Engineering with Computers (2025) 41:4575–4586
https://doi.org/10.1007/s00366-025-02203-2

ORIGINAL ARTICLE

Topology optimisation of transient compressible flow

Diego Hayashi Alonso[1] · Patrick E. Farrell[2,3] · Julio Romano Meneghini[1] · Emílio Carlos Nelli Silva[4]

Received: 8 May 2025 / Accepted: 21 August 2025 / Published online: 11 October 2025

Abstract
We present the first formulation of topology optimisation for transient compressible flow. We discretise the equations with a discontinuous Galerkin method, due to its similarity to the popular finite volume method, and its capacity to model discontinuities in the flow. The discretised topology optimisation problem is solved through an integer linear programming algorithm. We present numerical examples that optimise the shape of a nozzle and a pipe joint in the presence of a time-varying inflow.

Keywords Fluid topology optimisation · Discontinuous Galerkin · Subsonic flow · Transient compressible flow

1 Introduction

Topology optimisation is a valuable approach to design, due to its potential to discover solutions with any material distribution inside the design domain. Initially devised for solids, its extension to fluid problems began with Borrvall and Petersson [1], and this initial work has been then gradually extended to more complex fluid flow physics, such as for the Navier–Stokes equations [2], non-Newtonian effects [3, 4], thermal-fluid interactions [5, 6], and turbulence effects [7]. Applications include, for example, mixers [8], arterial by-pass grafts [9], bladed rotors [10], fluid diodes [11], and rectifiers [12]. Different approaches for implementing the topology optimisation method have also been considered throughout the years, including pseudo-density [1], level-set [13] and topological derivative [14] approaches. In this work we adopt the pseudo-density approach [15].

When considering compressible flow physics, the fluid density becomes dependent on its thermodynamic state. Since the density appears in all equations, this means that all equations become thermodynamically coupled, resulting in higher computational costs stemming from nonlinearities, interdependency between the fluid properties, and wall effects [16]. The contribution of the compressible behavior is measured by the Mach number (Ma): essentially, the compressible behavior of the fluid becomes important starting from $\mathrm{Ma} \approx 0.3$ [17], below which the fluid may be considered as incompressible. In terms of compressible flows, they may be characterised differently depending on the Mach number, due to changes in the fluid behavior [18]: subsonic ($0.3 \lesssim \mathrm{Ma} \lesssim 0.8$), transonic ($0.8 \lesssim \mathrm{Ma} \lesssim 1.2$, characterised by a transition in fluid flow behavior), sonic ($\mathrm{Ma} = 1$), supersonic ($1.2 \lesssim \mathrm{Ma} \lesssim 5$, characterized by the appearance of oblique shock waves) and hypersonic ($\mathrm{Ma} \gtrsim 5$, when the heat transfer becomes intense enough to induce molecular dissociation and/or ionisation of the gas). The most common types of compressible flows in turbomachinery range

Patrick E. Farrell, Julio Romano Meneghini and Emílio Carlos Nelli Silva have contributed equally to this work.

✉ Diego Hayashi Alonso
diego.alonso@usp.br

Patrick E. Farrell
patrick.farrell@maths.ox.ac.uk

Julio Romano Meneghini
jmeneg@usp.br

Emílio Carlos Nelli Silva
ecnsilva@usp.br

[1] Department of Mechanical Engineering, Polytechnic School of the University of São Paulo, Av. Prof. Mello Moraes, 2231, São Paulo, SP 05508-030, Brazil

[2] Mathematical Institute, University of Oxford, Woodstock Road, Oxford, Oxfordshire OX2 6GG, UK

[3] Mathematical Institute, Charles University, Sokolovská 49/83, 18675 Prague, Czechia

[4] Department of Mechatronics and Mechanical Systems Engineering, Polytechnic School of the University of São Paulo, Av. Prof. Mello Moraes, 2231, São Paulo, SP 05508-030, Brazil

from subsonic to supersonic. The use of topology optimisation to design for compressible flow started with the laminar subsonic case with Sá et al. [16]. This original incursion was then considered in a finite differences-based sensitivity approach [19], in an integer programming-based geometry trimming approach [20] (i.e., using a integer programming algorithm and trimmed meshes for the simulations), and in turbulent flow [21]. However, these works are all based on steady state formulations.

Another important effect, when evaluating fluid flow dynamics, is transience. Transient interactions arise such as from changes in boundary values (e.g. inlet velocity changes), flow oscillations [22], and moving boundaries. In terms of topology optimisation, transient effects have only been considered for laminar incompressible flows [23, 24], although body forces [15] and heat transfer [25] effects have already been considered. These previous works only consider incompressible flows.

In this work, we employ the discontinuous Galerkin (DG) approach [26]. DG methods are finite element formulations that do not assume the continuity of function values between the elements of the mesh. They can also be said to be conceptually similar to the finite volume method [27], and are even thought of as a hybrid of the finite volume and finite element methods [28]. This similarity is also apparent when considering the conservation of variables during the iterations of the simulation, since both DG and finite volume methods enforce the local conservation of the variables, whereas the usual continuous finite element method enforces a global conservation. Some further characteristics of DG methods are that the boundary conditions are weakly imposed through numerical fluxes, which may improve fluid flow simulation results at lower mesh resolutions [28]; they facilitate the solution of problems featuring discontinuities, such as with compressible fluid flows with higher Mach numbers [29]; they have good parallel scaling [30] due to their compact stencil (which requires only interface neighbour information) [31]; they are able to handle unstructured and non-conforming grids [31]; and there are further potential advantages with respect to high-order accuracy on unstructured meshes, and local hp-refinement [28]. Up to now, in fluid flow topology optimisation, DG formulations have only been considered for incompressible Stokes flow [32].

In this work we develop a topology optimisation method for transient compressible flow, discretised with DG. The fluid is treated as an ideal gas. In terms of the topology optimisation setup, the design variable is pointwise, and the classic Borrvall–Petersson material model is set for the inverse permeability [1]. The numerical implementation of the developed formulation is performed in the finite element-based computing platform FEniCS [33], with the use of the `dolfin_dg` library [34] for the derivation of the specific DG terms, and the sensitivities being automatically obtained from the adjoint method (dolfin-adjoint library) [35, 36]. Both forward and adjoint models are solved with the sparse direct solver UMFPACK [37]. Finally, we employ as optimisation algorithm TOBS [38, 39], which is based on a sequential integer linear optimisation algorithm.

This paper is organised as follows: in Sect. 2, the continuous formulation is presented; in Sect. 3, the DG formulation is presented; in Sect. 4, the boundary value problem is defined; in Sect. 5, the topology optimisation formulation is detailed; in Sect. 6, the numerical implementation is described; in Sect. 7, numerical results are presented; and in Sect. 8, some conclusions are offered.

2 Continuous formulation

The fluid flow equations for transient compressible flow consist of the continuity, linear momentum, and energy equations. The complete set of equations for an ideal gas, including compressibility and transient effects, and neglecting external body forces, is:

$$\frac{\partial \rho}{\partial t} + \nabla\cdot(\rho \boldsymbol{v}) = 0, \tag{1}$$

$$\frac{\partial (\rho \boldsymbol{v})}{\partial t} + \nabla\cdot[(\rho \boldsymbol{v}) \otimes \boldsymbol{v}] = \nabla\cdot \boldsymbol{T} + \boldsymbol{f}_r(\alpha), \tag{2}$$

$$\frac{\partial (\rho e_{\text{total}})}{\partial t} + \nabla\cdot(\rho e_{\text{total}} \boldsymbol{v}) = \nabla\cdot[k_{th} \nabla T_{\text{K}}] + \nabla\cdot[\boldsymbol{T}\boldsymbol{v}], \tag{3}$$

where $\boldsymbol{v}$ is the velocity of the fluid flow, p is the (static) pressure of the fluid flow, T_{K} is the absolute temperature of the fluid, $\rho = \rho(p, T_{\text{K}}) = \frac{pM_{\text{molar}}}{R_U T_{\text{K}}}$ is the density of the fluid according to the ideal gas model (M_{molar} is the molar mass of the gas, and R_U is the universal gas constant (8.314 J/(K mol))), $e_{\text{total}} = u_{\text{int}} + \frac{\boldsymbol{v}\cdot\boldsymbol{v}}{2}$ is the total energy, $u_{\text{int}} = \frac{p}{(\gamma-1)\rho}$ is the internal energy, c_p is the specific heat at constant pressure, c_v is the specific heat at constant volume (for an ideal gas, $c_v = c_p - \frac{R_U}{M_{\text{molar}}}$), $\gamma = \frac{c_p}{c_v}$ is the ratio of the specific heats, k_{th} is the thermal conductivity, and are the inner products (one- and two-dimensional versions, respectively), $\otimes$ is the outer product, $\boldsymbol{f}_r(\alpha)$ is a resistance force considered to model the solid material for topology optimisation (Sect. 5).

