Shape optimization of peristaltic pumping

Shawn W. Walker *, Michael J. Shelley

New York University, Courant Institute, 251 Mercer Street, Mail Code: 0711, New York City, NY 10012-1185, United States

ARTICLE INFO

ABSTRACT

Transport is a fundamental aspect of biology and peristaltic pumping is a fundamental mechanism to accomplish this; it is also important to many industrial processes. We present a variational method for optimizing the wave shape of a peristaltic pump. Specifically, we optimize the wave profile of a two dimensional channel containing a Navier–Stokes fluid with no assumption on the wave profile other than it is a traveling wave (e.g. we do not assume it is the graph of a function). Hence, this is an infinite-dimensional optimization problem. The optimization criteria consists of minimizing the input fluid power (due to the peristaltic wave) subject to constraints on the average flux of fluid and area of the channel. Sensitivities of the cost and constraints are computed variationally via shape differential calculus and we use a sequential quadratic programming (SQP) method to find a solution of the first order KKT conditions. We also use a merit-function based line search in order to balance between decreasing the cost and keeping the constraints satisfied when updating the channel shape. Our numerical implementation uses a finite element method for computing a solution of the Navier–Stokes equations, adjoint equations, as well as for the SQP method when computing perturbations of the channel shape. The walls of the channel are deformed by an explicit front-tracking approach. In computing functional sensitivities with respect to shape, we use L^2-type projections for computing boundary stresses and for geometric quantities such as the tangent field on the channel walls and the curvature; we show error estimates for the boundary stress and tangent field approximations. As a result, we find optimized shapes that are not obvious and have not been previously reported in the peristaltic pumping literature. Specifically, we see highly asymmetric wave shapes that are far from being sine waves. Many examples are shown for a range of fluxes and Reynolds numbers up to $Re = 500$ which illustrate the capabilities of our method.

© 2009 Elsevier Inc. All rights reserved.

1. Introduction

Peristalsis is present in the esophagus, intestines, and other parts of the human body [1–3] and is able to move material through the body by sending traveling waves of contraction along the esophageal and intestinal walls. Peristaltic pumping falls into the category of a positive displacement pump because the channel/tube walls deform in order to move the interior fluid. Peristalsis is common in other biological contexts, such as blood flow in small vessels [4] and the transport of urine from the kidney to the bladder. In the female reproductive system, embryonic transport is achieved by peristalsis in the intrauterine tubes. In this case, understanding the causes of inefficient pumping may help illuminate some causes of infertility [5,6]. Another example occurs in the respiratory system. The transport of mucus from the lungs [7] may depend

* Corresponding author. Tel.: +1 212 998 3040; fax: +1 212 995 4121.
  E-mail addresses: walker@cims.nyu.edu (S.W. Walker), shelley@cims.nyu.edu (M.J. Shelley).
  URLs: http://www.cims.nyu.edu/~walker/ (S.W. Walker), http://www.math.nyu.edu/faculty/shelley/ (M.J. Shelley).

on peristalsis and it would be useful to know what affects this transport in order to better understand causes of diseases, such as cystic fibrosis. In addition, peristalsis may be linked to early lung development in the embryo [8].

In industry, peristalsis is used in mechanical pumps to move very viscous or non-Newtonian fluids through flexible deformable tubes. One common method is to have 'rollers' move over and locally compress the tube to induce a net motion of the fluid. Knowing the optimal shape of the deformation would be useful for designing the mechanical parts of the pump, like the rollers, so as to take advantage of efficient pumping profiles.

There is a large body of work on the fluid dynamics of peristalsis. Various studies seek to isolate characteristics of pumping, such as trapping of fluid into boluses and reflux of particles against the peristaltic wave [9–11], elastic wall effects on pumping [12–14], and analysis of swimming micro-organisms [15]. Some experimental work on peristalsis can be found in [16–18,5]. More recently, peristalsis of non-Newtonian fluids has been investigated [19–25] and is important for understanding real biological systems. However, little work has been done on optimizing peristalsis, though some comparisons and analysis of different wave shapes are given in [26–28].

In this paper, we present a variational method for optimizing peristaltic pumping in a two dimensional periodic channel whose upper and lower walls move to pump fluid. No prior assumption is made on the wall motion, except that it is a traveling wave in shape. Hence, we consider an infinite-dimensional optimization problem. To the best of our knowledge, optimizing the general wave shape in peristaltic pumping has never been done. Our optimization algorithm consists of finding the shape of the traveling wave such that the input fluid power is minimized subject to a given amount of mass flux and given channel volume. Wall deformations are not penalized in this paper. We first pose the method at the continuous level and use a sequential quadratic programming (SQP) method to find a minimizer. We do not assume a form for the wave shape nor do we assume it is a graph. Our numerical implementation uses explicit front-tracking for deforming the channel walls, with occasional re-meshing of the domain, and a finite element method for solving the fluid PDE and adjoint systems. In computing the sensitivities of the cost and constraints, we use computationally cheap L^2-type projections for computing boundary stresses and curvature. We also give an error estimate for our boundary stress approximation. As a result, we are able to compute optimal peristaltic pumping profiles not previously reported in the literature. In particular, for high prescribed flux, we find wave profiles that are not graphs of a function. Most wave-forms assumed in the literature have relatively mild displacements [26–28,12–14] or are sinusoidal [9–11,25]. Moreover, our optimization method can be generalized to include more complicated fluid models, such as generalized Newtonian and viscoelastic flow models.

In Section 2 we state the optimization problem. Section 3 provides a brief overview of shape differential calculus and Section 4 gives a sensitivity analysis for peristaltic pumping (Appendix gives further details). Our continuous level optimization algorithm is given in Section 5 followed by the details of our finite element implementation in Section 6. We conclude with our numerical results in Section 7 and a brief discussion in Section 8.

2. Fluid problem

First we state the physical problem. Consider a periodic channel of length L filled with fluid (see Fig. 1). Let Ω_{ALL} be a large 'hold-all' domain that contains Ω. We call Ω the fluid domain with positively oriented boundary given as a union $\partial\Omega = \overline{\Gamma} \cup \overline{\Gamma}_P$ of disjoint open sets Γ and Γ_P, where Γ refers to the top and bottom walls of the channel and Γ_P denotes the periodic end-sections of the domain. Note that Γ and Γ_P consist of two disjoint pieces: $\Gamma := \Gamma^+ \cup \Gamma^-$ and $\Gamma_P := \Gamma_P^+ \cup \Gamma_P^-$. The upper and lower boundaries Γ^+, Γ^- may or may not be mirror images. The shape of the boundary Γ is time-varying in the lab frame. We assume this motion to be a traveling wave in the direction $\mathbf{e}_1$ (along the channel), with wave speed denoted by $c > 0$.

2.1. Governing equations in lab frame

Henceforth, we scale lengths by L, velocities by c, and time by $T := L/c$. Thus, the length of Ω is 1 in the $\mathbf{e}_1$ direction. Let the fluid flow (in the fixed lab frame) obey the unsteady Navier–Stokes equations. Writing in non-dimensional form gives:

$$\partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u} - [\nabla \cdot \boldsymbol{\sigma}] = \mathbf{0}, \quad in \ \Omega,$$
$$\nabla \cdot \mathbf{u} = 0, \quad in \ \Omega \tag{1}$$

where $\boldsymbol{\sigma}$ is the stress tensor and $\mathbf{u}$ is the velocity. The Newtonian stress tensor is defined by

$$\boldsymbol{\sigma} := -pI + \frac{1}{Re}D(\mathbf{u}), \tag{2}$$

where p is the pressure, $D(\mathbf{u}) := \nabla \mathbf{u} + (\nabla \mathbf{u})^\dagger$ (i.e. the symmetrized deformation gradient), and $Re = \rho cL/\mu$. Note: ρ and μ are the density and dynamic viscosity, respectively. We do not specify the velocity boundary conditions yet; this is addressed in the next section.

2.2. Governing equations in wave frame

We rewrite the fluid equations with respect to a frame of reference translating at the wave speed c. Define the coordinate transformation from the lab frame $\mathbf{x}$ to the wave frame $\tilde{\mathbf{x}}$ as

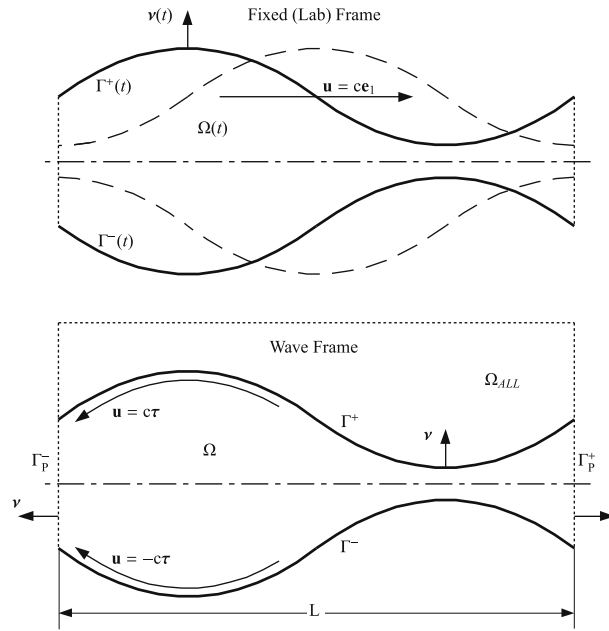$$\tilde{\mathbf{x}} := \mathbf{x} - \mathbf{e}_1 t, \tag{3}$$

Fig. 1. Peristaltic pumping of a Navier–Stokes fluid in dimensional variables. Top figure shows a 2D channel with upper and lower boundaries moving as a traveling wave. Bottom figure depicts the same setup, except in a coordinate frame moving with the traveling wave. The wavelength of the traveling wave is assumed to be L with speed c and period $T := L/c$. Note: Ω_{ALL} is a set containing Ω (only partially shown). The boundary of Ω is partitioned as $\partial\Omega = \overline{\Gamma} \cup \overline{\Gamma}_P$, where Γ and Γ_P are disjoint open sets. We denote the top and bottom walls by Γ^+ and Γ^-, respectively; likewise, the right and left ends are denoted by Γ_P^+ and Γ_P^-. The boundary is assumed to be positively oriented, with unit tangent vector τ; the outward unit normal vector is denoted ν. The boundary conditions for the fluid (in the lab frame) consist of the no-slip condition for velocity on Γ and periodic conditions for velocity and pressure on Γ_P. Since the shape of the wave is assumed to be static (see Section 2.2), the velocity is tangential on Γ in the wave frame. Moreover, assuming the walls are inextensible implies that the velocity on Γ is proportional to the tangent vector.

where the speed of translation is scaled to the non-dimensional unit wave speed. Next assume that, relative to the wave frame, the fluid variables

$$
\begin{aligned}
\tilde{\mathbf{u}}(\tilde{\mathbf{x}}) &:= \mathbf{u}(t,\mathbf{x}) - \mathbf{e}_1, \\
\tilde{p}(\tilde{\mathbf{x}}) &:= p(t,\mathbf{x}),
\end{aligned}
\tag{4}
$$

are independent of time. The fluid stress in the wave frame is also independent of time, and is given by (Galilean invariance)

$$
\tilde{\boldsymbol{\sigma}}(\tilde{\mathbf{x}}) := -\tilde{p}(\tilde{\mathbf{x}})I + \frac{1}{Re}D_{\tilde{\mathbf{x}}}(\tilde{\mathbf{u}}(\tilde{\mathbf{x}})) = -p(t,\mathbf{x})I + \frac{1}{Re}D_{\mathbf{x}}(\mathbf{u}(t,\mathbf{x})) = \boldsymbol{\sigma}(\mathbf{x},t).
\tag{5}
$$

The motion of Γ is assumed to be independent of time when viewed in the wave frame. This gives that the velocity $\tilde{\mathbf{u}}$ is completely tangential (on Γ). In addition, we assume that Γ is (locally) inextensible, implying that velocity is proportional to the oriented unit tangent vector τ (see Fig. 1).

