\nabla T\cdot\boldsymbol{n} = U(T - T_\infty) & \text{on } \Gamma^{\mathrm{U}},\\ -k(\hat{\rho})\nabla T\cdot\boldsymbol{n} = 0 & \text{on } \Gamma^{\mathrm{out}}\cup\Gamma^{\mathrm{q}}, \end{cases}$$

where T_∞ is the ambient temperature of the hot gas, U is the heat transfer coefficient, and $c = c_{\mathrm{v}}\rho_{\mathrm{f}}$, in which c_{v} and ρ_{f} are the specific heat capacity and the material density of the coolant, respectively. The thermal conductivity $k(\hat{\rho}) = k_{\mathrm{f}} + (k_{\mathrm{s}} - k_{\mathrm{f}})\hat{\rho}$ is linearly interpolated between the solid (k_{s}) and fluid (k_{f}) conductivities.

3. Discretization

The state problems are solved with the finite element (FE) method on the approximated domain Ω_h. The FE approximated hydrostatic pressure, velocity and temperature are denoted with p_h, $\boldsymbol{u}_h$ and T_h, respectively. The mesh is structured and made up of n_{e} equally-sized and cuboid-shaped eight-noded HEX-elements. The same mesh is used for both the flow and the heat problem, and both use standard Galerkin methods.

For the design problem, an element-wise constant design density ρ_h is used. The corresponding regularized, physical density $\hat{\rho}_h = \hat{\rho}_h(\rho_h)$ is also approximated as element-wise constant, with element values denoted $\hat{\rho}_e$.

3.1. The FE flow problem

Two different approaches are investigated for solving the flow problem, and a comparison between them is to be found in Section 4.2.1.

3.1.1. Penalty approach

In the penalty approach, the volume continuity equation in $(\mathbb{F})$ is not considered as a separate equation of state. Instead, the penalty parameter $\lambda \geq 0$ is introduced, and the pressure is

$$p = -\lambda\nabla\cdot\boldsymbol{u}, \tag{2}$$

which embodies the near-incompressible assumption, in the sense that $\lambda \to \infty \Rightarrow \nabla\cdot\boldsymbol{u} \to 0$. The velocity is approximated as continuous and element-wise trilinear (Q1) and the resulting matrix formulation reads

$$\mathbf{K}_{\mathrm{F}}(\hat{\rho}_h)\mathbf{u} = \mathbf{f}_{\mathrm{F}}, \tag{3}$$

where $\mathbf{u} \in \mathbb{R}^{3n_{\mathrm{n}}}$ is the global velocity vector, and n_{n} is the number of nodes in the domain. The element stiffness matrix and load vector are

$$\begin{aligned}\mathbf{K}^e_{\mathrm{F}}(\hat{\rho}_e) &= \int_{\Omega_e} \alpha(\hat{\rho}_e)\mathbf{N}_{\mathrm{F}}^{\mathsf{T}}\mathbf{N}_{\mathrm{F}}\,\mathrm{d}V + \int_{\Omega_e} \mathbf{B}_{\mathrm{F}}^{\mathsf{T}}\mathbf{C}_{\mu}\mathbf{B}_{\mathrm{F}}\,\mathrm{d}V + \int_{\Omega_e} \mathbf{B}_{\mathrm{F}}^{\mathsf{T}}\mathbf{C}_{\lambda}\mathbf{B}_{\mathrm{F}}\,\mathrm{d}V,\\ \mathbf{f}^e_{\mathrm{F}} &= \int_{\Gamma^{\mathrm{in}}\cap\Gamma_e} \mathbf{N}_{\mathrm{F}}^{\mathsf{T}}\boldsymbol{t}_{\mathrm{F}}\,\mathrm{d}A,\end{aligned} \tag{4}$$

where $\mathbf{N}_{\mathrm{F}}$ is the flow shape function matrix (such that $\boldsymbol{u}_h|_{\Omega_e} = \mathbf{N}_{\mathrm{F}}\mathbf{u}_e$) and $\mathbf{B}_{\mathrm{F}}$ its gradient, $\mathbf{C}_{\mu}$ and $\mathbf{C}_{\lambda}$ are constitutive matrices for the viscosity and the penalty contributions, respectively, and the element domain $\Omega_e \subset \Omega_h$.

The use of Q1 elements requires reduced integration on the penalty term in (4) to avoid volumetric locking (Hughes 2000). The penalty approach has shown good capability of approximating the velocity, despite the possible occurrence of pressure instabilities (Johnson and Pitkäranta 1982). However, the performance of this method depends on the choice of the penalty parameter λ in (2). This parameter cannot be too low, or otherwise the flow will not be incompressible (enough). And if the parameter is too large, the convergence of the linear state problem is affected, even when using direct methods such as Cholesky factorization. Hughes (2000) suggests a value of $10^7 < \lambda/\mu < 10^9$. Given the specific problem and parameters herein, a good number is found to be in the range 10^9–10^{10}.

3.1.2. Mixed approach

In the mixed approach, where both the velocity and the pressure are unknown, the volume continuity equation in ($\mathbb{F}$) is considered as a separate equation of state. A pressure-stabilized Q1–Q0 method is used, in which the pressure is approximated as element-wise constant (Q0) and the pressure-jump between adjacent elements is penalized to ensure stability and accuracy of the method (Hughes and Franca 1987). The matrix formulation of this approach reads

$$\begin{bmatrix}\mathbf{K}_{\mathrm{FF}}(\hat{\rho}_h) & \mathbf{K}_{\mathrm{FP}}\\ \mathbf{K}_{\mathrm{FP}}^{\mathsf{T}} & \mathbf{K}_{\mathrm{PP}}\end{bmatrix}\begin{bmatrix}\mathbf{u}\\ \mathbf{p}\end{bmatrix} = \begin{bmatrix}\mathbf{f}_{\mathrm{F}}\\ \mathbf{0}\end{bmatrix},$$

where $\mathbf{p} \in \mathbb{R}^{n_{\mathrm{e}}}$ is the global pressure vector. The element contributions are

$$\begin{aligned}\mathbf{K}^e_{\mathrm{FF}}(\hat{\rho}_e) &= \int_{\Omega_e} \alpha(\hat{\rho}_e)\mathbf{N}_{\mathrm{F}}^{\mathsf{T}}\mathbf{N}_{\mathrm{F}}\,\mathrm{d}V + \int_{\Omega_e} \mathbf{B}_{\mathrm{F}}^{\mathsf{T}}\mathbf{C}_{\mu}\mathbf{B}_{\mathrm{F}}\,\mathrm{d}V,\\ \mathbf{K}^e_{\mathrm{FP}} &= -\int_{\Omega_e} \operatorname{div}\mathbf{N}_{\mathrm{F}}^{\mathsf{T}}\,\mathrm{d}V,\\ \mathbf{K}^e_{\mathrm{PP}} &= -\delta h_{\max}\sum_{f\in F_e} A_{ef}\mathbf{c}_e\mathbf{c}_e^{\mathsf{T}},\end{aligned} \tag{5}$$

where A_{ef} is the area of the face between elements e and f, F_e is the set of faces on $\Gamma_e \setminus \Gamma_h$, $h_{\max}$ is the largest element side length (which is the same for all elements in the uniform mesh used), and the non-negative number δ is a user-specified parameter that should be large enough to get rid of oscillations in the pressure field, but small enough to avoid (roughly) constant pressure solutions—leading, similarly to what happens with a too small penalty parameter, to designs without channels. The vector $\mathbf{c}_e \in \mathbb{R}^{n_{\mathrm{f}}}$

is defined from the pressure jump between two adjacent elements as

$$[\![p_h]\!] = p_e - p_f = \underbrace{[1 \quad 0 \quad \dots \quad -1 \quad 0]}_{\mathbf{c}_e} \mathbf{p}_e,$$

where $\mathbf{p}_e$ collects p_e and the neighbouring element pressures, such that $n_\mathrm{f} \leq 7$ in 3D, excluding faces on Γ_h.

3.2. The FE heat problem

The temperature is approximated as continuous and element-wise trilinear. The matrix version of the heat problem reads

$$\mathbf{K}_\mathrm{T}(\hat{\rho}_h, \boldsymbol{u}_h)\mathbf{t} = \mathbf{f}_\mathrm{T},$$

where $\mathbf{t} \in \mathbb{R}^{n_\mathrm{n}}$ is the global temperature vector. The element stiffness matrix and load vector in the heat problem are

$$\begin{aligned} \mathbf{K}_\mathrm{T}^e(\hat{\rho}_e, \boldsymbol{u}_h) &= \int_{\Omega_e} k(\hat{\rho}_e)\mathbf{B}_\mathrm{T}^\mathsf{T}\mathbf{B}_\mathrm{T}\,\mathrm{d}V + \int_{\Omega_e} c\mathbf{N}_\mathrm{T}^\mathsf{T}\boldsymbol{u}_h^\mathsf{T}\mathbf{B}_\mathrm{T}\,\mathrm{d}V + \int_{\Gamma^\mathrm{U}\cap\Gamma_e} U\mathbf{N}_\mathrm{T}^\mathsf{T}\mathbf{N}_\mathrm{T}\,\mathrm{d}A, \\ \mathbf{f}_\mathrm{T}^e &= \int_{\Gamma^\mathrm{U}\cap\Gamma_e} UT_\infty\mathbf{N}_\mathrm{T}^\mathsf{T}\,\mathrm{d}A, \end{aligned} \tag{6}$$

where $\mathbf{N}_\mathrm{T}$ is the temperature shape function matrix (such that $T_h|_{\Omega_e} = \mathbf{N}_\mathrm{T}\mathbf{t}^e$) and $\mathbf{B}_\mathrm{T}$ its gradient.

The convection term in the heat problem can cause oscillations in the temperature solution for high flow speeds. A cheap way of taking care of these oscillations is by adding the Streamline-Upwind (SU) stabilization

$$\mathbf{K}_\mathrm{SU}^e(\boldsymbol{u}_h) = \int_{\Omega_e} c^2\tau(\boldsymbol{u}_h)\mathbf{B}_\mathrm{T}^\mathsf{T}\boldsymbol{u}_h\boldsymbol{u}_h^\mathsf{T}\mathbf{B}_\mathrm{T}\,\mathrm{d}V$$

to (6). The stabilization parameter, evaluated at each Gauss point, is

$$\begin{aligned} \tau(\boldsymbol{u}_h) &= \frac{h_\mathrm{max}}{2c\|\boldsymbol{u}_h\|^2}\mathbf{X}^\mathsf{T}\boldsymbol{u}_h, \\ [\mathbf{X}]_i &= \coth(Pe_i) - \frac{1}{Pe_i}, \\ Pe_i &= \frac{ch_\mathrm{max}[\boldsymbol{u}_h]_i}{k_\mathrm{f}}, \end{aligned}$$

where $\|\cdot\|$ is the Euclidean norm, Pe_i the directional Péclet number and $[\cdot]_i$ represents the ith component of the vector.

3.3. Voxelization

The original TO implementation in PETSc (Aage, Andreassen, and Lazarov 2015) only allows cuboid geometries with a uniform element size. The aim is to extend the analysis beyond cuboid geometries, and to that end a so-called voxelization method is employed, to handle more general geometries. The voxelization method used is based on the work by Zhang *et al.* (2021), where standard triangle language (STL) files are used for the geometry description.