The fluid stress tensor $\boldsymbol{T}$ is given by:

$$\boldsymbol{T} = 2\mu\boldsymbol{\epsilon} - p\boldsymbol{I} + \lambda(\nabla\cdot\boldsymbol{v})\boldsymbol{I}, \quad \boldsymbol{\epsilon} = \frac{1}{2}(\nabla\boldsymbol{v} + \nabla\boldsymbol{v}^T) \tag{4}$$

where μ is the dynamic viscosity of the fluid, and $\lambda = -\frac{2}{3}\mu$ is the dilatational viscosity of the fluid, and $\boldsymbol{I}$ is the identity matrix.

3 Discontinuous Galerkin formulation

The DG formulation is derived from the weak form of (1)–(3). Due to the discontinuous nature of the DG variables, the boundary integral terms, which come from Gauss' divergence theorem and integration by parts is split into internal and external facet integrations. The internal facet integration arises from the difference between the fluxes from one element to the other (i.e., the *jump* between them), whose contribution is equally split for both neighbouring elements. Furthermore, some additional terms are included in the equations, in order to maintain the continuity of the fluxes around the facets and avoid instabilities [26, 40], the continuity of the solution, and the stability of the method. As a first step to derive the DG formulation, (1)–(3) are rewritten by separating the convective, dissipative and external terms [34]. Thus, by stacking the resulting equations [34],

$$\nabla\cdot\boldsymbol{F}_c(\boldsymbol{u}) = \nabla\cdot\boldsymbol{F}_d(\boldsymbol{u},\nabla\boldsymbol{u}) + \boldsymbol{F}_e(\boldsymbol{u}), \tag{5}$$

where $\boldsymbol{F}_c(\boldsymbol{u})$ is the convective part, $\boldsymbol{F}_d(\boldsymbol{u},\nabla\boldsymbol{u})$ is the dissipative part, and $\boldsymbol{F}_e(\boldsymbol{u})$ is the external part. These parts are given as:

$$\boldsymbol{F}_c(\boldsymbol{u}) = \begin{bmatrix} \rho\boldsymbol{v} \\ p\boldsymbol{I} + (\rho\boldsymbol{v})\otimes\boldsymbol{v} \\ \rho e_{\text{total}}\boldsymbol{v} + (p\boldsymbol{I})\boldsymbol{v} \end{bmatrix}, \tag{6}$$

$$\boldsymbol{F}_d(\boldsymbol{u},\nabla\boldsymbol{u}) = \begin{bmatrix} \boldsymbol{0} \\ \mu(\nabla\boldsymbol{v} + \nabla\boldsymbol{v}^T) + \lambda\ \text{tr}(\nabla\boldsymbol{v})\boldsymbol{I} \\ k_{th}\nabla T_{\text{K}} + [\mu(\nabla\boldsymbol{v} + \nabla\boldsymbol{v}^T) + \lambda\ \text{tr}(\nabla\boldsymbol{v})\boldsymbol{I}]\boldsymbol{v} \end{bmatrix}, \tag{7}$$

$$\boldsymbol{F}_e(\boldsymbol{u}) = \begin{bmatrix} -\frac{\partial\rho}{\partial t} \\ -\frac{\partial(\rho\boldsymbol{v})}{\partial t} + \boldsymbol{f}_r(\alpha) \\ -\frac{\partial(\rho e_{\text{total}})}{\partial t} \end{bmatrix}, \tag{8}$$

where tr is the matrix trace ($\text{tr}(\nabla\boldsymbol{v}) = \nabla\cdot\boldsymbol{v}$), $\boldsymbol{u} = \begin{bmatrix} p \\ \boldsymbol{v} \\ T_{\text{K}} \end{bmatrix}$ is the state vector, and $\nabla\boldsymbol{u} = \begin{bmatrix} \nabla p \\ \nabla\boldsymbol{v} \\ \nabla T_{\text{K}} \end{bmatrix}$ is its gradient. Also, the divergence operator in eq. (5) is defined, when considering a generic stacked term $\boldsymbol{F} = \begin{bmatrix} \boldsymbol{F_1} \\ \boldsymbol{F_2} \\ \boldsymbol{F_3} \end{bmatrix}$, as $\nabla\cdot\boldsymbol{F} = \begin{bmatrix} \nabla\cdot\boldsymbol{F_1} \\ \nabla\cdot\boldsymbol{F_2} \\ \nabla\cdot\boldsymbol{F_3} \end{bmatrix}$.

In terms of (5), the DG formulation considered in this work for transient compressible flow is given as:

$$\begin{aligned}
&-\int_\Omega \boldsymbol{F}_c(\boldsymbol{u}){:}(\nabla\boldsymbol{w})\,d\Omega + \int_\Omega \boldsymbol{F}_d(\boldsymbol{u},\nabla\boldsymbol{u}){:}(\nabla\boldsymbol{w})\,d\Omega \\
&\quad - \int_\Omega \boldsymbol{F}_e(\boldsymbol{u})\cdot\boldsymbol{w}\,d\Omega \\
&+\int_{\Gamma_{\text{int}}} [\langle\boldsymbol{F}_c(\boldsymbol{u})\rangle + \alpha_{\text{LF}}(\boldsymbol{u})[\![\boldsymbol{u}]\!]_{\otimes,\boldsymbol{n}}]{:}[\![\boldsymbol{w}]\!]_{\otimes,\boldsymbol{n}}\,d\Gamma_{\text{int}} \\
&-\int_{\Gamma_{\text{int}}} [\langle\boldsymbol{F}_d(\boldsymbol{u},\nabla\boldsymbol{u})\rangle - \boldsymbol{\alpha}_{\text{SIPG}}(\boldsymbol{u})[\![\boldsymbol{u}]\!]_{\otimes,\boldsymbol{n}}]{:}[\![\boldsymbol{w}]\!]_{\otimes,\boldsymbol{n}}\,d\Gamma_{\text{int}} \\
&\quad - \int_{\Gamma_{\text{int}}} \langle\boldsymbol{F}_d(\boldsymbol{u},\nabla\boldsymbol{w})\rangle{:}[\![\boldsymbol{u}]\!]_{\otimes,\boldsymbol{n}}\,d\Gamma_{\text{int}} \\
&+\sum_i \int_{\Gamma_{\text{ext},i}} [\boldsymbol{F}_c(\boldsymbol{u}|_{b,i}) + \alpha_{\text{LF},b}(\boldsymbol{u},\boldsymbol{u}|_{b,i})(\boldsymbol{u}-\boldsymbol{u}|_{b,i})]: \\
&\quad (\boldsymbol{w}\otimes\boldsymbol{n})\,d\Gamma_{\text{ext},i} \\
&-\sum_i \int_{\Gamma_{\text{ext},i}} \{\boldsymbol{F}_d(\boldsymbol{u}|_{b,i},\nabla\boldsymbol{u})|_{N,i} \\
&\quad - \boldsymbol{\alpha}_{\text{SIPG},b}(\boldsymbol{u}|_{b,i})[(\boldsymbol{u}-\boldsymbol{u}|_{b,i})\otimes\boldsymbol{n}]\}{:}(\boldsymbol{w}\otimes\boldsymbol{n})\,d\Gamma_{\text{ext},i} \\
&-\sum_i \int_{\Gamma_{\text{ext},i}} \boldsymbol{F}_d(\boldsymbol{u}|_{b,i},\nabla\boldsymbol{w})|_{N,i}: \\
&\quad [(\boldsymbol{u}-\boldsymbol{u}|_{b,i})\otimes\boldsymbol{n}]\,d\Gamma_{\text{ext},i} = 0,
\end{aligned} \tag{9}$$

where $\sum_i$ represents the sum over all boundaries of the domain, Ω is the whole computational domain, Γ_{int} are the internal boundaries of each element, $\Gamma_{\text{ext},i}$ is the external boundary of the domain labeled i, $\boldsymbol{n}$ is the unit normal vector over a given facet of the element, $\boldsymbol{w} = \begin{bmatrix} \boldsymbol{w}_p \\ \boldsymbol{w}_v \\ \boldsymbol{w}_{T_{\text{K}}} \end{bmatrix}$ is the vector of test functions, $\nabla\boldsymbol{w} = \begin{bmatrix} \nabla\boldsymbol{w}_p \\ \nabla\boldsymbol{w}_v \\ \nabla\boldsymbol{w}_{T_{\text{K}}} \end{bmatrix}$ is its gradient, $|_{b,i}$ means to substitute the boundary value (from the Dirichlet boundary condition), $|_{N,i}$ means to substitute the boundary value (from the Neumann boundary condition), the DG tensor jump operator (over $\boldsymbol{u}$) is $[\![\boldsymbol{u}]\!]_{\otimes,\boldsymbol{n}} = (\boldsymbol{u}^+ - \boldsymbol{u}^-)\otimes\boldsymbol{n}$, and the DG average operator (over $\boldsymbol{u}$) is $\langle\boldsymbol{u}\rangle = \frac{1}{2}(\boldsymbol{u}^+ + \boldsymbol{u}^-)$. The boundaries and the use of the superscripts "+" and "-" are illustrated in Fig. 1.