Deriving the transformed system proceeds as follows. Since $\tilde{\mathbf{u}}(\tilde{\mathbf{x}}) = \mathbf{u}(t,\tilde{\mathbf{x}} + \mathbf{e}_1 t) - \mathbf{e}_1$, we have that

$$
\begin{aligned}
\mathbf{0} = \partial_t\tilde{\mathbf{u}}(\tilde{\mathbf{x}}) &= \partial_t\mathbf{u}(t,\tilde{\mathbf{x}} + \mathbf{e}_1 t) + (\mathbf{e}_1 \cdot \nabla_{\mathbf{x}})\mathbf{u}(t,\tilde{\mathbf{x}} + \mathbf{e}_1 t), \\
&= \partial_t\mathbf{u}(t,\mathbf{x}) + (\mathbf{e}_1 \cdot \nabla_{\mathbf{x}})\mathbf{u}(t,\mathbf{x}).
\end{aligned}
\tag{6}
$$

Ergo, the fluid momentum equation (starting in the lab frame) becomes

$$
\partial_t\mathbf{u} + (\mathbf{e}_1 \cdot \nabla)\mathbf{u} + ((\mathbf{u} - \mathbf{e}_1) \cdot \nabla)\mathbf{u} - [\nabla \cdot \boldsymbol{\sigma}] = ((\mathbf{u}(t,\mathbf{x}) - \mathbf{e}_1) \cdot \nabla)\mathbf{u} - [\nabla \cdot \boldsymbol{\sigma}] = (\tilde{\mathbf{u}} \cdot \nabla)\tilde{\mathbf{u}} - [\nabla \cdot \tilde{\boldsymbol{\sigma}}] = \mathbf{0},
\tag{7}
$$

where in the last line we transform to the wave frame. Therefore, after dropping the $\sim$ notation, the governing fluid equations in the wave frame now reads as:

$$
\begin{aligned}
(\mathbf{u} \cdot \nabla)\mathbf{u} - [\nabla \cdot \boldsymbol{\sigma}] &= \mathbf{0}, \quad in \ \Omega, \\
\nabla \cdot \mathbf{u} &= 0, \quad in \ \Omega, \\
\mathbf{u} &= +\tau, \quad on \ \Gamma^+, \\
\mathbf{u} &= -\tau, \quad on \ \Gamma^-,
\end{aligned}
\tag{8}
$$

with periodic boundary conditions for $\mathbf{u}$ and p applied on Γ_P. Assuming sufficient smoothness of the velocity implies $\boldsymbol{\sigma}|_{\Gamma_P^+} = \boldsymbol{\sigma}|_{\Gamma_P^-}$. Note that $\mathbf{u} = \pm\tau$ on Γ because τ is the positively oriented tangent vector. Note that this is the free-pumping regime of peristalsis because there is no net rise in pressure per wavelength of the channel.

2.3. Relevant quantities

Our optimization algorithm will involve three main quantities: (1) the average mass flow rate Q produced by the pump in the fixed lab frame, (2) the net power J dissipated by the pump, and (3) the total volume (area) A in one wavelength of the channel. These quantities are described in the following sections.

In order to avoid excessive notation when writing integrals, we will usually refrain from writing the function arguments, such as the (x, y) coordinates of a vector field. In this case, we also drop the differential notation, i.e. dx, dy. The function arguments and differential measure are implied by the domain of integration.

2.3.1. Mass flow rate

The non-dimensional average rate of mass flow per wavelength in the lab frame is given by

$$Q(\Omega, \mathbf{u}) = \int_0^1 \int_{\Theta(x)} (\mathbf{u}(x, y) + \mathbf{e}_1) \cdot \mathbf{e}_1 \, dy \, dx = \int_\Omega (\mathbf{u} + \mathbf{e}_1) \cdot \mathbf{e}_1 = \int_\Omega \mathbf{u} \cdot \mathbf{e}_1 + |\Omega|, \tag{9}$$

where $\Theta(x)$ is a vertical slice of Ω at x and $\mathbf{u}$ is the fluid velocity in the wave frame. The dimensional mass flow rate per wavelength is obtained by scaling: $\rho c L \cdot Q$. If the shape of the channel is a rectangle, then $\mathbf{u} = -\mathbf{e}_1$, and so $Q = 0$. If the channel is fully occluded (i.e. the top and bottom walls touch), then $Q = |\Omega|$. For all other channel shapes, $0 \leqslant Q \leqslant |\Omega|$.

2.3.2. Power loss

The non-dimensional fluid input power functional (in the lab frame) is given by

$$J(\Omega, (\mathbf{u}, p)) = \int_{\partial\Omega} (\mathbf{u} + \mathbf{e}_1) \cdot \boldsymbol{\sigma}\nu = \int_\Gamma (\mathbf{u} + \mathbf{e}_1) \cdot \boldsymbol{\sigma}\nu, \tag{10}$$

where ν is the outer unit normal to the fluid domain Ω, and $\mathbf{u}$ and $\boldsymbol{\sigma}$ are the velocity and stress tensor in the wave-frame. The dimensional power loss is obtained by scaling: $\rho c^3 L \cdot J$. Note, $J > 0$.

2.4. The minimization problem

The optimization problem is as follows. We wish to find the shape of the fluid domain Ω such that J is minimized subject to the constraints that $Q = C_Q$ and $|\Omega| = C_A$ for fixed positive constants C_Q and C_A. More precisely, let $\mathcal{O}$ be the set of admissible shapes for the fluid domain Ω:

$$\mathcal{O} = \left\{ \Omega \subset \Omega_{ALL}, \ \Omega \text{ is periodic and simply connected} : Q(\Omega, \mathbf{u}(\Omega)) = C_Q, \ |\Omega| = C_A, \ \Gamma \text{ is } C^2 \right\}, \tag{11}$$

where $(\mathbf{u}, p)$ is a solution of (8). Then the minimization problem is stated as follows. Find an *optimal pair* $(\Omega^*, (\mathbf{u}^*, p^*))$ such that

$$J(\Omega^*, (\mathbf{u}^*(\Omega^*), p^*(\Omega^*))) = \min_{\Omega \in \mathcal{O}} J(\Omega, (\mathbf{u}(\Omega), p(\Omega))), \tag{12}$$

where $(\mathbf{u}(\Omega), p(\Omega))$ solves (8) on Ω. The set $\mathcal{O}$ is general, in that we are not restricting the optimization to a small set of parameters (i.e. we are not assuming a known form for Γ nor do we assume that Γ is the graph of a function; see our results in Section 7).

3. Overview of shape sensitivity analysis

In order to optimize the general shape of the peristaltic pump, we make use of the shape derivative via the speed method. This gives us a method of deforming (or transforming) the fluid domain that is convenient for computing the derivative of domain or boundary functionals with respect to the deformation. In this section, we briefly review the general shape derivative for 2 or 3 dimensional domains; details can be found in [29–34]. In the following sections, and in computing the shape derivatives (see Section 4), we assume *sufficient smoothness* to make the arguments valid.

3.1. The velocity (speed) method

Let Ω be a smooth domain, whose shape we wish to optimize and is contained in a larger, fixed 'hold-all' domain: $\Omega \subset \Omega_{ALL} \subset \mathbb{R}^d$ for $d = 2$ or 3 (see Fig. 2). Let $\mathbf{V}(s, \cdot) : \Omega_{ALL} \to \mathbb{R}^d$ be a smooth velocity field for $s \in [0, s_{final}]$ and assume it is smooth with respect to s and $\mathbf{V} = 0$ on $\partial\Omega_{ALL}$. Next, let $\mathbf{a} \in \Omega_{ALL}$ be fixed and $\boldsymbol{\Phi}(\cdot; \mathbf{a}) : [0, s_{final}] \to \Omega_{ALL}$ be a vector function of s that satisfies the initial value problem:

$$\frac{d}{ds} \boldsymbol{\Phi}(s; \mathbf{a}) = \mathbf{V}(s, \boldsymbol{\Phi}(s; \mathbf{a})), \quad s \in [0, s_{final}],$$
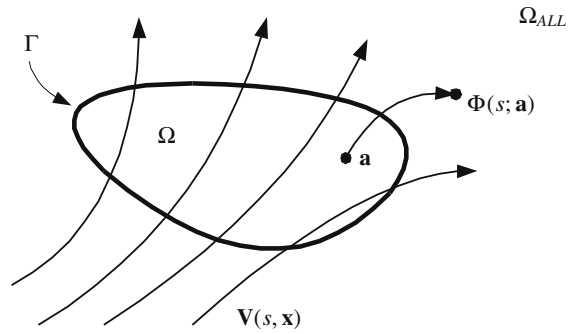$$\boldsymbol{\Phi}(0; \mathbf{a}) = \mathbf{a}, \quad \mathbf{a} \in \Omega_{ALL}. \tag{13}$$

Fig. 2. Velocity field **V** deforms a domain. The deformation evolves for increasing s (i.e. s is a 'fake' time variable). The Lagrangian flow map is denoted $\Phi(s; \cdot)$ and is the map that takes points in the initial configuration to the deformed configuration.

The vector **a** can be thought of as a *material* label for the point $\Phi(s; \mathbf{a})$ parameterized by s. Thus, $\Phi(s; \cdot) : \Omega_{ALL} \to \Omega_{ALL}$ and $\Phi(s; \mathbf{a})$ tracks the motion of **a** as s varies. Note that $\Phi(s; \cdot)$ is a smooth bijective transformation that depends on **V** and $\Phi(0; \mathbf{a})$ is the identity map on Ω_{ALL}. We use this map to define a parameterized family of domains and corresponding boundaries by $\Omega_s = \Phi(s; \Omega)$ and $\Gamma_s = \Phi(s; \Gamma)$. Note that $\Phi(0; \Omega) = \Omega_0 = \Omega$ and $\Phi(0; \Gamma) = \Gamma_0 = \Gamma$. In the following sections, we review how functionals that depend on Ω_s vary with changes in s.

3.2. Shape derivative operations

Let $J(\Omega_s) := \int_{\Omega_s} f(\Omega_s)$. The material derivative (first variation) of the functional J depending on Ω_s is defined as

$$\delta J(\Omega; \mathbf{V}) := \lim_{s \to 0} \frac{J(\Omega_s(\mathbf{V})) - J(\Omega)}{s}. \tag{14}$$

The material derivative of a function that depends on Ω_s (e.g. $f(\Omega_s)(\cdot) : \Omega_s \to \mathbb{R}$) is

$$\dot{f}(\Omega; \mathbf{V})(\cdot) := \lim_{s \to 0} \frac{f(\Phi(s; \cdot), s) - f(\cdot, s)}{s}. \tag{15}$$

The *shape* derivative of a function (Eulerian partial derivative) is simply

$$f'(\Omega; \mathbf{V}) := \dot{f}(\Omega; \mathbf{V}) - \nabla_{\mathbf{x}} f(\Omega) \cdot \mathbf{V}|_{s=0}. \tag{16}$$

In other words, the shape derivative is just the partial derivative with respect to deforming the domain but without the convective effect of moving the domain. This is especially important if f is the solution of a PDE in the domain Ω.

In the rest of this paper, we will drop the '$s = 0$' notation in $\mathbf{V}|_{s=0}$ and just write **V**. Only the value at $s = 0$ is important for our calculations. We now state the *Reynolds transport theorem* in the context of shape differentiation, which says that the material derivative in (14) can be written as

$$\delta J(\Omega; \mathbf{V}) = \int_{\Omega} f'(\Omega; \mathbf{V}) + \int_{\Gamma} f(\Omega)(\mathbf{V} \cdot \mathbf{v}). \tag{17}$$

Note that in standard fluid mechanics, f' is just the partial derivative of f with respect to time.

3.3. The Hadamard–Zolésio structure theorem

The following theorem states that the shape derivative of a functional can always be represented as an integral on the boundary of a measure multiplied by the normal component of the perturbation.

