Struct Multidisc Optim (2015) 51:801–811
DOI 10.1007/s00158-014-1176-x

RESEARCH PAPER

On Chebyshev's method for topology optimization of Stokes flows

Anton Evgrafov

Received: 13 April 2014 / Revised: 31 July 2014 / Accepted: 16 September 2014 / Published online: 16 January 2015

Abstract We present a locally cubically convergent algorithm for topology optimization of Stokes flows based on a Chebyshev's iteration globalized with Armijo linesearch. The characteristic features of the method include the low computational complexity of the search direction calculation, evaluation of the objective function and constraints needed in the linesearch procedure as well as their high order derivatives utilized for obtaining higher order rate of convergence. Both finite element and finite volumes discretizations of the algorithm are tested on the standard two-dimensional benchmark problems, in the case of finite elements both on structured and quasi-uniform unstructured meshes of quadrilaterals. The algorithm outperforms Newton's method in nearly all test cases. Finally, the finite element discretization of the algorithm is tested within a continuation/adaptive mesh refinement framework.

Keywords Topology optimization · Stokes flows · Chebyshev's algorithm

1 Introduction

Given a bounded Lipschitz domain $\Omega \subset R^d$, $d \in \{2, 3\}$, $\mathbf{f} \in [L^{6/5}(\Omega)]^d$, $\mathbf{u}_0 \in [H^{1/2}(\partial\Omega)]^d$ satisfying the compatibility condition $\int_{\partial\Omega} \mathbf{u}_0 \cdot \mathbf{n} = 0$, and a number $\gamma \in (0, 1)$ we consider the minimization problem introduced by Borrvall and Petersson (2003):

$$\begin{aligned} &\underset{(\mathbf{u},\rho)}{\text{minimize}} \quad J(\mathbf{u}, \rho), \\ &\text{subject to} \quad (\mathbf{u}, \rho) \in \mathscr{U}_{\text{ad}} \times \mathscr{D}_{\text{ad}}, \end{aligned} \tag{1}$$

where

$$\begin{aligned} J(\mathbf{u}, \rho) &= \frac{1}{2}\int_\Omega [|\nabla \mathbf{u}|^2 + \alpha(\rho)|\mathbf{u}|^2 - 2\mathbf{f}\cdot\mathbf{u}], \\ \mathscr{U}_{\text{ad}} &= \{\, \mathbf{u} \in [H^1(\Omega)]^d | \operatorname{div}\mathbf{u} = 0, T\mathbf{u} = \mathbf{u}_0 \,\}, \\ \mathscr{D}_{\text{ad}} &= \{\, \rho \in L^\infty(\Omega) | 0 \le \rho \le 1, \int_\Omega \rho = \gamma|\Omega| \,\}, \end{aligned} \tag{2}$$

$T : [H^1(\Omega)]^d \to [H^{1/2}(\partial\Omega)]^d$ is the trace operator, and $|\cdot|$ is used to denote the Euclidian norm on $\mathbb{R}^d$, Frobenius norm on $\mathbb{R}^{d\times d}$, and Lebesgue measure on measurable subsets of $\mathbb{R}^d$. A non-negative coefficient $\alpha : [0, 1] \to \mathbb{R}_+$ featuring in (2) has a physical meaning of inverse permeability of porous material and is typically taken to be $\alpha(\rho) = \bar{\alpha}q[(q + 1)/(\rho + q) - 1]$, for some large positive parameters $\bar{\alpha}$ (inverse permeability of the ersatz "solid" material) and q (design parametrization parameter related to the sharpness of the solid/flow interface); see (Borrvall and Petersson 2003).

Due to its simple structure and practical importance the problem has attracted significant attention in the community, see for example (Aage et al. 2008; Othmer 2008; Challis and Guest 2009) for various computational approaches to this and a related and even more practically important problem for Navier–Stokes flows.

A typical computation then consists of solving a sequence of problems (1) for increasing values of q, until a solution $(\mathbf{u}^*, \rho^*)$ is obtained with ρ^* being an approxi-

Electronic supplementary material The online version of this article (doi:10.1007/s00158-014-1176-x) contains supplementary material, which is available to authorized users.

A. Evgrafov (✉)
Department of Mathematical Sciences, Norwegian University of Science and Technology, N–7491 Trondheim, Norway
e-mail: anton.evgrafov@math.ntnu.no

mation to a characteristic function χ_{ω^*} of some subset $\omega^* \subset \Omega$. In this case, ω^* can be viewed as an optimal domain for a Stokes flow; see (Borrvall and Petersson 2003) for more details. The whole procedure relies upon enforcing the boundary conditions through penalties, the idea which may be traced at least to Babuška (1973). As a result, despite the common practice of keeping the penalty parameter $\bar{\alpha}$ fixed (though some authors also recommend relating it to the inverse of the Reynolds number in the case of Navier–Stokes problems (Kondoh et al. 2012)), this parameter should be increased as one refines the discretization of (1) in order to improve the accuracy of the computed solutions. We refer to Evgrafov (2005) for the discussion of the validity of the limiting process $\bar{\alpha} \to +\infty$ in the context of topology optimization.

Recently Carlsson et al. (2009) and Evgrafov (2014) proposed applying Newton's iteration embedded into a general globalized optimization framework, such as SQP, to a smoothed problem obtained by eliminating the design variables from (1). If such an elimination can be carried out computationally inexpensively and without ruining the sparsity of the problem, which is true for problems without global design regularization, the resulting practical algorithm attains fast local convergence due to the immediate availability of the second order information and its sparsity. Unlike the "1-step" or simultaneous analysis and design based approaches, the method operates in a smaller space, see Evgrafov (2014) for a discussion of advantages and disadvantages of the method. In fact, the complexity of the Newton's direction finding subproblem in such a method is virtually the same as the cost of computing a steepest descent direction in the traditional nested framework (Borrvall and Petersson 2003) when direct linear solvers are utilized; for iterative solvers drawing the comparison is a slightly more delicate issue owing to the differences in spectral properties of the matrices involved. The method, which is briefly reviewed in Section 2, demonstrates fast convergence, which is almost independent of the utilized PDE discretization or the mesh refinement level.

The main purpose of this note is to further improve the rate of convergence of the method proposed in Evgrafov (2014). Namely, the availability of the higher order derivatives of the objective function/constraints for very little computational effort allows us to apply Chebyshev's instead of Newton's algorithm, see Nechepurenko (1954), to the system of the first order necessary optimality conditions for a slight modification of (1) formulated in the state space. Thereby we obtain a method demonstrating third order local convergence, as may be expected (Nechepurenko 2004). One iteration of Chebyshev's algorithm is computationally relatively inexpensive since it requires solving two linear algebraic systems with the same matrix. Thus if the direct solvers are used, the additional computational effort when compared to the Newton's algorithm is negligible. Even in the case of Krylov subspace solvers one may reuse the computed preconditioner and utilize deflation techniques to speed up the solution of the second system.

We end this brief introduction by mentioning that there are locally cubically convergent methods, which only require derivatives of the objective function/constraints up to the second order, see for example (Kleinmichel 1968; Bartish 1969). There is also some recent interest and preliminary, but encouraging results on the use of the inexact versions of the methods in Halley class (which includes the Chebyshev's method), see for example (Steihaug and Suleiman 2013; Gundersen and Steihaug 2012; Deng and Zhang 2004) and references therein.

2 State space Newton's method for dissipated power minimization of Stokes flows

Instead of (1), we consider a barrier-type problem

$$\begin{aligned} \underset{(\mathbf{u},\rho)}{\text{minimize}} \quad & J_\mu(\mathbf{u},\rho) = J(\mathbf{u},\rho) - \mu \int_\Omega \log[(1-\rho)\rho], \\ \text{subject to} \quad & (\mathbf{u},\rho) \in \mathscr{U}_{\text{ad}} \times \mathscr{D}_{\text{ad}}, \end{aligned} \tag{3}$$

for a small barrier parameter $\mu > 0$. The first order necessary optimality conditions for (3) is a system of non-linear equations in $(\mathbf{u}, p, \rho, \lambda) \in \mathscr{U}_{\text{ad}} \times L^2_0(\Omega) \times \mathscr{D}_{\text{ad}} \times \mathbb{R}$:

$$-\operatorname{div} \nabla \mathbf{u} + \alpha(\rho)\mathbf{u} + \nabla p = \mathbf{f}, \tag{4}$$

$$\operatorname{div} \mathbf{u} = 0, \tag{5}$$

$$\underbrace{\frac{1}{2}\alpha'(\rho)\mathbf{u}\cdot\mathbf{u} - \frac{\mu}{\rho} + \frac{\mu}{1-\rho}}_{J'_\rho} + \lambda = 0, \tag{6}$$

$$\int_\Omega \rho = \gamma|\Omega|. \tag{7}$$

Note that at an arbitrary point $(\mathbf{u}, \rho) \in \mathscr{U}_{\text{ad}} \times \mathscr{D}_{\text{ad}}$ satisfying the inequality $J_\mu(\mathbf{u}, \rho) < \infty$ the expression J'_ρ, the ρ-derivative of J_μ, does not define a bounded linear functional on $L^\infty(\Omega)$. However, when ρ solves (6) for a given $(\mathbf{u}, \lambda)$, which we will denote by writing $\rho_{\mathbf{u},\lambda}$, we have the inclusion

$$\frac{\mu}{\rho} - \frac{\mu}{1-\rho} = \frac{1}{2}\alpha'(\rho)\mathbf{u}\cdot\mathbf{u} + \lambda \in L^3(\Omega), \tag{8}$$

owing to the Sobolev embedding $H^1(\Omega) \subset L^6(\Omega)$, which is valid in dimensions 2 and 3. Since both quantities in the left hand side cannot be large simultaneously, we deduce that both $\mu/(1-\rho)$ and μ/ρ are in $L^3(\Omega)$, and therefore

the barrier part of J_μ is three times differentiable with respect to ρ.[1]