The second and fourth lines of (9) correspond to the convective numerical flux according to the local Lax–Friedrichs scheme (aka. Rusanov scheme) [31]. In practical terms, this numerical flux term serves so as to include an additional diffusivity in the equations, by using a locally-selected weight. The parameter $\alpha_{\text{LF}}(\boldsymbol{u})$ is a measure of the propagation speed on the facets, configuring a local dissipation. The parameter $\alpha_{\text{LF},b}(\boldsymbol{u},\boldsymbol{u}|_{b,i})$ is its corresponding value for the boundaries. These two parameters are given as:

$$\begin{aligned}
\alpha_{\text{LF}}(\boldsymbol{u}) &= \frac{1}{2}\max(\max_i|\lambda_{e,i}(\boldsymbol{u})^+|, \max_i|\lambda_{e,i}(\boldsymbol{u})^-|) \\
\alpha_{\text{LF},b}(\boldsymbol{u},\boldsymbol{u}|_{b,i}) &= \frac{1}{2}\max(\max_i|\lambda_{e,i}(\boldsymbol{u})|, \max_i|\lambda_{e,i}(\boldsymbol{u}|_{b,i})|),
\end{aligned} \tag{10}$$

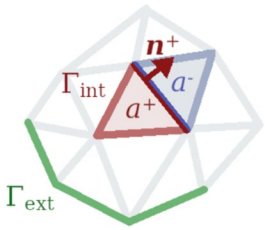


Fig. 1 Graphical representation of the notation used in the DG formulation for a part of a sample computational domain, where Γ_{ext} represents the boundary of the full computational domain, Γ_{int} represents the boundary of the illustrated element that is composed of only internal facets, a^+ is the variable according to the current element, a^- is the variable according to the neighbouring element, and $\boldsymbol{n}^+$ is the normal vector for one of the facets of the current element

where $\lambda_{e,i}$ is one of the eigenvalues of the convective part [41]. When considering incompressible flow, the list of eigenvalues is $\boldsymbol{\lambda}_e(\boldsymbol{u}) = [\boldsymbol{v}\cdot\boldsymbol{n}]$, and, when considering compressible flow, the list of eigenvalues is $\boldsymbol{\lambda}_e(\boldsymbol{u}) = \begin{bmatrix} \boldsymbol{v}\cdot\boldsymbol{n} - c \\ \boldsymbol{v}\cdot\boldsymbol{n} \\ \boldsymbol{v}\cdot\boldsymbol{n} + c \end{bmatrix}$, where $c = \sqrt{\frac{\gamma p}{\rho}}$ is the speed of sound.

The terms of (9) that include $\boldsymbol{\alpha}_{\mathrm{SIPG}}(\boldsymbol{u})$ and $\boldsymbol{\alpha}_{\mathrm{SIPG},b}(\boldsymbol{u}|_{b,i})$ correspond to the interior penalty in the Symmetric Interior Penalty Galerkin (SIPG) formulation [42], which is applied only to the dissipative part of the problem. This formulation essentially controls the imposition of Dirichlet boundary conditions and imposes additional stabilisations in the equations, by using an additional penalty parameter. They are given as:

$$\begin{aligned} \boldsymbol{\alpha}_{\mathrm{SIPG}}(\boldsymbol{u}) &= c_{ip}\frac{p^2}{\min(h^+, h^-)}\langle \boldsymbol{G}(\boldsymbol{u})\rangle \\ \boldsymbol{\alpha}_{\mathrm{SIPG},b}(\boldsymbol{u}|_{b,i}) &= c_{ip}\frac{p^2}{h}\boldsymbol{G}(\boldsymbol{u}|_{b,i}), \end{aligned} \tag{11}$$

where $h = \frac{V_{\mathrm{local}}}{A_{\mathrm{facet}}}$ is a measure of the element size in the mesh (V_{local} is the volume of each element, and A_{facet} is the area of each facet), $p \geqslant 1$ is the degree of the DG element, $c_{ip} > 0$ is the interior penalty parameter (which needs be chosen as a sufficiently large value; however, not too high, given that it affects the numerical convergence), and $\boldsymbol{G}(\boldsymbol{u}) = \frac{\partial \boldsymbol{F}_d}{\partial \boldsymbol{u}}$ is the homogeneity tensor.

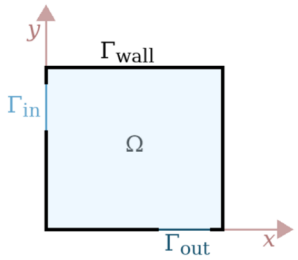
4 Boundary value problem

The external boundaries of a sample computational domain are shown in Fig. 2, and the boundary value problem is stated as:

$$\begin{aligned} [\text{Eqs. (1)-(3)}] \quad & \text{in } \Omega \\ \boldsymbol{v} = \boldsymbol{v}_{\mathrm{in}} \quad & \text{on } \Gamma_{\mathrm{in}} \\ T_{\mathrm{K}} = T_{\mathrm{K,in}} \quad & \text{on } \Gamma_{\mathrm{in}} \\ \boldsymbol{v} = \boldsymbol{0} \quad & \text{on } \Gamma_{\mathrm{wall}} \\ \nabla T_{\mathrm{K}}\cdot\boldsymbol{n} = 0 \quad & \text{on } \Gamma_{\mathrm{wall}} \\ p = p_{\mathrm{out}} \quad & \text{on } \Gamma_{\mathrm{out}} \\ \nabla T_{\mathrm{K}}\cdot\boldsymbol{n} = 0 \quad & \text{on } \Gamma_{\mathrm{out}}, \end{aligned} \tag{12}$$

where the inlet boundary is Γ_{in}, the outlet boundary is Γ_{out},

Fig. 2 Boundaries



the wall boundaries are Γ_{wall}, the inlet velocity is $\boldsymbol{v}_{in}$, the inlet temperature is $T_{\mathrm{K},in}$, and the outlet (static) pressure is p_{out}. The walls are set to be adiabatic, and the outlet is set for the heat fluxes to be zero. The Dirichlet conditions in eq. (12) are enforced weakly with a standard Nitsche approach.

5 Topology optimisation formulation

The topology optimisation formulation is defined as follows.

5.1 Material model

The material model corresponds to how the design variable (pseudo-density) affects the overall simulation. It corresponds to three effects: blocking the fluid flow inside the modelled solid material zones, given by the $\boldsymbol{f}_r(\alpha)$ term; and imposing adiabatic walls for the modelled solid material, imposed by not including any additional term in the energy equation [16]. The definition of $\boldsymbol{f}_r(\alpha)$, which is the resistance force to the fluid flow, is:

$$\boldsymbol{f}_r(\alpha) = -\kappa(\alpha)\boldsymbol{v}, \tag{13}$$

where $\kappa(\alpha)$ is the inverse permeability.

An interpolation function is used for $\kappa(\alpha)$ in order to allow a continuous transition between the solid ($\alpha = 0$) and fluid ($\alpha = 1$) states, better conditioning the topology optimisation problem. The interpolation function in this work is adopted from Borrvall and Petersson [1] as:

$$\kappa(\alpha) = \kappa_{\max} + (\kappa_{\min} - \kappa_{\max})\alpha\frac{1+q}{\alpha+q}, \tag{14}$$

where $\kappa_{\min} = 0$ kg/(m^{3}s) is the minimum permeability value, $\kappa_{\max}$ is a sufficiently high value to block the fluid flow in the modelled solid material, and q is a penalisation parameter that controls the convexity of $\kappa(\alpha)$.

