Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 386 (2021) 114177
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

Topology optimization of Stokes flow with traction boundary conditions using low-order finite elements

Carl-Johan Thore

Division of Solid Mechanics, Department of Management and Engineering, Linköping University, 581 83 Linköping, Sweden

Received 20 May 2021; received in revised form 31 August 2021; accepted 6 September 2021
Available online 7 October 2021

Abstract

We consider topology optimization of Stokes flow with traction boundary conditions using finite elements with low-order velocity-approximation and an element-wise constant hydrostatic pressure. The finite element formulation is stabilized using a penalty on the jump in pressure between adjacent elements. Convergence of solutions to the finite element-discretized topology optimization problem is shown, and several optimization problems are solved using a preconditioned conjugate gradient solver for the finite element matrix problem. Stable convergence to high-quality designs without an excessive number of linear solver iterations is observed, and it is seen that the finite element formulation is not particularly sensitive to the choice of the pressure jump penalty parameter, thus making it a practically useful method.

Keywords: Topology optimization; Stokes flow; Stabilized finite elements

1. Introduction

Topology optimization (TO) in fluid flow problems is a research area which receives a lot of academic and industrial interest [1]. In this article we consider TO problems where the flow is governed by the Stokes–Brinkman model proposed by Borrvall and Petersson [2]. In contrast to most work on TO with this model we consider problems where traction and not velocity is prescribed on the potential inlet and/or outlet regions of the design domain. An advantage of prescribing traction rather than velocity is that there is then no need to specify the exact location of those regions *a priori*. For example, in a coupled elasticity and flow TO problem it might be useful to be able to vary the location of the inlets and outlets during the optimization process to achieve a good compromise between efficient flow and high stiffness.

A drawback of traction boundary conditions is that it precludes the use of certain computationally attractive low-order finite elements (FEs) such as the Crouzeix–Raviart P1–P0 element. (While the mass matrix in the Stokes–Brinkman system ensures, with a non-zero lower bound on the coefficient, coercivity also in the discrete setting, the element can exhibit locking-type phenomena [3, Section 5].) The purpose of this work is therefore to investigate the performance of a stabilized FE method [4–7] which enables the use of low-order velocity and element-wise constant hydrostatic pressure approximations. The method, originally proposed by Hughes and Franca [4], penalizes jumps in the pressure between adjacent FEs resulting in satisfaction of a discrete inf–sup condition.

E-mail address: carl-johan.thore@liu.se.

https://doi.org/10.1016/j.cma.2021.114177

It is simple to implement, the main complications being the need for neighbourhood data for the FE mesh; gives a sparse, symmetric contribution to the system matrix; works with iterative linear solvers; and has only one free parameter, the selection of which appears not to be particularly tricky as we shall see. An interesting property of the method is that, as pointed out in [7], at least for P1–P0 elements one obtains H_{div}-stability which may be helpful to improve stability also in "grey" regions of the design domain where Darcy-like flow might occur. Finally, the method works for any FE for which a discrete Korn's inequality is satisfied, giving a lot of freedom for the choice of velocity approximation; we show below the use of both P1–P0 and Q1–P0 elements in 2D and 3D.

Higher order stable elements such as the Taylor–Hood P2–P1 lead (for a given mesh density) to a much larger number of degrees of freedom which is problematic for large-scale (3D) TO problems. Adding to that, stabilization may anyway be needed to handle the low permeability limit [8], and in the case of Navier–Stokes flow (to which the method studied herein is also applicable) to handle high-speed flows. For two-dimensional problems, the non-conforming, stable P1–P0 element due to Kouhia and Stenberg [9] is an attractive alternative to the tested method: it is as simple to implement as the conforming (but unfortunately useless) P1–P0 element, and introduces no additional parameters into the FE problem. Unfortunately, it does not have a counterpart in 3D. As for alternative stabilization methods, Ref. [10] could be considered.

The first article on TO in flow problems using stabilized FE methods appears to be due to Guest and Prévost [11] who solved problems similar to [2] but using a different design-parametrization interpolating between Darcy and Stokes flow. Since then, several publications have mentioned the use of stabilization; Refs. [12–16] for example use Galerkin/least squares stabilization to allow for equal-order velocity and pressure approximation. We have not, however, found any publications on TO in flow problems where stabilization is a central theme treated in detail.

2. The continuum problem

The flow in the design domain Ω (a bounded, connected set with Lipschitz boundary) is governed by a state problem of the (weak) form: Find $(\boldsymbol{u}, p) \in \boldsymbol{V} \times L^2(\Omega)$ such that

$$a(\rho; \boldsymbol{u}, \boldsymbol{v}) + (-p, \text{div}\boldsymbol{v}) + (q, \text{div}\boldsymbol{u}) = \ell(\boldsymbol{v}), \quad \forall (\boldsymbol{v}, q) \in \boldsymbol{V} \times L^2(\Omega), \tag{1}$$

where the design $\rho \in L^\infty(\Omega, [0, 1])$, $\boldsymbol{V} = \{\boldsymbol{v} \in \boldsymbol{H}_1(\Omega) \mid \boldsymbol{v}_{\Gamma_u} = \boldsymbol{0}\}$, with $\Gamma_u \subset \partial\Omega$ such that $|\Gamma_u| > 0$, and $(\cdot, \cdot)$ denotes the $L^2(\Omega)$ inner product. The functional a is defined as

$$a(\rho; \boldsymbol{u}, \boldsymbol{v}) = \int_\Omega \alpha(\rho)\boldsymbol{u} : \boldsymbol{v} \, \text{d}V + \int_\Omega 2\mu\boldsymbol{\varepsilon}(\boldsymbol{u}) : \boldsymbol{\varepsilon}(\boldsymbol{v}) \, \text{d}V, \tag{2}$$

where $\boldsymbol{\varepsilon}(\boldsymbol{u}) = \frac{1}{2}(\nabla\boldsymbol{u} + \nabla\boldsymbol{u}^\mathsf{T})$ and $\mu > 0$ is the viscosity. The function $\alpha : [0, 1] \to [0, \overline{\alpha}]$, $\overline{\alpha} < \infty$, is defined a.e. in Ω as

$$\alpha(\rho) = \overline{\alpha} - \overline{\alpha}(1 - \rho)\frac{1 + q}{1 - \rho + q}, \tag{3}$$

where $q > 0$. This function is chosen such that large values of $\overline{\alpha}$ give large resistance to flow in solid parts ($\rho \approx 1$) of the design domain. The load-functional is defined by

$$\ell(\boldsymbol{v}) = \int_{\Gamma_t \equiv \partial\Omega \setminus \Gamma_u} \boldsymbol{t} : \boldsymbol{v} \, \text{d}A,$$

with traction $\boldsymbol{t} \in \boldsymbol{L}^2(\Gamma_t)$.

The design problem is to maximize the "fluid compliance", i.e.

$$\sup_{\rho \in \mathcal{H}} \ell(\boldsymbol{u}(\rho)), \tag{4}$$

where $\boldsymbol{u}(\rho)$ is part of the solution to (1), and

$$\mathcal{H} = \left\{\rho \in L^\infty(\Omega) \mid 0 \le \rho \le 1 \text{ a.e. in } \Omega, \ \int_\Omega \rho \, \text{d}V = \gamma|\Omega|\right\},$$

in which $\gamma \in (0, 1)$ is the allowed fraction of solid material in the design domain. One physical interpretation of (4) is that we seek to maximize the ($\boldsymbol{t}$-weighted) average velocity at the inlet and/or outlet. Since $\sup_{\rho\in\mathcal{H}} \ell(\boldsymbol{u}(\rho)) = -2\inf_{\rho\in\mathcal{H}} -\frac{1}{2}\ell(\boldsymbol{u}(\rho))$ and the minimization problem is of the same form as the second of (32) in [2], existence of a solution follows from Theorem 3.2 in that reference.