The Newton's system for (4)–(7) at a point $(\mathbf{u}, p, \rho_{\mathbf{u},\lambda}, \lambda)$ may be formally stated as follows: find $(\mathbf{d_u}, d_p, d_\rho, d_\lambda) \in [H_0^1(\Omega)]^d \times L_0^2(\Omega) \times L^\infty(\Omega) \times \mathbb{R}$ such that

$$\begin{aligned} -\operatorname{div} \nabla \mathbf{d_u} + \alpha(\rho_{\mathbf{u},\lambda})\mathbf{d_u} + \alpha'(\rho_{\mathbf{u},\lambda})d_\rho \mathbf{u} + \nabla d_p = \mathbf{f} \\ + \operatorname{div} \nabla \mathbf{u} - \alpha(\rho_{\mathbf{u},\lambda})\mathbf{u} - \nabla p, \end{aligned} \tag{9}$$

$$\operatorname{div} \mathbf{d_u} = -\operatorname{div} \mathbf{u}, \tag{10}$$

$$\alpha'(\rho_{\mathbf{u},\lambda})\mathbf{u} \cdot \mathbf{d_u} + J''_{\rho,\rho} d_\rho + d_\lambda = 0, \tag{11}$$

$$\int_\Omega d_\rho = -\int_\Omega [\rho_{\mathbf{u},\lambda} - \gamma], \tag{12}$$

where

$$J''_{\rho,\rho} = \frac{1}{2}\alpha''(\rho_{\mathbf{u},\lambda})\mathbf{u} \cdot \mathbf{u} + \frac{\mu}{(\rho_{\mathbf{u},\lambda})^2} + \frac{\mu}{(1-\rho_{\mathbf{u},\lambda})^2}. \tag{13}$$

Unfortunately, the linear operator in the left hand side of (9)–(12) is not a locally Lipschitz function of the parameter ρ, when ρ is considered as an independent variable. Since this may tamper with the local quadratic convergence of the Newton's algorithm, we eliminate d_ρ from this system by taking a Schur complement with respect to $J''_{\rho,\rho} \geq 8\mu$ and obtain a Newton's system in a smaller state space $[H_0^1(\Omega)]^d \times L_0^2(\Omega) \times \mathbb{R}$:

$$\begin{aligned} -\operatorname{div} \nabla \mathbf{d_u} + \mathbf{A}\mathbf{d_u} + \nabla d_p + \rho'_{\mathbf{u}} d_\lambda = \mathbf{f} \\ + \operatorname{div} \nabla \mathbf{u} - \alpha(\rho_{\mathbf{u},\lambda})\mathbf{u} - \nabla p, \end{aligned} \tag{14}$$

$$\operatorname{div} \mathbf{d_u} = -\operatorname{div} \mathrm{u}, \tag{15}$$

$$\int_\Omega \rho'_{\mathbf{u}} \cdot \mathbf{d_u} + \rho'_\lambda d_\lambda = -\int_\Omega [\rho_{\mathbf{u},\lambda} - \gamma], \tag{16}$$

where

$$\begin{aligned} \mathbf{A} &= \alpha(\rho_{\mathbf{u},\lambda})\mathbb{I} - \frac{[\alpha'(\rho_{\mathbf{u},\lambda})]^2}{J''_{\rho,\rho}} \mathbf{u} \otimes \mathbf{u}, \\ \rho'_\lambda &= -\frac{1}{J''_{\rho,\rho}}, \qquad \rho'_{\mathbf{u}} = -\frac{\alpha'(\rho_{\mathbf{u},\lambda})}{J''_{\rho,\rho}} \mathbf{u}. \end{aligned} \tag{17}$$

It is easy to verify that in this space the non-constant elements of the operator in the left hand side of (14)–(16), namely $\mathbf{A}$, $\rho'_{\mathbf{u}}$, ρ'_λ are locally Lipschitz functions of $(\mathbf{u}, p, \lambda)$, see (Evgrafov 2014) for details. Therefore one can expect local quadratic convergence of the Newton's algorithm with unit steps towards points of local minimum, which satisfy the second order sufficient conditions.

[1] In dimension 2, the inclusion in (8) can be further strengthened and thus higher order derivatives may be obtained without assuming any further regularity of $\mathbf{u}$. Alternatively, since the zeroth order term $\alpha(\rho)\mathbf{u}$ in (4) is a priori in $[L^2(\Omega)]^d$ one may appeal to the standard regularity results for the Stokes problem, see for example (Kellogg and Osborn 1976; Amrouche and Girault 1991), to assert that $\mathbf{u} \in [H^s(\Omega)]^d$, $1 < s \leq 2$, under the appropriate additional assumptions on $\mathbf{f}$, $\mathbf{u}_0$, and Ω, and in this way further strengthen the inclusion (8) using the Sobolev inequality.

To globalize the convergence of this method we utilize two strategies. Firstly, we perform backtracking linesearch with respect to the augmented Lagrangian function

$$\begin{aligned} \psi(\theta) = J_\mu(\mathbf{u} + \theta\mathbf{d_u}, \rho_{\mathbf{u}+\theta\mathbf{d_u}}) \\ - \int_\Omega (p + \theta d_p)\operatorname{div}[\mathbf{u} + \theta\mathbf{d_u}] + \frac{\nu_p}{2}\|\operatorname{div}[\mathbf{u} + \theta\mathbf{d_u}]\|^2_{L^2(\Omega)} \\ + [\lambda + \theta d_\lambda] \int_\Omega (\rho_{\mathbf{u}+\theta\mathbf{d_u},\lambda+\theta d_\lambda} - \gamma) \\ + \frac{\nu_\lambda}{2}\left[\int_\Omega (\rho_{\mathbf{u}+\theta\mathbf{d_u},\lambda+\theta d_\lambda} - \gamma)\right]^2, \end{aligned} \tag{18}$$

where $\nu_\lambda > 0$, $\nu_p > 0$ are penalty parameters. Note that the search goes along a curve in the full space, owing to the non-linear dependence of $\rho_{\mathbf{u},\lambda}$ on its arguments through the equation (6). Secondly, when the Newton's system is unsolvable (we have not experienced this in our numerical computations) or when the Newton's direction is not a direction of descent for ψ, we compute a modified Newton's direction by replacing $\mathbf{A}$ in (14) with

$$\tilde{\boldsymbol{A}} = \begin{cases} \mathbf{A}, & \text{if } \alpha(\rho_{\mathbf{u},\lambda}) \geq \frac{[\alpha'(\rho_{\mathbf{u},\lambda})]^2}{J''_{\rho,\rho}} \mathbf{u} \cdot \mathbf{u}, \\ \mathbf{0}, & \text{otherwise.} \end{cases} \tag{19}$$

If the resulting direction still fails to be a descent direction for ψ, we also increase some or all of the penalty parameters ν_λ, ν_p.

3 Chebyshev's method for (3)

Chebyshev's method for solving a system of equations $F(x) = 0$ is based on the iteration $x^{k+1} = x^k - \{\mathbb{I} + 0.5[F'(x^k)]^{-1}F''(x^k)[F'(x^k)]^{-1}F(x^k)\}[F'(x^k)]^{-1}F(x^k)$. Nechepurenko (2004) has shown that under certain assumptions, including that of local Lipschitz continuity of F'', the method attains local third order convergence, which is much faster than the second order local convergence rate of Newton's iteration. Unlike other methods based on the the second order derivatives of F, such as Halley's method (method of tangent hyperbolas), see for example (Mertvecova 1953), Chebyshev's iteration requires solving two systems of equations with the same operator $F'(x)$ in the left hand side.