5.2 Topology optimisation problem

The topology optimisation problem is formulated as:

$$\boxed{\begin{array}{l}\min\limits_{\alpha} J(\boldsymbol{u}(\alpha),\alpha)\\ \text{such that}\\ \quad \text{Fluid volume constraint: } \displaystyle\int_{\Omega_\alpha} \alpha\, d\Omega_\alpha \leqslant fV_0\\ \quad \text{Box constraint of } \alpha\text{: } 0 \leqslant \alpha \leqslant 1\\ \quad \text{Other constraints}\end{array}}, \tag{15}$$

where f is the specified volume fraction, $V_0 = \int_{\Omega_\alpha} d\Omega_\alpha$ is the volume of Ω_α (the design domain), $J(\boldsymbol{u}(\alpha),\alpha)$ is the objective function, and $\boldsymbol{u}(\alpha)$ is the state vector obtained by solving the boundary value problem (12)), which indirectly depends on the design variable α.

We consider energy dissipation Φ as the objective function (i.e., $J = \Phi$) [1]. For transient compressible flow, it becomes the following time average:

$$\begin{aligned}\Phi = \frac{1}{t_f - t_0}\int_{t_0}^{t_f} \Bigg\{ & \\ \int_\Omega \left[\frac{1}{2}\mu(\nabla\boldsymbol{v} + \nabla\boldsymbol{v}^T):(\nabla\boldsymbol{v} + \nabla\boldsymbol{v}^T)\right] d\Omega & \\ + \int_\Omega \lambda(\nabla\cdot\boldsymbol{v})(\boldsymbol{I}:\nabla\boldsymbol{v})\, d\Omega - \int_\Omega \boldsymbol{f}_r(\alpha)\cdot\boldsymbol{v}\, d\Omega \Bigg\} dt,\end{aligned} \tag{16}$$

where t_0 and t_f are the initial and final times of the simulation.

In cases where there is more than one outlet, and the outlets are located at different distances from the inlet, the topology optimisation algorithm may, depending on the fluid velocity, decide to, for example, simply close the farthest outlet and redirect the whole fluid flow to the nearest outlet, in order to minimize the energy dissipation. Therefore, if a certain amount of fluid is required to pass through each of the outlets, additional constraints must be imposed. In the case where there are only two outlets, if a constraint is imposed on one of the outlets, its counterpart is implicitly imposed on the other outlet, due to the continuity equation. This type of constraint is considered in one of the numerical examples, which features two outlets (Sect. 7.2), and is based on Deng et al. [24]. In this numerical example, the outlet that is chosen for imposing the additional constraint (i.e., the *constrained outlet*) is the upper outlet. Since including an equality constraint on the flow would probably be a quite strong imposition, it may be relaxed into an inequality constraint. Also note that using an equality constraint *per se* would actually be infeasible in the binary variable-based algorithm considered in this work, due to the intrinsic discreteness of the design variable, which means that this relaxation of the constraint makes it possible to use a binary variable-based algorithm with it. Define $m_{\text{out},1}$ to be the total mass that exits through the constrained outlet in a given time interval $[t_{1,0}, t_{1,f}]$, $m_{\text{out},1,\text{target}}$ the target mass to exit through the constrained outlet, m the total mass that enters through the inlet, and m_{eps} a small tolerance for the imposition of the constraint. The inequality constraint, which corresponds to the relaxation of $m_{\text{out},1} = m_{\text{out},1,\text{target}}$, becomes:

$$\left(\frac{m_{\text{out},1}}{m_{\text{out},1,\text{target}}} - 1\right)^2 \leqslant m_{\text{eps}}, \tag{17}$$

where $m_{\text{out},1} = \int_{t_{1,0}}^{t_{1,f}} \int_{\Gamma_{\text{out},1}} \rho\boldsymbol{v}\cdot\boldsymbol{n}\, d\Gamma_{\text{out},1} dt$, and $m = -\int_{t_0}^{t_f} \int_{\Gamma_{\text{in}}} \rho\boldsymbol{v}\cdot\boldsymbol{n} d\Gamma_{\text{in}} dt$. In the numerical examples, we are setting $m_{\text{out},1,\text{target}} = \frac{1}{3}m$ and $m_{\text{eps}} = 1.7 \times 10^{-5}$.

We numerically approximate the time integration of a given function J by the trapezoidal rule:

$$J = \sum_{i=0}^{n_t-1} \frac{1}{2}(J|_{t_{i+1}} + J|_{t_i})\Delta t, \tag{18}$$

where Δt is the size of the time step, i is the number of the current time, n_t is the total number of time steps, and $J|_{t_i}$ is the value of the function at time t_i.

5.3 Sensitivity analysis

The sensitivity of J is given by composing the sensitivities of the corresponding time steps. By following (18),

$$\frac{dJ}{d\alpha} = \sum_{i=0}^{n_t-1} \frac{1}{2}\left(\frac{dJ|_{t_{i+1}}}{d\alpha} + \frac{dJ|_{t_i}}{d\alpha}\right)\Delta t, \tag{19}$$

where the sensitivity for $J|_{t_i}$ is given from the adjoint method.

In order to consider the adjoint method for $J|_{t_i} = J|_{t_i}(\boldsymbol{u}_i(\alpha),\alpha)$, the first part is to consider the derivative with respect to α, and the chain rule for derivatives:

$$\frac{dJ|_{t_i}}{d\alpha} = \frac{\partial J|_{t_i}}{\partial \alpha} + \frac{\partial J|_{t_i}}{\partial \boldsymbol{u}_i}\frac{d\boldsymbol{u}_i}{d\alpha}. \tag{20}$$

Then, the Jacobian term $\frac{d\boldsymbol{u}_i}{d\alpha}$ can be obtained by differentiating the equilibrium equation $F = 0$, which corresponds to (9) with respect to α, and applying the chain rule again (considering $F = F(\boldsymbol{u}_i(\alpha), \boldsymbol{u}_{i-1}(\alpha), \alpha)$). Therefore,

$$\begin{aligned} &\frac{dF}{d\alpha} = 0 \implies \frac{\partial F}{\partial \boldsymbol{u}_i}\frac{d\boldsymbol{u}_i}{d\alpha} + \frac{\partial F}{\partial \boldsymbol{u}_{i-1}}\frac{d\boldsymbol{u}_{i-1}}{d\alpha} + \frac{\partial F}{\partial \alpha} = 0 \\ &\implies \frac{d\boldsymbol{u}_i}{d\alpha} = -\left(\frac{\partial F}{\partial \boldsymbol{u}_i}\right)^{-1}\left[\frac{\partial F}{\partial \alpha} + \frac{\partial F}{\partial \boldsymbol{u}_{i-1}}\frac{d\boldsymbol{u}_{i-1}}{d\alpha}\right]. \end{aligned} \tag{21}$$

By substituting (21) in (20) for time step t_i, $\frac{dJ|_{t_i}}{d\alpha}$ can be computed by:

$$\begin{aligned} \frac{dJ|_{t_i}}{d\alpha} = \frac{\partial J|_{t_i}}{\partial \alpha} - \frac{\partial J|_{t_i}}{\partial \boldsymbol{u}_i}\left(\frac{\partial F|_{t_i}}{\partial \boldsymbol{u}_i}\right)^{-1}\left[\frac{\partial F|_{t_i}}{\partial \alpha}\right. \\ \left. + \frac{\partial F|_{t_i}}{\partial \boldsymbol{u}_{i-1}}\frac{d\boldsymbol{u}_{i-1}}{d\alpha}\right], \end{aligned} \tag{22}$$

which is the direct (tangent linear) model.