Theorem 1 (See Theorem 3.5 and Corollary 1 in [30]). *The Eulerian (or directional) derivative of a shape functional J always has a representation of the form*

$$\delta J(\Omega; \mathbf{V}) = \langle g, \mathbf{V} \cdot \mathbf{v} \rangle_{\Gamma},$$

where $\langle \cdot, \cdot \rangle_{\Gamma}$ is a duality pairing on Γ and g is a measure defined on Γ. In the case when g is sufficiently smooth, we have

$$\delta J(\Omega; \mathbf{V}) = \int_{\Gamma} g(\mathbf{V} \cdot \mathbf{v}).$$

We make note of Theorem 1 because our shape derivative calculation slightly deviates from this (see Remark 1).

4. Sensitivity analysis for peristaltic wave shape

In the following, we note that Γ^+ and Γ^- are disjoint periodic surfaces (curves). So any integration by parts on Γ^+ (or Γ^-) will have no boundary terms.

4.1. Computing the shape derivative of J

We can manipulate J into an integral over the bulk:

$$J(\mathbf{u}, p) = \int_{\partial\Omega} (\mathbf{u} + \mathbf{e}_1) \cdot \boldsymbol{\sigma}v, = \int_{\Omega} \{(\mathbf{u} \cdot \nabla)\mathbf{u} - [\nabla \cdot \boldsymbol{\sigma}]\} \cdot (\mathbf{u} + \mathbf{e}_1) + \int_{\Omega} \mathbf{e}_1 \cdot [\nabla \cdot \boldsymbol{\sigma}] + \int_{\partial\Omega} \mathbf{u} \cdot (\boldsymbol{\sigma}v),$$

$$= \int_{\Omega} [(\mathbf{u} \cdot \nabla)\mathbf{u}] \cdot (\mathbf{u} + \mathbf{e}_1) + \int_{\Omega} \boldsymbol{\sigma} : \nabla\mathbf{u} = \int_{\Omega} \boldsymbol{\sigma} : \nabla\mathbf{u}, \tag{18}$$

where we used (8), divergence theorem, integration by parts, and (2). By (84), the convective term vanishes, and the remaining term is the rate of viscous dissipation.

4.1.1. PDE for the shape derivative

Computing δJ will require the shape derivative of the solution $(\mathbf{u}, p)$ of (8), which is denoted $(\mathbf{u}', p')$ and solves the following linear, variable coefficient PDE:

$$(\mathbf{u}' \cdot \nabla)\mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u}' - [\nabla \cdot \boldsymbol{\sigma}'] = \mathbf{0}, \quad in \ \Omega,$$
$$\nabla \cdot \mathbf{u}' = 0, \quad in \ \Omega, \tag{19}$$
$$\mathbf{u}' = \dot{\mathbf{u}} - [(\mathbf{V} \cdot \nabla)\mathbf{u}], \quad on \ \Gamma,$$

where the boundary condition on Γ is given by the definition of $\mathbf{u}'$; $\dot{\mathbf{u}}$ is given by Eq. (79) in Appendix. Just as in (8), we have periodic boundary conditions on Γ_P and the stress tensor $\boldsymbol{\sigma}'$ is similarly given by

$$\boldsymbol{\sigma}' = -p'\mathbf{I} + \frac{1}{Re}D(\mathbf{u}'). \tag{20}$$

One can derive (19) by (formally) applying the shape derivative operator to (8) and commuting derivatives.

4.1.2. Computing δJ

Making note of the definition of $\boldsymbol{\sigma}$ and periodicity, we have:

$$\delta J = \int_{\Omega} \boldsymbol{\sigma}' : \nabla\mathbf{u} + \int_{\Omega} \boldsymbol{\sigma} : \nabla\mathbf{u}' + \int_{\Gamma} (\mathbf{V} \cdot v)\boldsymbol{\sigma} : \nabla\mathbf{u},$$

$$= -\int_{\Omega} (\nabla \cdot \boldsymbol{\sigma}') \cdot \mathbf{u} + \int_{\partial\Omega} \mathbf{u} \cdot \boldsymbol{\sigma}'v - \int_{\Omega} (\nabla \cdot \boldsymbol{\sigma}) \cdot \mathbf{u}' + \int_{\partial\Omega} \mathbf{u}' \cdot \boldsymbol{\sigma}v + \int_{\Gamma} (\mathbf{V} \cdot v)\boldsymbol{\sigma} : \nabla\mathbf{u},$$

$$= -\int_{\Omega} \{[(\mathbf{u}' \cdot \nabla)\mathbf{u}] + [(\mathbf{u} \cdot \nabla)\mathbf{u}']\} \cdot \mathbf{u} + \int_{\Gamma} \mathbf{u} \cdot \boldsymbol{\sigma}'v - \int_{\Omega} [(\mathbf{u} \cdot \nabla)\mathbf{u}] \cdot \mathbf{u}' + \int_{\Gamma} \mathbf{u}' \cdot \boldsymbol{\sigma}v + \int_{\Gamma} (\mathbf{V} \cdot v)\boldsymbol{\sigma} : \nabla\mathbf{u},$$

where we performed an integration by parts and used (19) and (8). Next, by (85), (87) and (16), we get

$$\delta J = \int_{\Gamma} \mathbf{u} \cdot \boldsymbol{\sigma}'v + \int_{\Gamma} \mathbf{u}' \cdot \boldsymbol{\sigma}v + \int_{\Gamma} (\mathbf{V} \cdot v)\boldsymbol{\sigma} : \nabla\mathbf{u},$$

$$= \int_{\Gamma} \{\mathbf{u} \cdot \boldsymbol{\sigma}'v + \dot{\mathbf{u}} \cdot \boldsymbol{\sigma}v - (\mathbf{V} \cdot v)[(v \cdot \nabla)\mathbf{u}] \cdot \boldsymbol{\sigma}v - [(\mathbf{V} \cdot \nabla_\Gamma)\mathbf{u}] \cdot \boldsymbol{\sigma}v + (\mathbf{V} \cdot v)\boldsymbol{\sigma} : \nabla\mathbf{u}\},$$

$$= \int_{\Gamma} \{\mathbf{u} \cdot \boldsymbol{\sigma}'v + \dot{\mathbf{u}} \cdot \boldsymbol{\sigma}v - (\mathbf{V} \cdot v)[(v \cdot \nabla)\mathbf{u}] \cdot v(v \cdot \boldsymbol{\sigma}v) - (\mathbf{V} \cdot v)[(v \cdot \nabla)\mathbf{u}] \cdot \tau(\tau \cdot \boldsymbol{\sigma}v)$$
$$\quad - [(\mathbf{V} \cdot \nabla_\Gamma)\mathbf{u}] \cdot \boldsymbol{\sigma}v + (\mathbf{V} \cdot v)(\tau \cdot D(\mathbf{u})v)(\tau \cdot \boldsymbol{\sigma}v)\},$$

$$= \int_{\Gamma} \{\mathbf{u} \cdot \boldsymbol{\sigma}'v + \dot{\mathbf{u}} \cdot \boldsymbol{\sigma}v - [(\mathbf{V} \cdot \nabla_\Gamma)\mathbf{u}] \cdot \boldsymbol{\sigma}v + (\mathbf{V} \cdot v)[\tau \cdot D(\mathbf{u})v - [(v \cdot \nabla)\mathbf{u}] \cdot \tau](\tau \cdot \boldsymbol{\sigma}v)\},$$

$$= \int_{\Gamma} \{\mathbf{u} \cdot \boldsymbol{\sigma}'v + \dot{\mathbf{u}} \cdot \boldsymbol{\sigma}v - [(\mathbf{V} \cdot \nabla_\Gamma)\mathbf{u}] \cdot \boldsymbol{\sigma}v + (\mathbf{V} \cdot v)([(\tau \cdot \nabla)\mathbf{u}] \cdot v)(\tau \cdot \boldsymbol{\sigma}v)\},$$

where we used (80) and (83). Note that $[\nabla_\Gamma\mathbf{u} : \boldsymbol{\sigma}] = \tau \cdot \boldsymbol{\sigma}(\partial_\tau\mathbf{u}) = (\tau \cdot \boldsymbol{\sigma}v)[(\tau \cdot \nabla)\mathbf{u}] \cdot v$, because $\nabla_\Gamma = \tau\partial_\tau$ for a 1D boundary. So we get:

$$\delta J = \int_{\Gamma} \mathbf{u} \cdot \boldsymbol{\sigma}'v + \dot{\mathbf{u}} \cdot \boldsymbol{\sigma}v + (\mathbf{V} \cdot v)\nabla_\Gamma\mathbf{u} : \boldsymbol{\sigma} - [(\mathbf{V} \cdot \nabla_\Gamma)\mathbf{u}] \cdot \boldsymbol{\sigma}v, \tag{21}$$

4.1.3. Adjoint PDE for δJ

The $\sigma' v$ term in (21) depends implicitly on the shape perturbation $\mathbf{V}$. Therefore, to avoid computing a solution of (19) for every perturbation $\mathbf{V}$, we rewrite that term in a more convenient form using the following adjoint problem. Let $\mathbf{S}$ be the stress tensor:

$$\mathbf{S} = -\pi I + \frac{1}{Re} D(\mathbf{r}), \tag{22}$$

where $(\pi, \mathbf{r})$ solves the following adjoint PDE:

$$-[\nabla \mathbf{r}]^\dagger \mathbf{u} - (\mathbf{u} \cdot \nabla) \mathbf{r} - [\nabla \cdot \mathbf{S}] = \mathbf{0}, \quad in\ \Omega,$$
$$\nabla \cdot \mathbf{r} = 0, \quad in\ \Omega, \tag{23}$$
$$\mathbf{r} = \mathbf{u}, \quad on\ \Gamma,$$

with periodic boundary conditions applied on Γ_P. Note that $[\nabla \mathbf{r}]^\dagger \mathbf{u} = [\partial_i r_j] u_j$ using Einstein summation. In the next section, we write δJ in a way that only involves solving (23) once for all perturbations.

4.1.4. Evaluating δJ with the adjoint

We make use of the shape derivative PDE (19) and integration by parts:

$$0 = \int_\Omega \{(\mathbf{u}' \cdot \nabla)\mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u}'\} \cdot \mathbf{r} - \int_\Omega (\nabla \cdot \sigma') \cdot \mathbf{r},$$
$$= -\int_\Omega \{[\nabla \mathbf{r}]^\dagger \mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{r}\} \cdot \mathbf{u}' + \int_\Gamma (\mathbf{u}' \cdot v)(\mathbf{u} \cdot \mathbf{r}) + \int_\Omega \sigma' : \nabla \mathbf{r} - \int_{\partial\Omega} \mathbf{r}^\dagger \sigma' v,$$
$$= \int_\Omega (\nabla \cdot \mathbf{S}) \cdot \mathbf{u}' + \int_\Omega \frac{1}{Re} D(\mathbf{u}') : \nabla \mathbf{r} - \int_\Omega p'(\nabla \cdot \mathbf{r}) - \int_\Gamma \mathbf{r}^\dagger \sigma' v + \int_\Gamma (\mathbf{u}' \cdot v)(\mathbf{u} \cdot \mathbf{r}),$$
$$= \int_\Omega (\nabla \cdot \mathbf{S}) \cdot \mathbf{u}' + \frac{1}{Re} \int_\Omega D(\mathbf{u}') : \nabla \mathbf{r} - \int_\Gamma \mathbf{r}^\dagger \sigma' v + \int_\Gamma (\mathbf{u}' \cdot v)(\mathbf{u} \cdot \mathbf{r}),$$
$$= \int_\Omega (\nabla \cdot \mathbf{S}) \cdot \mathbf{u}' + \int_\Omega \nabla \mathbf{u}' : \frac{1}{Re} D(\mathbf{r}) - \int_\Gamma \mathbf{r}^\dagger \sigma' v + \int_\Gamma (\mathbf{u}' \cdot v),$$
$$= \int_\Omega (\nabla \cdot \mathbf{S}) \cdot \mathbf{u}' + \int_\Omega \nabla \mathbf{u}' : \mathbf{S} - \int_\Gamma \mathbf{u} \cdot \sigma' v,$$

where we used the symmetry of $D(\cdot)$, the original PDE (8), the adjoint stress (22) and solution (23), (84), and the fact that $\nabla \cdot \mathbf{u}' = 0$. Performing another integration by parts, we have

$$0 = \int_\Omega (\nabla \cdot \mathbf{S}) \cdot \mathbf{u}' + \int_\Omega -\mathbf{u}' \cdot (\nabla \cdot \mathbf{S}) + \int_{\partial\Omega} (\mathbf{u}')^\dagger \mathbf{S} v - \int_\Gamma \mathbf{u} \cdot \sigma' v = \int_\Gamma \mathbf{u}' \cdot \mathbf{S} v - \int_\Gamma \mathbf{u} \cdot \sigma' v.$$