The voxelization approach eliminates the need for advanced and time-consuming manual meshing procedures, while still allowing general geometries to be analysed. Also, the element quality will turn out to be really good, since all elements are cuboids of similar size, and will be very near perfect cubes

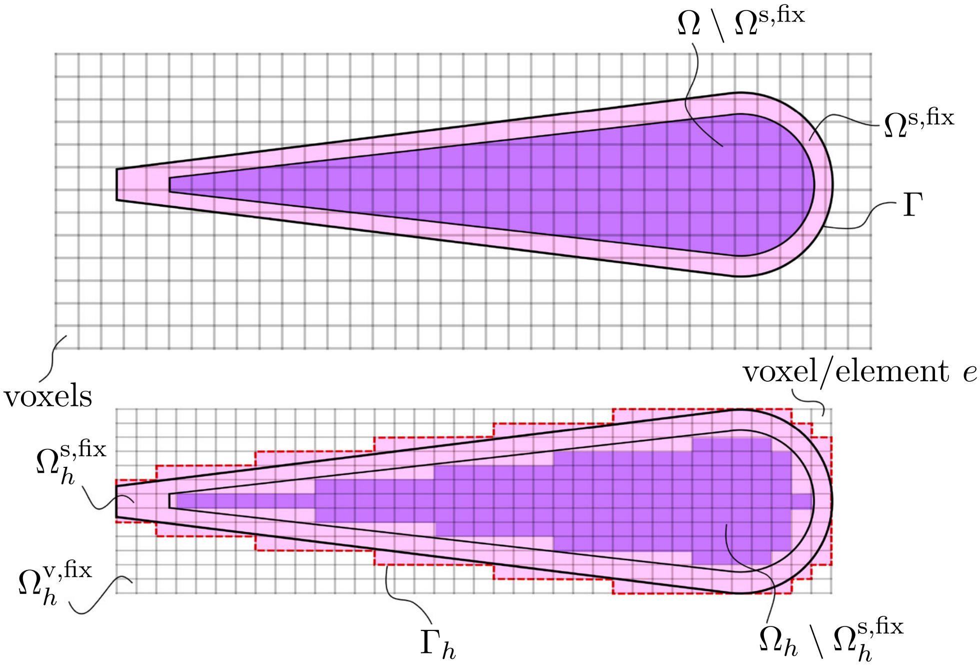

$\Omega \setminus \Omega^{\mathrm{s,fix}}$
$\Omega^{\mathrm{s,fix}}$
Γ
voxels
voxel/element e
$\Omega_h^{\mathrm{s,fix}}$
$\Omega_h^{\mathrm{v,fix}}$
Γ_h
$\Omega_h \setminus \Omega_h^{\mathrm{s,fix}}$

Figure 3. Voxelization of (a 2D version of) the geometry, later described in Section 4.1.

if the number of elements in each direction is chosen wisely. This is particularly important for iterative linear solvers, whose performance can be greatly affected by mesh quality.

The output of the voxelization, illustrated in Figure 3, is a cuboid mesh consisting of voxels. After the mesh is scaled and translated, it encloses Ω perfectly, and the voxels are recognized as the FEs used to discretize Ω, by assigning elements intersecting $\Omega^{\mathrm{s,fix}}$ to $\Omega_h^{\mathrm{s,fix}}$, the remaining elements intersecting Ω to the designable domain $\Omega_h \setminus \Omega_h^{\mathrm{s,fix}}$, and the rest to $\Omega_h^{\mathrm{v,fix}}$.

The total (cuboid) domain is $\Omega_h^{\mathrm{tot}} = \Omega_h \cup \Omega_h^{\mathrm{v,fix}}$, but the state problems are solved on Ω_h only, by means of prescribing a zero-value to degrees-of-freedom (DOFs) outside Ω_h. Furthermore, $\rho_e = \hat{\rho}_e = 0$ and $\rho_e = \hat{\rho}_e = 1$ are prescribed for elements in $\Omega_h^{\mathrm{v,fix}}$ and $\Omega_h^{\mathrm{s,fix}}$, respectively.

Implementation-wise, two different STL files, describing Ω and $\Omega^{\mathrm{s,fix}}$, are needed to make the partitioning of domains. The result is two vectors identifying elements in $\Omega_h^{\mathrm{s,fix}}$ and the designable domain $\Omega_h \setminus \Omega_h^{\mathrm{s,fix}}$, respectively.

3.3.1. Boundary detection algorithm

The imposing of the boundary conditions is refined compared with Zhang *et al.* (2021), in which the conditions are imposed on a set of elements closest to the boundary. Herein, the boundary conditions are instead only imposed on sets of nodes belonging to the voxelized boundary Γ_h, by means of a boundary detection algorithm.

For each boundary condition, except adiabatic and traction-free boundaries, an STL file is provided, which is voxelized using the algorithm by Zhang *et al.* (2021). In the numerical example, two STL files are provided describing Γ^{U} and Γ^{in}, respectively. The output from the voxelization consists of vectors identifying HEX-elements intersecting the boundary definitions in the STL files. To identify which nodes of each HEX-element are on Γ_h, an algorithm to identify all nodes lying on Γ_h is used, see Figure 4. The result is a vector ($\mathbf{b}$) that can be used together with the output from the voxelization process to identify the exact nodes to which the boundary conditions should be applied.

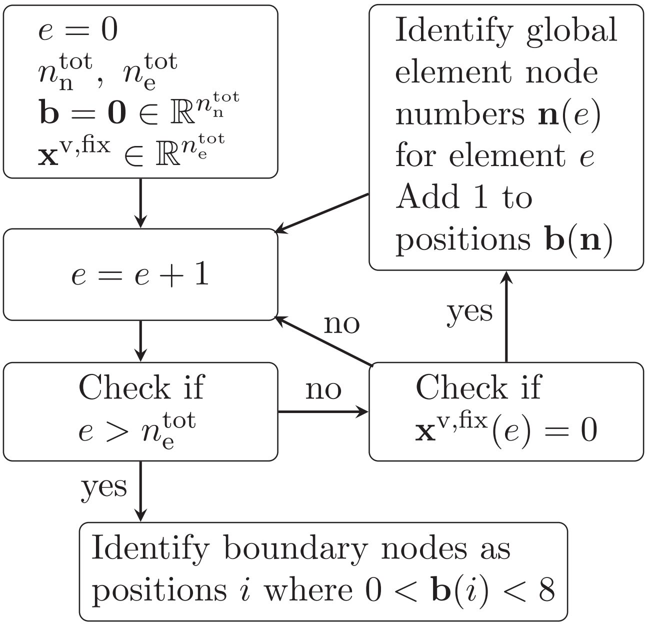

$e = 0$
$n_\mathrm{n}^\mathrm{tot}$, $n_\mathrm{e}^\mathrm{tot}$
$\mathbf{b} = \mathbf{0} \in \mathbb{R}^{n_\mathrm{n}^\mathrm{tot}}$
$\mathbf{x}^\mathrm{v,fix} \in \mathbb{R}^{n_\mathrm{e}^\mathrm{tot}}$
Identify global element node numbers $\mathbf{n}(e)$ for element e
Add 1 to positions $\mathbf{b}(\mathbf{n})$
$e = e + 1$
no
yes
Check if $e > n_\mathrm{e}^\mathrm{tot}$
no
Check if $\mathbf{x}^\mathrm{v,fix}(e) = 0$
yes
Identify boundary nodes as positions i where $0 < \mathbf{b}(i) < 8$

Figure 4. Flow chart for boundary node identification process. Here, e is the element number, $n_\mathrm{n}^\mathrm{tot}$ is the number of nodes in Ω_h^tot, n_e^tot is the number of elements in Ω_h^tot, $\mathbf{x}^\mathrm{v,fix}$ is a vector indicating if an element belongs to $\Omega_h^\mathrm{v,fix}$, and the output $\mathbf{b}$ is a vector indicating if a node is on the boundary or not.

3.4. The design problem

The objective is to minimize the maximum temperature in Ω_h. The maximum temperature is expected to be found on the convection boundary $\Gamma_h^\mathrm{U} = \Gamma^\mathrm{U} \cap \Gamma_h$, and to this end the maximum temperature is approximated using an objective function based on an ℓ^P-norm as

$$\phi_h\big(\mathbf{t}(\hat{\rho}_h, \mathbf{u}(\hat{\rho}_h))\big) = \left(\sum_{i=1}^{n_\mathrm{n}} [\mathbf{f}_\mathrm{T}]_i [\mathbf{t}(\hat{\rho}_h, \mathbf{u}(\hat{\rho}_h))]_i^P\right)^{1/P}, \tag{7}$$

where P is the power parameter, and $\mathbf{f}_\mathrm{T}$ is included in a compliance-like fashion to weigh each nodal temperature in order to obtain a measure only over the nodes on Γ_h^U.

The mass flow constraint reads

$$g_h\big(\mathbf{u}(\hat{\rho}_h)\big) = \int_{\Gamma_h^\mathrm{in}} \rho_\mathrm{f} \boldsymbol{u}_h(\hat{\rho}_h) \cdot (-\boldsymbol{n})\, \mathrm{d}A \leq \overline{m}, \tag{8}$$

where $-\boldsymbol{n}$ is the inward boundary normal, $\overline{m}$ is the specified maximum mass flow, and the surface integral is (exactly) evaluated using a 2 × 2 numerical integration scheme.

The complete design problem reads

$$(\mathbb{P}) \begin{cases} \displaystyle\min_{\rho_h:\Omega_h^\mathrm{tot} \to \mathbb{R}} & \phi_h\big(\mathbf{t}(\hat{\rho}_h, \mathbf{u}(\hat{\rho}_h))\big), \\ \text{s.t.} & \begin{cases} g_h\big(\mathbf{u}(\hat{\rho}_h)\big) \leq \overline{m}, \\ \rho_h \in [0,1] \text{ in } \Omega_h, \\ \rho_h = 1 \text{ in } \Omega_h^\mathrm{s,fix}, \\ \rho_h = 0 \text{ in } \Omega_h^\mathrm{v,fix}. \end{cases} \end{cases}$$

Table 1. Continuation parameters.

$[\overline{\alpha}_{c1}, \overline{\alpha}_{c2}, \overline{\alpha}_{c3}, \overline{\alpha}_{c4}]$ (kg/m^{3}s)	$[q_{c1}, q_{c2}, q_{c3}, q_{c4}]$ (–)
$\{10^8, 10^8, 10^9, 10^{10}\}$	{100, 10, 10, 10}

Note: The notation $(\cdot)_{ci}$ indicates the *i*th continuation step.

Table 2. Numerical and MMA parameters for the design problem.

R (mm)	P	$\overline{m}$ (g/s)	$s^{\text{init}}_{c1,c2}$	$s^{\text{init}}_{c3,c4}$	$s^{\text{decr}}_{c1,c2}$	$s^{\text{decr}}_{c3,c4}$	$s^{\text{incr}}_{c1,c2}$	$s^{\text{incr}}_{c3,c4}$
1.5	8	1	0.5	0.3	0.7	0.5	1.2	1.1

Note: The notation $(\cdot)_{ci}$ indicates the *i*th continuation step.

Table 3. Numerical values for the flow problem.

$t_F \cdot \boldsymbol{n}(:= t_{\text{in}})$ on Γ_{in} (kPa)	$t_F \times \boldsymbol{n}$ on Γ_{in} (kPa)	μ (kg/ms)	λ (–)
−400	$\mathbf{0}$	1	10^9

Table 4. Numerical values for the heat problem.

k_f, k_s (W/mK)	T_∞ (K)	U (W/m^{2}K)	T_0 (K)	c_v (J/kg·K)	ρ_f (kg/m^3)
0.031, 10.2	1373.15	3500	673.15	1001	5

4. Numerical examples

PETSc (v. 3.16.3) (Smith 2011) is used for the numerical calculations. The TO implementation is inspired by the code described in Aage, Andreassen, and Lazarov (2015), and runs on the high performance computing clusters Sigma (NSC 2023a) and Tetralith (NSC 2023b), where each computational node has two Intel® Xeon® Gold 6130 CPUs for a total of 32 cores and 96 GiB of RAM.

The design problem is solved using the method of moving asymptotes (MMA) solver (Svanberg 1987), with analytical gradients computed using the discrete adjoint method (see the appendix). The initial guess is $\rho_e = 0.5$ everywhere. To avoid getting stuck in poor local minima, while achieving close-to-binary-valued designs, a couple of continuation steps for the Brinkman parameters $\overline{\alpha}$ and q in (1) is needed, see Table 1. Each continuation step runs for a maximum of 250 iterations, except for the final step, where 500 iterations is allowed. A stopping criteria is set to $\|\boldsymbol{\rho}^k - \boldsymbol{\rho}^{k-1}\|_\infty < 0.001$, where $\boldsymbol{\rho}^k$ is the design density vector in MMA iteration k, but this criteria was never met during this work. MMA and design problem parameters are to be found in Table 2.

The majority of the time to convergence of the optimization problem is spent solving the flow problem in each MMA iteration. The penalty approach is used to produce the designs shown in Section 4.2, and the solution of the state problem is obtained using Cholesky factorization from the parallel MUMPS suite (v. 5.4.1) (Amestoy *et al.* 2001, 2019). An iterative refinement process with three steps is implemented to achieve a slightly better residual at a very cheap cost. The flow problem parameters are to be found in Table 3.

The heat state problem is solved by the iterative `FGMRES` solver, with a multigrid preconditioner (`PCMG`). Parameters for the heat problem are to be found in Table 4.