The adjoint model is given by computing $\frac{\partial J|_{t_i}}{\partial \boldsymbol{u}_i}\left(\frac{\partial F|_{t_i}}{\partial \boldsymbol{u}_i}\right)^{-1}$ first, which can be done if the conjugate transpose is applied to (22). Then, the adjoint variable $\boldsymbol{\lambda}_{J|_{t_i}}$ (Lagrange multiplier of the weak form) is introduced in order to compute $\left(\frac{\partial F|_{t_i}}{\partial \boldsymbol{u}_i}\right)^{-1^*}\left(\frac{\partial J|_{t_i}}{\partial \boldsymbol{u}_i}\right)^*$. Therefore,

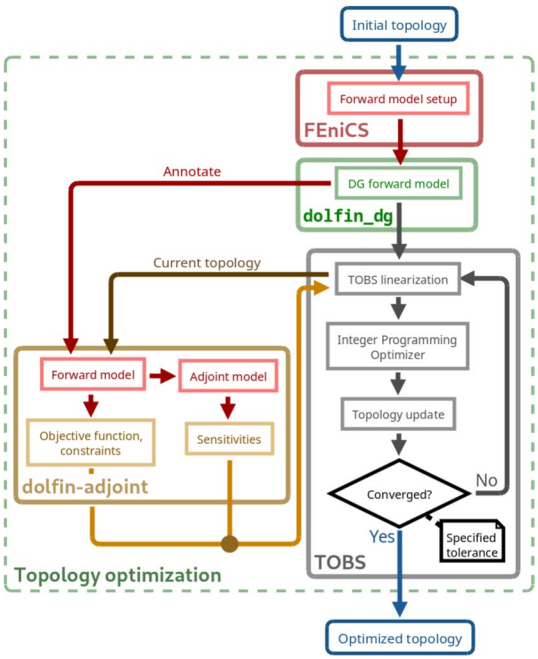


Fig. 3 Flowchart of the numerical implementation

$$\begin{aligned} \left(\frac{dJ|_{t_i}}{d\alpha}\right)^* = \left(\frac{\partial J|_{t_i}}{\partial \alpha}\right)^* - \left[\left(\frac{\partial F|_{t_i}}{\partial \alpha}\right)^*\right. \\ \left. + \left(\frac{d\boldsymbol{u}_{i-1}}{d\alpha}\right)^*\left(\frac{\partial F|_{t_i}}{\partial \boldsymbol{u}_{i-1}}\right)^*\right]\boldsymbol{\lambda}_{J|_{t_i}}, \end{aligned} \tag{23}$$

$$\left(\frac{\partial F|_{t_i}}{\partial \boldsymbol{u}_i}\right)^*\boldsymbol{\lambda}_{J|_{t_i}} = \left(\frac{\partial J|_{t_i}}{\partial \boldsymbol{u}_i}\right)^* \text{(adjoint equation)}, \tag{24}$$

where " * " is the conjugate transpose. Note that (23) and (24) are computed with automatic differentiation in this work. The term $\frac{d\boldsymbol{u}_{i-1}}{d\alpha}$ can be substituted by a similar equation to (21) (considering $F = F(\boldsymbol{u}_{i-1}(\alpha), \boldsymbol{u}_{i-2}(\alpha), \alpha)$), recursively until reaching $\boldsymbol{u}_0$. Then, the terms involving the computation of inverse matrices are converted into adjoint variables, such as how it was done for (24).

6 Numerical implementation of the topology optimisation problem

The implementation of the topology optimisation problem is illustrated in Fig. 3. It begins with the initial topology, i.e. the initial distribution of the design variable. Then, the first step is to define the forward model in the finite element-based platform FEniCS [33], in the form of the convective, dissipative and external terms ((6)–(8)). These terms are used as the input for the dolfin_dg library [34], which is able to use them to automatically derive the SIPG formulation (9). This equation can be defined by hand in FEniCS; however, having an automated way to define it allows further flexibility when changing the involved physics and boundary conditions of the problem. We then employ the dolfin-adjoint library [35, 36], which annotates the forward model and symbolically derives its corresponding adjoint model. Next, the topology optimisation procedure starts, by using the TOBS algorithm [38, 39]. TOBS is a binary variable-based topology optimisation solver, which makes use of the continuous problem derivatives, and performs a linearisation of the problem at every topology optimisation iteration, relaxes the constraints and includes a truncation error constraint. After defining the whole topology optimisation integer variable subproblem, an efficient external procedure (e.g., CPLEX®, from IBM), is used. The simulations in FEniCS are solved with a Newton solver. At the end of the optimisation loop, a specified tolerance is verified, concluding the procedure.

We discretise in time with the implicit Euler method. Given a variable a and that the corresponding equations may be represented as $\frac{\partial a}{\partial t} = f(a)$, the discrete equations become

$\frac{a_k - a_{k-1}}{\Delta t} = f(a_k)$, where k is the index of the current time step. The time integrations of the objective function and the mass constraint are performed according to the trapezoidal rule for numerical integration.

Since the flow is compressible, it is not necessary to satisfy the LBB (Ladyžhenskaya-Babuška-Brezzi) condition for numerical stability. Therefore, all state variables are discretised with DG1 elements (dP_1 interpolation). Since the optimised topology is continuously distributed in the design domain, and it is not expected for it to feature discontinuities, the design variable is discretised with continuous elements (P_1 interpolation). Note that, in the DG1 elements (dP_1 interpolation), the nodes (vertex values) are not shared between elements, which allows element-to-element discontinuities in the DG formulation.

A Helmholtz-based pseudo-density filter is employed in the numerical examples [43]. In order to diminish the effects of mesh-dependency in the sensitivities, the sensitivities are scaled by an L^2 Riesz map [44].

7 Numerical results

In the numerical examples, transience originates from the time-varying inlet velocity profile $\boldsymbol{v}_{\text{in}}$. The fluid properties that are considered are $c_p = 1004$ J/(kg K), $\mu = 0.01$ Pa s, $k_{th} = 1.7$ W/(m K), and $M_{\text{molar}} = 25$ g/mol, while the boundary conditions are given by $T_{\text{K,in}} = 50$ K and $p_{\text{out}} = 10^5$ Pa. The inlet velocity profiles are parabolic. The Helmholtz filter radii are 0.00938 mm and 0.0284 mm, respectively, for the two sections of the numerical examples. The initial guess for the pseudo-density (design variable) distribution is the computational domain filled with only fluid. The parameters of the TOBS algorithm are set as $\varepsilon_{\text{relax}} = 0.05$ and $\beta_{\text{flip limit}} = 0.05$ [38, 39], and the algorithm is iterated until the design variable stops changing. The optimised topologies are the optimised distributions of the design variable, and it is possible to delineate the contours of the fluid part of the domain. The identified contours are considered as shown in Fig. 4, and are used to compute the objective function values after the topology optimisation.

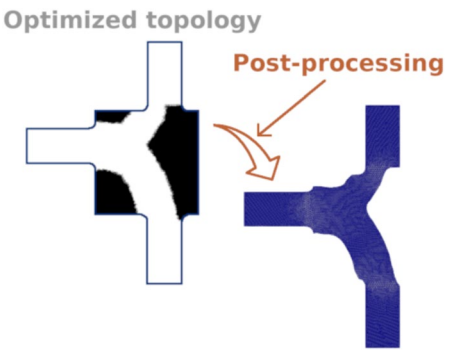


Fig. 4 Post-processing procedure

The Reynolds number is defined from the maximum local Reynolds number based on the size of the inlet, and the characterisation of the fluid flow as compressible depends on the Mach number. These two quantities are given as:

$$\text{Re}_\ell = \frac{\mu\,|\boldsymbol{v}|\,\ell_{\text{in}}}{\rho}, \text{ and} \tag{25}$$

$$\text{Ma}_\ell = \frac{|\boldsymbol{v}|}{v_s}, \tag{26}$$

where $v_s = \sqrt{\gamma \frac{R_U}{M_{\text{molar}}} T_K}$ is the speed of sound in the fluid.

Two numerical examples are presented, for a nozzle and a pipe joint. In both cases, the flow is compressible, which can be seen from the maximum local Mach numbers of about 0.5 and 0.4, respectively. The maximum local Reynolds numbers are about 50 and 40, respectively. The interior penalty parameter c_{ip} is set to 50 and 100 in the two examples, respectively.