In the present context we calculate the Chebyshev's correction $(\mathbf{d}^C_{\mathbf{u}}, d^C_p, d^C_\lambda) \in [H_0^1(\Omega)]^d \times L_0^2(\Omega) \times \mathbb{R}$ to the Newton's direction $(\mathbf{d_u}, d_p, d_\lambda)$ given by (14)–(16) by solving the following boundary value problem:

$$-\operatorname{div} \nabla \mathbf{d}^C_{\mathbf{u}} + \mathbf{A}\mathbf{d}^C_{\mathbf{u}} + \nabla d^C_p + \rho'_{\mathbf{u}} d^C_\lambda = \mathbf{t}^C_{\mathbf{u}}, \tag{20}$$

$$\operatorname{div} \mathbf{d}^C_{\mathbf{u}} = 0, \tag{21}$$

$$\int_\Omega \rho'_{\mathbf{u}} \cdot \mathbf{d}^C_{\mathbf{u}} + \rho'_\lambda d^C_\lambda = t^C_\lambda, \tag{22}$$

where

$$\begin{aligned}\mathbf{t}_{\mathbf{u}}^C = &-\frac{1}{2(J''_{\rho,\rho})^2}\left[\alpha''(\rho_{\mathbf{u},\lambda}) - \frac{\alpha'(\rho_{\mathbf{u},\lambda})J'''_{\rho,\rho,\rho}}{J''_{\rho,\rho}}\right]d_\lambda^2\mathbf{u} \\ &-\frac{\alpha'(\rho_{\mathbf{u},\lambda})}{(J''_{\rho,\rho})^2}\left[2\alpha''(\rho_{\mathbf{u},\lambda}) - \frac{\alpha'(\rho_{\mathbf{u},\lambda})J'''_{\rho,\rho,\rho}}{J''_{\rho,\rho}}\right]d_\lambda(\mathbf{u}\cdot\mathbf{d}_{\mathbf{u}})\mathbf{u} \\ &+\frac{\alpha'(\rho_{\mathbf{u},\lambda})}{J''_{\rho,\rho}}d_\lambda\mathbf{d}_{\mathbf{u}} + \frac{(\alpha'(\rho_{\mathbf{u},\lambda}))^2}{2J''_{\rho,\rho}}\left[|\mathbf{d}_{\mathbf{u}}|^2\mathbf{u} + 2(\mathbf{u}\cdot\mathbf{d}_{\mathbf{u}})\mathbf{d}_{\mathbf{u}}\right] \\ &-\frac{(\alpha'(\rho_{\mathbf{u},\lambda}))^2}{2(J''_{\rho,\rho})^2}\left[3\alpha''(\rho_{\mathbf{u},\lambda}) - \frac{\alpha'(\rho_{\mathbf{u},\lambda})J'''_{\rho,\rho,\rho}}{J''_{\rho,\rho}}\right](\mathbf{u}\cdot\mathbf{d}_{\mathbf{u}})^2\mathbf{u},\end{aligned} \tag{23}$$

$$\begin{aligned}t_\lambda^C = &\int_\Omega \frac{J'''_{\rho,\rho,\rho}}{2(J''_{\rho,\rho})^3}d_\lambda^2 + \int_\Omega \frac{\alpha'(\rho_{\mathbf{u},\lambda})}{2J''_{\rho,\rho}}|\mathbf{d}_{\mathbf{u}}|^2 \\ &-\int_\Omega \frac{1}{(J''_{\rho,\rho})^2}\left[\alpha''(\rho_{\mathbf{u},\lambda}) - \frac{\alpha'(\rho_{\mathbf{u},\lambda})J'''_{\rho,\rho,\rho}}{J''_{\rho,\rho}}\right]d_\lambda(\mathbf{u}\cdot\mathbf{d}_{\mathbf{u}}) \\ &-\int_\Omega \frac{\alpha'(\rho_{\mathbf{u},\lambda})}{2(J''_{\rho,\rho})^2}\left[2\alpha''(\rho_{\mathbf{u},\lambda}) - \frac{\alpha'(\rho_{\mathbf{u},\lambda})J'''_{\rho,\rho,\rho}}{J''_{\rho,\rho}}\right](\mathbf{u}\cdot\mathbf{d}_{\mathbf{u}})^2,\end{aligned} \tag{24}$$

and

$$J'''_{\rho,\rho,\rho} = \frac{1}{2}\alpha'''(\rho_{\mathbf{u},\lambda})|\mathbf{u}|^2 - \frac{2\mu}{\rho_{\mathbf{u},\lambda}^3} + \frac{2\mu}{(1-\rho_{\mathbf{u},\lambda})^3}. \tag{25}$$

Algorithm 1 Globalized Chebyshev iteration in state space

1: $(\mathbf{u}, p) \leftarrow$ solve (4), (5) with $\rho = 1$, $\lambda \leftarrow \lambda^0$ ▷ Initialization
2: **while True do**
3: $(\mathbf{d}_{\mathbf{u}}, d_p, d_\lambda) \leftarrow$ solve (14)–(16) ▷ Pure Newton's direction
4: **if** $\|(\mathbf{d}_{\mathbf{u}}, d_p, d_\lambda)\| < \varepsilon$ **then stop**
5: **end if**
6: $(\mathbf{d}_{\mathbf{u}}^C, d_p^C, d_\lambda^C) \leftarrow$ solve (20)–(22) ▷ Chebyshev's correction
7: **if** $(\mathbf{u}+\mathbf{d}_{\mathbf{u}}+\mathbf{d}_{\mathbf{u}}^C, p+d_p+d_p^C, \lambda+d_\lambda+d_\lambda^C)$ accepted by ψ **then**
8: $(\mathbf{u}, p, \lambda) \leftarrow (\mathbf{u}+\mathbf{d}_{\mathbf{u}}+\mathbf{d}_{\mathbf{u}}^C, p+d_p+d_p^C, \lambda+d_\lambda+d_\lambda^C)$
9: **else if** $(\mathbf{d}_{\mathbf{u}}, d_p, d_\lambda)$–descent dir. for ψ **then**
10: $(\mathbf{u}, p, \lambda) \leftarrow$ linesearch for ψ along $(\mathbf{d}_{\mathbf{u}}, d_p, d_\lambda)$
11: **else** ▷ Compute modified Newton's direction
12: $(\mathbf{d}_{\mathbf{u}}, d_p, d_\lambda) \leftarrow$ solve (14)–(16) with modified $\mathbf{A}$ (19)
13: $(\mathbf{u}, p, \lambda) \leftarrow$ linesearch for ψ along $(\mathbf{d}_{\mathbf{u}}, d_p, d_\lambda)$
14: **end if**
15: **end while**

The resulting algorithm is formally summarized as Algorithm 1.

4 Solving the linear systems

Utilizing the exact Chebyshev's direction (or the exact Newton's direction) is primarily feasible if a factorization of the left hand side of the system (14)–(15) is available as a result of an application of a direct solver. To keep the number of non-zero elements in the discretized operators to a minimum, instead of enforcing the zero mean pressure constraint $p, d_p, d_p^C \in L_0^2(\Omega)$ we simply prescribe the pressure to be zero at some point in the computational domain. The solution to (14)–(16) is obtained by computing a Schur complement with respect to the first two equations as follows. We first solve the systems

$$\begin{aligned}&-\operatorname{div}\nabla\mathbf{d}_{\mathbf{u}}^{(1)} + \mathbf{A}\mathbf{d}_{\mathbf{u}}^{(1)} + \nabla d_p^{(1)} = \mathbf{f} + \operatorname{div}\nabla\mathbf{u} - \alpha(\rho_{\mathbf{u},\lambda})\mathbf{u} - \nabla p, \\ &\operatorname{div}\mathbf{d}_{\mathbf{u}}^{(1)} = -\operatorname{div}\mathbf{u},\end{aligned} \tag{26}$$

and

$$\begin{aligned}&-\operatorname{div}\nabla\mathbf{d}_{\mathbf{u}}^{(2)} + \mathbf{A}\mathbf{d}_{\mathbf{u}}^{(2)} + \nabla d_p^{(2)} = \rho'_{\mathbf{u}}, \\ &\operatorname{div}\mathbf{d}_{\mathbf{u}}^{(2)} = 0,\end{aligned} \tag{27}$$

which requires only one matrix factorization. Using this information we obtain the Newton's direction

$$\begin{aligned}d_\lambda &= \frac{-\int_\Omega[\rho_{\mathbf{u},\lambda} - \gamma] - \int_\Omega \rho'_{\mathbf{u}}\cdot\mathbf{d}_{\mathbf{u}}^{(1)}}{\int_\Omega \rho'_\lambda - \int_\Omega \rho'_{\mathbf{u}}\cdot\mathbf{d}_{\mathbf{u}}^{(2)}}, \\ (\mathbf{d}_{\mathbf{u}}, d_p) &= \left(\mathbf{d}_{\mathbf{u}}^{(1)}, d_p^{(1)}\right) - d_\lambda\left(\mathbf{d}_{\mathbf{u}}^{(2)}, d_p^{(2)}\right).\end{aligned} \tag{28}$$

Furthermore, the Chebyshev's direction is found by solving the system

$$\begin{aligned}&-\operatorname{div}\nabla\mathbf{d}_{\mathbf{u}}^{(3)} + \mathbf{A}\mathbf{d}_{\mathbf{u}}^{(3)} + \nabla d_p^{(3)} = \mathbf{t}_{\mathbf{u}}^C, \\ &\operatorname{div}\mathbf{d}_{\mathbf{u}}^{(3)} = 0,\end{aligned} \tag{29}$$

after which we similarly to (28) compute

$$\begin{aligned}d_\lambda^C &= \frac{t_\lambda^C - \int_\Omega \rho'_{\mathbf{u}}\cdot\mathbf{d}_{\mathbf{u}}^{(3)}}{\int_\Omega \rho'_\lambda - \int_\Omega \rho'_{\mathbf{u}}\cdot\mathbf{d}_{\mathbf{u}}^{(2)}}, \\ (\mathbf{d}_{\mathbf{u}}^C, d_p^C) &= \left(\mathbf{d}_{\mathbf{u}}^{(3)}, d_p^{(3)}\right) - d_\lambda^C\left(\mathbf{d}_{\mathbf{u}}^{(2)}, d_p^{(2)}\right).\end{aligned} \tag{30}$$

Thus, in the considered case no new matrix factorizations is required and the additional computational work compared to finding the Newton's direction is truly negligible.