Therefore (see (21)), using the shape derivative PDE (19) gives

$$\delta J(\mathbf{u}, p; \mathbf{V}) = \int_\Gamma \dot{\mathbf{u}} \cdot \sigma v + \mathbf{u}' \cdot \mathbf{S} v + (\mathbf{V} \cdot v)\nabla_\Gamma \mathbf{u} : \sigma - [(\mathbf{V} \cdot \nabla_\Gamma)\mathbf{u}] \cdot \sigma v = \int_\Gamma \dot{\mathbf{u}} \cdot (\sigma v + \mathbf{S} v) - [(\mathbf{V} \cdot \nabla)\mathbf{u}] \cdot \mathbf{S} v + (\mathbf{V} \cdot v)\nabla_\Gamma \mathbf{u}$$
$$: \sigma - [(\mathbf{V} \cdot \nabla_\Gamma)\mathbf{u}] \cdot \sigma v. \tag{24}$$

Referring to the appendix (79), we have that $\dot{\mathbf{u}} = (\mathbf{u} \cdot \tau)(v \cdot \partial_\tau \mathbf{V})v$ (note: $(\mathbf{u} \cdot \tau) = \pm 1$ on Γ). Thus, further manipulation of (24) leads to

$$\delta J = \int_\Gamma (\mathbf{u} \cdot \tau)(v \cdot \partial_\tau \mathbf{V})(v \cdot [\sigma v + \mathbf{S} v]) - [(\mathbf{V} \cdot \nabla)\mathbf{u}] \cdot \mathbf{S} v + (\mathbf{u} \cdot \tau)\{(\mathbf{V} \cdot v)\tau \cdot (\sigma \partial_\tau \tau) - (\mathbf{V} \cdot \tau)[\partial_\tau \tau] \cdot \sigma v\},$$
$$= \int_\Gamma (\mathbf{u} \cdot \tau)(v \cdot \partial_\tau \mathbf{V})(v \cdot [\sigma v + \mathbf{S} v]) - [(\mathbf{V} \cdot \nabla)\mathbf{u}] \cdot \mathbf{S} v + (\mathbf{u} \cdot \tau)\{-(\mathbf{V} \cdot v)\kappa(\tau \cdot \sigma v) + (\mathbf{V} \cdot \tau)\kappa(v \cdot \sigma v)\}. \tag{25}$$

The remaining term can be written as follows:

$$[(\mathbf{V} \cdot \nabla)\mathbf{u}] \cdot v = (\mathbf{V} \cdot v)[(v \cdot \nabla)\mathbf{u}] \cdot v + (\mathbf{V} \cdot \tau)[\partial_\tau \mathbf{u}] \cdot v = (\mathbf{u} \cdot \tau)(\mathbf{V} \cdot \tau)[-\kappa v] \cdot v = -(\mathbf{u} \cdot \tau)(\mathbf{V} \cdot \tau)\kappa, \tag{26}$$

using previous results. The tangential component is given by

$$[(\mathbf{V} \cdot \nabla)\mathbf{u}] \cdot \tau = (\mathbf{V} \cdot v)[(v \cdot \nabla)\mathbf{u}] \cdot \tau + (\mathbf{V} \cdot \tau)[\partial_\tau \mathbf{u}] \cdot \tau = (\mathbf{V} \cdot v)[(v \cdot \nabla)\mathbf{u}] \cdot \tau,$$
$$= (\mathbf{V} \cdot v)\{Re\,\tau \cdot \sigma v - [(\tau \cdot \nabla)\mathbf{u}] \cdot v\} = (\mathbf{V} \cdot v)\{Re\,\tau \cdot \sigma v - (\mathbf{u} \cdot \tau)[\partial_\tau \tau] \cdot v\}, \tag{27}$$
$$= (\mathbf{V} \cdot v)\{Re\,\tau \cdot \sigma v + \kappa(\mathbf{u} \cdot \tau)\}.$$

So, Eq. (25) now becomes

$$\delta J = \int_{\Gamma} (\mathbf{u} \cdot \boldsymbol{\tau})\{(v \cdot \partial_{\tau}\mathbf{V})(v \cdot [\boldsymbol{\sigma}v + \mathbf{S}v]) - (\mathbf{V} \cdot v)\kappa(\boldsymbol{\tau} \cdot \boldsymbol{\sigma}v) + (\mathbf{V} \cdot \boldsymbol{\tau})\kappa(v \cdot \boldsymbol{\sigma}v)\} - [(\mathbf{V} \cdot \nabla)\mathbf{u}] \cdot v(v \cdot \mathbf{S}v) - [(\mathbf{V} \cdot \nabla)\mathbf{u}] \cdot \boldsymbol{\tau}(\boldsymbol{\tau} \cdot \mathbf{S}v)$$

$$= \int_{\Gamma} (\mathbf{u} \cdot \boldsymbol{\tau})\{(v \cdot \partial_{\tau}\mathbf{V})(v \cdot [\boldsymbol{\sigma}v + \mathbf{S}v]) - (\mathbf{V} \cdot v)\kappa(\boldsymbol{\tau} \cdot \boldsymbol{\sigma}v) + (\mathbf{V} \cdot \boldsymbol{\tau})\kappa(v \cdot \boldsymbol{\sigma}v)\}(\mathbf{u} \cdot \boldsymbol{\tau})(\mathbf{V} \cdot \boldsymbol{\tau})\kappa(v \cdot \mathbf{S}v) - (\mathbf{V} \cdot v)\{Re\,\boldsymbol{\tau} \cdot \boldsymbol{\sigma}v + \kappa(\mathbf{u} \cdot \boldsymbol{\tau})\}(\boldsymbol{\tau} \cdot \mathbf{S}v)$$

$$= \int_{\Gamma} (\mathbf{u} \cdot \boldsymbol{\tau})\{(v \cdot \partial_{\tau}\mathbf{V})v \cdot (\boldsymbol{\sigma}v + \mathbf{S}v) + \kappa[(\mathbf{V} \cdot \boldsymbol{\tau})v \cdot (\boldsymbol{\sigma}v + \mathbf{S}v) - (\mathbf{V} \cdot v)\boldsymbol{\tau} \cdot (\boldsymbol{\sigma}v + \mathbf{S}v)]\} - Re \int_{\Gamma} (\mathbf{V} \cdot v)(\boldsymbol{\tau} \cdot \boldsymbol{\sigma}v)(\boldsymbol{\tau} \cdot \mathbf{S}v), \tag{28}$$

where we recall that $\mathbf{u} \cdot \boldsymbol{\tau} = +1$ on Γ^+ and $\mathbf{u} \cdot \boldsymbol{\tau} = -1$ on Γ^-. Note that integrating by parts gives the following result

$$\delta J = \int_{\Gamma} [(\mathbf{u} \cdot \boldsymbol{\tau})\{v \cdot \partial_{\tau}(\boldsymbol{\sigma}v + \mathbf{S}v) - 2\kappa\boldsymbol{\tau} \cdot (\boldsymbol{\sigma}v + \mathbf{S}v)\} - Re(\boldsymbol{\tau} \cdot \boldsymbol{\sigma}v)(\boldsymbol{\tau} \cdot \mathbf{S}v)](\mathbf{V} \cdot v), \tag{29}$$

which is consistent with Theorem 1.

Remark 1. Even though (29) satisfies the structure theorem (see Theorem 1), Eq. (28) is more convenient because it requires less regularity of the solution of the Navier–Stokes equations (8) and adjoint system (23). Thus, we actually use (28) in computing δJ in our optimization algorithm (see Section 5). This requires more regularity of $\mathbf{V}$ but is compensated by solving the variational problem in (43).

4.2. Computing the shape derivative of Q

Next, we compute the derivative of Q with respect to deforming the domain. We split the calculation into two pieces. Recall from (9) (in the wave frame):

$$Q_A(\mathbf{u}, p) = \int_{\Omega} 1, \qquad Q_B(\mathbf{u}, p) = \int_{\Omega} \mathbf{u} \cdot \mathbf{e}_1, \qquad Q(\mathbf{u}, p) = Q_A + Q_B. \tag{30}$$

By formula (17), we have

$$\delta Q_A(\Omega; \mathbf{V}) = \int_{\partial\Omega} \mathbf{V} \cdot v, \qquad \delta Q_B(\Omega; \mathbf{V}) = \int_{\Omega} \mathbf{u}' \cdot \mathbf{e}_1 + \int_{\partial\Omega} \mathbf{u} \cdot \mathbf{e}_1(\mathbf{V} \cdot v), \tag{31}$$

Just as in Section 4.1.3, we need an adjoint problem.

4.2.1. Adjoint PDE for δQ_B

Let $\mathbf{H}(\varrho, \mathbf{z})$ be the stress tensor defined in (22), where $(\varrho, \mathbf{z})$ solves the following adjoint system:

$$\begin{aligned} &-[\nabla\mathbf{z}]^{\dagger}\mathbf{u} - (\mathbf{u} \cdot \nabla)\mathbf{z} - [\nabla \cdot \mathbf{H}] = \mathbf{e}_1, \quad \text{in } \Omega \\ &\nabla \cdot \mathbf{z} = 0, \quad \text{in } \Omega \\ &\mathbf{z} = \mathbf{0}, \quad \text{on } \Gamma, \end{aligned} \tag{32}$$

where $\mathbf{u}$ solves (8) and $(\varrho, \mathbf{z})$ satisfies periodic boundary conditions on Γ_P.