Remark. It was recently shown [17], for the case of pure homogeneous Dirichlet boundary conditions, that an optimal ρ satisfies $\rho \in H_1(\text{supp}\,\boldsymbol{u})$; i.e., ρ is at least somewhat smooth in those parts of Ω where $\|\boldsymbol{u}\| > 0$ ($\|\cdot\|$ denoting the Euclidean norm). This makes precise the intuitively plausible idea that unlike for elasticity, designs with infinitely small features cannot be optimal for Stokes flow. □.

3. Finite element discretization

Given a family of conforming, shape-regular meshes $\{\mathcal{T}_h\}_h > 0$, $\mathcal{T}_h = \{K_e\}_{e=1,\ldots,m}$, on Ω we introduce dense, conforming finite-dimensional approximation spaces $\boldsymbol{V}_h \subset \boldsymbol{V}$, $L_h \subset L^2(\Omega)$ (where L_h is the space of element-wise constant functions) and $\mathcal{H}_h \subset \mathcal{H}$ for the velocity, hydrostatic pressure and design, respectively. The discretized version of the design problem is

$$\sup_{\rho_h \in \mathcal{H}_h} \ell(\boldsymbol{u}_h(\rho_h)), \tag{5}$$

where $\boldsymbol{u}_h(\rho_h)$ is part of the solution to

$$a(\rho_h; \boldsymbol{u}_h, \boldsymbol{v}) + (-p_h, \text{div}\boldsymbol{v}) + (q, \text{div}\boldsymbol{u}_h) + J(p_h, q) = \ell(\boldsymbol{v}), \quad \forall (\boldsymbol{v}, q) \in \boldsymbol{V}_h \times L_h. \tag{6}$$

The pressure jump penalty function J is defined as

$$J(p, q) = \delta \sum_K h_{\partial K} \int_{\partial K \setminus \partial\Omega} [p][q]\, \mathrm{d}A, \tag{7}$$

in which the parameter $\delta \geq 0$, ∂K is the boundary of element K, and $h_{\partial K}$ is the length or area of ∂K. The term $[p] = p_{|K_e} - p_{|K_f}$ is the jump in p over the common boundary $\partial K_e \cap \partial K_f$ of two elements K_e and K_f.

The presence of the pressure jump penalty parameter δ which must be specified by the user is a drawback (shared by most FE stabilization methods) of the method which must be carefully considered; for example, while the method is stable for arbitrarily large δ:s, choosing too large values relative to h, or the local mesh size in a highly non-uniform mesh, can change the physics of the problem – essentially enforcing a constant-pressure solution – so much that it leads to strange, non-physical optimized designs; see Fig. 3.

The issue of selecting the stabilization parameter for this kind of method was discussed by Norburn and Silvester [6]. They noted that if a good value of δ is found on a sufficiently fine mesh, then this value is likely to work on finer meshes as well. This hypothesis is corroborated by our numerical experiments. As for the selection of a good value to start with, it was suggested [6] to minimize the condition number of the Schur complement of the system matrix. A somewhat indirect way of doing this is to try to solve the linear system with an iterative linear solver and adjust δ by looking at the number of iterations required to achieve a certain accuracy; see Fig. 3.

Kechkar and Silvester [5] proposed a local version of our method applicable to meshes formed from macroelements. For such meshes, the sum in (7) can be restricted to the edges/faces internal to the macroelements. Theoretically [5, Theorem 3.1], this method is less sensitive to "large" values of δ, but a drawback is the existence of a critical lower value $\delta_0 > 0$ below which the method becomes unstable. This is in contrast to the global method used here which is stable for any $\delta > 0$.

Remark. Many research papers (see Introduction) on TO with Stokes or Navier–Stokes flow use stabilization methods to allow for equal-order approximation of velocity and hydrostatic pressure. However, since the hydrostatic pressure is only an auxiliary variable (the actual pressure on a surface element with normal $\boldsymbol{n}$ is more accurately described by $\|\boldsymbol{\sigma}\boldsymbol{n}\|$, where $\boldsymbol{\sigma} = -p\boldsymbol{I} + 2\mu\boldsymbol{\varepsilon}(\boldsymbol{u})$) it does not seem necessary in general to require that it be approximated to the same accuracy as the velocity for problems of the type studied herein (Fig. 5 shows that there is at least one case in which one can get a correct design even with a highly unstable approximation of the hydrostatic pressure). The main appeal of equal-order methods would instead seem to be ease of implementation, though a detailed numerical comparison with the method studied herein would make for interesting future work. □

3.1. Finite element convergence

As for the relation between the design problem (4) and its FE-discretized version (5) we show here, inspired by [2], that every sequence of FE-solutions admits a subsequence converging (weakly* as $h \to 0$) to a solution to

the original problem (4). This implies that every sequence of FE-solutions converges to the set of solutions to (4), and if this problem happens to have a unique solution that the entire sequence converges to this solution.

For the convergence proof we make use of the following lemma:

Lemma 1. *Let* $\{(\boldsymbol{u}_h, p_h)\}$ *be a sequence of solutions to the FE-problem* (6) *converging weakly to a limit* $(\boldsymbol{u}, p)$. *Then*

$$(q, div\, \boldsymbol{u}) = \lim_{h\to 0} [(q_h, div\, \boldsymbol{u}_h) + J(p_h, q_h)] = 0, \quad \forall q \in L^2(\Omega). \tag{8}$$

Proof. Since L_h is dense in $L^2(\Omega)$ there exists, by definition, for every $q \in L^2(\Omega)$ and $\varepsilon > 0$ a $q_h = q_h(q) \in L_h$ such that $\|q - q_h\|_{L^2(\Omega)} < \varepsilon$ for small enough h. Furthermore, since $H^1(\Omega)$ is dense in $L^2(\Omega)$, there exists for every $q \in L^2(\Omega)$ and $\varepsilon > 0$ a function $\tilde{q} = \tilde{q}(q) \in H^1(\Omega)$ such that $\|q - \tilde{q}\|_{L^2(\Omega)} < \varepsilon$. Combining these two facts we find that for every $\varepsilon > 0$,

$$\|q_h - \tilde{q}\|_{L^2(\Omega)} = \|q_h - \tilde{q} + q - q\|_{L^2(\Omega)} \le \|q_h - q\|_{L^2(\Omega)} + \|q - \tilde{q}\|_{L^2(\Omega)} \le \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon \tag{9}$$

for h small enough.

Choosing $\boldsymbol{v} = \boldsymbol{0}$ in (6) gives

$$(q_h, \operatorname{div} \boldsymbol{u}_h) + J(p_h, q_h) = 0, \quad \forall q_h \in L_h.$$

Then for every $q \in L^2(\Omega)$ and $\varepsilon > 0$, we have, for q_h such that $q_h \to q$ and small enough h,

$$\begin{aligned} |(q, \operatorname{div} \boldsymbol{u})| &= |(q, \operatorname{div} \boldsymbol{u}) - (q_h, \operatorname{div} \boldsymbol{u}_h) - J(p_h, q_h)| \le \\ &|(q, \operatorname{div} \boldsymbol{u} - \operatorname{div} \boldsymbol{u}_h) + (q - q_h, \operatorname{div} \boldsymbol{u}_h)| + |J(p_h, q_h)| \le \frac{\varepsilon}{3} + \frac{\varepsilon}{3} + |J(p_h, q_h)|, \end{aligned} \tag{10}$$

where the second inequality follows from the weak convergence of $\boldsymbol{u}_h$ to $\boldsymbol{u}$ and the strong convergence of q_h to q together with the boundedness of $\{\operatorname{div} \boldsymbol{u}_h\}$.