5 Benchmarking Chebyshev's iteration

In this section we compare the performance of the Chebyshev's and Newton's algorithms on a set of two-dimensional benchmarks described in detail in Borrvall and Petersson (2003).

We discretize Algorithm 1 using the approach proposed for the Newton's method in Evgrafov (2014). Namely, we select a stable discretization[2] of the Stokes equations and use it for representing the discrete versions of $(\mathbf{u}, p)$ and the search directions $(\mathbf{d}_{\mathbf{u}}, d_p)$, $(\mathbf{d}_{\mathbf{u}}^C, d_p^C)$. The quantity $\rho_{\mathbf{u},\lambda}$ is not stored, but rather recomputed on the fly by solving the scalar (6) at points determined by the selected discretization of the Newton's direction finding subproblem (14), (15).

[2] In Evgrafov (2014) we have verified that the state space Newton's algorithm performs virtually independently of the selected discretization/mesh refinement.

Finite element discretization In this note we use two popular stable mixed finite element families on quadrilaterals, namely $[Q2]^2/Q1$ (Hood and Taylor 1973) and $[Q2\text{-iso-}Q1]^2/Q1$ (Bercovier and Pironneau 1979).[3] We do not store discretized $\rho_{\mathbf{u},\lambda}$ but instead recompute it by solving (11) at Gaussian quadrature points whenever needed for evaluating integrals.

Within the Newton's method framework we typically need to integrate non-polynomial functions (terms involving $\rho_{\mathbf{u},\lambda}$) times polynomials (terms involving $\mathbf{u}$ or $\mathbf{d_u}$). For the pure Newton's direction, the polynomials are at most piecewise bi-quadratic for $[Q2\text{-iso-}Q1]^2/Q1$ elements or piecewise bi-quartic for $[Q2]^2/Q1$ elements, which require tensor product Gaussian quadratures of at least degree 2 or 3 over every $\mathbf{u}$-element (thus a reiterated Gaussian quadrature of degree 2 over the macro-element) to compute the integral over the polynomial part exactly. In Evgrafov (2014) we used a rule of thumb of adding another quadrature point along each coordinate axis for evaluating all the non-polynomial integrals, thus leading to tensor product Gaussian quadratures of order 4 for Taylor–Hood elements.

In the case of Chebyshev discretization, integrating the expression (23), which features in the right hand side of (20), times a test function involves some non-polynomial terms including $\rho_{\mathbf{u},\lambda}$ times piecewise tensor-products of polynomials of the order up to 12 in the case of Taylor–Hood elements or of order 6 in the case of Bercovier–Pironneau elements. To address this we raise the quadrature power to 8 and 5, respectively; however, we report the performance of the Taylor–Hood discretization with the quadrature of order 4 as well. A general comment based on our computational experience is that for coarse discretizations the accuracy of the quadrature plays a much more important role for reducing the KKT residuals to near machine accuracy than for the finer discretizations, which is not surprizing as the volumetric Lebesgue measure of the elements where $\rho_{\mathbf{u},\lambda}$ deviates significantly from a constant becomes smaller and smaller near optimal solutions for fine discretizations.

We have implemented the FEM-discretized Newton's and Chebyshev's algorithms in C++ using deal.II libraries (Bangerth et al. 2007). Umfpack (Davis 2004) is utilized as a linear solver.

Finite volume discretization We utilize the standard staggered grid approach, see for example Griebel et al. (1997). Since the divergence is exactly zero at the discrete level, we remove the terms involving divergence from the augmented Lagrangian function or the right hand side of the Newton's system. Additionally, we integrate $|\mathrm{curl}\mathbf{u}|^2$ instead of $|\nabla\mathbf{u}|^2$ when evaluating J or its derivatives in our implementation. All the terms involving $\rho_{\mathbf{u},\lambda}$ are evaluated only at the vorticity nodes, see Fig. 1, and then interpolated back to u_1 and u_2 nodes, where momentum conservation is formulated. The algorithm is implemented in Matlab (we used an excellent educational Navier–Stokes code (Seibold 2008) as a basis for the Stokes solver) and utilizes the built-in LDL factorization based on MA27 (Duff 2002) for solving all the linear systems. The full code including setups for the benchmarks is available as Online Resource 1.

For both discretizations We normally initialize $(\mathbf{u}, p)$ to the solution of the pure Stokes problem ($\rho = 1$) in the domain, and put $\lambda^0 = 100.0$. In the case of the "rugby ball" problem, such a starting point satisfies the first order optimality conditions. To escape this critical point we initialize $(\mathbf{u}, p)$ to be the solution of Brinkmann's equations corresponding to $\rho = 1.0 - 0.25\chi_{B(0.1)}$ instead, where $\chi_{B(0.1)}$ is the characteristic function for the ball of radius 0.1 around the center of Ω.

We begin the iterations by using the modified Newton's search direction, see (19), and we do this until the linesearch accepts unit steplength and $\|\mathbf{d_u}\|_{[H^1(\Omega)]^d}/\|\mathbf{u}\|_{[H^1(\Omega)]^d} < 10^{-1}$. This normally happens within the first few iterations, after which we proceed as outlined in Algorithm 1. We terminate the algorithm when either $\|\mathbf{d_u}\|_{[H^1(\Omega)]^d}/\|\mathbf{u}\|_{[H^1(\Omega)]^d} < 10^{-10}$ (success) or the limit of iterations has been exceeded (failure). For the benchmark problems presented in this study we have empirically found out that typically 50 iterations are sufficient for algorithm's

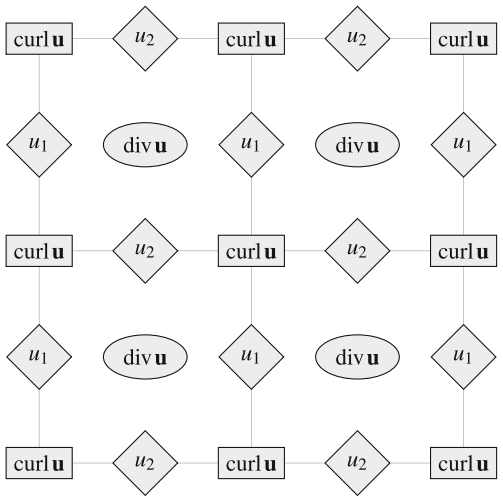

Fig. 1 Standard staggered FVM discretization utilized in this work. Pressure nodes coincide with div$\mathbf{u}$ nodes. Non-linear terms involving $\rho_{\mathbf{u},\lambda}$ are evaluated at curl$\mathbf{u}$ nodes. Bilinear ansatz is used for interpolating all quantities between their staggered locations

[3] This discretization has been used in the original paper of Borrvall and Petersson (2003). *Note that we count the number of elements differently*: in (Borrvall and Petersson 2003) the number of $\mathbf{u}$-elements is reported, whereas we list the number of macroelements. As a result the number of degrees of freedom in $[Q2]^2/Q1$ and $[Q2\text{-iso-}Q1]^2/Q1$ elements is the same on the same discretization level. In other words our 100×100 discretization $[Q2\text{-iso-}Q1]^2/Q1$ corresponds to 200×200 $\mathbf{u}$-elements.

convergence, and therefore we use 50 as the iteration limit throughout our computations. Occasionally the algorithm stagnates–in the sense that a discrete Newton's directions becomes a direction of ascent for the discretized augmented Lagrangian function with a very small positive directional derivative on the order of the discretization error. This is particularly true on coarse meshes and can often be rectified by increasing the number of quadrature points. The algorithm still makes progress utilizing the modified Newton's directions, but the fast local rate of convergence is of course lost in this case. Fortunately, this happens at near-optimal points. To address this issue at least partly, we modify the algorithm as follows. In the linesearch routine, if $|\psi'(0)| < 10^{-8}$ and the search direction is based on non-modified Hessian, we accept the unit step. On the other hand, if a modified Newton's step is taken but the relative improvement in the merit function after linesearch is less than 10^{-8} we stop the algorithm and declare that it has stagnated. Strictly speaking these criteria should be adjusted with the discretization choice and mesh refinement level. However, in our numerical simulations we observe that stagnation occurs only occasionally, provided that the mesh is "fine enough."