4.2.2. Evaluating δQ_B with the adjoint

In the following, we make use of (84) and the boundary conditions of (32):

$$\begin{aligned} \int_{\Omega} \mathbf{u}' \cdot \mathbf{e}_1 &= -\int_{\Omega} \{[\nabla\mathbf{z}]^{\dagger}\mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{z}\} \cdot \mathbf{u}' - \int_{\Omega} \mathbf{u}' \cdot (\nabla \cdot \mathbf{H}), \\ &= \int_{\Omega} \{(\mathbf{u}' \cdot \nabla)\mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u}'\} \cdot \mathbf{z} - \int_{\Gamma} (\mathbf{u}' \cdot v)(\mathbf{u} \cdot \mathbf{z}) - \int_{\partial\Omega} \mathbf{u}' \cdot (\mathbf{H}v) + \int_{\Omega} \nabla\mathbf{u}' : \mathbf{H}, \\ &= \int_{\Omega} \{(\mathbf{u}' \cdot \nabla)\mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u}'\} \cdot \mathbf{z} - \int_{\Gamma} \mathbf{u}' \cdot (\mathbf{H}v) + \int_{\Omega} \boldsymbol{\sigma}' : \nabla\mathbf{z}, \\ &= \int_{\Omega} \{(\mathbf{u}' \cdot \nabla)\mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u}' - [\nabla \cdot \boldsymbol{\sigma}']^{\dagger}\} \cdot \mathbf{z} - \int_{\Gamma} \mathbf{u}' \cdot (\mathbf{H}v) + \int_{\partial\Omega} \mathbf{z} \cdot \boldsymbol{\sigma}'v, \\ &= -\int_{\Gamma} \mathbf{u}' \cdot (\mathbf{H}v) = -\int_{\Gamma} \dot{\mathbf{u}} \cdot (\mathbf{H}v) + \int_{\Gamma} [(\mathbf{V} \cdot \nabla)\mathbf{u}] \cdot (\mathbf{H}v), \end{aligned} \tag{33}$$

where we used (19) in the last line. Simplifying, we get:

$$\begin{aligned} \int_{\Omega} \mathbf{u}' \cdot \mathbf{e}_1 &= -\int_{\Gamma} (\mathbf{u} \cdot \boldsymbol{\tau})\dot{\boldsymbol{\tau}} \cdot (\mathbf{H}v) + \int_{\Gamma} \{[(\mathbf{V} \cdot \nabla)\mathbf{u}] \cdot v(v \cdot \mathbf{H}v) + [(\mathbf{V} \cdot \nabla)\mathbf{u}] \cdot \boldsymbol{\tau}(\boldsymbol{\tau} \cdot \mathbf{H}v)\}, \\ &= -\int_{\Gamma} (\mathbf{u} \cdot \boldsymbol{\tau})[(v \cdot \partial_{\tau}\mathbf{V}) + (\mathbf{V} \cdot \boldsymbol{\tau})\kappa](v \cdot \mathbf{H}v) + \int_{\Gamma} (\mathbf{V} \cdot v)\{Re\,\boldsymbol{\tau} \cdot \boldsymbol{\sigma}v + \kappa(\mathbf{u} \cdot \boldsymbol{\tau})\}(\boldsymbol{\tau} \cdot \mathbf{H}v), \\ &= \int_{\Gamma} (\mathbf{u} \cdot \boldsymbol{\tau})\{-(v \cdot \partial_{\tau}\mathbf{V})(v \cdot \mathbf{H}v) + \kappa[(\mathbf{V} \cdot v)(\boldsymbol{\tau} \cdot \mathbf{H}v) - (\mathbf{V} \cdot \boldsymbol{\tau})(v \cdot \mathbf{H}v)]\} + Re \int_{\Gamma} (\mathbf{V} \cdot v)(\boldsymbol{\tau} \cdot \boldsymbol{\sigma}v)(\boldsymbol{\tau} \cdot \mathbf{H}v), \end{aligned} \tag{34}$$

where we used (26) and (27). Therefore, putting the above results together, we get

$$\delta Q = \int_{\Gamma} (\mathbf{u} \cdot \mathbf{e}_1 + 1)(\mathbf{V} \cdot v) + \int_{\Gamma} (\mathbf{u} \cdot \tau)\{-(v \cdot \partial_\tau \mathbf{V})(v \cdot \mathbf{H}v) + \kappa[(\mathbf{V} \cdot v)(\tau \cdot \mathbf{H}v) - (\mathbf{V} \cdot \tau)(v \cdot \mathbf{H}v)]\} + Re \int_{\Gamma} (\mathbf{V} \cdot v)(\tau \cdot \sigma v)(\tau \cdot \mathbf{H}v). \tag{35}$$

Of course, we can obtain a formula that is consistent with Theorem 1 by integrating by parts:

$$\delta Q = \int_{\Gamma} [(\mathbf{u} \cdot \mathbf{e}_1 + 1) + (\mathbf{u} \cdot \tau)\{v \cdot \partial_\tau(\mathbf{H}v) + 2\kappa(\tau \cdot \mathbf{H}v)\} + Re(\tau \cdot \sigma v)(\tau \cdot \mathbf{H}v)](\mathbf{V} \cdot v). \tag{36}$$

But Eq. (35) is more convenient in our optimization method (see Remark 1).

5. Optimization algorithm

We use a sequential quadratic programming (SQP) method to find a local minimizer Ω^* in the set of admissible shapes $\mathcal{O}$ (recall (11)). In other words, we solve for a shape flow that will deform the channel's walls Γ in such a way as to reduce the cost J subject to constraints on the average flux Q and area $A(\Omega) := |\Omega|$. A related approach for gradient flows involving Laplacian type PDE constraints can be found in [35,36]. Other approaches using level sets can be found in [37–41], as well as boundary integral methods in [42–44]. See [45,31,29,34,46,47] for other methods.

5.1. Lagrange multipliers

In order to handle the constraint on the admissible shapes, we use Lagrange multipliers with the following Lagrangian functional:

Definition 1. Let $\lambda, \mu \in \mathbb{R}$ and define the Lagrangian to be:

$$\mathcal{L}(\Omega, \lambda, \mu) = J(\Omega) + \lambda(Q(\Omega) - C_Q) + \mu(A(\Omega) - C_A). \tag{37}$$

$\mathcal{L}$ depends on $(\mathbf{u}, p), (\mathbf{r}, \pi)$, and $(\mathbf{z}, \varrho)$; we omit writing the explicit dependence on velocity and pressure to avoid excessive notation.

The first order optimality conditions are given by

$$\frac{\partial \mathcal{L}}{\partial \lambda} = (Q - C_Q) = 0,$$
$$\frac{\partial \mathcal{L}}{\partial \mu} = (A - C_A) = 0, \tag{38}$$
$$\delta \mathcal{L}(\Omega, \lambda, \mu; \mathbf{V}) = \delta J(\Omega; \mathbf{V}) + \lambda \delta Q(\Omega; \mathbf{V}) + \mu \delta A(\Omega; \mathbf{V}) = 0,$$

where $\mathbf{V}$ is an arbitrary perturbation of Γ. Note: $\mathbf{V}$ is periodic on Γ_P; only the boundary Γ is allowed to move. Next, plugging in (28) and (35), performing some integration by parts, and rearranging gives:

$$0 = \delta \mathcal{L}$$
$$= \int_{\Gamma} (\mathbf{V} \cdot v)[\lambda \mathbf{u} \cdot \mathbf{e}_1 + \lambda + \mu + (\mathbf{u} \cdot \tau)\{v \cdot \partial_\tau(\lambda \mathbf{H}v + \sigma v + \mathbf{S}v) + 2\kappa\tau \cdot (\lambda \mathbf{H}v - \sigma v - \mathbf{S}v)\} + Re(\tau \cdot \sigma v)[\tau \cdot (\lambda \mathbf{H}v - \mathbf{S}v)]], \tag{39}$$

which gives an optimality condition for the minimizing shape.

5.2. Newton's method

Finding a solution to (38) can be accomplished (formally) by using a Newton method which consists of iteratively solving the following Newton-KKT system [48]

$$\begin{bmatrix} \delta^2 \mathcal{L}(\mathbf{V}, \cdot) & \delta Q(\mathbf{V}) & \delta A(\mathbf{V}) \\ \delta Q^\dagger(\cdot) & 0 & 0 \\ \delta A^\dagger(\cdot) & 0 & 0 \end{bmatrix} \begin{bmatrix} \varphi \\ \lambda \\ \mu \end{bmatrix} = \begin{bmatrix} -\delta \mathcal{L}(\mathbf{V}) \\ -(Q - C_Q) \\ -(A - C_A) \end{bmatrix}, \tag{40}$$

where $\mathbf{V}$ can be thought of as an infinite-dimensional row index, and $\delta^2 \mathcal{L}(\mathbf{V}, \varphi)$ is the Hessian bilinear form. Solving (40) gives a local perturbation φ of the domain shape, which may reduce J but will also move toward satisfying the constraints. Iterating this will create a flow that deforms the domain Ω in order to obtain a (local) solution of (12).

In our method, we avoid computing $\delta^2 \mathcal{L}(\mathbf{V}, \boldsymbol{\varphi})$ by replacing it with a positive definite inner product over an appropriate function space $\mathbb{V}$:

Definition 2 (*Shape Perturbation Space*). Let $H^1(\Gamma)$ be the usual Hilbert space [49,50] given by

$$H^1(\Gamma) = \left\{ \mathbf{v} : \Gamma \to \mathbb{R}^2 : \int_\Gamma |\mathbf{v}|^2 < \infty, \quad \text{and} \quad \int_\Gamma |\nabla_\Gamma \mathbf{v}|^2 < \infty \right\}, \tag{41}$$

where Γ is periodic. The perturbation space is taken to be a restriction of $H^1(\Gamma)$:

$$\mathbb{V} = \left\{ \mathbf{v} \in H^1(\Gamma) : \mathbf{v} = 0, \quad \text{on } \overline{\Gamma_P} \cap \overline{\Gamma^-}, \quad \text{and} \quad \mathbf{v} \cdot \mathbf{e}_1 = 0, \quad \text{on } \overline{\Gamma_P} \cap \overline{\Gamma^+} \right\}. \tag{42}$$

In other words, the shape is pinned at the lower left corner and the top wall is prevented from sliding in the $\mathbf{e}_1$ direction. This restriction is needed to prevent the channel domain from rigidly translating and to keep the periodic boundary Γ_P as a vertical line segment.

Our algorithm involves solving the following variational formulation (instead of solving (40)):

Variational Formulation 1 (H^1 *Shape Perturbation*). Assume that the curvature is point-wise bounded (i.e. $\kappa \in L^\infty(\Gamma)$) and that the solutions of Eqs. (8), (23) and (32) are sufficiently regular (see Theorem 3) such that $\delta J(\mathbf{V})$, $\delta Q(\mathbf{V})$, and $\delta A(\mathbf{V})$ are well-defined for all $\mathbf{V}$ in $\mathbb{V}$. Then we solve the following variational formulation to obtain a perturbation of the shape: find $\boldsymbol{\varphi}$ in $\mathbb{V}$ and λ, μ in $\mathbb{R}$ such that

$$\begin{aligned}
\langle \mathbf{V}, \boldsymbol{\varphi} \rangle_\mathbb{V} + \lambda \delta Q(\mathbf{V}) + \mu \delta A(\mathbf{V}) &= -\delta J(\mathbf{V}), \\
\delta Q(\boldsymbol{\varphi}) &= -(Q - C_Q), \\
\delta A(\boldsymbol{\varphi}) &= -(A - C_A),
\end{aligned} \tag{43}$$

for all $\mathbf{V}$ in $\mathbb{V}$. The inner product is given by $\langle \mathbf{V}, \boldsymbol{\varphi} \rangle_\mathbb{V} = \int_\Gamma \mathbf{V} \cdot \boldsymbol{\varphi} + \int_\Gamma \nabla_\Gamma \mathbf{V} : \nabla_\Gamma \boldsymbol{\varphi}$.

Remark 2 (*Cost Reduction*). The solution $\boldsymbol{\varphi}$ in (43) gives a perturbation for updating the shape which will try to decrease the cost J while maintaining the constraints. To see why, set $\mathbf{V} = \boldsymbol{\varphi}$ in (43). Assuming the current shape lies on the constraint manifold (i.e. $Q = C_Q$, $A = C_A$), combining the equations in (43) implies $\delta J(\boldsymbol{\varphi}) = -\langle \boldsymbol{\varphi}, \boldsymbol{\varphi} \rangle_\mathbb{V} < 0$ for $\boldsymbol{\varphi} \neq \mathbf{0}$.

Remark 3 (*Solvability*). The system (43) has a unique solution provided the following 'inf-sup' condition is true [51,52]:

$$\sup_{\mathbf{V} \in \mathbb{V}} \frac{\lambda \delta Q(\mathbf{V}) + \mu \delta A(\mathbf{V})}{\|\mathbf{V}\|_\mathbb{V}} \geqslant c_0(|\lambda| + |\mu|), \quad \text{for all } \lambda, \mu \in \mathbb{R}, \tag{44}$$

which is equivalent [53] to the following condition:

$$\text{the } 2 \times 2 \text{ matrix } \begin{bmatrix} \delta Q(\mathbf{V}) & \delta Q(\mathbf{W}) \\ \delta A(\mathbf{V}) & \delta A(\mathbf{W}) \end{bmatrix} \text{ is non-singular for suitably chosen functions } \mathbf{V}, \mathbf{W} \in \mathbb{V}. \tag{45}$$

Proving (45) is not trivial due to the dependence of δQ on $\sigma \nu$, $\mathbf{H}\nu$, and κ. Note that for the trivial case when Ω is a rectangular channel (horizontal walls Γ), the matrix in (45) is singular. This corresponds to the case where $Q = 0$ and the velocity field is zero in the fixed lab frame.