To bound the jump penalty term in (10) we write it as

$$|J(p_h, q_h)| = |J(p_h, q_h - \tilde{q}) + J(p_h, \tilde{q})| = |J(p_h, q_h - \tilde{q})|,$$

where $J(p_h, \tilde{q}) = 0$ since $\tilde{q} \in H^1(\Omega)$ can have no jumps. Using Cauchy–Schwartz then gives

$$\begin{aligned} |J(p_h, q_h - \tilde{q})| &\le \delta \sum_K h_{\partial K}^{1/2} \|p_h\|_{L^2(\partial K)} h_{\partial K}^{1/2} \|[q_h - \tilde{q}]\|_{L^2(\partial K)} \le \\ &\delta \left(\sum_K h_{\partial K} \|p_h\|^2_{L^2(\partial K)} \right)^{1/2} \left(\sum_K h_{\partial K} \|[q_h - \tilde{q}]\|^2_{L^2(\partial K)} \right)^{1/2}. \end{aligned} \tag{11}$$

A scaled trace inequality [18, (10.3.8)] applied to each edge/face, together with the shape regularity of the mesh (allowing us to bound the element diameter in terms of $h_{\partial K}$) and the fact that p_h and q_h are element-wise constant give the estimates

$$\begin{aligned} \|[p_h]\|^2_{L^2(\partial K)} &\le c_K h_{\partial K}^{-1} \sum_{\tilde{K} \in \mathcal{T}_{\partial K}} \|p_h\|^2_{L^2(\tilde{K})} \\ \|[q_h - \tilde{q}]\|^2_{L^2(\partial K)} &\le c_K h_{\partial K}^{-1} \sum_{\tilde{K} \in \mathcal{T}_{\partial K}} \left(\|q_h - \tilde{q}\|^2_{L^2(\tilde{K})} + h_{\partial K} \|\nabla \tilde{q}\|^2_{L^2(\tilde{K})} \right), \end{aligned} \tag{12}$$

where $\mathcal{T}_{\partial K}$ consists of K and the elements sharing an edge/face with K and the constant c_K depends on the shape and number of edges/faces of K. Substituting in (11) gives

$$\begin{aligned} &|J(p_h, q_h)| = |J(p_h, q_h - \tilde{q})| \le \\ &\delta \left(\sum_K c_K \sum_{\tilde{K} \in \mathcal{T}_{\partial K}} \|p_h\|^2_{L^2(\tilde{K})} \right)^{1/2} \left(\sum_K c_K \sum_{\tilde{K} \in \mathcal{T}_{\partial K}} \left(\|q_h - \tilde{q}\|^2_{L^2(\tilde{K})} + h^2_{\partial K} \|\nabla \tilde{q}\|^2_{L^2(\tilde{K})} \right) \right)^{1/2} \le \\ &\delta c \|p_h\|_{L^2(\Omega)} \left(\|q_h - \tilde{q}\|^2_{L^2(\Omega)} + h^2 \|\nabla \tilde{q}\|^2_{L^2(\Omega)} \right)^{1/2} \end{aligned}$$

for a constant c. Since $\{p_h\}$ is a convergent, hence bounded sequence in $L^2(\Omega)$; q_h tends strongly to $\tilde{q}$ in $L^2(\Omega)$ according to (9); and $\|\nabla\tilde{q}\|_{L^2(\Omega)}$ is bounded since $\tilde{q} \in H^1(\Omega)$, it follows that $|J(p_h, q_h)|$ tends to zero as $h \to 0$. Returning to (10) we see then that $|(q, \operatorname{div} \boldsymbol{u})| \leq \varepsilon$ for small enough h. □

We can now prove the following:

Theorem 2. *Every sequence $\{\rho_h^*\}$ of solutions to (5) has a subsequence converging weakly* in $L^\infty(\Omega)$ to a solution ρ^* to (4).*

Proof. Let $\{\rho_h\}$ be a sequence of feasible points in (5). Since $\rho_h \in \mathcal{H}_h \subset \mathcal{H}$ for all h and the latter set is weakly* compact in $L^\infty(\Omega)$ we can extract a subsequence converging weakly* to some $\rho \in \mathcal{H}$. From [7, Theorem 2] (which is valid also in our setting since $\alpha(\rho) \geq 0$) we get, denoting the left-hand side in (6) by $B_h(\cdot, \cdot)$ and using the non-negativity of $J(q, q)$, the existence of $\gamma > 0$ such that

$$\gamma(\|\boldsymbol{u}_h\|_{H^1(\Omega)} + \|p_h\|_{L^2(\Omega)}) \leq \sup_{(\boldsymbol{v},q)\in \boldsymbol{V}_h\times L_h} \frac{B_h((\boldsymbol{u}_h, p_h), (\boldsymbol{v}, q))}{(\|\boldsymbol{v}\|^2_{H^1(\Omega)} + \|q\|^2_{L^2(\Omega)})^{1/2}} \leq \sup_{(\boldsymbol{v},q)\in \boldsymbol{V}_h\times L_h} \frac{\ell(\boldsymbol{v})}{\|\boldsymbol{v}\|_{\boldsymbol{H}^1(\Omega)}} = \sup_{\boldsymbol{v}\in \boldsymbol{V}_h} \frac{\ell(\boldsymbol{v})}{\|\boldsymbol{v}\|_{\boldsymbol{H}^1(\Omega)}} < \infty, \tag{13}$$

where the last inequality follows from the continuity, hence boundedness of ℓ. This shows that any sequence of FE-solutions $(\boldsymbol{u}_h, p_h) = (\boldsymbol{u}_h(\rho_h), p_h(\rho_h))$ is bounded, hence admits a subsequence converging weakly to a limit in $\boldsymbol{V} \times L^2(\Omega)$.

Now consider a weakly* convergent sequence of solutions $\{\rho_h^*\}$, with limit ρ^*, to the FE-discretized design problem (5) such that $\{(\boldsymbol{u}_h(\rho_h^*), p_h(\rho_h^*))\}$ converges weakly. There holds, by definition, that

$$\ell(\boldsymbol{u}_h(\rho_h^*)) \geq \ell(\boldsymbol{u}_h(\rho_h)) \quad \forall \rho_h \in \mathcal{H}_h. \tag{14}$$

The density of $\mathcal{H}_h$ implies that there exists for every $\rho \in \mathcal{H}$ a $\rho_h(\rho) \in \mathcal{H}_h$ tending strongly in $L^2(\Omega)$ to ρ. Substitution in (14) gives

$$\ell(\boldsymbol{u}_h(\rho_h^*)) \geq \ell(\boldsymbol{u}_h(\rho_h(\rho))) \quad \forall \rho \in \mathcal{H}. \tag{15}$$

We now proceed by first showing that the limit of the right-hand side is $\ell(\boldsymbol{u}(\rho))$ and then that the limit of the left-hand side is bounded above by $\ell(\boldsymbol{u}(\rho^*))$.