The adjoint problems are solved with the same solvers as the respective state problem. The symbolic factorization of the flow stiffness matrix is done once in the beginning of the optimization process, while the numerical factorization is done every iteration, but is saved during the MMA iteration for use in the adjoint flow problem, making that solution time negligible.

The value of the dynamic viscosity μ in Table 3 is not based on the fluid material parameter, which is several orders of magnitude lower. Instead, a modelling choice has been made to use an

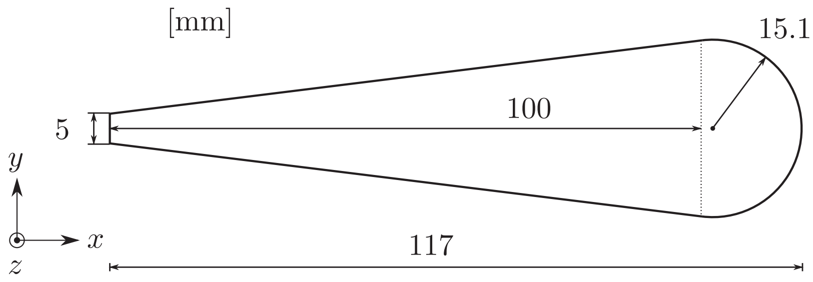


Figure 5. 2D geometry to be extruded in the z-direction (units: millimetres).

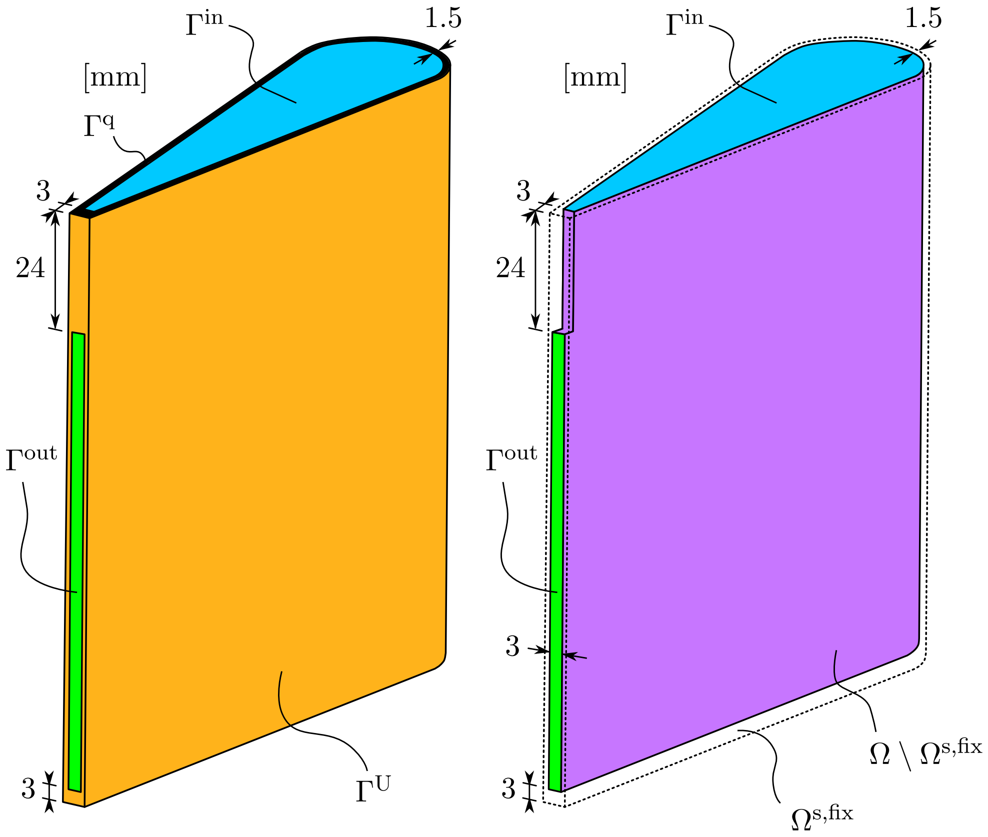


Figure 6. Illustration of (left) boundary definitions and (right) the designable domain $\Omega \setminus \Omega^{\text{s,fix}}$ with inlet and outlet. The bottom (not shown here) is adiabatic (part of Γ^{q}).

increased value, mainly to get a smooth convergence of the model. It is stressed that this choice is not a fundamental aspect of this work.

4.1. Geometry

A tapered 2D shape has been extruded in the normal direction to obtain a geometry resembling that of a true guide vane. The height is 120 mm, and the other dimensions are to be found in Figure 5. The boundary definitions are given in Figure 6, and the domain definitions are illustrated in Figure 7.

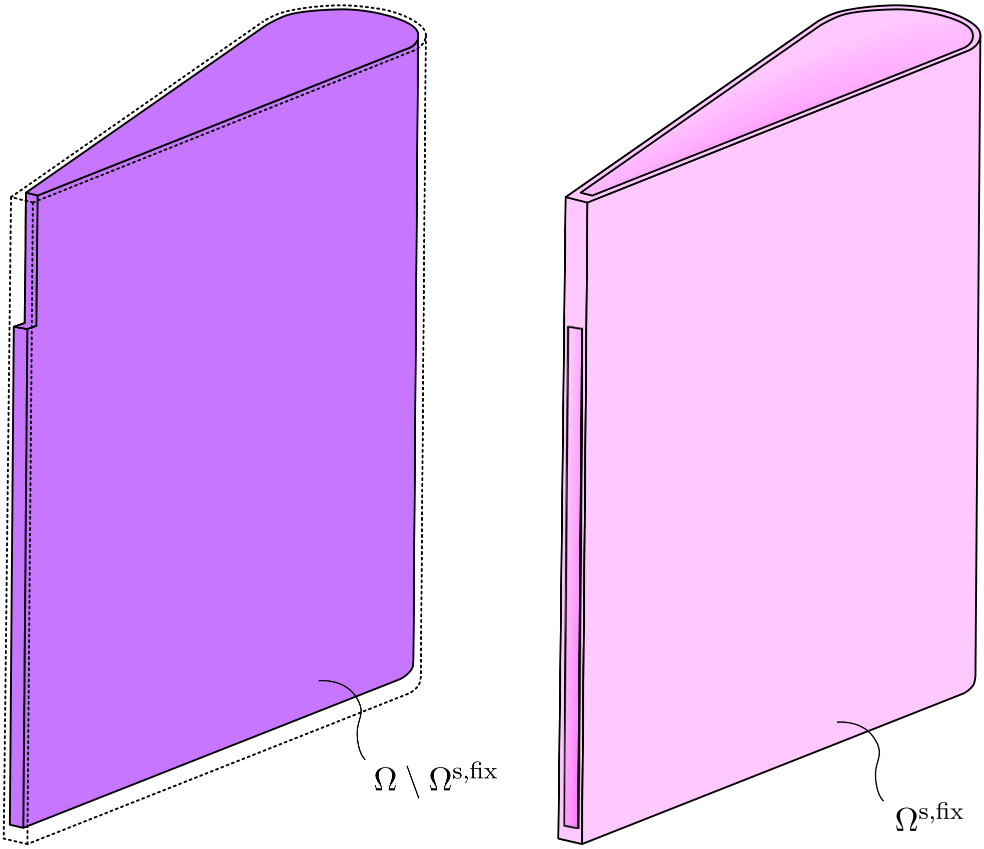

$\Omega \setminus \Omega^{\mathrm{s,fix}}$
$\Omega^{\mathrm{s,fix}}$

Figure 7. Illustration of (left) the design domain and (right) the fixed domain.

The complete mesh (covering the total domain Ω_h^{tot}) is 240 × 30 × 200 elements, utilizing symmetry in the y-direction. For Ω_h, the number of elements is 910,410, and the number of nodes is 969,443.

4.2. Results

The visualization in Figure 8 shows the solid domain $\Omega_h^{\mathrm{s}} = \{\Omega_e \mid \hat{\rho}_e \geq 0.5\}$ after each continuation step. The colour indicates the relative normal distance from the symmetry plane, to illustrate the depth in the figure. The first two iteration steps show designs without continuous channels from inlet to outlet, due to the low value of $\overline{\alpha}$, which makes significant flow through solid parts possible. The channels are more clearly defined when raising the impermeability parameter $\overline{\alpha}$ in steps 3 and 4.

The iteration history in Figure 9 shows good convergence, with some oscillations in the fourth continuation step. Disregarding the spikes in the maximum temperature in Figure 9(a), which reach 1390K but are cut from the figure for visualization reasons, the appearance of the objective function and the maximum temperature are similar, indicating good agreement between the two measures. Probably, the number of iterations in the initial phase could have been reduced, based on the flat appearance of the objective function during the final parts of the first two continuation steps.

An enlarged version of Figure 8(d) appears in Figure 10, together with a temperature plot of the solid domain with flow streamlines, also temperature coloured. Figure 11 shows isosurfaces of the optimized solid domain $\Omega_h^{\mathrm{s,opt}} = \{\boldsymbol{x} \in \Omega_h \mid \hat{\rho}_h(\boldsymbol{x}) \in [1/2, 1]\}$ and the optimized flow domain

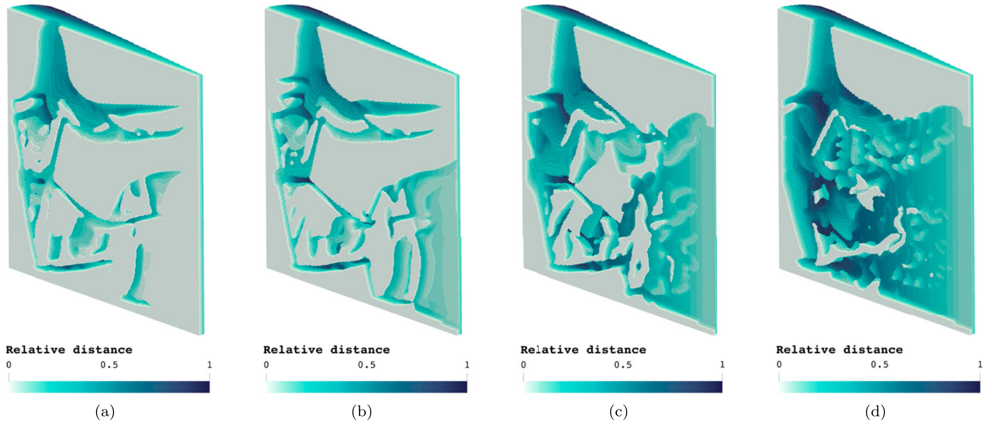


Figure 8. Ω_h^s after each continuation step, coloured by the relative normal distance to the symmetry plane: (a) Step 1; (b) Step 2; (c) Step 3; (d) Step 4.

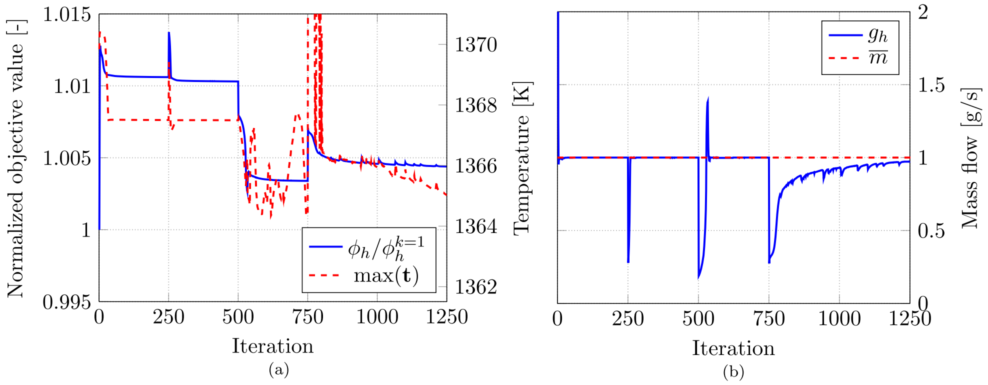


Figure 9. Iteration history. The spikes are due to the change of parameters at each continuation step. The objective function is normalized by its value in MMA iteration $k = 1$: (a) normalized objective value and maximum nodal temperature in Ω_h; (b) constraint function value g_h, and mass flow limit $\overline{m}$.