Remark 4 (*Choice of $\mathbb{V}$*). Choosing $\mathbb{V}$ in Definition 2 ensures that (43) is well-defined. In particular, it is needed since $\delta J(\mathbf{V})$ and $\delta Q(\mathbf{V})$ involve tangential derivatives of $\mathbf{V}$. Recall that this was done to avoid differentiating the stress terms (see Remark 1). The practical effect of this is to avoid taking unnecessarily small steps in our optimization routine as we deform Ω towards an optimal shape (see our iterative algorithm in Section 5.3). This also prevents small scale oscillations from developing in the shape [31]. Thus, choosing the inner product space $\mathbb{V}$ in (42) acts as a pre-conditioner for our optimization algorithm [35,36]. Ideally, one would like to choose $\langle \mathbf{V}, \boldsymbol{\varphi} \rangle_\mathbb{V} = \delta^2 \mathcal{L}(\mathbf{V}, \boldsymbol{\varphi})$ as this would give a true Newton method (i.e. faster convergence) but we avoid this for simplicity.

5.3. SQP method

Our algorithm is essentially an SQP method, where we have replaced the Hessian term by a positive definite bilinear form. The details of our iterative method are described in Algorithm 1. Basically, one first computes the solutions to the Navier–Stokes equation (8) and adjoint PDEs (23) and (32). Next, the shape derivatives are calculated and we solve (43) to obtain a descent direction for changing the domain shape. Lastly, we use a line search (see Section 5.4) to update the domain shape. This process is iterated until a desired convergence criteria is reached.

Algorithm 1. Semi-Discrete Shape Flow

Define: $\Omega^0 \subset \Omega_{ALL}$ with partition $\partial\Omega^0 = \overline{\Gamma^0} \cup \overline{\Gamma_P^0}$
Compute $C_Q := Q(\Omega^0)$ and $C_A := A(\Omega^0)$
Initialize Penalty: $\zeta^0 := 0$
for $k = 0, 1, 2, \ldots$ **do**
  Solve Navier–Stokes: Let $(\mathbf{u}^k, p^k)$ solve (8) on Ω^k. Let $\boldsymbol{\sigma}^k$ be the stress tensor
  Solve First Adjoint: Let $(\mathbf{r}^k, \pi^k)$ solve (23) on Ω^k. Let $\mathbf{S}^k$ be the stress tensor
  Solve Second Adjoint: Let $(\mathbf{z}^k, \varrho^k)$ solve (32) on Ω^k. Let $\mathbf{H}^k$ be the stress tensor
  Compute curvature κ^k of Γ^k
  Compute δJ, δQ, and δA using $\kappa^k, \boldsymbol{\sigma}^k, \mathbf{S}^k$, and $\mathbf{H}^k$
  Solve For Shape Perturbation: Let $(\boldsymbol{\varphi}^{k+1}, \lambda^{k+1}, \mu^{k+1})$ solve (43) on Γ^k
  Extend Velocity: Create a smooth periodic extension of $\boldsymbol{\varphi}^{k+1}$ to all of Ω^k (necessary because of the interior mesh)
  Line Search: Execute Algorithm 2 to compute step size α^{k+1}
  Update Shape: Let $\mathbf{x}^{k+1} := \mathbf{x}^k + \alpha^{k+1}\boldsymbol{\varphi}^{k+1}(\mathbf{x}^k)$, for all points $\mathbf{x}^k$ in $\Omega^k \subset \mathbb{R}^d$
  Define $\Omega^{k+1} := \{\mathbf{x}^{k+1} \in \mathbb{R}^d\}$
end for

The fully discrete algorithm follows directly from Algorithm 1. One need only apply a spatial discretization when solving each PDE (see Section 6). A convergence criteria can be based on the relative change in the cost J between successive iterations or on the relative change in the shape of Ω.

5.4. Line search with merit function

The optimization process in Algorithm 1 may produce iterates that violate the constraints. Therefore, we must balance reducing the cost versus violating the constraints. This can be done by choosing the step size at each iteration through a merit-function based line search criteria [48]. Consider the following merit function:

$$\theta(\Omega, \zeta) = J(\Omega) + \zeta(|Q(\Omega) - C_Q| + |A(\Omega) - C_A|), \tag{46}$$

where ζ is a penalty parameter for the constraint violation and is updated at each optimization step. At the kth optimization step, we define a step size α to be acceptable if

$$\theta(\Omega^{k+1}(\alpha), \zeta^{k+1}) \leqslant \theta(\Omega^k, \zeta^{k+1}) + \xi\alpha\delta\theta(\Omega^k, \zeta^{k+1}; \boldsymbol{\varphi}), \quad \xi \in [0, 1], \tag{47}$$

where $\delta\theta$ is the shape derivative of θ, ζ^{k+1} is sufficiently large, $\boldsymbol{\varphi}$ is the current descent direction, and we use the abuse of notation $\Omega^{k+1}(\alpha) := \Omega^k + \alpha\boldsymbol{\varphi}$ to denote the domain update. Our line search method is described in Algorithm 2.

Algorithm 2. Line Search With Merit Function

1: Set parameter ξ
2: Input: $\Omega^k, (\boldsymbol{\varphi}^{k+1}, \lambda^{k+1}, \mu^{k+1})$, and ζ^k
3: Update Penalty: $\zeta^{k+1} := \max(\zeta^k, |\lambda^{k+1}|, |\mu^{k+1}|)$
4: Compute $\theta(\Omega^k, \zeta^{k+1})$ by (46) and $\delta\theta(\Omega^k, \zeta^{k+1}; \boldsymbol{\varphi}^{k+1})$ by (48)
5: Initialize: $\alpha^{k+1} := 1$, ACCEPT := FALSE.
6: **while** ACCEPT = FALSE **do**
7:   Compute $\Omega^{k+1}(\alpha^{k+1}) := \Omega^k + \alpha^{k+1}\boldsymbol{\varphi}^{k+1}$
8: Compute $\theta(\Omega^{k+1}, \zeta^{k+1})$ by (46)
9:   **if** $\theta(\Omega^{k+1}, \zeta^{k+1}) \leqslant \theta(\Omega^k, \zeta^{k+1}) + \xi\alpha^{k+1}\delta\theta(\Omega^k, \zeta^{k+1}; \boldsymbol{\varphi}^{k+1})$ **then**
10:      ACCEPT := TRUE
11:   **else**
12:      $\alpha^{k+1} := \alpha^{k+1}/2$
13:   **end if**
14: **end while**
15: **return** α^{k+1} and ζ^{k+1}

Updating the penalty parameter by $\zeta^{k+1} := \max(\zeta^k, |\lambda^{k+1}|, |\mu^{k+1}|)$ guarantees that $\delta\theta(\Omega^k, \zeta^{k+1}; \boldsymbol{\varphi}^{k+1}) \leqslant 0$ by the following theorem.

Theorem 2. *Let $(\varphi^{k+1}, \lambda^{k+1}, \mu^{k+1})$ be generated by Algorithm 1; in particular, $(\varphi^{k+1}, \lambda^{k+1}, \mu^{k+1})$ is a solution of the variational problem (43). Then, for $\zeta \geqslant 0$, the shape derivative of $\theta(\Omega^k, \zeta)$ (46) in the direction φ^{k+1} is given by*

$$\delta\theta(\Omega^k, \zeta; \varphi^{k+1}) = \delta J(\varphi^{k+1}) - \zeta(|Q(\Omega^k) - C_Q| + |A(\Omega^k) - C_A|). \tag{48}$$

Moreover, we have that

$$\delta\theta(\Omega^k, \zeta; \varphi^{k+1}) \leqslant -\langle \varphi^{k+1}, \varphi^{k+1} \rangle_{\mathbb{V}} - (\zeta - \max(|\lambda^{k+1}|, |\mu^{k+1}|))(|Q(\Omega^k) - C_Q| + |A(\Omega^k) - C_A|). \tag{49}$$

Proof. The proof of the finite dimensional version can be found in [48, Theorem 18.2]. For the infinite-dimensional setting, one must assume that the functionals J and Q are sufficiently smooth (and analytic) with respect to the flow parameter s (recall Section 3.1) so that a Taylor expansion in α makes sense. □

Computing $\delta\theta(\Omega^k, \zeta; \varphi^{k+1})$ (48) is straightforward by using previously known quantities.

6. Finite element method

The finite element method [54–56,51,52] offers a flexible way of solving the Navier–Stokes system (8) and adjoint problems (23) and (32), in addition to computing the sensitivities and a descent direction for deforming the domain. We note here that accuracy in computing the PDE solutions is especially important to ensure consistency between J and δJ (likewise, Q and δQ). For example, our discrete approximation of the continuous level δJ equation may not accurately represent changes in the discrete approximation of J if our discretization is too coarse. Thus, it is important that a sufficiently fine grid is used especially at higher Reynolds numbers.

In the following sections, we describe our method of computing the PDE solutions, the shape derivatives (28), (35), and the solution of (43).

6.1. Solving Navier–Stokes and adjoint systems

We use a mixed finite element method for solving the periodic Navier–Stokes system (8) and for the adjoint systems (23) and (32).

6.1.1. Velocity and pressure discretization

We denote a discrete approximation of the domain by Ω_h and partition it into a set of triangles labeled by $\top_{\Omega_h}$ and discretize velocity and pressure by the Taylor–Hood element

$$\begin{aligned}
\mathbb{U}_h &:= \{\mathbf{u} \in C(\overline{\Omega}_h) : \mathbf{u}|_T \in \mathcal{P}_2(T), \forall T \in \top_{\Omega_h}\}, \\
\mathbb{P}_h &:= \{p \in C(\overline{\Omega}_h) : p|_T \in \mathcal{P}_1(T), \forall T \in \top_{\Omega_h}\},
\end{aligned} \tag{50}$$

i.e. vector piecewise quadratic polynomials for velocity and piecewise linear polynomials for pressure. We solve the Navier–Stokes system using a Newton iteration. Details can be found in [54–56,51,52].

6.1.2. Enforcing the tangential boundary condition

In order to enforce the boundary condition $\mathbf{u} = \pm\tau$ on Γ, we use an averaging technique. First, let Γ_h be the top and bottom walls of the polygonal boundary of Ω_h and partition it into a set of edges denoted S_{Γ_h} (this set of edges is inherited from $\top_{\Omega_h}$). Define the piecewise linear space $\mathbb{B}_h$

$$\mathbb{B}_h := \{\mathbf{v} \in C(\Gamma_h) : \mathbf{v}|_E \in \mathcal{P}_1(E), \ \forall E \in \mathsf{S}_{\Gamma_h}\}, \tag{51}$$

i.e. the set of vector piecewise linear functions over the boundary Γ_h. Denote the unit tangent vector of Γ_h by τ_h which is piecewise constant. Now define a piecewise linear approximation of the tangent vector τ of Γ in the following way. Let $\hat{\tau} \in \mathbb{B}_h$ and $\mathbf{x}_i$ be the position of a vertex of Γ_h with adjacent edges labeled as E_i^- and E_i^+. The nodal values of $\hat{\tau}$ are given by:

$$\zeta_i := \frac{|E_i^+|}{|E_i^-| + |E_i^+|} \tau_h|_{E_i^-} + \frac{|E_i^-|}{|E_i^-| + |E_i^+|} \tau_h|_{E_i^+}, \quad \Rightarrow \quad \hat{\tau}(\mathbf{x}_i) := \frac{\zeta_i}{|\zeta_i|} \tag{52}$$

i.e. a local average of the piecewise constant tangent vector which is then normalized. Without the normalization, equation (52) is similar to the so-called *gradient recovery* formula [57–59], except it is applied to the tangential derivative of a parametrization of Γ_h, namely τ_h. From [57–59], and provided Γ is a C^3 curve, it is straightforward to show that

$$\|\tau - (\hat{\tau} \circ F)\|_{L^\infty(\Gamma)} \leqslant Ch^2,$$

where $F : \Gamma \to \Gamma_h$ is a continuous, piecewise smooth locally defined map [60]. Thus, it is advantageous to enforce the boundary condition in (8) using the approximation of the tangent vector $\hat{\tau}$. One can also use the $L^2(\Gamma_h)$ projection of τ_h onto $\mathbb{B}_h$; this has the same order of accuracy as the method in (52), but is a little more expensive because a mass matrix must be inverted.