The velocity $\boldsymbol{u}_h = \boldsymbol{u}_h(\rho_h(\rho))$ in (15) is part of the solution to

$$a(\rho_h(\rho); \boldsymbol{u}_h, \boldsymbol{v}_h) + (-p_h, \operatorname{div} \boldsymbol{v}_h) + (q_h, \operatorname{div} \boldsymbol{u}_h) + J(p_h, q_h) = \ell(\boldsymbol{v}_h), \quad \forall (\boldsymbol{v}_h, q_h) \in \boldsymbol{V}_h \times L_h. \tag{16}$$

The density of $\boldsymbol{V}_h$ and L_h implies that there are functions $\boldsymbol{v}_h(\boldsymbol{v}) \in \boldsymbol{V}_h$ and $q_h(q) \in L_h$ tending strongly to $\boldsymbol{v}$ and q, respectively. Substitution into (16) gives, for every $(\boldsymbol{v}, q) \in \boldsymbol{V} \times L^2(\Omega)$,

$$a(\rho_h(\rho); \boldsymbol{u}_h, \boldsymbol{v}_h(\boldsymbol{v})) + (-p_h, \operatorname{div} \boldsymbol{v}_h(\boldsymbol{v})) + (q_h(q), \operatorname{div} \boldsymbol{u}_h) + J(p_h, q_h(q)) = \ell(\boldsymbol{v}_h(\boldsymbol{v})). \tag{17}$$

To see that the limiting state $(\boldsymbol{u}, p)$ satisfies (1), so that $\boldsymbol{u} = \boldsymbol{u}(\rho)$, we now consider an arbitrary subsequence $\{\rho_{\tilde{h}}(\rho)\}$ such that $\rho_{\tilde{h}}(\rho)$ tends to ρ pointwise a.e. (as allowed by [19, Theorem 4.9]) and such that $\{(\boldsymbol{u}_{\tilde{h}}(\rho_{\tilde{h}}(\rho)), p_{\tilde{h}}(\rho_{\tilde{h}}(\rho)))\}$ converges weakly and $\boldsymbol{u}_{\tilde{h}}(\rho_{\tilde{h}}(\rho))$ converges strongly in $\boldsymbol{L}^2(\Omega)$ (as allowed by the Rellich–Kondrachov theorem [19, Theorem 9.16]). Since we have already established (8), the main difficulty is the convergence of $a(\rho_h(\rho); \boldsymbol{u}_h, \boldsymbol{v}_h(\boldsymbol{v}))$ to $a(\rho; \boldsymbol{u}, \boldsymbol{v})$. To this end, consider the design-dependent term in the definition (2) of a (the other term is treated similarly, and more simply since there is no design-dependence). Adding and subtracting terms give (omitting arguments to ease notation)

$$\left|\int_\Omega \alpha(\rho_{\tilde{h}})\boldsymbol{u}_{\tilde{h}} : \boldsymbol{v}_{\tilde{h}}\, \mathrm{d}V - \int_\Omega \alpha(\rho)\boldsymbol{u} : \boldsymbol{v}\, \mathrm{d}V\right| =$$

$$\left|\int_\Omega \alpha(\rho_{\tilde{h}})(\boldsymbol{u}_{\tilde{h}} - \boldsymbol{u}) : \boldsymbol{v}_{\tilde{h}}\, \mathrm{d}V + \int_\Omega \alpha(\rho_{\tilde{h}})\boldsymbol{u} : (\boldsymbol{v}_{\tilde{h}} - \boldsymbol{v})\, \mathrm{d}V + \int_\Omega (\alpha(\rho_{\tilde{h}}) - \alpha(\rho))\boldsymbol{u} : \boldsymbol{v}\, \mathrm{d}V\right| \leq$$

$$\overline{\alpha}\|\boldsymbol{u}_{\tilde{h}} - \boldsymbol{u}\|_{\boldsymbol{L}^2(\Omega)}\|\boldsymbol{v}_{\tilde{h}}\|_{\boldsymbol{L}^2(\Omega)} + \overline{\alpha}\|\boldsymbol{u}\|_{\boldsymbol{L}^2(\Omega)}\|\boldsymbol{v}_{\tilde{h}} - \boldsymbol{v}\|_{\boldsymbol{L}^2(\Omega)} + \int_\Omega |(\alpha(\rho_{\tilde{h}}) - \alpha(\rho))\boldsymbol{u} : \boldsymbol{v} - 0|\, \mathrm{d}V$$

using Cauchy–Schwartz. Since $\overline{\alpha} < \infty$, the first two terms tend to zero by the strong convergence of $\boldsymbol{u}_{\tilde{h}}$ and $\boldsymbol{v}_{\tilde{h}}$. As for the third term we note that $(\alpha(\rho_{\tilde{h}}) - \alpha(\rho))\boldsymbol{u} : \boldsymbol{v}$ tends to zero point-wise a.e. due to the continuity of $\alpha(\cdot)$ and the almost everywhere boundedness of $\boldsymbol{u} : \boldsymbol{v}$. Since we also have

$$|(\alpha(\rho_{\tilde{h}}) - \alpha(\rho))\boldsymbol{u} : \boldsymbol{v}| \leq 2\overline{\alpha}|\boldsymbol{u} : \boldsymbol{v}| \in L^1(\Omega), \tag{18}$$

convergence to zero of the third term follows from Lebesque's dominated convergence theorem. Recalling (8) and using the continuity of ℓ we find that the limiting state $(\boldsymbol{u}, p)$ of the subsequence of solutions to (17) satisfies

$$a(\rho; \boldsymbol{u}, \boldsymbol{v}) + (-p, \operatorname{div} \boldsymbol{v}) + (q, \operatorname{div} \boldsymbol{u}) = \ell(\boldsymbol{v}), \quad (\boldsymbol{v}, q) \in \boldsymbol{V} \times L^2(\Omega); \tag{19}$$

i.e., $(\boldsymbol{u}, p) = (\boldsymbol{u}(\rho), p(\rho))$. Since our subsequence was arbitrary and (19) has a unique solution it follows that the whole sequence $\{\boldsymbol{u}_h(\rho_h(\rho))\}$ converges weakly to $\boldsymbol{u}(\rho)$. Then the continuity of ℓ and (15) gives

$$\limsup_{h\to 0} \ell(\boldsymbol{u}_h(\rho_h^*)) \geq \ell(\boldsymbol{u}(\rho)), \quad \forall \rho \in \mathcal{H}. \tag{20}$$

To conclude the proof we need to show that the left-hand side in (20) is bounded above by $\ell(\boldsymbol{u}(\rho^*))$; i.e. that $\ell(\boldsymbol{u}_h(\cdot))$ is weakly* sequentially upper semi-continuous. To this end we first choose $\boldsymbol{v} = \boldsymbol{u}_h = \boldsymbol{u}_h(\rho_h^*)$ and $q = p_h = p_h(\rho_h^*)$ in (6) to get

$$a(\rho_h^*; \boldsymbol{u}_h, \boldsymbol{u}_h) + (-p_h, \operatorname{div} \boldsymbol{u}_h) + (p_h, \operatorname{div} \boldsymbol{u}_h) + J(p_h, p_h) = \ell(\boldsymbol{u}_h),$$

and thus

$$\ell(\boldsymbol{u}_h) = -\ell(\boldsymbol{u}_h) + 2\ell(\boldsymbol{u}_h) = \underbrace{-a(\rho_h^*, \boldsymbol{u}_h, \boldsymbol{u}_h) + 2\ell(\boldsymbol{u}_h)}_{-2\Pi(\rho_h, \boldsymbol{u}_h)} - J(p_h, p_h). \tag{21}$$

As shown in [2, Theorem 3.1], Π is sequentially lower-semi-continuous, hence $-\Pi$ is sequentially upper semi-continuous, for weak*$\times$ weak convergence. Using this fact and the non-negativity of $J(p_h, p_h)$ (21) gives

$$\limsup_{h\to 0} \ell(\boldsymbol{u}_h) \leq -2\Pi(\rho^*, \boldsymbol{u}).$$

Since, by Lemma 1, the limiting velocity $\boldsymbol{u}$ is divergence-free we get

$$\limsup_{h\to 0} \ell(\boldsymbol{u}_h(\rho_h^*)) \leq -2\Pi(\rho^*, \boldsymbol{u}) \leq -2 \inf_{\boldsymbol{v}\in\boldsymbol{V}_{div}} \Pi(\rho^*, \boldsymbol{v}) = \ell(\boldsymbol{u}(\rho^*)),$$

where $\boldsymbol{V}_{div} = \{\boldsymbol{v} \in \boldsymbol{V} \mid (\operatorname{div} \boldsymbol{u}, q) = 0, \forall q \in L^2(\Omega)\}$, and thus from (20) that

$$\ell(\boldsymbol{u}(\rho^*)) \geq \limsup_{h\to 0} \ell(\boldsymbol{u}_h(\rho_h^*)) \geq \ell(\boldsymbol{u}(\rho)), \quad \forall \rho \in \mathcal{H};$$

i.e. ρ^* solves (4). □

Remark 3. Papadopoulos and Süli [17, Theorem 4] recently showed that if a local isolated minimum exists, there is a sequence of solutions converging to it in a stronger sense than showed here. Their proof assumed a stable FE method but can probably be modified to hold also for stabilized methods. □