$\Omega_h^{\mathrm{f,opt}} = \{\boldsymbol{x} \in \Omega_h \mid \hat{\rho}_h(\boldsymbol{x}) \in [0, 1/2]\}$. It seems to be optimal to only have one large channel occupied by some free-floating islands (solid subdomains disconnected from Γ_h), clearly visible in Figures 10(b) and 11(b) as regions with considerably lower temperature than other solid regions nearby. Probably, the role of these islands is to slow the flow, so that the mass flow constraint is fulfilled. The free-floating islands makes it impossible to realize the obtained designs. To achieve designs without these islands, it is necessary to include something in the optimization problem that prohibits the forming of islands, either in an heuristic way where islands are detected and removed, or by improving the structural integrity of the design by introducing a linear elasticity field.

In Figure 10(b), it can be seen that the fluid temperature at the outlet is above 1200K, indicating that the coolant has taken up almost as much thermal energy as possible. It is also noted that the minimum temperature in Ω_h is lower than the inlet temperature T_0. This numerical artefact seems to have little impact on the resulting design.

The temperature plot in Figure 12(a) shows that the highest temperatures are found on the leading edge, while the largest temperature gradients are found close to the inflow boundary, visible in Figure 12(b). The way of imposing the inlet temperature boundary condition has significant impact on the temperature gradients close to the inlet, since not only the inlet is restricted to T_0, but also the

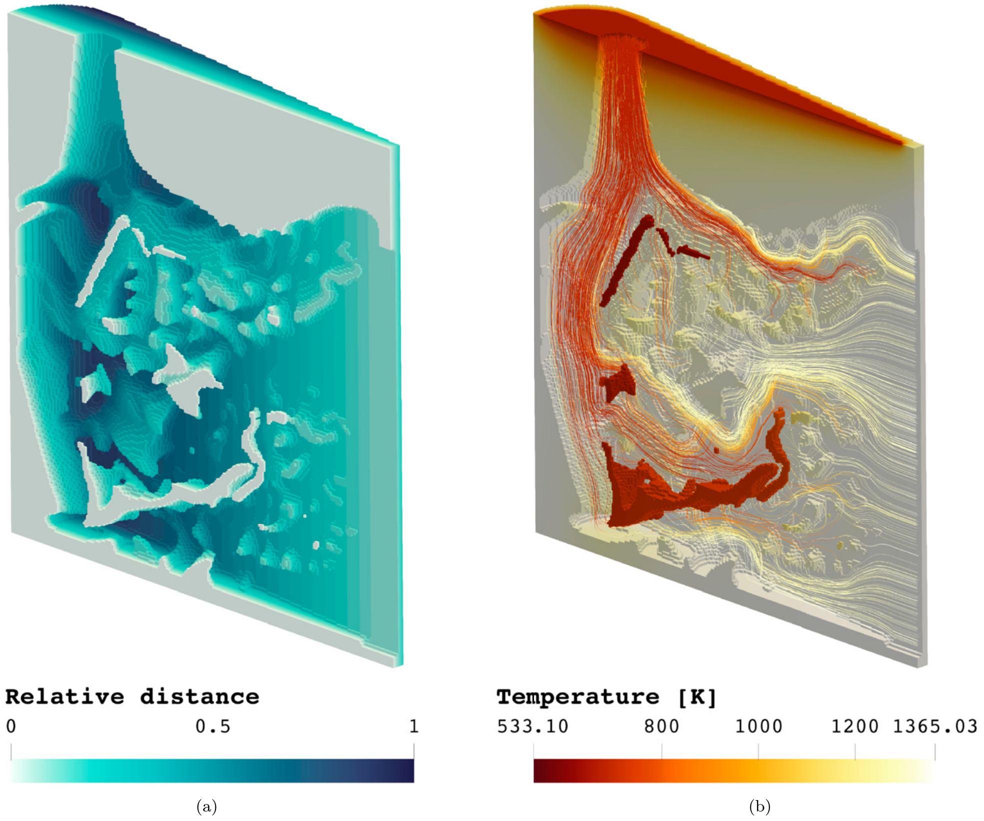


Figure 10. $\Omega_h^{\mathrm{s,opt}}$ after the fourth continuation step. The temperature range is [min(T), max(T)]: (a) the relative normal distance to the symmetry plane; (b) the temperature of the solid domain and the flow streamlines.

entire top surface, in an unnatural way. This effect is extra influential due to the relatively small inlet-to-top area ratio. However, this is not a fundamental aspect of the model; it is believed that this effect could be attenuated with more accurate boundary conditions for the heat problem. Apart from the top region, the temperature distribution visible in Figure 12 is fairly uniform, which can be expected when minimizing the maximum temperature.

The final volume fraction is around 76%, with respect to Ω_h. A binary-measure is calculated as $\sum_{e=1}^{n_\mathrm{e}} 4\hat{\rho}_e(1-\hat{\rho}_e)/n_\mathrm{e} = 0.143$, which is interpreted as sufficient in this context.

4.2.1. Mixed versus penalty

The penalty approach, used to produce the optimized designs herein, is compared with the mixed Q1–Q0 approach in terms of the time it takes to solve the flow state problem once in the beginning of the optimization process, depending on the number of cores used to run the example. Included in the comparison are three different set-ups: the penalty approach solved with the same settings as described in Section 4 together with the REDISTRIBUTE preconditioner, the mixed approach solved with an iterative solver, and the mixed approach with the REDISTRIBUTE preconditioner solved with an iterative solver. For this comparison, an optimized version of PETSc (v. 3.18.1), linked with Intel Math Kernel Library (v. 2018) and MUMPS (v. 5.5.1) (Mumps Technologies 2022), was used on the Tetralith cluster (NSC 2023b).

The REDISTRIBUTE preconditioner works by eliminating the fixed DOF from the system matrix. By doing this, a smaller linear system is obtained, compared with imposing the Dirichlet boundary conditions by zeroing rows and columns using the PETSc function

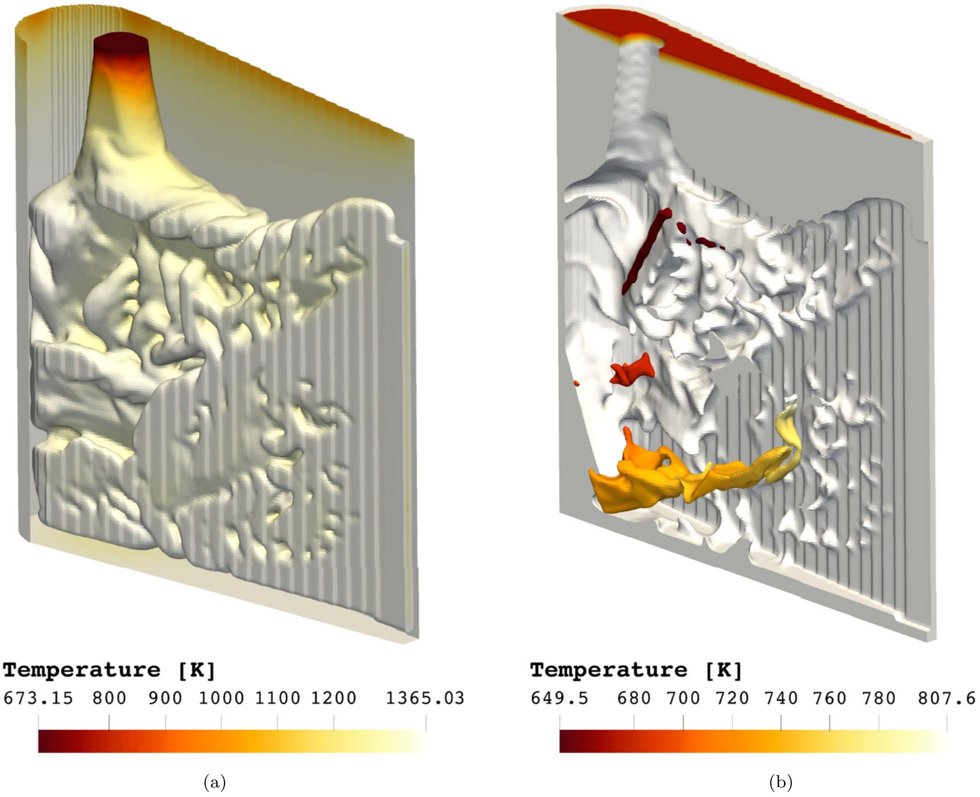
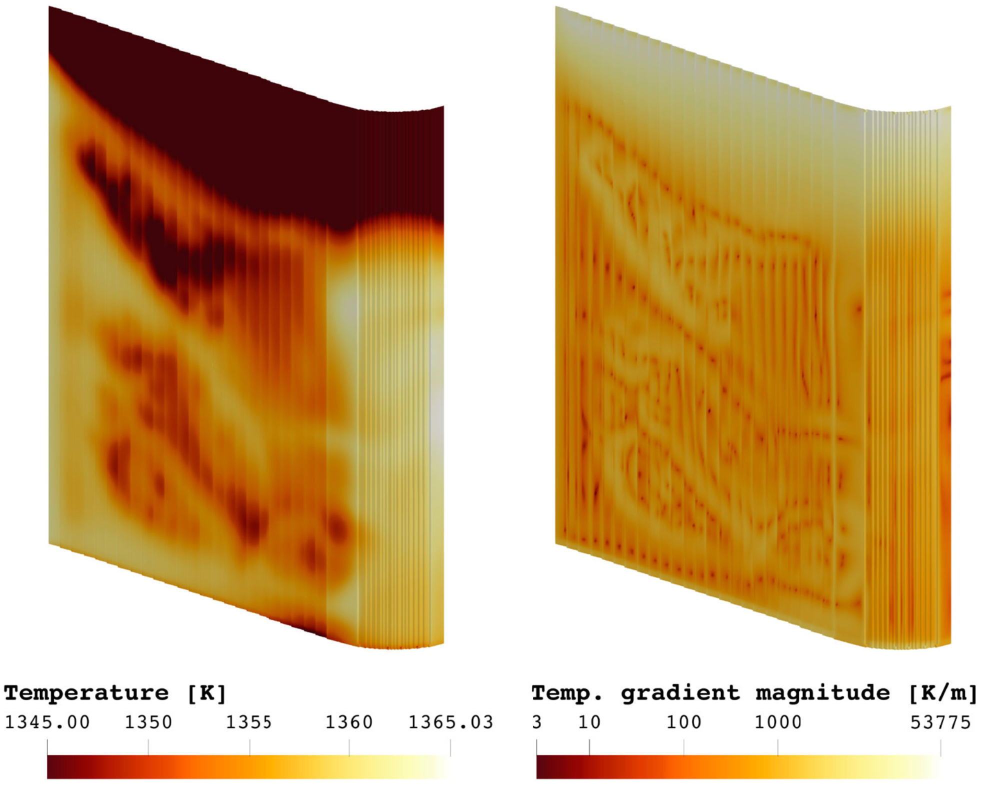


Figure 11. Isosurfaces of the optimized design after the fourth continuation step: (a) $\Omega_h^{\text{f,opt}}$ inside parts of Γ_h; (b) $\Omega_h^{\text{s,opt}}$ with free-floating islands reflected across the symmetry plane.

MatZeroRowsColumns. This reduction in problem size is believed to be able to reduce the computational time for both the mixed and penalty approaches.

An implementation of the mixed approach is done in PETSc based on the DMStag object, which represents a staggered grid, in this case having velocity DOF at the vertices and the pressure DOF associated with the 'element' (DMSTAG_ELEMENT). The matrix problem is solved using the BCGSL solver with the FIELDSPLIT preconditioner, with both parts using the ASM preconditioner. Two stopping tolerances—10^{-10} and 10^{-6}—are used for the relative residual $\mathcal{L}^2$-norm of the preconditioned problem. The smaller tolerance matches the relative residual norm from the direct solver used for the penalty approach, while the larger tolerance is provided as a reference for what possibly can be earned by stopping the iterative solver a bit earlier. The user-specified pressure stabilization parameter in (5) is $\delta = 10^{-2}$.

Figure 13 shows times for solving the state problem once, using the same geometry and mesh as in Section 4.2. The times for assembly of the system matrix for the two approaches are quite similar and negligible, and hence not accounted for. It should be added that the adjoint problem most often took a bit longer to solve than the state problem for the examples treated herein. It is clear that the penalty approach is competitive when fewer—here 32 cores, which correspond to one computational node on Tetralith (NSC 2023b)—computational resources are available. For visualization reasons, the penalty approach without using the REDISTRIBUTE preconditioner is not included in Figure 13. This approach requires 158 seconds on 32 cores and 70 seconds on 256 cores to solve the flow state problem.