These stopping conditions are unnecessarily stringent as the final error will be dominated by the errors in the discretization (small parameter h), approximation of the Dirichlet boundary conditions (small parameter $\bar{\alpha}^{-1}$), relaxation of the "0–1" affine penalty to $\alpha(\rho)$ (small parameter q^{-1}), or the barrier function (small parameter μ). Nevertheless, owing to the fast local convergence of the algorithm the

Table 1 Numerical performance of the state space Newton's/Chebyshev's algorithm on the two-dimensional benchmark problems from Borrvall and Petersson (2003)

Prob. →	Diffuser			Pipe bend			Rugby ($\gamma = 0.8$)			Dbl. pipe (short)			Dbl. pipe (long)		
Discr. ↓	J^*_μ	#it	Time	J^*_μ	#it	Time	J^*_μ	#it	Time	J^*_μ	#it	Time	J^*_μ	#it	Time
Borrvall and Petersson (2003):															
50×50	30.59	29	≈ 29	10.01	64	≈ 64									
100×100	30.46	33	≈ 33	9.76	85	≈ 85	31.75	47	≈ 47	25.67	61	≈ 61	27.64	236	≈ 236
Challis and Guest (2009):															
100×100	36.57	197	≈ 47.6	13.18	717	≈ 52.1	38.02	127	≈ 57.2						
200×200	36.16	725	≈ 188.5	12.99	706	≈ 51.7	37.50	278	≈ 112.8	31.68	595	≈ 53.0	32.68	681	≈ 58.6
$[Q2]^2/Q1$ FEM, Newton, 4-point quadrature:															
100×100	30.45033	12	17.7	9.643469	33	41.9	31.42813	13	16.5	21.89791	27	37.7	23.86315	27	35.9
200×200	30.45239	12	16.1	9.642835	29	33.9	31.42744	13	16.7	21.90054	26	34.2	23.86641	26	31.3
400×400	30.45305	12	15.8	9.642701	29	32.6	31.42740	13	15.6	21.90193	25	32.7	23.86787	27	33.1
$[Q2]^2/Q1$ FEM, Chebyshev, 4-point quadrature:															
100×100	30.45033	11	17.7	9.643468	28	38.1	31.42813	12	17.1	21.89857	50	99.0	23.86315	24	34.2
200×200	30.45239	11	14.3	9.642835	24	29.8	31.42744	12	15.8	21.90054	23	32.2	23.86641	24	31.0
400×400	30.45305	11	14.4	9.642701	23	25.8	31.42740	12	14.4	21.90193	23	29.8	23.86787	24	31.3
$[Q2]^2/Q1$ FEM, Newton, 8-point quadrature:															
100×100	30.45033	12	19.2	9.643457	28	36.1	31.42812	13	19.2	21.89826	50	100.5	23.86310	26	36.3
200×200	30.45239	12	16.8	9.642834	29	36.0	31.42744	13	17.1	21.90047	26	36.8	23.86642	26	35.6
400×400	30.45305	12	16.7	9.642701	29	32.0	31.42740	13	15.9	21.90193	26	34.9	23.86787	26	32.6
$[Q2]^2/Q1$ FEM, Chebyshev, 8-point quadrature:															
100×100	30.45033	11	18.0	9.643457	24	32.3	31.42812	12	17.8	21.90004	50	102.5	23.86310	24	34.2
200×200	30.45239	11	16.4	9.642834	24	31.6	31.42744	11	15.5	21.90047	24	36.2	23.86642	23	33.9
400×400	30.45305	11	14.6	9.642701	23	27.0	31.42740	12	15.6	21.90193	22	28.2	23.86787	23	30.6
$[Q2\text{-iso-}Q1]^2/Q1$ FEM, Newton, iterated 5-point quadrature:															
100×100	30.45033	12	20.3	9.643457	29	42.6	31.42813	13	20.7	21.89813	26	42.7	23.86310	27	41.3
200×200	30.45239	12	18.0	9.642833	29	36.8	31.42744	13	16.6	21.90047	25	36.2	23.86642	27	36.6
400×400	30.45305	13	17.1	9.642701	29	32.9	31.42740	13	16.2	21.90193	26	33.4	23.86787	27	33.6
$[Q2\text{-iso-}Q1]^2/Q1$ FEM, Chebyshev, iterated 5-point quadrature:															
100×100	30.45033	11	20.0	9.643457	23	36.9	31.42813	11	17.7	21.89813	27	45.9	23.86310	25	39.2
200×200	30.45239	11	17.6	9.642833	24	31.1	31.42744	11	15.0	21.90047	22	33.4	23.86642	24	32.6
400×400	30.45305	11	15.6	9.642701	24	28.0	31.42740	11	13.9	21.90193	21	29.1	23.86787	23	29.5
FVM, Newton:															
100×100	30.55573	15	20.9	9.691091	41	87.6	31.55465	15	21.7	22.13683	37	59.1	23.99473	27	47.8
200×200	30.47737	13	20.2	9.652791	33	67.8	31.45903	13	18.5	21.95593	25	32.0	23.89662	28	47.3
400×400	30.45881	14	26.2	9.644621	32	93.1	31.43537	14	32.6	21.91517	25	36.4	23.87475	28	87.9
800×800	30.45458	14	13.6	9.643010	29	33.5	31.42939	16	32.2	21.90533	27	31.6	23.86966	28	45.8
1600×1600	30.45360	14	35.4	9.642712	30	106.8	31.42789	12	37.1	21.90306	27	43.7	23.86859	28	125.4
FVM, Chebyshev:															
100×100	30.55573	15	22.7	9.691091	25	46.9	31.55465	16	24.9	22.13687	31	52.7	23.99473	25	44.8
200×200	30.47737	13	21.2	9.652791	29	58.0	31.45903	14	20.3	21.95593	24	32.4	23.89662	25	41.2
400×400	30.45881	13	11.9	9.644621	29	78.7	31.43537	14	15.0	21.91517	22	37.7	23.87475	25	48.1
800×800	30.45458	13	22.4	9.643010	25	67.7	31.42939	14	27.1	21.90533	23	27.2	23.86966	25	51.2
1600×1600	30.45360	13	37.0	9.642712	23	78.0	31.42789	11	31.4	21.90306	23	39.3	23.86859	25	78.8

All computations are done on uniformly refined structured grids. The algorithmic termination is shown as success, stagnation, or failure (number of allowed iterations has been exceeded). All times are reported relative to solving one Stokes problem. See Section 5 for more details and discussion

strictness of the stopping criteria has very little effect on the number of algorithmic iterations.

For other algorithmic parameters we set $\bar{\alpha} = 2.5 \cdot 10^4$, $q = 0.1$, $\mu = 10^{-3}$. We use a non-monotone heuristic strategy for updating the penalty parameters ν_λ, ν_p of the augmented Lagrangian penalty function ψ. At the beginning of every iteration we put $(\nu_\lambda, \nu_p) = (|\lambda| + \delta, \|p\|_{L^2_0(\Omega)} + \delta)$, where $\delta = 10.0$ in our computations. In backtracking/Armijo's linesearch, we use $c_1 = 10^{-4}$ in the sufficient decrease condition, and we successively halve the steplength until this condition is satisfied. The results of the numerical benchmarks are summarized in Table 1 and the first two block rows of Table 2; the snapshots of some of the optimal designs are shown in the top row in Fig. 2. Whether the algorithm terminates successfully, stagnates, or runs out of interations, it converges towards the optimal solutions with the same topology as the ones reported in (Borrvall and Petersson 2003; Challis and Guest 2009). It is undoubtedly possible to fine-tune the algorithmic parameters to further reduce the number of iterations in either Newton's or Chebyshev's cases. The purpose of this work is rather to illustrate the effect of the "freely" available high-order search direction on the algorithmic performance. Before proceeding further, we would like to make a few comments about Tables 1 and 2:

Mesh size:

- For the test case "double pipe (long)" the number of elements along the x axis should be multiplied by a factor 1.5;
- Quasi-uniform unstructured meshes of quadrilaterals have been generated using Gmsh (Geuzaine and Remacle 2009). Mesh #1, #2, and #3 have approximately 10K, 40K, and 160K quadrilateral elements ($\times 1.5$ in the case of "double pipe (long)" problem), which corresponds to the number of elements used in the structured grids;
- The mesh resolutions taken from Borrvall and Petersson (2003) are measured in **u**-elements, and therefore should be divided by 2 to get to the resolution in terms of $[Q2\text{-iso-}Q1]^2/Q1$ macro-elements, see footnote 3;
- The mesh resolutions taken from Challis and Guest (2009) are rounded to the nearest hundred, see the cited paper for details.

Timings:

- In order to get "architecture-independent" results we report the total running time of the algorithm relative to one solution of the Stokes problem, even excluding the finite element assembly time for the latter in the case of FEM implementation. Note that this metric is especially unfair to the FVM discretization as the number of non-zeros in the discretized Jacobian of the Newton's system is much larger than that for the pure Stokes or Brinkmann system due to the presence of the terms in the momentum equations, which explicitly couple u_1 and u_2. Also note that we were not able to obtain very reliable timings of our algorithm on the multi-user server, on which we performed our computations, and some outliers could be easily spotted in Table 1;
- For the results taken from Borrvall and Petersson (2003), we simply "guess" the relative timing to be the number of algorithmic iterations. This is a very optimistic estimate as it is based on the assumption that all calculations other than solving the Stokes/Brinkmann system take zero time;
- For the results taken from (Challis and Guest 2009), we estimate the time relative to the solution of the Stokes problem on the full computational domain based on the data reported in the cited paper. This is done because the level set algorithm utilized in the cited paper needs to solve the flow equations only on a smaller flow domain, and explains the fact that the estimated timing is much smaller than the number of algorithmic iterations. Note that the timing is still proportional to the number of algorithmic iterations.