6.1.3. Finite element discretization error

We will use the following result [54,52] in Section 6.2 to estimate the error in approximating the boundary stress.

Theorem 3. *Suppose there exists a unique solution* $(\mathbf{u}, p)$ *of (8) in the sense of distributions with the following regularity:*

$$\mathbf{u} \in H^3(\Omega), \quad p \in H^2(\Omega). \tag{53}$$

Then we have the following error estimate for the Taylor–Hood element (50):

$$\|\mathbf{u} - \mathbf{u}_h\|_{H^1(\Omega)} + \|p - p_h\|_{L^2(\Omega)} \leqslant Ch^2 (\|\mathbf{u}\|_{H^3(\Omega)} + \|p\|_{H^2(\Omega)}), \tag{54}$$

where C is an independent constant and h is the maximum size of an edge in the mesh $\top_{\Omega_h}$.

Remark 5. The regularity assumption in (53) is guaranteed if the domain is C^3 [61]. In our numerical results (see Section 7), the domain shapes appear to be smooth. Lastly, the Navier–Stokes and adjoint equation solvers were tested against a known exact solution to confirm the error decay given by (54).

6.2. Stress approximation

Computing the shape derivatives, namely δJ and δQ, requires the stresses on the boundary $\boldsymbol{\sigma}v$, $\mathbf{S}v$, and $\mathbf{H}v$. In our method, we use an L^2 type projection property similar to [62,63]. For computing $\boldsymbol{\sigma}v$, we have the following weak formulation of (8):

$$\int_\Gamma (\boldsymbol{\sigma}v) \cdot \mathbf{v} = \int_\Omega [(\mathbf{u} \cdot \nabla)\mathbf{u}] \cdot \mathbf{v} + \int_\Omega \boldsymbol{\sigma}(\mathbf{u}, p) : \nabla \mathbf{v}, \quad \text{for all } \mathbf{v} \in [H^1(\Omega)]^2, \tag{55}$$

where $(\mathbf{u}, p)$ is the solution of (8). At the discrete level, the right hand side of (55) can be viewed as given data and $\boldsymbol{\sigma}v$ is treated as an unknown. To state the discrete variational formulation of (55), we need the following finite element space:

$$\mathbb{W}_h := \left\{ \mathbf{v} \in C(\overline{\Omega}_h) : \mathbf{v}|_T \in \mathcal{P}_1(T), \ \forall T \in \top_{\Omega_h}, \text{ and } \mathbf{v} = 0 \text{ at all interior nodes} \right\}, \tag{56}$$

i.e. the set of vector piecewise linear functions over the triangulation $\top_{\Omega_h}$ with all interior nodal values set to zero. Recall the definition of $\mathbb{B}_h$ (51); note that $\mathbb{B}_h$ is the restriction of $\mathbb{W}_h$ to the boundary Γ_h. Therefore, given the discrete solution $(\mathbf{u}_h, p_h)$, find a discrete approximation of the boundary stress $(\boldsymbol{\sigma}v)_h \in \mathbb{B}_h$ such that

$$\int_{\Gamma_h} (\boldsymbol{\sigma}v)_h \cdot \mathbf{v}_h = \int_{\Omega_h} [(\mathbf{u}_h \cdot \nabla)\mathbf{u}_h] \cdot \mathbf{v}_h + \int_{\Omega_h} \boldsymbol{\sigma}_h : \nabla \mathbf{v}_h, \quad \text{for all } \mathbf{v}_h \in \mathbb{W}_h, \tag{57}$$

where $\boldsymbol{\sigma}_h = -\mathbf{I}p_h + (1/Re)D(\mathbf{u}_h)$. This gives a square mass matrix on the left side which is trivial to invert compared to solving (8), (23), (32). Computation of the boundary stresses $(\mathbf{S}v)_h$ and $(\mathbf{H}v)_h$ is done similarly.

Using averaging, least-squares, and L^2-type projections to compute stresses is a classical technique in the finite element literature; for instance, see [64,65]. Our approximation of the boundary stress is based on a method in [62,63] which was used to compute the boundary flux for Laplace's equation given the discrete solution of the Dirichlet problem. Following a similar outline as in [63], we derive error estimates for the boundary stress approximation (57) in the following sections. More recent methods of post-processing the solution to obtain flux information over the entire domain can be found in [66].

6.2.1. Preliminary estimates

In the following, we require that $\top_{\Omega_h}$ be a quasi-uniform, shape regular triangulation of Ω (see [55,56]). We also assume that $\Omega_h = \Omega$ to avoid unnecessary technicalities (see Remark 6).

Lemma 1 (trace inverse estimate). *Let* $S_\top = S_\top(\Omega, h)$ *denote a strip of triangles in* $\top_{\Omega_h}$ *such that each triangle* $T \in S_\top$ *has at least one vertex on* Γ. *Let* $\mathbf{v}_h \in \mathbb{W}_h$. *Then,*

$$\begin{aligned} \|\nabla \mathbf{v}_h\|_{L^2(S_\top)} &\leqslant Ch^{-1/2} \|\mathbf{v}_h\|_{L^2(\Gamma)}, \\ \|\mathbf{v}_h\|_{L^2(S_\top)} &\leqslant Ch^{1/2} \|\mathbf{v}_h\|_{L^2(\Gamma)}, \end{aligned} \tag{58}$$

where C is a constant independent of h.

Proof. Using basic inverse estimates [55,56] (i.e. norm equivalence on finite dimensional spaces) gives the assertion [63]. □

Lemma 2. *The following interpolation estimates hold:*

$$\int_\Gamma (\boldsymbol{\sigma}v - \mathcal{I}(\boldsymbol{\sigma}v)) \cdot \mathbf{v}_h \leqslant Ch^{3/2} (\|\mathbf{u}\|_{H^3(\Omega)} + \|p\|_{H^2(\Omega)}) \|\mathbf{v}_h\|_{L^2(\Gamma)}, \tag{59}$$

where $\mathcal{I}$ *is the interpolation operator onto* $\mathbb{B}_h$.

Proof. The Bramble–Hilbert lemma [56] gives

$$\|\boldsymbol{\sigma v} - \mathcal{I}(\boldsymbol{\sigma v})\|_{L^2(\Gamma)} \leqslant Ch^{3/2}\|\boldsymbol{\sigma v}\|_{H^{3/2}(\Gamma)}. \tag{60}$$

Then using standard Sobolev embedding [50] and the smoothness of the domain (see Remark 5) gives

$$\int_\Gamma (\boldsymbol{\sigma v} - \mathcal{I}(\boldsymbol{\sigma v})) \cdot \mathbf{v}_h \leqslant \|\boldsymbol{\sigma v} - \mathcal{I}(\boldsymbol{\sigma v})\|_{L^2(\Omega)}\|\mathbf{v}_h\|_{L^2(\Gamma)} \leqslant Ch^{3/2}(\|\mathbf{u}\|_{H^3(\Omega)} + \|p\|_{H^2(\Omega)})\|\mathbf{v}_h\|_{L^2(\Gamma)}. \qquad \square$$

Lemma 3. *Suppose that* $\mathbf{v}_h \in \mathbb{W}_h$. *Then the following estimate holds*

$$\int_\Gamma (\boldsymbol{\sigma v} - (\boldsymbol{\sigma v})_h) \cdot \mathbf{v}_h \leqslant Ch^{3/2}(\|\mathbf{u}\|_{H^3(\Omega)} + \|p\|_{H^2(\Omega)})\|\mathbf{v}_h\|_{L^2(\Gamma)}. \tag{61}$$

Proof. Using the variational forms (55) and (57), we have that

$$\begin{aligned}
\int_\Gamma (\boldsymbol{\sigma v} - (\boldsymbol{\sigma v})_h) \cdot \mathbf{v}_h &= \int_\Omega (\boldsymbol{\sigma} - \boldsymbol{\sigma}_h) : \nabla \mathbf{v}_h + \int_\Omega ([(\mathbf{u}\cdot\nabla)\mathbf{u}] - [(\mathbf{u}_h\cdot\nabla)\mathbf{u}_h]) \cdot \mathbf{v}_h \\
&= -\int_\Omega (p - p_h)\nabla\cdot\mathbf{v}_h + \frac{1}{Re}\int_\Omega (D(\mathbf{u}) - D(\mathbf{u}_h)) : \nabla\mathbf{v}_h + \int_\Omega ([(\mathbf{u}-\mathbf{u}_h)\cdot\nabla)\mathbf{u}] + [(\mathbf{u}_h\cdot\nabla)(\mathbf{u}-\mathbf{u}_h)]) \cdot \mathbf{v}_h \\
&\leqslant C_2\left(\|p - p_h\|_{L^2(\Omega)} + \frac{1}{Re}\|\nabla\mathbf{u} - \nabla\mathbf{u}_h\|_{L^2(\Omega)}\right)\|\nabla\mathbf{v}_h\|_{L^2(S_\top)} + \|\mathbf{u}-\mathbf{u}_h\|_{H^1(\Omega)}\|\nabla\mathbf{u}\|_{L^2(\Omega)}\|\mathbf{v}_h\|_{H^1(S_\top)} \\
&\quad + \|\mathbf{u}_h\|_{H^1(\Omega)}\|\nabla\mathbf{u} - \nabla\mathbf{u}_h\|_{L^2(\Omega)}\|\mathbf{v}_h\|_{H^1(S_\top)} \\
&\leqslant C_3 h^2(\|\mathbf{u}\|_{H^3(\Omega)} + \|p\|_{H^2(\Omega)})\|\mathbf{v}_h\|_{H^1(S_\top)} \leftarrow \text{ by Eq. (54)} \\
&\leqslant C_4 h^{3/2}(\|\mathbf{u}\|_{H^3(\Omega)} + \|p\|_{H^2(\Omega)})\|\mathbf{v}_h\|_{L^2(\Gamma)} \leftarrow \text{ by Lemma 1}.
\end{aligned} \tag{62}$$

Note: we used the classical fact that (see [52,67])

$$\left|\int_\Omega [(\mathbf{w}\cdot\nabla)\mathbf{u}]\cdot\mathbf{v}\right| \leqslant C\|\mathbf{w}\|_{H^1(\Omega)}\|\nabla\mathbf{u}\|_{L^2(\Omega)}\|\mathbf{v}\|_{H^1(\Omega)}. \qquad \square$$

Remark 6. For the case where the discrete domain only approximates the true domain ($\Omega_h \neq \Omega$), one must account for the difference by considering a locally defined map $F : \Omega \to \Omega_h$ and use a variational crime argument [56,68,60]. It can be shown that the error due to approximating the domain with a polygon is $O(h^2)$.