(a)

(b)

Figure 12. The temperature and its gradient on the convection boundary Γ^{U}: (a) temperature; (b) temperature gradient with log scale.

The mixed approach is indeed faster when using the REDISTRIBUTE preconditioner, although it is at the same time more unstable and prone to breakdowns, and has a worse convergence relative to the problem size compared with the mixed approach without the REDISTRIBUTE preconditioner. It is not obvious that the mixed set-up using the REDISTRIBUTE preconditioner benefits from also using the FIELDSPLIT as the inner preconditioner, but no better inner preconditioner has been found.

Included in Figure 13 is an illustration of the trade-off for when the mixed approach is competitive with the penalty approach. The solid black trade-off line is half of the time it takes to solve the state flow problem once with the penalty approach using the REDISTRIBUTE preconditioner, and it should be interpreted as the maximum time the mixed approach can take without being inferior to the penalty approach.

When comparing the computational times for the penalty and the mixed approach, there are a number of things to keep in mind. First, when using the penalty approach, solving the adjoint flow problem for a given design is essentially free in terms of computational time compared with solving the state problem. To be competitive on the presented flow problem, with one state and one adjoint flow problem, the iterative solver must thus be (roughly) twice as fast.

Secondly, the times for the mixed approach are most likely not the best that can be achieved, and can probably be improved by further experiments with combinations of the plethora of solvers and preconditioners, and settings for these, offered by PETSc. In addition, iterative linear solvers are well-suited for GPU computations, which might offer further speed-up. In practice, the stopping tolerance for the iterative solver may be overly strict; lowering the tolerance can result in non-negligible savings

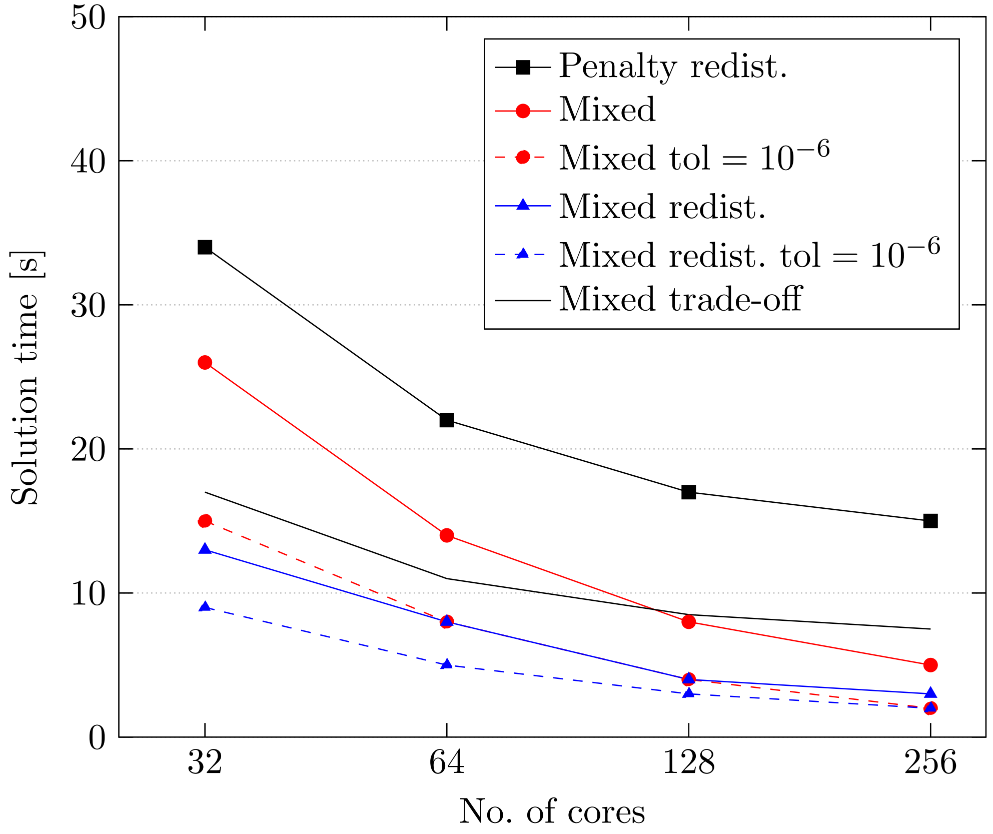


Figure 13. Timing for different approaches and solver settings for the flow state problem.

on computational time—illustrated in Figure 13—though at the same time, less accurate solves might lead to less accurate derivatives and thus more MMA iterations. The possibility of using early stopping for iterative solvers in TO has been investigated recently (Amir, Stolpe, and Sigmund 2010; Amir, Aage, and Lazarov 2014; Limkilde, Evgrafov, and Gravesen 2018).

Finally, the performance of iterative solvers tends to degrade as the design becomes closer to binary-valued. Based on this, it can be expected that times will increase as the optimization process progresses. The experience from the presented numerical examples is that the solution time rises by 10%–30% because of this. However, this effect will probably be counteracted when reusing the flow solution from the previous MMA iteration as an initial guess for the iterative solver, since the change in design, and thus the change in the flow, is reduced between later MMA iterations. Based on experience from the presented numerical examples, the time earned from this can compensate for the time lost owing to the design becoming closer to binary-valued.

5. Result comparison

An investigation of the validity of the flow and temperature response in the TO model is conducted by comparing the TO model with two other models: a high-fidelity CFD simulation using RANS equations and a low-fidelity Stokes flow model, both using a mesh with segregated domains (body-fitted meshes) and solved by conjugate heat transfer analysis.

The computational mesh for the segregated meshing approach consists of about 7.7 million solid and fluid tetrahedral elements and is created as follows: first, an STL file encapsulating the fluid volume, defined by the isosurface $\hat{\rho} = 0.5$, is exported from ParaView (v. 5.8.1) to HyperMesh® (v. 2017.3), where it is remeshed with triangular elements with an approximate size of 3.5×10^{-4} m. The STL file defining Ω, used in the voxelization process, is then imported and meshed with the same element size. Finally, 3D tetrahedral meshes are generated for the fluid and solid parts, and linked via a common solid–fluid interface. This is referred to as a segregated mesh, in contrast to the monolithic mesh used in the TO model.

The RANS simulations are performed utilizing the finite-volume based software ANSYS® Fluent® (v. 2022 R2). The SST turbulence model developed by Menter (1994) is used to perform a steady-state RANS simulation. This implies that the RANS equations along with two additional transport equations for the turbulence kinetic energy and the specific dissipation rate are solved and the model blends effectively the k–ω formulation for the near wall region with the k–ε formulation in the far field (ANSYS, Inc 2022). The velocity and pressure coupling in the momentum equations is done by means of the `COUPLED` scheme in ANSYS Fluent. The spatial discretization for gradients is least-squares cell-based, while pressure, momentum, energy and turbulent quantities use a second-order upwind scheme.

When performing the simulation, residuals level out to 10^{-4} for all quantities, *i.e.* continuity, momentum, energy, k and ω. In addition, velocity and temperatures at local points are monitored to make sure that the solution is stable and converged, which is achieved after about 3000 iterations.

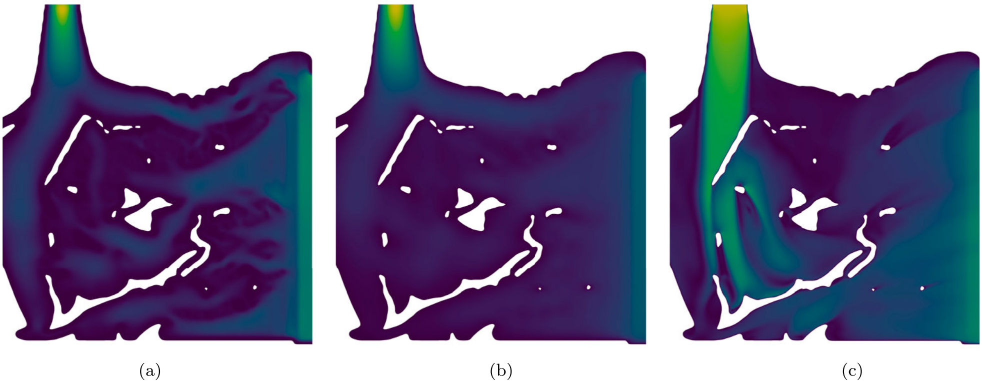
The boundary conditions and the material properties for the two segregated models are exactly those described in Section 4, except for the inlet condition and the viscosity for the RANS model, which use a mass flow boundary condition at the inlet, where 1 g/s of coolant enters, and a viscosity set to 3.25×10^{-5} kg/ms, representing operating conditions. This means that the only difference between the Stokes TO model and the segregated Stokes model is the meshing approach. It is noted that the Brinkman term is not used for the segregated models. The purpose of the segregated Stokes model is to be able to study separately the effect of interpreting the TO design on a body-fitted mesh and the effect of changing to a more advanced flow model. The segregated Stokes simulations are performed using COMSOL® (v. 6.0).

It is noted that the process for generating body fitted meshes from TO designs is far from automatic. One reason is that the STL from ParaView can include errors, such as overlapping elements, duplicated elements or bad connectivity between elements. And even if the STL is flawless, such errors might appear after the remeshing. Another issue is that the remeshing of the STL surfaces introduces an arbitrariness when it comes to the interpretation of the fluid–solid interface, by the fact that the original isosurface $\hat{\rho} = 0.5$ is overridden by the remeshing in HyperMesh. There are several options for doing this remesh, including `shrink wrap` and `surface deviation` in the `automesh` panel.

Figures 14 and 15 show the velocity and the temperature at the fluid symmetry plane for the three different models, while Figure 16 shows the temperature on the convective boundary.

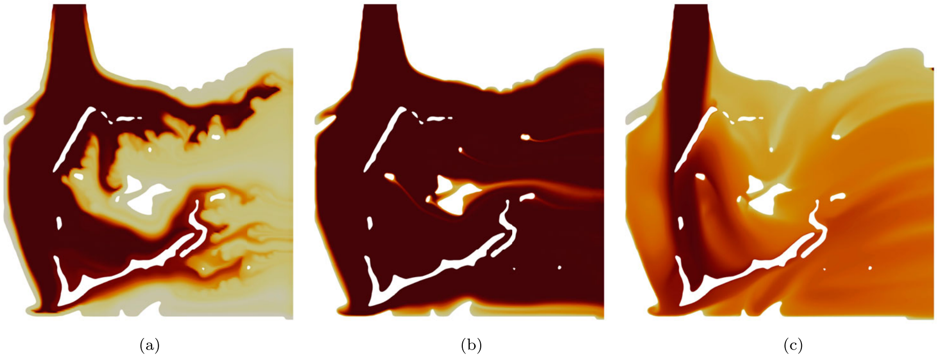
Figure 14 shows a fairly similar flow behaviour between the TO model and the segregated Stokes model. The RANS model on the other hand has a more diffusive and longer sustain of the momentum compared with the TO model, which excludes the convective terms. The exchange of momentum associated with convective acceleration also indicates a somewhat different trajectory of the flow compared with the Stokes models. This effect can also be seen in Figure 15(a), where a horizontal flow path closest to the inlet is clearly visible in the dark colour. A corresponding flow path is not visible in the RANS model in Figure 15(c). The consequence is a hotter surface in the RANS model at that height, as seen in Figure 16. These possible consequences of this effect may need to be further investigated in relation to the use of the Stokes equations in TO.

It is obvious from Figure 14 that the monolithic meshing causes large resistance to flow near the solid–fluid interface, owing to the Brinkman term. This is probably the cause of the very much higher flow velocities (maximum 52.0 versus 9.92 m/s) and mass flow (7.6 versus 1.0 g/s) in the segregated Stokes model. To avoid this effect, the isosurface can be defined much closer to the fluid bulk, *i.e.* making the flow channels narrower.