7.1 Nozzle

The first example is a nozzle under transient compressible flow, illustrated in Fig. 5. There is an inlet above, and an outlet below. The angle of the inlet velocity changes linearly in time from θ_0 to θ_1 ($\theta_1 = -\theta_0$), from a given parabolic inlet velocity profile at 0°, while always keeping the same flow rate. The maximum value of the inlet velocity at 0° is set as 40 m/s. The design domain is restricted to the height h_1. The time-related parameters of the simulation are $t_0 = 0$ s, $t_1 = 1$ s, $\Delta t = 0.05$ s and the initial state is set as the steady state simulation of the t_0 configuration. On the optimisation side, the specified maximum fluid volume fraction (f) is set as 70%, the maximum inverse permeability value ($\kappa_{\max}$) is set as $2.5 \times 10^{16}\mu$, and the penalisation parameter q is set as one. The mesh is composed of 25,600 elements and is shown in Fig. 6. The topology optimisation problem from (15) adjusted for this numerical example is:

$$\begin{aligned} &\min_{\alpha}\ \Phi(\boldsymbol{u}(\alpha), \alpha) \\ &\text{such that} \\ &\text{Fluid volume constraint: } \int_{\Omega_\alpha} \alpha\, d\Omega_\alpha \leqslant f V_0 \\ &\text{Box constraint of } \alpha\text{: } 0 \leqslant \alpha \leqslant 1 \end{aligned} \tag{27}$$

The topology optimisation results are shown in Fig. 7. The values are shown for $\Delta\Phi_p$, which is the percentage increase/decrease of the energy dissipation Φ with respect

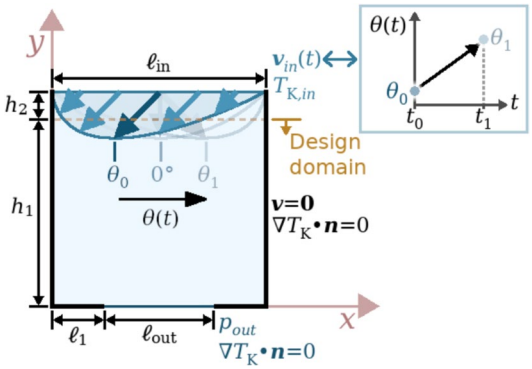
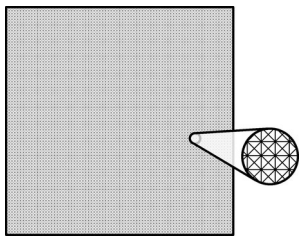


Fig. 5 Nozzle under transient flow ($h_1 = 0.875$ mm, $h_2 = 0.125$ mm, $\ell_1 = 0.25$ mm, $\ell_{in} = 1$ mm, $\ell_{out} = 0.5$ mm)

Fig. 6 Mesh for the nozzle case



to the steady state topology (0°) under the given transient conditions for θ_1. In all cases, this decrease is less than 1%. The first point of the plot from Fig. 7 corresponds to the steady state topology under the steady state conditions, which means that $\Delta\Phi_p$ is naturally zero in it. The remaining values for $\Delta\Phi_p$ are negative, which mean that the optimised topologies for transient flow dissipate less energy than the steady state topology operating under transient flow. The differences between the optimised topologies from Fig. 7 are better visualized in Fig. 8, which shows the tendency of having a thicker and taller channel the higher θ_1 is.

The local Mach number and density are shown in Fig. 9, for the steady state topology: the local Mach number reaches a maximum value of 0.51, showing that the fluid flow is clearly compressible and leads to the portrayed 26% variation in the fluid density.

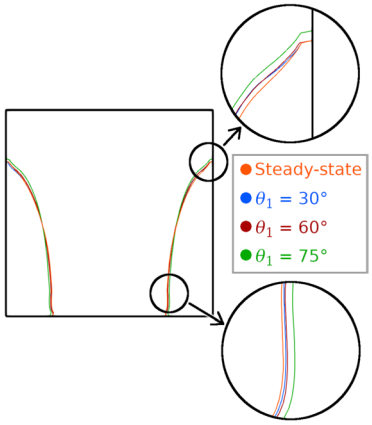


Fig. 8 Comparison of the contours of the optimised nozzles

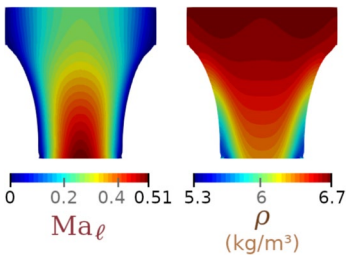
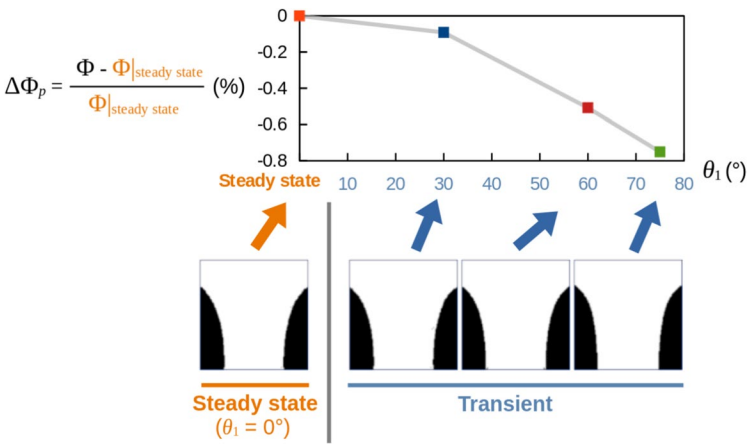


Fig. 9 Local Mach number and density for the steady state nozzle

The convergence curve of the optimised nozzle for $\theta_1 = 75°$ is shown in Fig. 10. The values are presented for $f_{ni} = \frac{f_i - \min(f_i)}{\max(f_i) - \min(f_i)}$, which represents the normalised function values (energy dissipation or volume constraint) between 0 and 1, where f_i represents the current function. In the case of the volume constraint, the value 1 represents the satisfied constraint.

The simulation results for $\theta_1 = 75°$ are shown in Fig. 11, for some time values. Note that, from the velocity values, due to $\theta_1 = 75°$, there is a quite abrupt change in the flow direction when the flow approaches the side walls. This probably means that, for the given volume constraint, it should be better to have the walls closer to the inlet, helping

Fig. 7 Change in the energy dissipation of the optimised nozzles



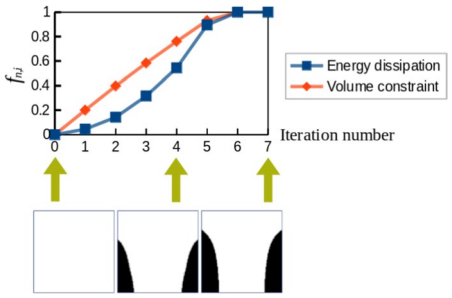


Fig. 10 Convergence curve of the optimised nozzle for $\theta_1 = 75°$

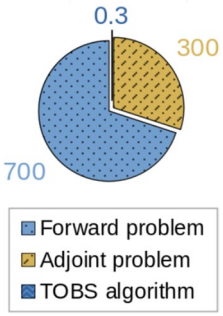
the fluid to smoothly change direction afterwards towards the outlet. Also, note that the simulation results are almost symmetric, aside for small differences, which is caused by the laminar behavior.

A brief insight on the computational cost of the performed topology optimisation is shown in Fig. 12, with the average/typical timings. The main cost is in the solution of the forward problem, while the solution of the adjoint problem takes a little less than half of this cost. These timings can be improved by employing a preconditioned iterative solver (such as GMRES) for the involved linear problem computations, instead of using the direct solver UMFPACK. Note that the computational cost of the TOBS algorithm is negligible when compared to the other two parts.

7.2 Pipe joint

The second example is a pipe joint, inspired by the problem presented by Deng et al. [24] for transient laminar incompressible flow. The present case is illustrated in Fig. 13. There is an inlet on the left side of the domain, and there are two outlets, one above and one below. The inlet velocity profile $\boldsymbol{v}_{in}$ is parabolic, with a time-dependent variation (see Fig. 13). The magnitude of the inlet parabolic velocity profile is changed according to $v_{\text{in,max}}(t)$ from Fig. 13, where $v_{\text{in,max},0} = 52$ m/s, $v_{\text{in,max},1} = 65$ m/s. The design domain is restricted in order not to include the inlet and outlet channels. The time-related parameters of the simulation are $t_0 = 0$ s, $t_1 = 0.5$ s, $t_2 = 1$ s, $t_3 = 2$ s, $t_4 = 2.5$ s, $t_5 =$

Fig. 12 Average/typical timings for each optimisation iteration (in seconds)



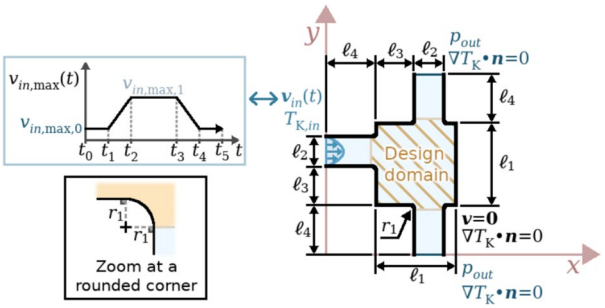
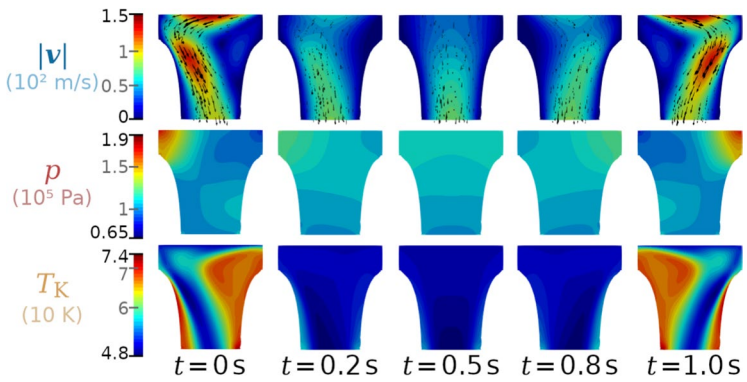


Fig. 13 Pipe joint under transient flow ($\ell_1 = 3$ mm, $\ell_2 = 1$ mm, $\ell_3 = 1.5$ mm, $\ell_4 = 2$ mm, $r_1 = 0.2$ mm)

3 s, $\Delta t = 0.05$ s and the initial state is set as the steady state simulation of the t_0 configuration. On the optimisation side, the specified maximum fluid volume fraction (f) is set as 50%, the maximum inverse permeability value ($\kappa_{\max}$) is set as $2.5 \times 10^{12}\mu$, and the penalisation parameter q is set as one. The mesh is composed of 11,815 elements and is shown in Fig. 14.