Objective function values:

- It is extremely important to keep in mind that the calculations are based on a numerical discretization of Algorithm 1 and not on applying a mathematical programming algorithm to a discretized version of (1). That is, we utilize "optimize, then discretize" rather than "discretize, then optimize" approach. Therefore, the fact that the algorithm terminates successfully with $\|\mathbf{d}_\mathbf{u}\|_{[H^1(\Omega)]^d} < \varepsilon$ does not imply that the resulting point $(\mathbf{u}, \rho_{\mathbf{u},\lambda})$ is an $O(\varepsilon)$-optimal for the discretized problem (1). *The computed solutions are only as accurate, as the utilized discretization scheme!*
- One can directly compare the objective function values reported in Borrvall and Petersson (2003) and the ones found by the present algorithm, even though we report the power dissipation J augmented with the positive barrier function;
- One cannot directly compare the objective function values reported in Challis and Guest (2009) and the ones found by the present algorithm, as we underimpose the no-slip Dirichlet boundary conditions in the present case. This is not unique to our algorithm for solving the topology optimization problem but rather a major weakness of the homogenization based topology optimization approach to fluid mechanics inherent in formulation (1).

Table 2 Numerical performance of the state space Newton's/Chebyshev's algorithm on the two-dimensional benchmark problems from (Borrvall and Petersson 2003)

Prob. →				Diffuser			Pipe bend			Rugby ($\gamma = 0.8$)			Dbl. pipe (short)			Dbl. pipe (long)		
Discr. ↓	$\bar{\alpha}$	q	μ	J^*_μ	#it	Time	J^*_μ	#it	Time	J^*_μ	#it	Time	J^*_μ	#it	Time	J^*_μ	#it	Time
Quasi-uniform grids, $[Q2]^2/Q1$ FEM, Newton, 8-point quadrature:																		
Mesh #1	$2.5\cdot10^4$	$1.0\cdot10^{-1}$	$1.0\cdot10^{-3}$	30.44900	12	17.7	9.643073	29	35.8	31.42842	13	17.4	21.89234	26	36.3	23.86987	33	44.0
Mesh #2	$2.5\cdot10^4$	$1.0\cdot10^{-1}$	$1.0\cdot10^{-3}$	30.45344	12	16.3	9.640938	29	33.8	31.42749	13	16.4	21.90251	26	34.1	23.86840	29	39.3
Mesh #3	$2.5\cdot10^4$	$1.0\cdot10^{-1}$	$1.0\cdot10^{-3}$	30.45279	12	15.7	9.642210	29	32.7	31.42740	13	15.9	21.90148	26	33.2	23.86743	27	33.3
Quasi-uniform grids, $[Q2]^2/Q1$ FEM, Chebyshev, 8-point quadrature:																		
Mesh #1	$2.5\cdot10^4$	$1.0\cdot10^{-1}$	$1.0\cdot10^{-3}$	30.44900	11	16.7	9.643073	22	29.2	31.42842	12	16.4	21.89234	22	32.9	23.86987	30	43.9
Mesh #2	$2.5\cdot10^4$	$1.0\cdot10^{-1}$	$1.0\cdot10^{-3}$	30.45344	11	15.7	9.640938	24	28.6	31.42749	12	15.6	21.90251	21	30.3	23.86840	26	36.7
Mesh #3	$2.5\cdot10^4$	$1.0\cdot10^{-1}$	$1.0\cdot10^{-3}$	30.45279	11	14.8	9.642210	24	26.5	31.42740	11	13.9	21.90148	20	27.3	23.86743	23	29.3
Continuation strategy #1, $[Q2]^2/Q1$ FEM, Newton, 8-point quadrature:																		
Refinement #1	$2.5\cdot10^4$	$1.0\cdot10^{-1}$	$1.0\cdot10^{-3}$	30.45179	7		9.642731	3		31.42752	3		21.90026	8		23.86862	3	
Refinement #2	$2.5\cdot10^4$	$1.0\cdot10^{-1}$	$1.0\cdot10^{-3}$	30.45296	1		9.642691	2		31.42744	2		21.90176	1		23.86848	3	
Refinement #3	$2.5\cdot10^4$	$1.0\cdot10^{-1}$	$1.0\cdot10^{-3}$	30.45323	2		9.642670	2		31.42741	2		21.90224	2		23.86836	3	
Refinement #4	$2.5\cdot10^4$	$1.0\cdot10^{-1}$	$1.0\cdot10^{-3}$	30.45330	2		9.642664	2		31.42741	2		21.90234	2		23.86833	2	
Continuation strategy #1, $[Q2]^2/Q1$ FEM, Chebyshev, 8-point quadrature:																		
Refinement #1	$2.5\cdot10^4$	$1.0\cdot10^{-1}$	$1.0\cdot10^{-3}$	30.45178	2		9.642731	2		31.42752	2		21.90024	1		23.86662	2	
Refinement #2	$2.5\cdot10^4$	$1.0\cdot10^{-1}$	$1.0\cdot10^{-3}$	30.45296	1		9.642691	2		31.42744	2		21.90177	2		23.86807	14	
Refinement #3	$2.5\cdot10^4$	$1.0\cdot10^{-1}$	$1.0\cdot10^{-3}$	30.45323	2		9.642670	2		31.42741	2		21.90224	1		23.86821	2	
Refinement #4	$2.5\cdot10^4$	$1.0\cdot10^{-1}$	$1.0\cdot10^{-3}$	30.45330	2		9.642664	2		31.42741	1		21.90234	2		23.86829	2	
Continuation strategy #2, $[Q2]^2/Q1$ FEM, Newton, 8-point quadrature:																		
Refinement #1	$2.5\cdot10^4$	$2.0\cdot10^{-1}$	$5.0\cdot10^{-4}$	31.32134	6		10.07666	7		32.42190	6		23.36283	5		25.04417	7	
Refinement #2	$2.5\cdot10^4$	$4.0\cdot10^{-1}$	$2.5\cdot10^{-4}$	31.83089	5		10.33390	8		32.99263	5		24.25412	5		25.74418	7	
Refinement #3	$2.5\cdot10^4$	$8.0\cdot10^{-1}$	$1.3\cdot10^{-4}$	32.08979	5		10.47179	5		33.28090	5		24.71847	5		26.10504	5	
Refinement #4	$2.5\cdot10^4$	$1.6\cdot10^{0}$	$3.1\cdot10^{-5}$	32.20128	4		10.53051	6		33.40492	5		24.92134	4		26.26188	6	
Continuation strategy #2, $[Q2]^2/Q1$ FEM, Chebyshev, 8-point quadrature:																		
Refinement #1	$2.5\cdot10^4$	$2.0\cdot10^{-1}$	$5.0\cdot10^{-4}$	31.32134	7		10.07665	5		32.42190	5		23.36283	7		25.04417	5	
Refinement #2	$2.5\cdot10^4$	$4.0\cdot10^{-1}$	$2.5\cdot10^{-4}$	31.83089	6		10.33343	7		32.99263	5		24.25412	5		25.74418	7	
Refinement #3	$2.5\cdot10^4$	$8.0\cdot10^{-1}$	$1.3\cdot10^{-4}$	32.08979	7		10.47183	5		33.28090	7		24.71847	7		26.10504	6	
Refinement #4	$2.5\cdot10^4$	$1.6\cdot10^{0}$	$3.1\cdot10^{-5}$	32.20128	4		10.53051	6		33.40492	4		24.92134	6		26.26188	6	
Continuation strategy #3, $[Q2]^2/Q1$ FEM, Newton, 8-point quadrature:																		
Refinement #1	$5.0\cdot10^4$	$2.0\cdot10^{-1}$	$5.0\cdot10^{-4}$	32.55751	11		9.518515	29		33.73643	10		25.63163	9		26.82841	17	
Refinement #2	$1.0\cdot10^5$	$4.0\cdot10^{-1}$	$2.5\cdot10^{-4}$	33.73063	8		11.37980	37		34.95537	14		27.92111	9		28.54689	29	
Refinement #3	$2.0\cdot10^5$	$8.0\cdot10^{-1}$	$1.3\cdot10^{-4}$	34.39721	8		11.75413	8		35.66936	16		29.29217	10		29.54514	15	
Refinement #4	$4.0\cdot10^5$	$1.6\cdot10^{0}$	$3.1\cdot10^{-5}$	34.80009	16		11.98308	16		36.03318	15		30.14311	12		30.15443	20	
Continuation strategy #3, $[Q2]^2/Q1$ FEM, Chebyshev, 8-point quadrature:																		
Refinement #1	$5.0\cdot10^4$	$2.0\cdot10^{-1}$	$5.0\cdot10^{-4}$	32.5751	9		9.466042	1		33.73643	8		25.63163	12		26.82840	15	
Refinement #2	$1.0\cdot10^5$	$4.0\cdot10^{-1}$	$2.5\cdot10^{-4}$	33.73063	8		11.38005	26		34.95537	16		27.92111	8		28.54688	21	
Refinement #3	$2.0\cdot10^5$	$8.0\cdot10^{-1}$	$1.3\cdot10^{-4}$	34.39722	6		11.75444	13		35.63153	16		29.29218	9		29.54509	25	
Refinement #4	$4.0\cdot10^5$	$1.6\cdot10^{0}$	$3.1\cdot10^{-5}$	34.80009	15		11.98365	18		36.03315	11		30.14311	8		30.15431	15	

Computations are done on quasi-uniform or adaptively refined unstructured grids of quadrilaterals. The algorithmic termination is shown as success, stagnation, or failure (number of allowed iterations has been exceeded). All times are reported relative to solving one Stokes problem. See Sections 5 and 6 for more details and discussion.