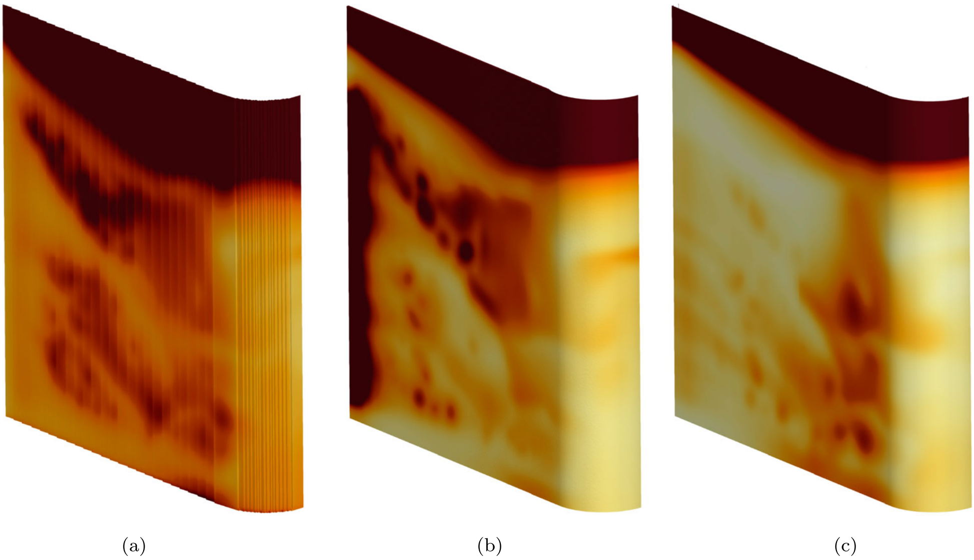
(a) (b) (c)

Figure 14. Relative flow velocities on the symmetry boundary for different models, ranging from zero (dark) to one (light). Maximal velocities are 9.92, 52.0 and 3.59 m/s, respectively: (a) Stokes TO model; (b) segregated Stokes model; (c) segregated RANS model.



(a) (b) (c)

Figure 15. Fluid temperatures on the symmetry boundary for different models, ranging from inlet temperature T_0 (dark) to ambient temperature T_∞ (light): (a) Stokes TO model; (b) segregated Stokes model; (c) segregated RANS model.



(a) (b) (c)

Figure 16. Solid temperatures on the convection boundary for different models, ranging from 1345K (dark) to ambient temperature $T\infty$ (light): (a) Stokes TO model; (b) segregated Stokes model; (c) segregated RANS model.

As indicated by Figure 15, the segregated Stokes model does not achieve mixing of cooling, *i.e.* on the symmetry plane the fluid temperature remains fairly similar to the inlet temperature. The RANS model, in contrast to the Stokes models, resembles a more diffusive mixing of the cooling when exposed to a hot solid surface, primarily associated with the role of convective acceleration and momentum exchange from the solid wall to the symmetry plane, see Figure 15(c).

In Figure 16, it can be seen that the TO model is cooler on the surface than the segregated Stokes model. This could be due to an overestimation of the cooling effect. This hypothesis is supported by the fact that the segregated Stokes model has a higher mass flow and higher flow speed closer to the fluid–solid interface, which most likely increases the heat transfer from the solid to the fluid.

6. Concluding remarks

A method for TO using complex design domains based on a voxelization technique with a novel boundary detection algorithm has been proposed. The use of voxelization circumvents complex and time-consuming generation of body-fitted meshes, and can give perfectly shaped elements, something that is appealing *e.g.* when using iterative solvers. This facilitates the possibilities for analysing complicated geometries with high resolution in a parallel computational environment. The method is demonstrated on an industrially relevant test case, with a geometry resembling a guide vane, and shows that the voxelization approach is a viable option when dealing with non-trivial design domains.

Furthermore, a demonstration has been made for a working HPC implementation of a large 3D example of a coupled flow–heat problem with a conjugate heat transfer approach, using PETSc (Smith 2011). The bottleneck is to solve the flow problem, and for that reason a comparison is included between the use of a penalty and a mixed FE approach that shows the penalty approach to be competitive if less resources are available and the problem size is not too large.

Finally, a comparison between the optimized designs from the TO model and two models using segregated meshes is provided, as an illustration on the methodology of performing a post-analysis validation of TO results. The key differences are believed to be due to the penalization of flow in intermediate design density regions, and the non-existing fluid mixing when using laminar flow models. The consequence is that the cooling effect is lower in the TO model. A possible remedy is to decrease the width of the cooling channels when creating the segregated mesh, which would lower the mass flow given a certain pressure drop.

Note

[1] Portable, Extensible Toolkit for Scientific Computation, https://petsc.org/.

Acknowledgments

The authors thank Prof. Anders Klarbring for his ideas and valuable input during the course of this work. Gratitude is extended to the SNIC support, and especially to application expert Dr Frank Bramkamp, for assistance concerning technical and implementational aspects.

Data availability statement

The STL files defining the 3D example geometry and the boundary conditions are provided as supplemental data. Additional data will be made available by the corresponding author upon reasonable request.

Disclosure statement

The authors declare that no conflict of interest regarding financial or personal relationships exists that could have influenced this work.

Funding

The computations were made possible by the Swedish National Infrastructure for Computing (SNIC) at the National Supercomputer Centre (NSC), partially funded by Linköping University, and by the Swedish Research Council [grant agreement no. 2018-05973]; the last named author also acknowledges financial support from Centrum för Industriell Informationsteknologi (CENIIT) at Linköping University [project ID no. 21.09], and the Swedish Research Council [grant agreement no. 2019-04615].

ORCID

Jonas Lundgren http://orcid.org/0000-0002-7759-2767
Jan-Erik Lundgren http://orcid.org/0000-0001-8641-6063
Hossein Nadali Najafabadi http://orcid.org/0000-0002-6693-2308
Carl-Johan Thore http://orcid.org/0000-0002-9180-2388

References

Aage, N., E. Andreassen, and B. S. Lazarov. 2015. “Topology Optimization Using PETSc: An Easy-to-Use, Fully Parallel, Open Source Topology Optimization Framework.” *Structural and Multidisciplinary Optimization* 51: 565–572. doi:10.1007/s00158-014-1157-0.

Aage, N., T. Poulsen, A. Gersborg-Hansen, and O. Sigmund. 2008. “Topology Optimization of Large Scale Stokes Flow Problems.” *Structural and Multidisciplinary Optimization* 35: 175–180. doi:10.1007/s00158-007-0128-0.

Alexandersen, J., and C. S. Andreasen. 2020. “A Review of Topology Optimisation for Fluid-Based Problems.” *Fluids* 5 (1): 29. doi:10.3390/fluids5010029.

Amestoy, P. R., A. Buttari, J.-Y. L’Excellent, and T. Mary. 2019. “Performance and Scalability of the Block Low-Rank Multifrontal Factorization on Multicore Architectures.” *ACM Transactions on Mathematical Software* 45 (1): 1–26. doi:10.1145/3242094.

Amestoy, P. R., I. S. Duff, J.-Y. L’Excellent, and J. Koster. 2001. “A Fully Asynchronous Multifrontal Solver Using Distributed Dynamic Scheduling.” *SIAM Journal on Matrix Analysis and Applications* 23 (1): 15–41. doi:10.1137/S0895479899358194.

Amir, O., N. Aage, and B. S. Lazarov. 2014. “On Multigrid-CG for Efficient Topology Optimization.” *Structural and Multidisciplinary Optimization* 49 (5): 815–829. doi:10.1007/s00158-013-1015-5.

Amir, O., M. Stolpe, and O. Sigmund. 2010. “Efficient Use of Iterative Solvers in Nested Topology Optimization.” *Structural and Multidisciplinary Optimization* 42 (1): 55–72. doi:10.1007/s00158-009-0463-4.

Andersson, O., A. Graichen, H. Brodin, and V. Navrotsky. 2017. “Developing Additive Manufacturing Technology for Burner Repair.” *Journal of Engineering for Gas Turbines and Power* 139 (3): Article ID 031506. doi:10.1115/1.4034235.

ANSYS, Inc. 2022. *ANSYS Fluent Theory Guide*. Release 2022 R2.

Badia, S., P. A. Martorell, and F. Verdugo. 2022. “Geometrical Discretisations for Unfitted Finite Elements on Explicit Boundary Representations.” *Journal of Computational Physics* 460: Article ID 111162. doi:10.1016/j.jcp.2022.111162.

Behrou, R., R. Ranjan, and J. K. Guest. 2019. “Adaptive Topology Optimization for Incompressible Laminar Flow Problems with Mass Flow Constraints.” *Computer Methods in Applied Mechanics and Engineering* 346: 612–641. doi:10.1016/j.cma.2018.11.037.

Bendsøe, M. P., and O. Sigmund. 2004. *Topology Optimization*. 2nd ed. Berlin: Springer.

Borrvall, T., and J. Petersson. 2003. “Topology Optimization of Fluids in Stokes Flow.” *International Journal for Numerical Methods in Fluids* 41 (1): 77–107. doi:10.1002/(ISSN)1097-0363.

Bourdin, B. 2001. “Filters in Topology Optimization.” *International Journal for Numerical Methods in Engineering* 50 (9): 2143–2158. doi:10.1002/(ISSN)1097-0207.

Bruns, T. E. 2007. “Topology Optimization of Convection-Dominated, Steady-State Heat Transfer Problems.” *International Journal of Heat and Mass Transfer* 50 (15): 2859–2873. doi:10.1016/j.ijheatmasstransfer.2007.01.039.

Burman, E., S. Claus, P. Hansbo, M. G. Larson, and A. Massing. 2015. “CutFEM: Discretizing Geometry and Partial Differential Equations.” *International Journal for Numerical Methods in Engineering* 104 (7): 472–501. doi:10.1002/nme.v104.7.

Dede, E. M. 2009. “Multiphysics Topology Optimization of Heat Transfer and Fluid Flow Systems.” In *Proceedings of the 2009 COMSOL Users Conference*. Boston, MA: COMSOL.

Dede, E. M., S. N. Joshi, and F. Zhou. 2015. “Topology Optimization, Additive Layer Manufacturing, and Experimental Testing of An Air-Cooled Heat Sink.” *Journal of Mechanical Design* 137 (11): Article ID 111403. doi:10.1115/1.4030989.

Dilgen, S. B., C. B. Dilgen, D. R. Fuhrman, O. Sigmund, and B. S. Lazarov. 2018. “Density Based Topology Optimization of Turbulent Flow Heat Transfer Systems.” *Structural and Multidisciplinary Optimization* 57 (5): 1905–1918. doi:10.1007/s00158-018-1967-6.

Fawaz, A., Y. Hua, S. Le Corre, Y. Fan, and L. Luo. 2022. “Topology Optimization of Heat Exchangers: A Review.” *Energy* 252: Article ID 124053. doi:10.1016/j.energy.2022.124053.

Gersborg-Hansen, A., M. Berggren, and B. Dammann. 2006. “Topology Optimization of Mass Distribution Problems in Stokes Flow.” In *IUTAM Symposium on Topological Design Optimization of Structures, Machines and Materials*, Vol. 137 of *Solid Mechanics and Its Applications*, 365–374. Dordrecht, The Netherlands: Springer. doi:10.1007/1-4020-4752-5_36.

Gibson, I., D. Rosen, B. Stucker, and M. Khorasani. 2021. *Additive Manufacturing Technologies*. 3rd ed.Cham, Switzerland: Springer. doi:10.1007/978-3-030-56127-7.

Glassman, I., R. A. Yetter, and N. G. Glumac. 2015. “Chapter 1—Chemical Thermodynamics and Flame Temperatures.” In *Combustion*, 5th ed., 1–40. Boston, MA: Academic Press.

Haertel, J. H. K., K. Engelbrecht, B. S. Lazarov, and O. Sigmund. 2018. “Topology Optimization of a Pseudo 3D Thermofluid Heat Sink Model.” *International Journal of Heat and Mass Transfer* 121: 1073–1088. doi:10.1016/j.ijheatmasstransfer.2018.01.078.

Haertel, J. H. K., and G. F. Nellis. 2017. “A Fully Developed Flow Thermofluid Model for Topology Optimization of 3D-Printed Air-Cooled Heat Exchangers.” *Applied Thermal Engineering* 119: 10–24. doi:10.1016/j.applthermaleng.2017.03.030.

Holka, Q., E. Toubiana, J. Cortial, B. Ghannam, and M. Nemer. 2022. “Density-Based Topology Optimization of a Surface Cooler in Turbulent Flow Using a Continuous Adjoint Turbulence Model.” *Structural and Multidisciplinary Optimization* 65: Article Number 223. doi:10.1007/s00158-022-03311-z.

Holman, J. P. 2010. *Heat Transfer*. 10th ed. Boston, MA: McGraw-Hill.

Hughes, T. J. R. 2000. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Mineola, NY: Dover Publications.