3.2. Matrix problem

The matrix problem corresponding to the FE-discretized state problem (6) is

$$\begin{pmatrix} \boldsymbol{M}(\rho_h) + \boldsymbol{K} & \boldsymbol{G} \\ \boldsymbol{G}^{\mathsf{T}} & -\boldsymbol{K}_p \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{p} \end{pmatrix} = \begin{pmatrix} \boldsymbol{f} \\ \boldsymbol{0} \end{pmatrix}, \tag{22}$$

where the (sparse, symmetric and positive semi-definite) matrix $\boldsymbol{K}_p$, arising from the pressure-jump penalty term $J(p, q)$, is in our implementation assembled by an outer loop over the elements and an inner loop over all n_e

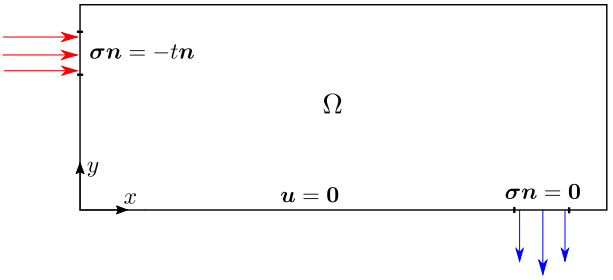

Fig. 1. Design domain with boundary conditions. Inflow on the right. Outflow on the bottom left. The size of the domain is $3 \times 1 \times 1$, $t = 1$, and $\boldsymbol{n}$ is the outward normal.

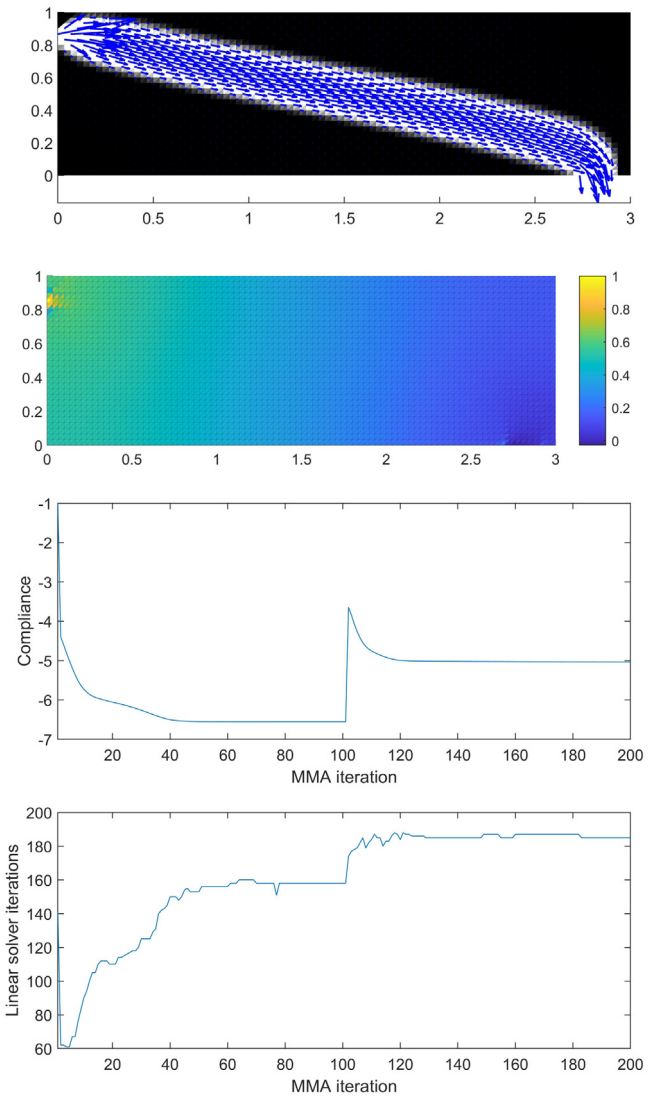

Fig. 2. Top: An optimized design with superimposed flow field. Mesh with 90×30 P1/P0 elements. Top middle: Normalized hydrostatic pressure field. Bottom middle: Normalized convergence history (since MMA solves minimization problems we maximize by minimizing the negative of the compliance). The jump at iteration 100 is due to the change of q from 0.01 to 0.1. Bottom: Number of CG iterations.

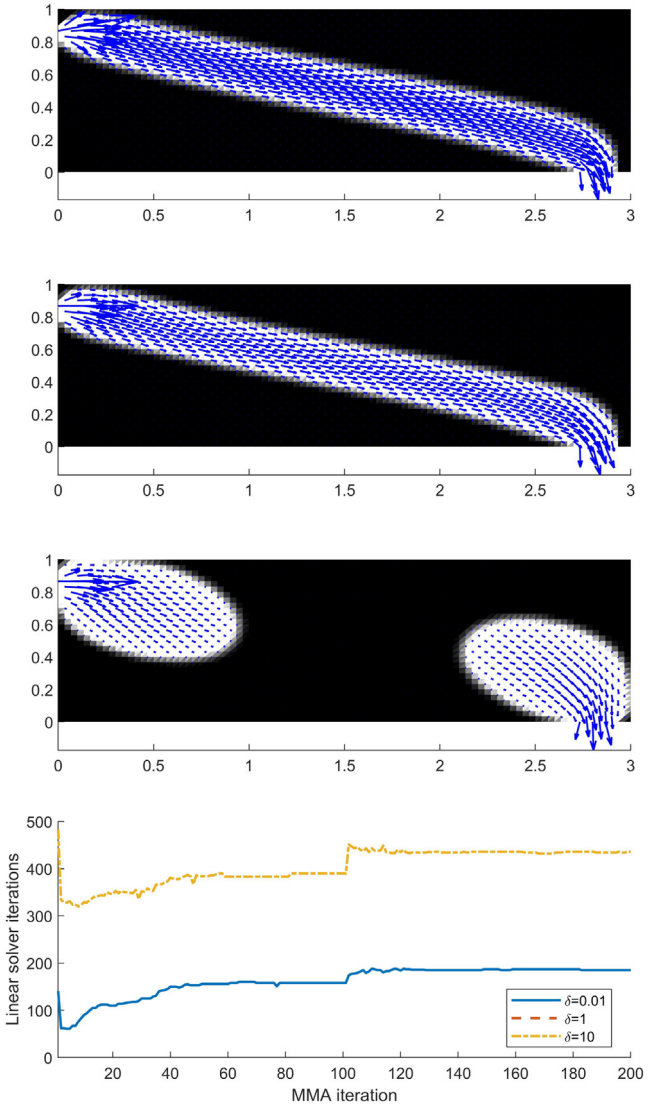

Fig. 3. From the top and down: Optimized designs with $\delta = 0.01$, 1 and 10. Only in the last case do we get a non-physical result. Bottom plot shows the number of linear solver iterations in each case. The graphs for $\delta = 0.01$ and 1 are identical and thus visually indistinguishable.

internal edges/faces. Formally this can be written as

$$\boldsymbol{K}_p = \delta \sum_{e=1}^{m} \sum_{i=1}^{n_e} \boldsymbol{c}_{ei}^{\mathsf{T}} h_{\partial K} \int_{\partial K_i \setminus \partial \Omega} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \mathrm{d}A \, \boldsymbol{c}_{ei},$$

where $\boldsymbol{c}_{ei} \in \mathbb{R}^m$ contains zeros expect for a one in position e and a minus one in the position corresponding to the neighbour across the i:th edge/face. In practice we assemble the matrix by first constructing it in triplet format and then converting it to compressed sparse column format [20]. Setting up this matrix requires information about the immediate neighbours of all elements in the mesh, data which is readily obtainable from many codes (Lo [21, p. 33] suggests algorithms for triangular and tetrahedral meshes, which are also readily adapted to quadrilateral and hexahedral meshes).