In most of the cases Chebyshev's iteration outperforms the Newton's iteration with respect to both the number of required iterations and the total running time. The effect seems to be more prominent for the examples where the optimal configurations contain long straight channels, such as the "pipe bend" and "two pipes" test cases, see Fig. 3, bottom. In these cases the linesearch accepts unit Newton's steps long before the fast decrease of the residuals occurs. On the other hand, for the "diffuser" and "rugby" test cases the Newton's algorithm behaves extremely well with quadratic convergence occurring almost immediately after the algorithm switches from the initial modified Newton's to pure Newton's directions, see Fig. 3, top. This makes it very difficult to outperform such a process.

Another general statement is that the discretization errors stemming from the interpolation of certain non-linear quantities between the staggered grids in the FVM discretization negatively affects the rate of convergence of Newton's and to an even larger degree the Chebyshev's iteration. Nevertheless, benchmarks in Table 1 demonstrate that utilizing Chebyshev's correction is beneficial within this framework as well.

Note that the level set method utilized in Challis and Guest (2009) does not require continuation with respect to the parameters $(\bar{\alpha}, q, \mu)$, which to a degree compensates for the large number of algorithmic iterations that it requires, which also seems to grow with mesh refinement.

6 Continuation and adaptive mesh refinement

We now apply the fast locally convergent algorithm developed in the previous sections in the framework of adaptive mesh refinement and parameter continuation. For simplicity, we use a residual-based error estimator for (4), (5) (note that (6) is preserved at the discrete level), see for

example (Verfürth 1989). Thus for every element T in the mesh we compute the a posteriori error indicator

$$\eta_T^2 = |T| \| - \operatorname{div} \nabla \mathbf{u} + \alpha(\rho)\mathbf{u} + \nabla p - \mathbf{f}\|^2_{[L^2(T)]^d} + \|\operatorname{div} \mathbf{u}\|^2_{L^2(T)} + \frac{1}{2} \sum_{e \in \partial T \setminus \partial\Omega} |e| \|[\nabla \mathbf{u} \cdot \mathbf{n}]_e\|^2_{L^2(e)}, \quad (31)$$

where $[\nabla \mathbf{u} \cdot \mathbf{n}]_e$ is the jump of the normal derivative of $\mathbf{u}$ across the edge e in the mesh. A conceptual continuation/adaptive mesh refinement algorithm is thus stated as Algorithm 2. Note that in step 5 of the algorithm we do not need to project the design field ρ, as it is recomputed on the new mesh using (6). Additionally, the interpolation does not preserve the conservation of mass (in the weak sense) constraint $\operatorname{div}\mathbf{u} = 0$, which is another reason for adding it to the right hand side of Newton's system and the merit function.

We test Algorithm 2 in the following fashion. In all cases, we start by computing the optimal solution on unstructured grid #1 using Algorithm 1. We then make four adaptive refinement steps (four iterations of the loop in Algorithm 2) by refining 20 % of cells with the largest indicator, and coarsening 3 % of cells with the smallest indicator. Deal II also needs to refine some adjacent cells (we refer to the library documentation for details), and this strategy leads to roughly speaking doubling of the number of cells at every iteration. For parameters $(\bar{\alpha}, q, \mu^{-1})$ we test the following strategies: #1: $(\bar{\alpha}, q, \mu^{-1})$ are constant on all meshes; #2: (q, μ^{-1}) are doubled before each refinement, but $\bar{\alpha}$ is constant; #3: all parameters are doubled before each refinement. We note that in the topology optimization literature typically a strategy analogous to our strategy #2 is utilized, however without adaptive mesh refinement or parameter μ, which is not found in the original problem (1), see for example (Borrvall and Petersson 2003; Aage et al. 2008).

The results of these benchmarks are summarized in Table 2 and some of the optimal designs obtained are shown in Fig. 2, middle and bottom rows. One can see that both Newton's and Chebyshev's algorithms perform incredibly well in the strategy #1, needing in most cases only two iterations for reducing the residuals in Algorithm 1 to the level described in the previous section.

For strategy #2, both algorithms also perform reasonably well, although both Newton's and Chebyshev's algorithm stagnated on most instances of "pipe bend" test case. Interestingly enough, Chebyshev's method in some benchmarks required one or two more iterations when compared to Newton's algorithm. Even ignoring the advantages of adaptive mesh refinement where most of the iterations are performed on coarse meshes and therefore are very computationally inexpensive, $12 + 5 + 5 + 7 + 4 = 33$ iterations needed by Chebyshev's algorithm to solve "rugby ball" problem to a very high precision can be compared with 384 iterations reported in (Aage et al. 2008) for this benchmark problem and a similar continuation procedure based on a

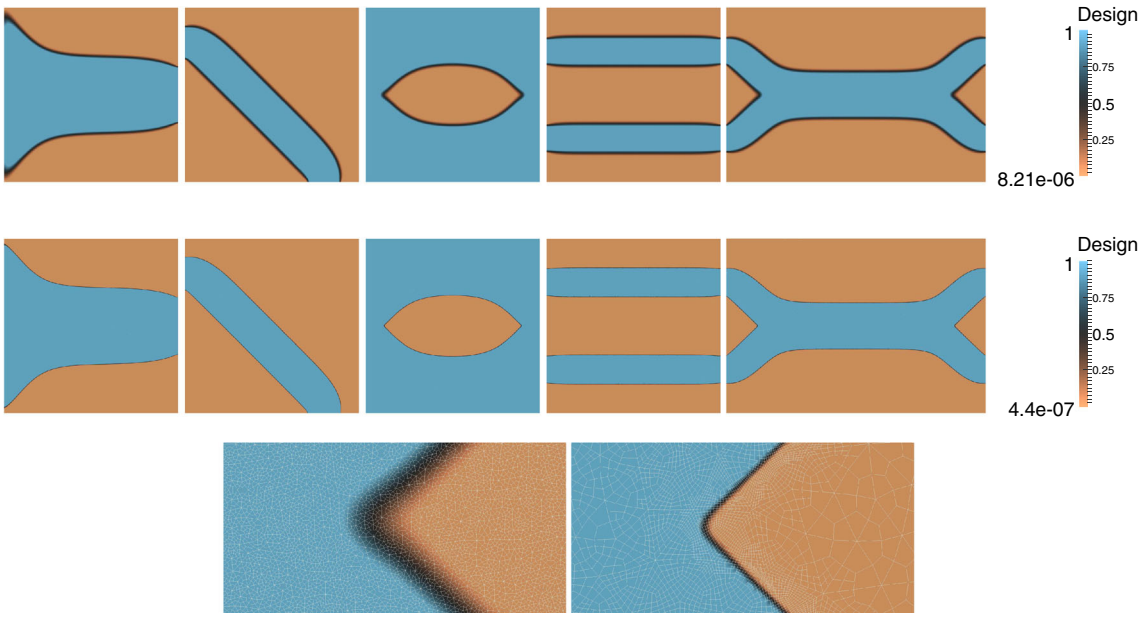

Fig. 2 Snapshots of the final designs computed using Chebyshev's iteration for some of the benchmark problems. Note: the field has been projected onto the space of piecewise linear polynomials for visualization purposes. In the *top* two rows, from *left* to *right*: "diffuser", "pipe bend", "rugby ($\gamma = 0.8$)", "dbl. pipe (short)", and "dbl. pipe (long)". *Top row*: designs computed on a quasi-uniform unstructured mesh #3; *middle row*: designs computed as a result of the adaptive mesh refinement/continuation strategy #2; bottom row: zoom into the leading edge of the "rugby ($\gamma = 0.8$)" design including the mesh lines for the quasi-uniform and adaptive meshes

first order algorithm and much less strict stopping criteria, resulting in a more than 10-fold speedup.

Strategy #3 is the most challenging one, with both methods struggling to reduce the residual to the requested tolerance on most problem instances. Additionally, increasing $\bar{\alpha}$ results in larger residuals in the "solid" regions of the domain, which are then refined in accordance with our refinement strategy. This results in meshes, which are much more quasi-uniform when compared with strategies #1 and #2.