Here, two cases are considered, by changing the time interval $[t_{1,0}, t_{1,f}]$, which is used to compute the additional mass constraint. In the first case, the whole time interval $[t_0, t_f]$ is considered, meaning that the total mass that flows through the upper outlet is being constrained. In the second case, the reduced time interval $[t_2, t_3]$ is considered, meaning that the mass that flows through the upper outlet is constrained in this time interval. This can be seen depicted in Fig. 15. Overall, this means that both cases impose that

Fig. 11 Simulation results of the optimised nozzle for $\theta_1 = 75°$



the optimised topologies need to connect the inlet to both outlets (i.e., closing one of them is not allowed), and that the second case has to allow a higher mass flow rate than the first case. The topology optimisation problem from (15) adjusted for this numerical example is given as:

$$\begin{aligned}
&\min_{\alpha} \ \Phi(\boldsymbol{u}(\alpha), \alpha) \\
&\text{such that} \\
&\quad \text{Fluid volume constraint:} \ \int_{\Omega_\alpha} \alpha \, d\Omega_\alpha \leqslant \ fV_0 \\
&\quad \text{Outflow mass constraint:} \\
&\quad \left(\frac{m_{\text{out},1}}{m_{\text{out},1,\text{target}}} - 1 \right)^2 \leqslant m_{\text{eps}} \\
&\quad \text{Box constraint of } \alpha\text{:} \ 0 \ \leqslant \ \alpha \leqslant \ 1
\end{aligned} \tag{28}$$

The optimised topologies are shown in Fig. 16. Note that the channel towards the upper outlet is thicker in the second case, allowing a larger mass flow (see Fig. 17), which is consistent with the imposed constraint. When comparing the energy dissipation values for both cases, it can be noticed that it is 5.3% higher in the first case. This is probably caused by the fact that the upper outlet is the nearest outlet with respect to the inlet, dissipating less energy.

Also note that, even at the lowest flow rate, such as in the end of the simulation ($t = 3$ s), the flow is still compressible for both optimised cases. This can be seen in Fig. 18, which portrays the local Mach number and density (20% variation in the density).

8 Conclusions

In this work, a topology optimisation formulation for transient compressible flow was presented, discretised with a DG method. Two numerical examples are presented to illustrate the developed methodology. The first numerical example is a nozzle, with an inlet velocity with a varying angle, which shows that the fluid exhibits compressible effects and that the transient conditions slightly change the optimised design, when compared to a steady-state configuration. Overall, the higher the varying angle, the taller and thicker the outlet part of the nozzle becomes. The second numerical example is a pipe joint, featuring one inlet and two outlets, and shows that different transient constraints

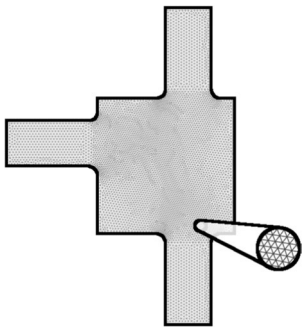


Fig. 14 Mesh for the pipe joint case

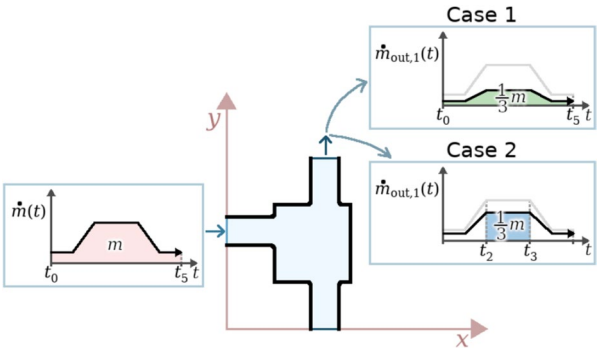
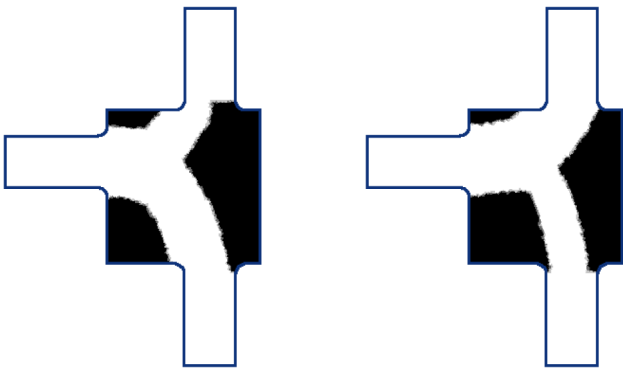


Fig. 15 Outflow mass constraints considered in each case



(a) Case 1. (b) Case 2.

Fig. 16 Optimised topologies for the pipe joint case

change the overall design—in this case, the upper pathway becomes larger when a larger mass is necessary to flow to the upper channel for a while.

There remain many important extensions of this work, including the incorporation of turbulence, different types of heat transfer, flow machine design, 3D configurations, and supersonic flows.

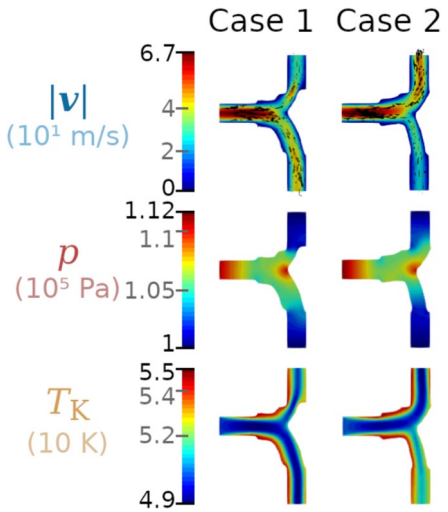


Fig. 17 Simulation result for the pipe joint case at $t = 1.5$ s

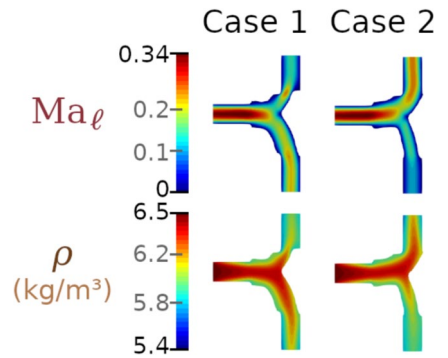


Fig. 18 Local Mach number and density for the pipe joint case at $t = 3$ s

Acknowledgements This research was partly supported by CNPq (Brazilian National Research Council) and FAPESP (São Paulo Research Foundation). The authors thank the supporting institutions. The first author thanks the financial support of FAPESP under grant 2023/10817-6. The fourth author thanks the financial support of CNPq (National Council for Research and Development) under grant 304508/2023-3. The first, third and fourth authors also gratefully acknowledge support of the RCGI—Research Centre for Greenhouse Gas Innovation, hosted by the University of São Paulo (USP) and sponsored by FAPESP – São Paulo Research Foundation (2020/15230-5) and Shell Brasil, and the strategic importance of the support given by ANP (Brazil's National Oil, Natural Gas and Biofuels Agency) through the R&DI levy regulation. The second author was funded by the Engineering and Physical Sciences Research Council [grant numbers EP/R029423/1 and EP/W026163/1], the Donatio Universitatis Carolinae Chair "Mathematical modelling of multicomponent systems", and the UKRI Digital Research Infrastructure Programme through the Science and Technology Facilities Council's Computational Science Centre for Research Communities (CoSeC). For the purpose of open access, the authors have applied a CC BY public copyright licence to any author accepted manuscript (AAM) arising from this submission.