Regarding (22) we note that it is only $\boldsymbol{M}(\rho_h)$ (stemming from the first term in (2)) which depends on the design, hence all the other matrices need only be computed once at the beginning of the optimization process. The fact that

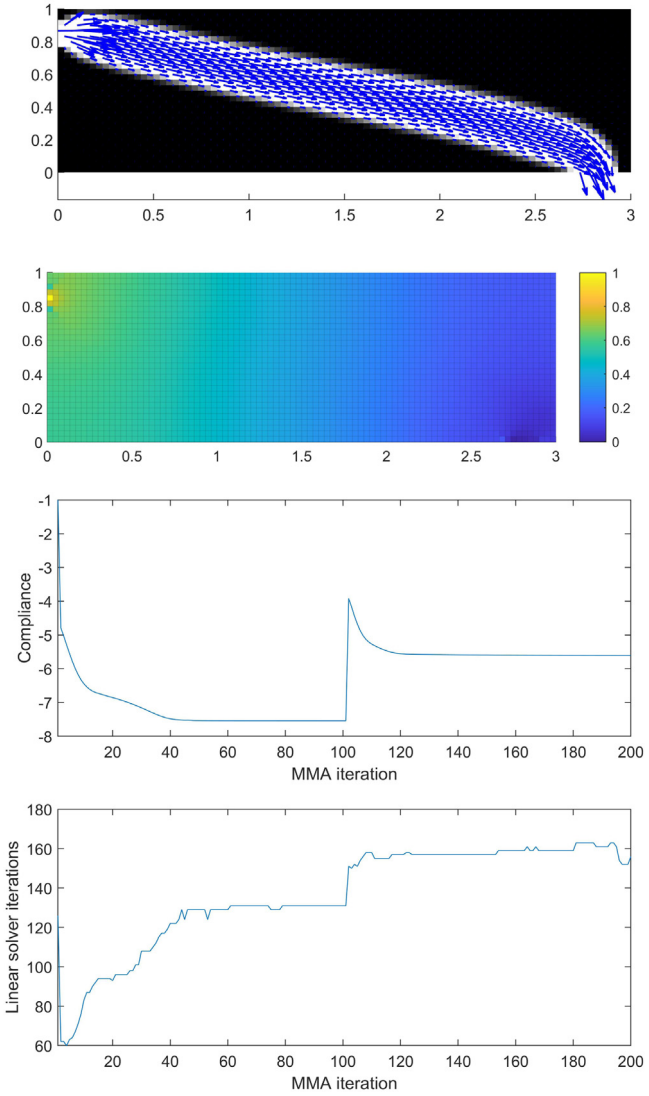

Fig. 4. Data for a 90 × 30 Q1–P0 mesh with $\delta = 0.01$.

$\boldsymbol{K}_p$ does not depend on the design implies that the expressions for the derivatives of the objective function in (5) are the same as in [2] since we are using an element-wise constant approximation of the design.

4. Numerical examples

The overall framework for solving (5) is implemented in Matlab (R2021a) with The Method of Moving Asymptotes (MMA) [22] as optimization solver. The MMA-solver parameters are set to default values, except `asyinit` $= 0.1$, `asyincr` $= 1.1$ and `asydecr` $= 0.3$. The maximum number of MMA iterations for a given set of problem parameters was set to 100.

Following [2], the linear system (22) is solved by applying a preconditioned conjugate gradient (CG) algorithm on the symmetric positive definite system for the pressure $\boldsymbol{p}$ obtained by eliminating the velocities from (22). (The 2D-examples below can actually be solved faster by a direct linear solver applied directly to (22), but for practical-scale problems iterative solvers are necessary, and so it is of interest to see how they perform already on small problems.) This approach was implemented using the code AMGCL v.1.4.0 [23] following [24] with OpenMP for parallelization. In the preconditioning step (c.f. [24, (8a), (8b)]), the velocity was solved for using algebraic multigrid

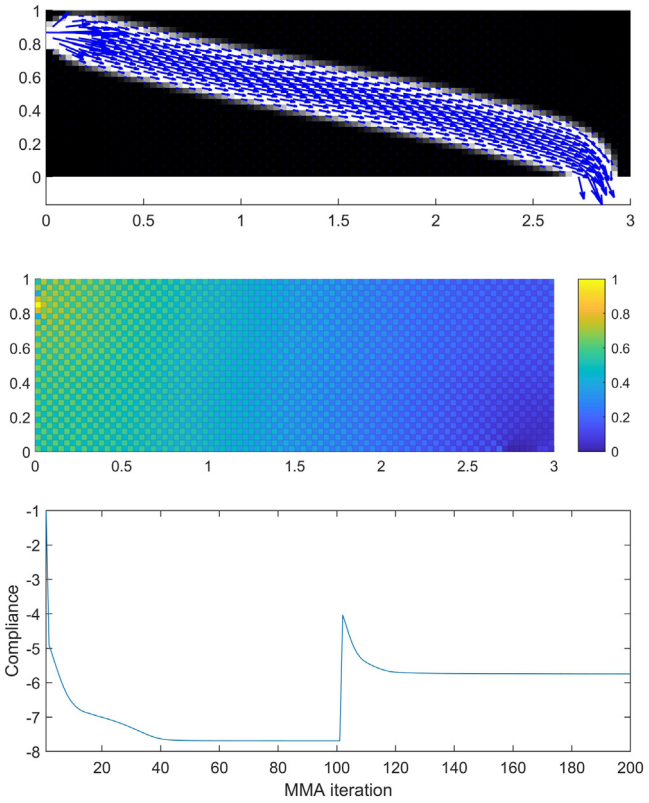

Fig. 5. Data for a 90 x 30 Q1–P0 mesh with $\delta = 0$ and a direct linear solver.

with smoothed aggregation and an incomplete LU factorization (`ilu0`) for coarsening and relaxation, respectively. Near null-space vectors for the velocity block were provided. For the pressure we used a diagonal preconditioner (`spai0`). The stopping criterion for the linear solver is of the form $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\| < 10^{-10}\|\boldsymbol{b}\|$. The mixed-precision approach suggested in [24] to speed up the linear solves did not work satisfactorily in our examples, so double precision was used for both solvers in the preconditioning step.

Aside from mesh data, the design problem (5) has five user-specified parameters. We set $\overline{\alpha} = 10^4$, $\mu = 1$, $\gamma = 0.7$, and vary the pressure jump penalty parameter δ in (7) in the examples below. To reduce the risk of obtaining poor local minima/stationary points we use continuation on the penalty parameter q in (3), starting with $q = 0.01$ and then at MMA iteration 100 switching to $q = 0.1$ (in fact, starting with $q = 0.1$ in the examples below yielded designs with slightly curved channels which are clearly not globally optimal for Stokes flow).