When one takes into account the fact that most of the computations are done on low resolution meshes, the savings are quite significant: for example on a multi-user Linux machine with 4×6 core Intel Xeon 2.67 GHz CPUs and 256Gb of memory, one "pipe bend" problem run using Newton's (Chebyshev's) method on the unstructured meshes #1, #2, and #3 take respectively 566 (453), 3 330 (2 860), and 29 100 (25 200) seconds.[4] On the same hardware, adaptive strategies #1, #2, and #3 utilizing Newon's (Chebyshev's) iteration take respectively 1 920 (1 740), 3 520 (3 360), and 11 500 (11 500) seconds.

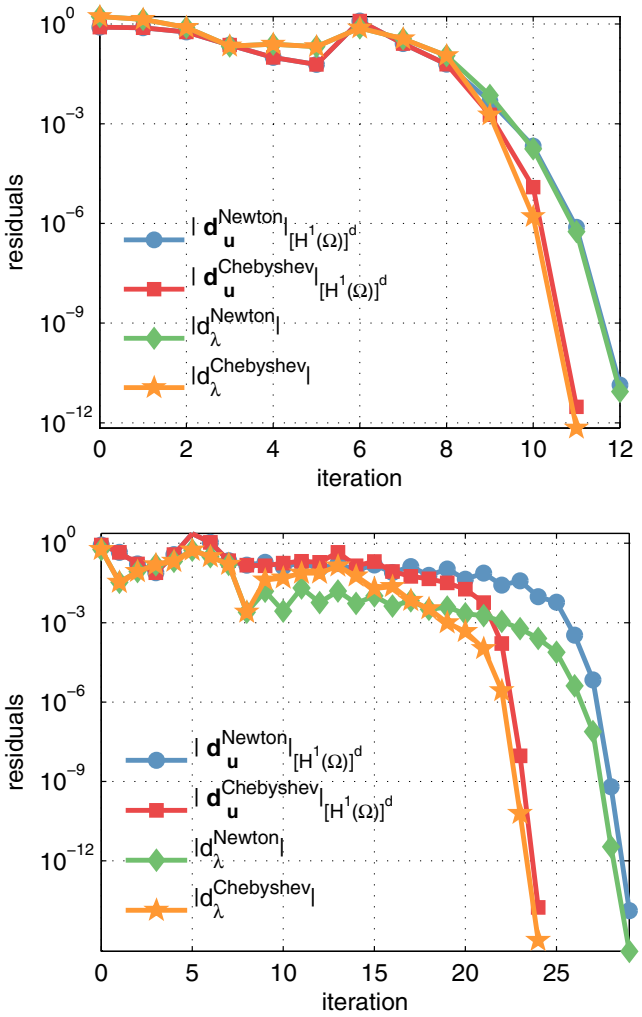

Fig. 3 Convergence of the residuals for the state space Newton/Chebyshev methods. *Top*: diffuser problem; *bottom*: pipe bend problem. In both cases, FEM discretization with 8-point quadratures on an unstructured Mesh #3 with approximately 160 K quadrilaterals is used. In the second case, faster-than-Newton local convergence of Chebyshev's iteration is particularly prominent

Algorithm 2 Continuation/adaptive mesh refinement

1: **repeat**
2: Apply Algorithm 1
3: Increase $(\bar{\alpha}, q, \mu^{-1})$; compute error indicators (31)
4: Refine/coarsen elements T with large/small η_T
5: Interpolate $(\mathbf{u}, p)$ onto the new mesh
6: **until** convergence

We end this brief discussion by stating that in our opinion a much more thorough analytical/numerical study, providing recommendations for updating many parameters involved in topology optimization problems such as (1) within the adaptive mesh refinement context, would be beneficial for further advancing the application of topology optimization to real-life problems.

7 Conclusions

We have presented, to the best of our knowledge, the first locally cubically convergent method for topology optimization. On quasi-uniform meshes the method demonstrated savings of 8 % to 17 % of iterations when compared with already fast Newton's algorithm, or 60 % to 90 % when compared with approaches based on the first order algorithms, even disregarding differences in stopping criteria.

The local cubic convergence is attained at the cost of solving one additional linear system only differing in its right hand side per algorithmic iteration when compared with the Newton's iteration. When direct solvers are used, the increase in the computational cost is negligible. If iterative solvers are utilized, one should use inexact versions of Newton's or higher order methods, but at least in the case of Chebyshev's iteration one can reuse the constructed preconditioners and utilize deflation techniques to speed up the solution of the second system.

Both Newton's and Chebyshev's iterations have been tested within the framework of parameter continuation and adaptive mesh refinement, where neither of the methods has decisively outperformed the other. Nevertheless we hope that the ideas utilized in deriving a higher order method for this simple benchmark problem are useful in other contexts pertinent to structural and multidisciplinary optimization problems. Additionally, Tables 1 and 2 could be used for benchmarking other optimization algorithms applied to (1).

[4] Note the lack of numerical scalability owing to the utilization of the direct linear solver.

Acknowledgments The author is grateful to Martin Berggren for pointing out the reference (Carlsson et al. 2009) to us.

References

Aage N, Poulsen TH, Gersborg-Hansen A, Sigmund O (2008) Topology optimization of large scale Stokes flow problems. Struct Multidiscip Optim 35(2):175–180

Amrouche C, Girault V (1991) On the existence and regularity of the solution of Stokes problem in arbitrary dimension. Proc Jpn Acad, Series A, Math Sci 67(5):171–175

Babuška I (1973) The finite element method with penalty. Math Comput 27(122):221–228

Bangerth W, Hartmann R, Kanschat G (2007) deal.II - a general purpose object oriented finite element library. ACM Trans Math Softw 33(4):24/1–24/27

Bartish MJ (1969) On some iterative methods of solving functional equations. Sibirskij matematiceskij zurnal 10(3):488–493

Bercovier M, Pironneau O (1979) Error estimates for finite element method solution of the Stokes problem in the primitive variables. Numer Math 33(2):211–224

Borrvall T, Petersson J (2003) Topology optimization of fluids in Stokes flow. Internat J Numer Methods Fluids 41(1):77–107

Carlsson J, Sandberg M, Szepessy A (2009) Symplectic Pontryagin approximations for optimal design. ESAIM: Math Model and Numer Anal 43(1):3–32

Challis VJ, Guest JK (2009) Level-set topology optimization of fluids in Stokes flow. Int J Numer Methods Eng 79:1284–1308

Davis TA (2004) Algorithm 832: UMFPACK, an unsymmetric-pattern multifrontal method. ACM Trans Math Softw 30(2):196–199

Deng N, Zhang H (2004) Theoretical efficiency of a new inexact method of tangent hyperbolas. Optim Meth and Softw 19(3–4):247–265

Duff IS (2002) MA57 a new code for the solution of sparse symmetric definite and indefinite systems. Technical Report RAL-TR-2002-024. Rutherford Appleton Laboratory

Evgrafov A (2005) The limits of porous materials in the topology optimization of Stokes flows. Appl Math Optim 52(3):263–277

Evgrafov A (2014) State space Newtons method for topology optimization. Comput Methods Appl Mech Eng 278:272–290

Geuzaine C, Remacle J-F (2009) Gmsh A 3-d finite element mesh generator with built-in pre-and post-processing facilities. Int J Numer Methods Eng 79(11):1309–1331

Griebel M, Dornseifer T, Neunhoeffer T (1997) Numerical simulation in fluid dynamics: a practical introduction. SIAM 3

Gundersen G, Steihaug T (2012) On diagonally structured problems in unconstrained optimization using an inexact super Halley method. J Comput Appl Math 236(15):3685–3695

Hood P, Taylor C (1973) Numerical solution of the Navier–Stokes equations using the finite element technique. Comput Fluids 1:1–28

Kellogg RB, Osborn JE (1976) A regularity result for the Stokes problem in a convex polygon. J Funct Anal 21(4):397–431

Kleinmichel H (1968) Stetige Analoga und Iterationsverfahren für nichtlineare Gleichungen in BANACHräumen. Math Nachr 37(5-6):313–343

Kondoh T, Matsumori T, Kawamoto A (2012) Drag minimization and lift maximization in laminar flows via topology optimization employing simple objective function expressions based on body force integration. Struct Multidiscip Optim 45(5):693–701

Mertvecova MA (1953) Analogue of the process of tangent hyperbolas for general functional equations. Dokl Akad Nauk SSSR 88:611–614

Nechepurenko MI (1954) On Chebyshev's method for functional equations. Uspekhi Mat Nauk 9(2):163–170

Nechepurenko MI (2004) Refinement of convergence conditions for the Chebyshev method. Sib Zh Vychisl Mat 7(3):249–260

Othmer C (2008) A continuous adjoint formulation for the computation of topological and surface sensitivities of ducted flows. Internat J Numer Methods Fluids 58(8)

Seibold B (2008) A compact and fast Matlab code solving the incompressible Navier–Stokes equations on rectangular domains. http://math.mit.edu/cse/codes/mit18086_navierstokes.pdf

Steihaug T, Suleiman S (2013) Rate of convergence of higher order methods. Appl Numer Math 67:230–242

Verfürth R (1989) A posteriori error estimators for the Stokes equations. Numer Math 55(3):309–325