Hughes, T. J. R., and L. P. Franca. 1987. “A New Finite Element Formulation for Computational Fluid Dynamics. Part VII: The Stokes Problem with Various Well-Posed Boundary Conditions: Symmetric Formulations that Converge for All Velocity/Pressure Spaces.” *Computer Methods in Applied Mechanics and Engineering* 65 (1): 85–96. doi:10.1016/0045-7825(87)90184-8.

Johnson, C., and J. Pitkäranta. 1982. “Analysis of Some Mixed Finite Element Methods Related to Reduced Integration.” *Mathematics of Computation* 38 (158): 375–400. doi:10.1090/mcom/1982-38-158.

Kambampati, S., and H. A. Kim. 2020. “Level Set Topology Optimization of Cooling Channels Using the Darcy Flow Model.” *Structural and Multidisciplinary Optimization* 61: 1345–1361. doi:10.1007/s00158-019-02482-6.

Koga, A. A., E. C. C. Lopes, H. F. Villa Nova, C. R. de Lima, and E. C. N. Silva. 2013. “Development of Heat Sink Device by Using Topology Optimization.” *International Journal of Heat and Mass Transfer*64: 759–772. doi:10.1016/j.ijheatmasstransfer.2013.05.007.

Kontoleontos, E. A., E. M. Papoutsis-Kiachagias, A. S. Zymaris, D. I. Papadimitriou, and K. C. Giannakoglou. 2013. “Adjoint-Based Constrained Topology Optimization for Viscous Flows, Including Heat Transfer.” *Engineering Optimization* 45 (8): 941–961. doi:10.1080/0305215X.2012.717074.

Lazarov, B. S., O. Sigmund, K. E. Meyer, and J. Alexandersen. 2018. “Experimental Validation of Additively Manufactured Optimized Shapes for Passive Cooling.” *Applied Energy* 226: 330–339. doi:10.1016/j.apenergy.2018.05.106.

Lei, T., J. Alexandersen, B. S. Lazarov, F. Wang, J. H. K. Haertel, S. De Angelis, S. Sanna, O. Sigmund, and K. Engelbrecht. 2018. “Investment Casting and Experimental Testing of Heat Sinks Designed by Topology Optimization.” *International Journal of Heat and Mass Transfer* 127: 396–412. doi:10.1016/j.ijheatmasstransfer.2018.07.060.

Li, H., X. Ding, F. Meng, D. Jing, and M. Xiong. 2019. “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: Article ID 118638. doi:10.1016/j.ijheatmasstransfer.2019.118638.

Limkilde, A., A. Evgrafov, and J. Gravesen. 2018. “On Reducing Computational Effort in Topology Optimization: We Can Go At Least This Far!” *Structural and Multidisciplinary Optimization* 58: 2481–2492. doi:10.1007/s00158-018-2121-1.

Lundgren, J., A. Klarbring, J.-E. Lundgren, and C.-J. Thore. 2019. “Topology Optimization of Periodic 3D Heat Transfer Problems with 2D Design.” *Structural and Multidisciplinary Optimization* 60 (6): 2295–2303. doi:10.1007/s00158-019-02319-2.

Marck, G., M. Nemer, and J.-L. Harion. 2013. “Topology Optimization of Heat and Mass Transfer Problems: Laminar Flow.” *Numerical Heat Transfer, Part B: Fundamentals* 63 (6): 508–539. doi:10.1080/10407790.2013.772001.

McConnell, C., and G. Pingen. 2012. “Multi-Layer, Pseudo 3D Thermal Topology Optimization of Heat Sinks.” In *Proceedings of the ASME 2012 International Mechanical Engineering Congress and Exposition*, Vol. 7: Fluids and Heat Transfer, Parts A, B, C and D, 2381–2392. doi:10.1115/IMECE2012-93093.

Menter, F. R. 1994. “Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications.” *AIAA Journal* 32 (8): 1598–1605. doi:10.2514/3.12149.

Mumps Technologies. 2022. *Users’ Guide v. 5.5.1*. https://graal.ens-lyon.fr/MUMPS/doc/userguide_5.5.1.pdf. Accessed: 2023-03-10.

NSC. 2023a. “Sigma HPC Cluster Web Page.” https://nsc.liu.se/systems/sigma/. Accessed: 2023-03-10.

NSC. 2023b. “Tetralith HPC Cluster Web Page.” https://nsc.liu.se/systems/tetralith/. Accessed: 2023-03-10.

Öberg, S., M. Odenberger, and F. Johnsson. 2022. “Exploring the Competitiveness of Hydrogen-Fueled Gas Turbines in Future Energy Systems.” *International Journal of Hydrogen Energy* 47 (1): 624–644. doi:10.1016/j.ijhydene.2021.10.035.

Pietropaoli, M., R. Ahlfeld, F. Montomoli, A. Ciani, and M. D’Ercole. 2017. “Design for Additive Manufacturing: Internal Channel Optimization.” *Journal of Engineering for Gas Turbines and Power*139 (10): Article ID 102101. doi:10.1115/1.4036358.

Pietropaoli, M., A. Gaymann, and F. Montomoli. 2020. “Three-Dimensional Fluid Topology Optimization and Validation of a Heat Exchanger with Turbulent Flow.” In *Proceedings of the ASME Turbo Expo 2020*, Vol. 7A: Heat Transfer. New York: ASME (American Society of Mechanical Engineers). doi:10.1115/GT2020-14479.

Pietropaoli, M., F. Montomoli, and A. Gaymann. 2019. “Three-Dimensional Fluid Topology Optimization for Heat Transfer.” *Structural and Multidisciplinary Optimization* 59: 801–812. doi:10.1007/s00158-018-2102-4.

Qian, X., and E. M. Dede. 2016. “Topology Optimization of a Coupled Thermal-Fluid System Under a Tangential Thermal Gradient Constraint.” *Structural and Multidisciplinary Optimization* 54: 531–551. doi:10.1007/s00158-016-1421-6.

Qian, S., W. Wang, C. Ge, S. Lou, E. Miao, and B. Tang. 2018. “Topology Optimization of Fluid Flow Channel in Cold Plate for Active Phased Array Antenna.” *Structural and Multidisciplinary Optimization* 57 (6): 2223–2232. doi:10.1007/s00158-017-1852-8.

Rogié, B., and C. S. Andreasen. 2023. “Design Complexity Tradeoffs in Topology Optimization of Forced Convection Laminar Flow Heat Sinks.” *Structural and Multidisciplinary Optimization* 66 (1): Article Number 6. doi:10.1007/s00158-022-03449-w.

Sato, Y., K. Yaji, K. Izui, T. Yamada, and S. Nishiwaki. 2018. “An Optimum Design Method for a Thermal–Fluid Device Incorporating Multiobjective Topology Optimization with An Adaptive Weighting Scheme.” *Journal of Mechanical Design* 140 (3): Article ID 031402. doi:10.1115/1.4038209.

Sigmund, O., and J. Petersson. 1998. “Numerical Instabilities in Topology Optimization: A Survey on Procedures Dealing with Checkerboards, Mesh-Dependencies and Local Minima.” *Structural Optimization* 16 (1): 68–75. doi:10.1007/BF01214002.

Smith, B. 2011. “PETSc (Portable, Extensible Toolkit for Scientific Computation).” In *Encyclopedia of Parallel Computing*, edited by D. Padua, 1530–1539. Boston, MA: Springer US. doi:10.1007/978-0-387-09766-4_87.

Sun, S., P. Liebersbach, and X. Qian. 2020. “3D Topology Optimization of Heat Sinks for Liquid Cooling.” *Applied Thermal Engineering* 178: Article ID 115540. doi:10.1016/j.applthermaleng.2020.115540.

Svanberg, K. 1987. “The Method of Moving Asymptotes—A New Method for Structural Optimization.” *International Journal for Numerical Methods in Engineering* 24 (2): 359–373. doi:10.1002/(ISSN)1097-0207.

Thore, C.-J., J. Lundgren, and J.-E. Lundgren. 2023. “A Mathematical Game for Topology Optimization of Cooling Systems.” *ZAMM—Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik* 103 (2): Article ID e202100086. doi:10.1002/zamm.202100086.

Thore, C.-J., J. Lundgren, J.-E. Lundgren, and A. Klarbring. 2022. “Topology Optimization for Minimum Temperature with Mass Flow and Stiffness Constraints.” *Computer Methods in Applied Mechanics and Engineering* 400: Article ID 115334. doi:10.1016/j.cma.2022.115334.

Yaji, K., T. Yamada, S. Kubo, K. Izui, and S. Nishiwaki. 2015. “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. doi:10.1016/j.ijheatmasstransfer.2014.11.005.

Yan, S., F. Wang, J. Hong, and O. Sigmund. 2019. “Topology Optimization of Microchannel Heat Sinks Using a Two-Layer Model.” *International Journal of Heat and Mass Transfer* 143: Article ID 118462. doi:10.1016/j.ijheatmasstransfer.2019.118462.

Yoon, G. H. 2010. “Topological Design of Heat Dissipating Structure with Forced Convective Heat Transfer.” *Journal of Mechanical Science and Technology* 24: 1225–1233. doi:10.1007/s12206-010-0328-1.

Zeng, S., B. Kanargi, and P. S. Lee. 2018. “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. doi:10.1016/j.ijheatmasstransfer.2018.01.039.

Zhang, Z.-D., O. Ibhadode, A. Bonakdar, and E. Toyserkani. 2021. “TopADD: A 2D/3D Integrated Topology Optimization Parallel-Computing Framework for Arbitrary Design Domains.” *Structural and Multidisciplinary Optimization* 64 (3): 1701–1723. doi:10.1007/s00158-021-02917-z.

Zhao, X., M. Zhou, O. Sigmund, and C. S. Andreasen. 2018. “A ‘Poor Man’s Approach’ to Topology Optimization of Cooling Channels Based on a Darcy Flow Model.” *International Journal of Heat and Mass Transfer* 116: 1108–1123. doi:10.1016/j.ijheatmasstransfer.2017.09.090.

Zhou, M., J. Alexandersen, O. Sigmund, and C. B. W. Pedersen. 2016. “Industrial Application of Topology Optimization for Combined Conductive and Convective Heat Transfer Problems.” *Structural and Multidisciplinary Optimization* 54: 1045–1060. doi:10.1007/s00158-016-1433-2.

Appendix. Sensitivities

The design problem is solved using gradient-based methods. Therefore, the sensitivities of the objective (7) and constraint (8) functions are derived using a discrete adjoint approach.

A.1 Objective and constraint function sensitivities

Consider the Lagrangian function

$$\tilde{\phi} = \phi_h + \boldsymbol{\lambda}_{\mathrm{F}}^{\mathsf{T}}\mathbf{r}_{\mathrm{F}} + \boldsymbol{\lambda}_{\mathrm{T}}^{\mathsf{T}}\mathbf{r}_{\mathrm{T}},$$

where ϕ_h is the objective function in $(\mathbb{P})_{\mathrm{D}}$, $\boldsymbol{\lambda}_i \in \mathbb{R}^{n_i \times 1}$ for $i = \{\mathrm{F}, \mathrm{T}\}$ are adjoint vectors, and $\mathbf{r}_i$ are the residuals

$$\mathbf{r}_{\mathrm{F}}(\hat{\boldsymbol{\rho}}, \mathbf{u}) = -\mathbf{K}_{\mathrm{F}}(\hat{\boldsymbol{\rho}})\mathbf{u} + \mathbf{f}_{\mathrm{F}},$$

$$\mathbf{r}_{\mathrm{T}}(\hat{\boldsymbol{\rho}}, \mathbf{t}, \mathbf{u}) = -\mathbf{K}_{\mathrm{T}}(\hat{\boldsymbol{\rho}}, \mathbf{u})\mathbf{t} + \mathbf{f}_{\mathrm{T}}.$$

The Lagrangian $\tilde{\phi}$ will be equal to the original objective function ϕ as long as the residuals are zero, *i.e.* the state problems are solved exactly. Differentiating the Lagrangian by the regularized FE design variable $\hat{\rho}_e$ gives the sensitivity

$$\nabla_{\hat{\rho}_e}\tilde{\phi} = \frac{\partial \phi_h}{\partial \hat{\rho}_e} + \boldsymbol{\lambda}_{\mathrm{F}}^{\mathsf{T}}\left(-\frac{\partial \mathbf{K}_{\mathrm{F}}}{\partial \hat{\rho}_e}\mathbf{u}\right) + \boldsymbol{\lambda}_{\mathrm{T}}^{\mathsf{T}}\left(-\frac{\partial \mathbf{K}_{\mathrm{T}}}{\partial \hat{\rho}_e}\mathbf{t}\right), \tag{A1}$$

where the first term disappears since there is no explicit design dependency in the objective function (7), and the adjoint vectors are obtained from the adjoint problems

$$\mathbf{K}_{\mathrm{T}}^{\mathsf{T}}\boldsymbol{\lambda}_{\mathrm{T}} = \frac{\partial \phi_h}{\partial \mathbf{t}}, \tag{A2}$$

$$\mathbf{K}_{\mathrm{F}}\boldsymbol{\lambda}_{\mathrm{F}} = -\mathbf{t}^{\mathsf{T}}\left(\frac{\partial \mathbf{K}_{\mathrm{T}}^{\mathsf{T}}}{\partial \mathbf{u}}\right)\boldsymbol{\lambda}_{\mathrm{T}}. \tag{A3}$$

Note the transpose in both instances of $\mathbf{K}_{\mathrm{T}}$ in the adjoint problems. The derivatives of the stiffness matrices in (A1) and the right-hand sides in the adjoint problems are provided in Sections A.2 and A.3, respectively.