The numerical examples are based on a version of the pipe bend problem from [2, Section 4.3]. The setup is shown in Fig. 1. The inlet is located at $x = 0$, $0.8 \leq y \leq 0.9$ and the outlet at $2.7 \leq x \leq 2.9$, $y = 0$.

In the first example, shown in Fig. 2, we use conforming P1–P0 triangular elements, i.e. the velocity is approximated as a continuous, element-wise linear function and the hydrostatic pressure as element-wise constant. As is well-known, this element locks without stabilization; this is also observed here, where setting $\delta = 0$ makes (22) impossible to solve. However, setting $\delta = 0.01$ yields smooth convergence and a sensible final design, which appears likely to be a global optimum, despite some oscillations in the hydrostatic pressure field around the inflow in the pressure plot of Fig. 2. The runtime for this example was approximately 1 min on a six-core laptop running at around 3.7 GHz.

As mentioned above, the presence of the pressure jump penalty parameter δ which must be specified by the user is a drawback of the method. Fortunately, it appears that the problem is not very sensitive to this choice and that poor choices can be detected easily at the beginning of the optimization process. Fig. 3 shows some designs for different δ. We note in particular that the solution process and the final design are essentially unaffected by increasing δ by a factor 100 from 0.01 to 1. Looking at the number of CG iterations, one notices that they increase

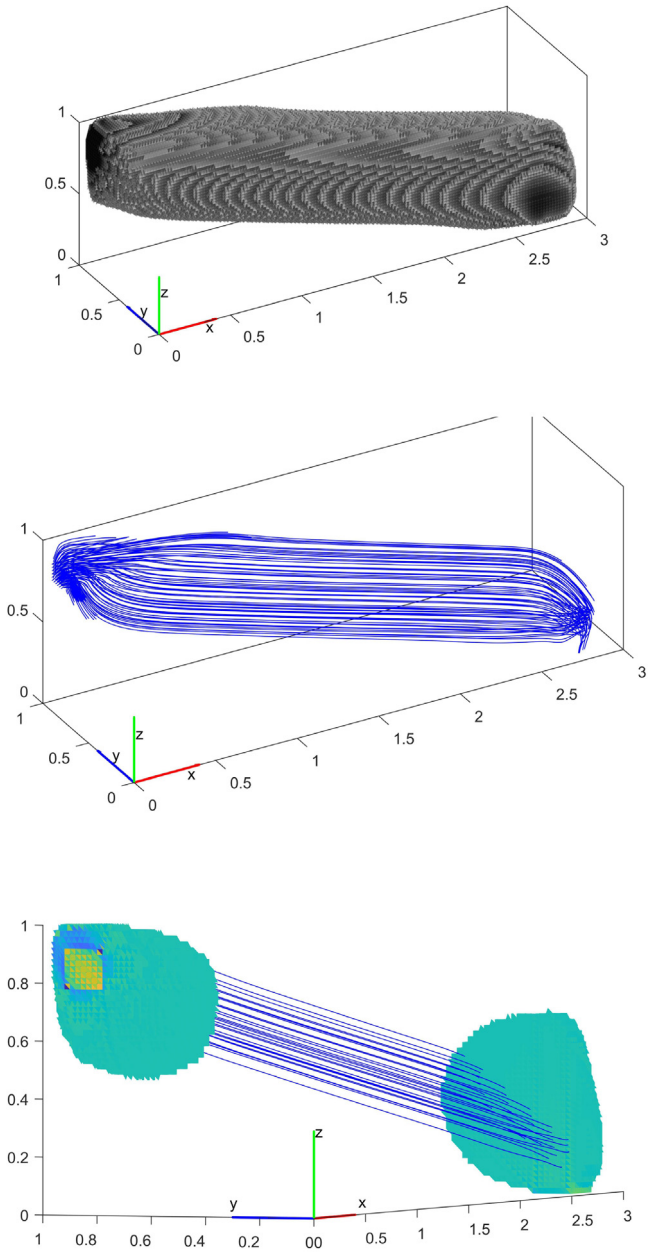

Fig. 6. From the top and down: Design ($1 - \rho_h$ threshold at 0.5), streamline plot of the flow field, and the hydrostatic pressure field (in a cut-up view showing regions around the inlet and outlet with some streamlines superimposed) for a tetrahedral P1–P0 mesh with $\delta = 0.01$.

significantly when going from $\delta = 1$ to $\delta = 10$. Since this difference is seen already for the initial design, this suggests that a strategy for selecting δ can be based on the number of CG iterations taken in the beginning of the optimization process.

Fig. 4 shows results when using 4-noded Q1–P0 elements with a bi-linear approximation of the velocity. This element is popular in the TO-literature, and both quad and hex-meshes are frequently used in the industry, so it is nice to see that the stabilization method works well for this element. We remark that, although we do not recommend it, it is actually also possible to use the Q1–P0 element with $\delta = 0$ and a direct linear solver (Ma57 [25]); see Fig. 5. Despite a pressure field which exhibits checkerboards in the entire design domain, we achieve smooth convergence to a very reasonable design. However, our iterative linear solver fails to converge at all.

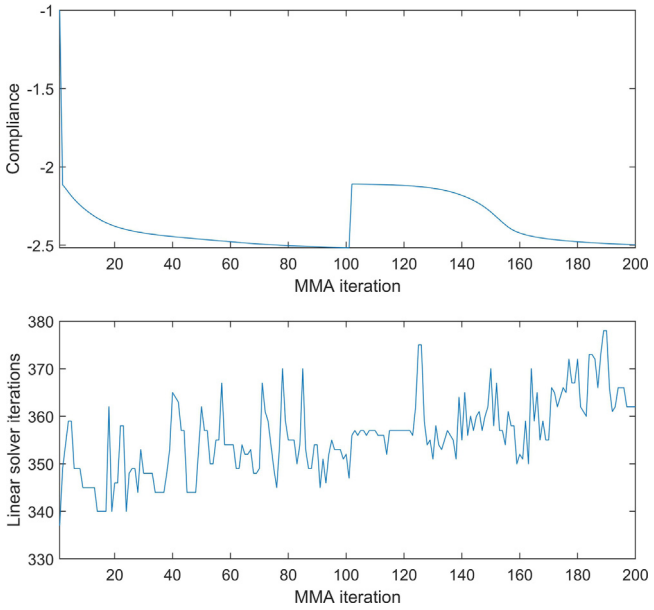

Fig. 7. Data for a tetrahedral P1–P0 mesh with $\delta = 0.01$.

Figs. 6 and 7 show results from a 3D version of the setup seen in Fig. 1 using 4-noded tetrahedral P1–P0 elements. The inflow region is defined by $x = 0$, $2.4 \leq y \leq 2.7$ and $0.8 \leq z \leq 0.9$, and the outflow by $2.7 \leq x \leq 2.9$, $0.1 \leq y \leq 0.2$ and $z = 0$. The mesh was obtained by splitting each of the elements in a $150 \times 50 \times 50$ 8-noded hexahedral mesh into six tetrahedra, giving a total of 2250000 tetrahedral elements and a total number of (velocity and pressure) degrees of freedom (DOFs) of around $3.4 \cdot 10^6$. The optimized design in Fig. 6 has a straight channel connecting inflow to outflow, suggesting global optimality. Again we see smooth convergence in the optimization process (Fig. 7) and a reasonable amount of CG iterations, despite some oscillations in the pressure at the inflow, and to a lesser degree at the outflow, seen in Fig. 6 (bottom). The runtime for this example was roughly 6 h on one compute node on a cluster equipped with two 2 Intel Xeon Gold 6130 processors, giving a total of 32 physical cores. As expected, the biggest part of the runtime was spent on solving the linear system (22).