Author contributions All authors contributed to the conceptualization and writing. The methodology, implementation and analysis were conducted by DHA. The supervision of the work was performed by PEF and JRM The first draft of the manuscript was written by DHA, and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript. ECNS, PEF and JRM were responsible for mentoring and funding acquisition.

Data availability No datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors declare no conflict of interest.

References

1. Borrvall T, Petersson J (2003) Topology optimization of fluids in Stokes flow. Int J Numer Meth Fluids 41(1):77–107
2. Gersborg-Hansen A, Sigmund O, Haber RB (2005) Topology optimization of channel flow problems. Struct Multidiscip Optim 30(3):181–192
3. Pingen G, Maute K (2010) Optimal design for non-Newtonian flows using a topology optimization approach. Comput Math Appl 59(7):2340–2350
4. Suarez MAA, Romero JS, Pereira A, Menezes I (2022) On the virtual element method for topology optimization of non-Newtonian fluid-flow problems. Eng Comput 38:5445–5466
5. Yaji K, Yamada T, Yoshino M, Matsumoto T, Izui K, Nishiwaki S (2016) Topology optimization in thermal-fluid flow using the lattice Boltzmann method. J Comput Phys 307:355–377
6. Galanos N, Papoutsis-Kiachagias E, Giannakoglou K (2025) A continuous adjoint cut? Cell formulation for topology optimization of bi? Fluid heat exchangers. Int J Numer Methods Heat Fluid Flow
7. Papoutsis-Kiachagias E, Kontoleontos E, Zymaris A, Papadimitriou D, Giannakoglou K (2011) Constrained topology optimization for laminar and turbulent flows, including heat transfer. CIRA, editor, EUROGEN, Evolutionary and Deterministic Methods for Design, Optimization and Control, Capua, Italy
8. Andreasen CS, Gersborg AR, Sigmund O (2009) Topology optimization of microfluidic mixers. Int J Numer Meth Fluids 61:498–513
9. Zhang B, Liu X (2015) Topology optimization study of arterial bypass configurations using the level set method. Struct Multidiscip Optim 51(3):773–798
10. Romero JS, Silva ECN (2014) A topology optimization approach applied to laminar flow machine rotor design. Comput Methods Appl Mech Eng 279:268–300
11. Lin S, Zhao L, Guest JK, Weihs TP, Liu Z (2015) Topology optimization of fixed-geometry fluid diodes. J Mech Des 137(8):081402–10814028
12. Jensen KE, Szabo P, Okkels F (2012) Topology optimization of viscoelastic rectifiers. Appl Phys Lett 100(23):234102
13. Duan X, Li F, Qin X (2016) Topology optimization of incompressible Navier-Stokes problem by level set based adaptive mesh method. Comput Math Appl 72(4):1131–1141
14. Sá LFN, Amigo RCR, Novotny AA, Silva ECN (2016) Topological derivatives applied to fluid flow channel design optimization problems. Struct Multidiscip Optim 54(2):249–264
15. Deng Y, Liu Z, Wu Y (2013) Topology optimization of steady and unsteady incompressible Navier-Stokes flows driven by body forces. Struct Multidiscip Optim 47(4):555–570
16. Sá LFN, Okubo Jr CM, Silva ECN (2021) Topology optimization of subsonic compressible flows. Struct Multidiscip Optim

17. White FM (2011) Fluid mechanics, 7th edn. McGraw-Hill, New York
18. Anderson JD (2003) Modern compressible flow: with historical perspective, 3rd edn. McGraw-Hill New York, New York
19. Okubo CM Jr, Sa LFNDS, Kiyono CY, Silva ECN (2022) A discrete adjoint approach based on finite differences applied to topology optimization of flow problems. Comput Methods Appl Mech Eng 389:114406
20. Maffei FS, Sá LFN, Moscatelli E, Picelli R, Meneghini JR, Silva ECN (2023) Integer programming topology optimization for subsonic compressible flows with geometry trimming. Int J Heat Mass Transf 201:123614
21. Garcia Rodriguez LF, Alonso DH, Silva ECN (2025) Turbulence effects in the topology optimization of compressible subsonic flow. Int J Numer Meth Fluids 97(1):44–68
22. Wein L, Seume JR, Herbst F (2018) Unsteady flow in a labyrinth seal. In: Proceedings of GPPS Forum 18 . Zug: Global Power and Propulsion Society
23. Kreissl S, Pingen G, Maute K (2011) Topology optimization for unsteady flow. Int J Numer Meth Eng 87(13):1229–1253
24. Deng Y, Liu Z, Zhang P, Liu Y, Wu Y (2011) Topology optimization of unsteady incompressible Navier-Stokes flows. J Comput Phys 230(17):6688–6708
25. Makhija D, Beran PS (2019) Time scale effects in topology optimization of the interior material distribution of a body subject to transient conjugate heat transfer. In: AIAA Scitech 2019 Forum, p. 1468
26. Hartmann R (2006) Discontinuous Galerkin methods for compressible flows: higher order accuracy, error estimation and adaptivity. VKI Lecture Series 0, 1–50
27. Dedner A, Kröner D, Rohde C, Schnitzer T, Wesenberg M (2003) Comparison of finite volume and discontinuous Galerkin methods of higher order for systems of conservation laws in multiple space dimensions, 573–589
28. Collis SS (2002) Discontinuous Galerkin methods for turbulence simulation. In: Proceedings of the summer program 2002, pp 155–167 . Center for Turbulence Research
29. Coburn N (1954) Discontinuities in compressible fluid flow. Math Mag 27(5):245–264
30. Altmann C, Beck AD, Hindenlang F, Staudenmaier M, Gassner GJ, Munz C-D (2013) An efficient high performance parallelization of a discontinuous Galerkin spectral element method. Fac Multicore-Chall III Asp New Parad Technol Parall Comput 37–47
31. Gassner GJ, Winters AR (2021) A novel robust strategy for discontinuous Galerkin methods in computational fluid mechanics: Why? When? What? Where?. Front Phys 8
32. Papadopoulos IP (2022) Numerical analysis of a discontinuous Galerkin method for the Borrvall-Petersson topology optimization problem. SIAM J Numer Anal 60(5):2538–2564
33. Logg A, Mardal K-A, Wells G (2012) Automated solution of differential equations by the finite element method: the FEniCS book vol. 84. Springer, Berlin . https://fenicsproject.org/documentation/
34. Houston P, Sime N (2018) Automatic symbolic computation for discontinuous Galerkin finite element methods. SIAM J Sci Comput 40(3):327–357
35. Farrell PE, Ham DA, Funke SW, Rognes ME (2013) Automated derivation of the adjoint of high-level transient finite element programs. SIAM J Sci Comput 35(4):369–393
36. Mitusch S, Funke S, Dokken J (2019) dolfin-adjoint 2018.1: automated adjoints for FEniCS and Firedrake. J Open Source Softw 4(38):1292
37. Davis TA (2004) Algorithm 832: UMFPACK v4.3—an unsymmetric-pattern multifrontal method. ACM Trans Math Softw 30(2):196–199
38. Souza BC, Yamabe PVM, Sá LFN, Ranjbarzadeh S, Picelli R, Silva ECN (2021) Topology optimization of fluid flow by using integer linear programming. Struct Multidiscip Optim
39. Sivapuram R, Picelli R (2018) Topology optimization of binary structures using integer linear programming. Finite Elem Anal Des 139:49–61
40. Landet T, Mardal K-A, Mortensen M (2020) Slope limiting the velocity field in a discontinuous Galerkin divergence-free two-phase flow solver. Comput Fluids 196:104322
41. Rohde A (2001) Eigenvalues and eigenvectors of the Euler equations in general geometries. In: 15th AIAA computational fluid dynamics conference, p 2609
42. Hartmann R, Houston P (2008) An optimal order interior penalty discontinuous Galerkin discretization of the compressible Navier-Stokes equations. J Comput Phys 227(22):9670–9685
43. Lazarov BS, Sigmund O (2010) Filters in topology optimization based on Helmholtz-type differential equations. Int J Numer Meth Eng 86(6):765–781
44. Schwedes T, Ham DA, Funke SW, Piggott MD (2017) Mesh dependence in PDE-constrained optimisation—an application in Tidal Turbine array layouts, 1st edn. Springer, Switzerland