The sensitivity of the mass flow constraint (8) becomes

$$\nabla_{\hat{\rho}_e}\left(g_h(\mathbf{u})\right) = \frac{\partial g_h}{\partial \mathbf{u}}\frac{\partial \mathbf{u}}{\partial \hat{\rho}_e}.$$

By using the fact that

$$\frac{\partial \mathbf{K}_{\mathrm{F}}}{\partial \hat{\rho}_e}\mathbf{u} + \mathbf{K}_{\mathrm{F}}\frac{\partial \mathbf{u}}{\partial \hat{\rho}_e} = \mathbf{0},$$

the sensitivity becomes

$$\nabla_{\hat{\rho}}\left(g_h(\mathbf{u})\right) = -\boldsymbol{\lambda}_g^{\mathsf{T}}\frac{\partial \mathbf{K}_{\mathrm{F}}}{\partial \hat{\rho}_e}\mathbf{u},$$

where the adjoint vector $\boldsymbol{\lambda}_g$ is the solution to

$$\mathbf{K}_{\mathrm{F}}\boldsymbol{\lambda}_g = \frac{\partial g_h}{\partial \mathbf{u}}. \tag{A4}$$

This adjoint problem shows it to be unnecessary to solve, since it resembles the flow state problem to such a degree that the solution vector in (A4) becomes a pure scaling of the solution vector $\mathbf{u}$ in (3). The details are provided in Section A.3.3.

A.2 Element stiffness matrix sensitivities

In (A1), the derivatives of the fluid and heat stiffness matrices with respect to $\hat{\rho}_e$ are required.

The derivative of the fluid stiffness matrix with respect to $\hat{\rho}_e$ is

$$\frac{\partial \mathbf{K}_{\mathrm{F}}}{\partial \hat{\rho}_e} = \frac{\partial \alpha(\hat{\boldsymbol{\rho}})}{\partial \hat{\rho}_e}\int_{\Omega_e}\mathbf{N}_{\mathrm{F}}^{\mathsf{T}}\mathbf{N}_{\mathrm{F}}\,\mathrm{d}V,$$

where $\alpha(\hat{\boldsymbol{\rho}})$ is specified in (1), and thus

$$\frac{\partial \alpha(\hat{\boldsymbol{\rho}})}{\partial \hat{\rho}_e} = \overline{\alpha}\frac{1+q}{(1+q(1-\hat{\rho}_e))^2}.$$

The derivative of the heat stiffness matrix with respect to $\hat{\rho}_e$ is

$$\frac{\partial \mathbf{K}_{\mathrm{T}}}{\partial \hat{\rho}_e} = \frac{\partial k(\hat{\boldsymbol{\rho}})}{\partial \hat{\rho}_e} \int_{\Omega_e} \mathbf{B}_{\mathrm{T}}^{\mathsf{T}} \mathbf{B}_{\mathrm{T}} \, \mathrm{d}V,$$

where

$$\frac{\partial k(\hat{\boldsymbol{\rho}})}{\partial \hat{\rho}_e} = k_s - k_f.$$

A.3 Adjoint right-hand sides

A.3.1 Heat adjoint

In (A2), the derivative of the objective function (7) with respect to the nodal temperatures **t** is required. It reads

$$\frac{\partial \phi_h}{\partial t_j} = \left(\sum_{i=1}^{n_{\mathrm{n}}} [\mathbf{f}_{\mathrm{T}}]_i [\mathbf{t}]_i^P \right)^{1/P-1} [\mathbf{f}_{\mathrm{T}}]_j [\mathbf{t}]_j^{P-1}, \quad \text{for } j = 1, \dots, n_{\mathrm{n}}.$$

A.3.2 Fluid adjoint

In (A3), the derivative of the global heat stiffness matrix with respect to the nodal velocities **u** is required. The right-hand side of the fluid adjoint matrix problem is assembled at element level, and therefore the derivative of the element heat stiffness matrix with respect to the nodal velocity component u_e^k is given as

$$\frac{\partial \mathbf{K}_{\mathrm{T}}^e}{\partial u_e^k} = \frac{\partial \mathbf{K}_c^e}{\partial u_e^k} + \frac{\partial \mathbf{K}_{SU}^e}{\partial u_e^k},$$

where

$$\frac{\partial [\mathbf{K}_c^e]_{ij}}{\partial u_e^k} = \int_{\Omega_e} c[\mathbf{N}_{\mathrm{T}}]_i [\mathbf{N}_{\mathrm{F}}]_{mk} [\mathbf{B}_{\mathrm{T}}]_{mj} \, \mathrm{d}V,$$

which is non-symmetric, and

$$\frac{\partial [\mathbf{K}_{SU}^e]_{ij}}{\partial u_e^k} = \int_{\Omega_e} \frac{\partial \tau}{\partial u_e^k} [\mathbf{B}_{\mathrm{T}}^{\mathsf{T}} \mathbf{u}_e (\mathbf{u}_e)^{\mathsf{T}} \mathbf{B}_{\mathrm{T}}]_{ij} \, \mathrm{d}V + \int_{\Omega_e} c^2 \tau \frac{\partial [\mathbf{B}_{\mathrm{T}}^{\mathsf{T}} \mathbf{u}_e (\mathbf{u}_e)^{\mathsf{T}} \mathbf{B}_{\mathrm{T}}]_{ij}}{\partial u_e^k} \, \mathrm{d}V, \tag{A5}$$

which is symmetric. The element nodal velocity vector $\mathbf{u}_e$ contains 24 components u_e^k. The derivative of the stabilization parameter τ in (A5) is

$$\frac{\partial \tau}{\partial u_e^k} = \frac{h}{2c} \left(\frac{\partial \left(\|\boldsymbol{u}_h\|^{-2} \right)}{\partial u_e^k} \mathbf{X}^{\mathsf{T}} \boldsymbol{u}_h + \frac{1}{\|\boldsymbol{u}_h\|^2} \left(\frac{\partial \mathbf{X}^{\mathsf{T}}}{\partial u_e^k} \boldsymbol{u}_h + \mathbf{X}^{\mathsf{T}} \frac{\partial \boldsymbol{u}_h}{\partial u_e^k} \right) \right), \tag{A6}$$

where

$$\frac{\partial \boldsymbol{u}_h}{\partial u_e^k} = \frac{\partial [\boldsymbol{u}_h]_i}{\partial u_e^k} = [\mathbf{N}_{\mathrm{F}}]_{ik}, \tag{A7}$$

and thus

$$\begin{aligned} \frac{\partial \left(\|\boldsymbol{u}_h\|^{-2} \right)}{\partial u_e^k} &= \frac{\partial \left(\|\boldsymbol{u}_h\|^{-2} \right)}{\partial \|\boldsymbol{u}_h\|} \frac{\partial \|\boldsymbol{u}_h\|}{\partial \boldsymbol{u}_h} \frac{\partial \boldsymbol{u}_h}{\partial u_e^k} \\ &= \frac{-2}{\|\boldsymbol{u}_h\|^3} \frac{\boldsymbol{u}_h^{\mathsf{T}}}{\|\boldsymbol{u}_h\|} \frac{\partial \boldsymbol{u}_h}{\partial u_e^k} = \sum_{i=1}^{3} \frac{-2[\boldsymbol{u}_h^{\mathsf{T}}]_i}{\|\boldsymbol{u}_h\|^4} [\mathbf{N}_{\mathrm{F}}]_{ik}. \end{aligned}$$

The derivative in the second term in (A6) is, for $i = 1,2,3$, given as

$$\frac{\partial [\mathbf{X}]_i}{\partial u_e^k} = \frac{\partial [\mathbf{X}]_i}{\partial Pe_i} \frac{\partial Pe_i}{\partial [\boldsymbol{u}_h]_i} \frac{\partial [\boldsymbol{u}_h]_i}{\partial u_e^k} = \left(1 - \coth^2(Pe_i) + \frac{1}{Pe_i^2} \right) \frac{ch_{\max}}{k_f} [\mathbf{N}_{\mathrm{F}}]_{ik}.$$

The final expression for the stabilization parameter derivative with respect to the element nodal velocities reads

$$\frac{\partial \tau}{\partial u_e^k} = \frac{h_{\max}}{2c\|\boldsymbol{u}_h\|^2} \sum_{i=1}^{3} \left(\frac{-2[\boldsymbol{u}_h^{\mathsf{T}}]_i}{\|\boldsymbol{u}_h\|^2} \mathbf{X}^{\mathsf{T}} \boldsymbol{u}_h + \frac{ch_{\max}[\boldsymbol{u}_h^{\mathsf{T}}]_i}{k_f} \left(1 - \coth^2(Pe_i) + \frac{1}{Pe_i^2} \right) + [\mathbf{X}^{\mathsf{T}}]_i \right) [\mathbf{N}_{\mathrm{F}}]_{ik}.$$

The derivative in the second term in (A5) can be evaluated by use of (A7), which gives

$$\frac{\partial [\mathbf{B}_{\mathrm{T}}^{\mathsf{T}} \boldsymbol{u}_h \boldsymbol{u}_h^{\mathsf{T}} \mathbf{B}_{\mathrm{T}}]_{ij}}{\partial u_e^k} = \mathbf{B}_{\mathrm{T}}^{\mathsf{T}} \left(\frac{\partial \boldsymbol{u}_h}{\partial u_e^k} \boldsymbol{u}_h^{\mathsf{T}} + \boldsymbol{u}_h \frac{\partial \boldsymbol{u}_h^{\mathsf{T}}}{\partial u_e^k} \right) \mathbf{B}_{\mathrm{T}}$$

$$= [\mathbf{B}_{\mathrm{T}}]_{mi}\Big([\mathbf{N}_{\mathrm{T}}]_{mk}[\boldsymbol{u}_h]_n + [\boldsymbol{u}_h]_m[\mathbf{N}_{\mathrm{F}}]_{nk}\Big)[\mathbf{B}_{\mathrm{T}}]_{nj}.$$

A.3.3 Mass flow constraint adjoint

The right-hand side of (A4) is

$$\frac{\partial g_h}{\partial \mathbf{u}} = \mathop{\mathsf{A}}_{e=1}^{n_e} \int_{\Gamma_e^{\mathrm{in}}} \rho_f \mathbf{N}_{\mathrm{F}}^{\mathsf{T}}(-\mathbf{n}_e)\,\mathrm{d}A,$$

where A is the assembly operator, n_e is the number of elements that have faces in Γ_h^{in}, and $\mathbf{n}_e$ is the local element outward normal.

The right-hand side of (A4) is a pure scaling of the load vector in (3), since both $\mathbf{t}_{\mathrm{F}}^e$ and $-\mathbf{n}_e$ are oriented in the same direction (into the domain, normal to the boundary). This means that the constraint adjoint vector $\boldsymbol{\lambda}_g$ will be a pure scaling of the state solution vector $\mathbf{u}$ in (3).

The scaling factor for the specific problem treated herein becomes

$$\frac{\rho_f \|\mathbf{n}_e\|}{\|\mathbf{t}_{\mathrm{F}}^e\|} = \frac{\rho_f}{|t_{\mathrm{in}}|},$$

where the constant input traction t_{in} is given in Table 3.