Fig. 8 shows data for two different meshes using 8-noded Q1–P0 elements with element-wise bi-linear velocity and element-wise constant pressure. The runtime for the $150 \times 50 \times 50$ mesh was roughly 5.4 h, dominated by the solution of the linear system which took around 90 s to solve once. The runtime for the $210 \times 70 \times 70$ mesh, with roughly 4 million velocity and pressure DOFs, was around 14 h, and solving the linear system once took around 230 s. We remark that the ratio between the number of nodes and the number of faces is much larger in the tetrahedral mesh used in Fig. 6 than in the hexahedral meshes, so $\boldsymbol{K}_p$ is much sparser for the latter meshes.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The 3D computations were enabled by the Swedish National Infrastructure for Computing (SNIC) at the National Super Computer Centre partially funded by the Swedish Research Council through grant no. 2018-05973. The research was partially funded by the Swedish Research Council, grant no. 2019-0461, and CENIIT, Sweden. We thank the anonymous reviewers for very useful comments. Thanks to Krister Svanberg for sharing his MMA code.

Replications of results

A complete description of the mathematical model and values for all its parameters are provided in the article.

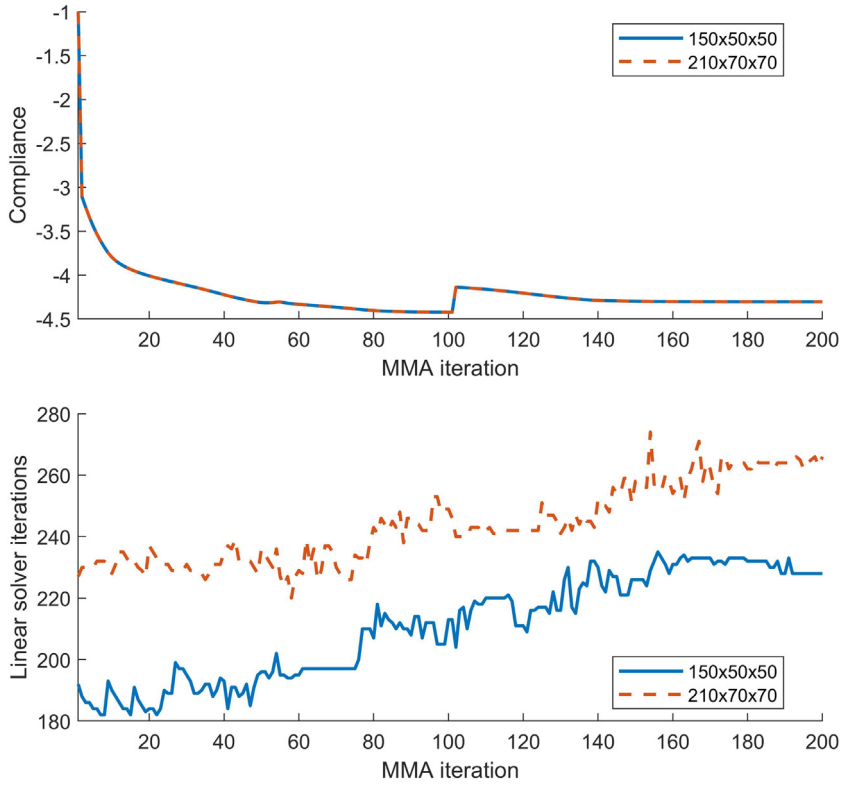

Fig. 8. Data for two different meshes with hexahedral Q1–P0 elements and $\delta = 0.01$.

References

[1] J. Alexandersen, C.S. Andreasen, A review of topology optimisation for fluid-based problems, Fluids 5 (1) (2020) 29.
[2] T. Borrvall, J. Petersson, Topology optimization of fluids in Stokes flow, Internat. J. Numer. Methods Fluids 41 (1) (2003) 77–107.
[3] R.S. Falk, M.E. Morley, Equivalence of finite element methods for problems in elasticity, SIAM J. Numer. Anal. 27 (6) (1990) 1486–1505.
[4] T.J.R. Hughes, L.P. Franca, A new finite element formulation for computational fluid dynamics: VII. The Stokes problem with various well-posed boundary conditions: symmetric formulations that converge for all velocity/pressure spaces, Comput. Methods Appl. Mech. Engrg. 65 (1) (1987) 85–96.
[5] N. Kechkar, D. Silvester, Analysis of locally stabilized mixed finite element methods for the Stokes problem, Math. Comp. 58 (197) (1992) 1–10.
[6] S. Norburn, D. Silvester, Stabilised vs. stable mixed methods for incompressible flow, Comput. Methods Appl. Mech. Engrg. 166 (1–2) (1998) 131–141.
[7] E. Burman, P. Hansbo, A unified stabilized method for Stokes' and Darcy's equations, J. Comput. Appl. Math. 198 (1) (2007) 35–51.
[8] K.A. Mardal, X.-C. Tai, R. Winther, A robust finite element method for Darcy–Stokes flow, SIAM J. Numer. Anal. 40 (5) (2002) 1605–1631.
[9] R. Kouhia, R. Stenberg, A linear nonconforming finite element method for nearly incompressible elasticity and Stokes flow, Comput. Methods Appl. Mech. Engrg. 124 (3) (1995) 195–212.
[10] P.B. Bochev, C.R. Dohrmann, M.D. Gunzburger, Stabilization of low-order mixed finite elements for the Stokes equations, SIAM J. Numer. Anal. 44 (1) (2006) 82–101.
[11] J.K. Guest, J.H. Prévost, Topology optimization of creeping fluid flows using a Darcy–Stokes finite element, Internat. J. Numer. Methods Engrg. 66 (3) (2006) 461–484.
[12] C.S. Andreasen, A.R. Gersborg, Ole O. Sigmund, Topology optimization of microfluidic mixers, Internat. J. Numer. Methods Fluids 61 (5) (2009) 498–513.
[13] S. Kreissl, K. Maute, Levelset based fluid topology optimization using the extended finite element method, Struct. Multidiscip. Optim. 46 (3) (2012) 311–326.
[14] N. Aage, B.S. Lazarov, Parallel framework for topology optimization using the method of moving asymptotes, Struct. Multidiscip. Optim. 47 (4) (2013) 493–505.

[15] J. Alexandersen, O. Sigmund, N. Aage, Large scale three-dimensional topology optimisation of heat sinks cooled by natural convection, Int. J. Heat Mass Transfer 100 (2016) 876–891.
[16] K Yaji, S Yamasaki, S Tsushima, T Suzuki, Kikuo K Fujita, Topology optimization for the design of flow fields in a redox flow battery, Struct. Multidiscip. Optim. 57 (2) (2018) 535–546.
[17] I Papadopoulos, E Süli, Numerical analysis of a topology optimization problem for Stokes flow, 2021, arXiv:2102.10408.
[18] S.C. Brenner, L.R. Scott, The Mathematical Theory of Finite Element Methods, Springer, 2008.
[19] H. Brezis, Functional Analysis, Sobolev Spaces and Partial Differential Equations, Springer Science & Business Media, 2010.
[20] T. Davis, Direct Methods for Sparse Linear Systems, SIAM, 2006.
[21] D.S.H. Lo, Finite Element Mesh Generation, CRC Press, 2014.
[22] K. Svanberg, MMA and GCMMA, versions september 2007, 2007, URL https://people.kth.se/~krille/.
[23] D. Demidov, AMGCL: An efficient, flexible, and extensible algebraic multigrid implementation, Lobachevskii J. Math. 40 (5) (2019) 535–546.
[24] D. Demidov, L. Mu, B. Wang, Accelerating linear solvers for Stokes problems with C++ metaprogramming, J. Comput. Sci. 49 (2021).
[25] I. Duff, MA57—A code for the solution of sparse symmetric definite and indefinite systems, ACM Trans. Math. Software 30 (2004) 118–144.