6.2.2. Boundary stress error estimate

Theorem 4. *Let the solution* $(\mathbf{u}, p)$ *of* (8) *satisfy the regularity property* (53). *Assume the finite element mesh is shape regular and quasi-uniform. If the boundary stress is approximated by* (57), *then*

$$\|\boldsymbol{\sigma v} - (\boldsymbol{\sigma v})_h\|_{L^2(\Gamma)} \leqslant Ch^{3/2}(\|\mathbf{u}\|_{H^3(\Omega)} + \|p\|_{H^2(\Omega)}). \tag{63}$$

Proof. First, using (61) and (59), we get

$$\begin{aligned}
\|(\boldsymbol{\sigma v})_h - \mathcal{I}(\boldsymbol{\sigma v})\|_{L^2(\Gamma)}^2 &= \int_\Gamma ((\boldsymbol{\sigma v})_h - \boldsymbol{\sigma v}) \cdot ((\boldsymbol{\sigma v})_h - \mathcal{I}(\boldsymbol{\sigma v})) + \int_\Gamma (\boldsymbol{\sigma v} - \mathcal{I}(\boldsymbol{\sigma v})) \cdot ((\boldsymbol{\sigma v})_h - \mathcal{I}(\boldsymbol{\sigma v})) \\
&\leqslant C_1 h^{3/2}(\|\mathbf{u}\|_{H^3(\Omega)} + \|p\|_{H^2(\Omega)})\|(\boldsymbol{\sigma v})_h - \mathcal{I}(\boldsymbol{\sigma v})\|_{L^2(\Gamma)},
\end{aligned} \tag{64}$$

which implies $\|(\boldsymbol{\sigma v})_h - \mathcal{I}(\boldsymbol{\sigma v})\|_{L^2(\Gamma)} \leqslant C_1 h^{3/2}(\|\mathbf{u}\|_{H^3(\Omega)} + \|p\|_{H^2(\Omega)})$. Together with (60), we get

$$\|\boldsymbol{\sigma v} - (\boldsymbol{\sigma v})_h\|_{L^2(\Gamma)} \leqslant \|\boldsymbol{\sigma v} - \mathcal{I}(\boldsymbol{\sigma v})\|_{L^2(\Gamma)} + \|\mathcal{I}(\boldsymbol{\sigma v}) - (\boldsymbol{\sigma v})_h\|_{L^2(\Gamma)} \leqslant C_2 h^{3/2}(\|\mathbf{u}\|_{H^3(\Omega)} + \|p\|_{H^2(\Omega)}). \qquad \square$$

Remark 7. Implementation of (57) is straightforward. The finite element matrices, using the Taylor–Hood element, are already assembled when solving (8). One need only combine the rows of the discrete column vector representing the right hand side of (8) appropriately to reduce the test function $\mathbf{v}_h$ from piecewise quadratic to piecewise linear.

The error analysis given here can be modified to obtain an error estimate of $O(h^2 \ln(1/h))$ if the solution is more regular, i.e. $(\mathbf{u}, p) \in W^{3,\infty}(\Omega) \times W^{2,\infty}(\Omega)$ [69] (see [63] in the case of Laplace's equation). In our numerical tests of convergence, we first computed a discrete solution of (8) (when the exact solution is C^∞) and then computed $(\boldsymbol{\sigma v})_h$ from (57) using the discrete solution of (8). In this case, we appear to get $O(h^2)$ in $L^2(\Gamma)$ when comparing the approximation $(\boldsymbol{\sigma v})_h$ to the exact C^∞ solution.

6.3. Curvature approximation

We also use an L^2 projection for computing the curvature, which takes advantage of the variational structure of the curvature vector κv. This follows by multiplying equation (76) by a smooth test function $\mathbf{v}$ and integrating by parts:

$$\int_\Gamma \kappa v \cdot \mathbf{v} = \int_\Gamma -\Delta_\Gamma \mathbf{X} \cdot \mathbf{v} = \int_\Gamma \nabla_\Gamma \mathbf{X} : \nabla_\Gamma \mathbf{v} = \int_\Gamma \nabla_\Gamma \cdot \mathbf{v}. \tag{65}$$

The last term in Eq. (65) makes sense even when Γ is a polygonal curve. Hence, our finite element method for computing curvature is the following. Recall the finite element space $\mathbb{B}_h$ and define the following discrete variational problem: find $\mathbf{K} \in \mathbb{B}_h$ (a piecewise linear approximation of the curvature) such that

$$\int_{\Gamma_h} \mathbf{K} \cdot \mathbf{v} = \int_{\Gamma_h} \nabla_\Gamma \cdot \mathbf{v}, \quad \forall \mathbf{v} \in \mathbb{B}_h. \tag{66}$$

Similarly to the previous section, this gives a mass matrix on the left side of (66) that is cheap to invert. If we assume that Γ_h interpolates an underlying smooth curve, then this approximation of the curvature appears to be $O(h^2)$ in the $L^2(\Gamma)$ norm by our own numerical tests. This type of approximation is analyzed in [70].

6.4. Computing the descent direction

The system in (43) is a standard saddle point system and can be solved by straightforward techniques. The finite element space $\mathbb{V}_h \subset \mathbb{V}$ is given by (51): $\mathbb{V}_h := \mathbb{B}_h$. More details can be found in [55,56,51].

6.5. Method for deforming the domain

The shape perturbation φ that solves (43) is only defined on the upper and lower walls Γ of the domain Ω. Since the domain is partitioned into a set of triangles, we must extend φ to all of Ω in order to update the interior mesh properly. We do this in two steps. First, we define φ on the periodic boundary Γ_P of Ω by linear interpolation of φ between $\overline{\Gamma_P} \cap \overline{\Gamma^-}$ and $\overline{\Gamma_P} \cap \overline{\Gamma^+}$, which acts to vertically stretch/compress Γ_P. This fully defines φ on $\partial\Omega$. Second, we extend φ by letting φ_{ext} solve the following vector Laplace equation (harmonic extension) [71]:

$$\begin{aligned} -\Delta\varphi_{\text{ext}} &= 0, \quad \text{in } \Omega, \\ \varphi_{\text{ext}} &= \varphi, \quad \text{on } \partial\Omega. \end{aligned} \tag{67}$$

This will not affect the updated shape of the domain (because $\varphi_{\text{ext}} = \varphi$ on Γ). It is a classical result that the solution of (67) minimizes $\int_\Omega |\nabla\varphi_{\text{ext}}|^2$ [49], which is desirable because large gradients in the velocity cause mesh distortion [71]. This allows for smoothly updating the mesh node positions in the interior at each optimization step. However, after several steps the mesh may still become distorted, so we use an optimization based mesh smoothing method every few iterations [72].

Of course, in the case of large deformations even these methods will fail. Thus, whenever the mesh quality [73] gets sufficiently bad, we re-mesh the domain using the program 'Triangle' [74].

7. Numerical results

In our numerical experiments, we introduce a non-dimensional scaling of the cost functional J:

$$J_{\text{scaled}} := \frac{Re}{50} J. \tag{68}$$

We use this scaling uniformly for all our experiments so that J_{scaled} and Q have the same order of magnitude (note that J contains $1/Re$ by the definition (2) of σ). This ensures that the terms in the merit function (46) have similar weight. If the cost

Table 1
List of differences in the optimized power. The left side gives the percent difference between the symmetric and asymmetric solutions, i.e. $(J_{\text{sym}} - J_{\text{asym}})/J_{\text{asym}}$. The right side gives the percent difference between the flat bottom wall and asymmetric solutions, i.e. $(J_{\text{flat}} - J_{\text{asym}})/J_{\text{asym}}$. All differences are positive (except occlusion index 1, $Re = 20$) which means that the symmetric and flat bottom wall case is more costly in terms of input fluid power. This indicates that the asymmetric case is better able to minimize the viscous dissipation. For the anomalous case, the asymmetric and symmetry enforced shapes are very close; compare occlusion index 1 of Fig. 3(b) with Fig. 9(b). Also note the difference is very small.

Occ. index	Symmetry enforced				Flat bottom wall			
	$Re = 0.01$ (%)	$Re = 20$ (%)	$Re = 200$ (%)	$Re = 500$ (%)	$Re = 0.01$ (%)	$Re = 20$ (%)	$Re = 200$ (%)	$Re = 500$ (%)
1	2.97	−0.36	1.01	15.4	1.33	0.40	0.77	0.23
2	8.83	3.45	7.10	13.5	5.22	1.38	0.30	0.66
3	32.1	29.1	14.2	23.2	24.8	22.9	2.30	3.13
4	24.5	24.8	30.8	84.3	13.0	13.6	4.10	4.07
5	43.5	41.9	51.4	92.3	9.70	8.50	2.35	4.14
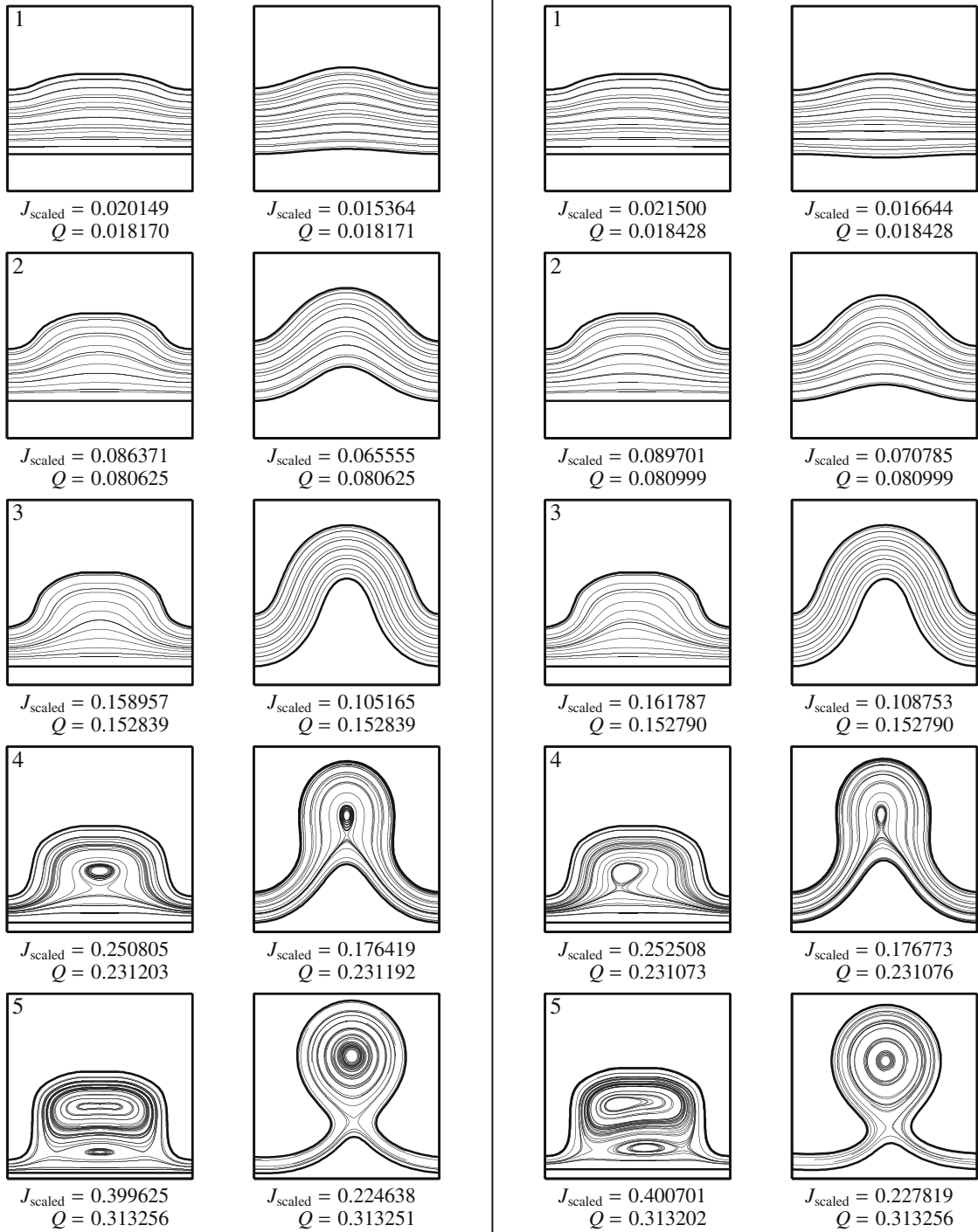
(a) *Re* = 0.01: Initial Guess (left) and Optimized Shape (right).   (b) *Re* = 20: Initial Guess (left) and Optimized Shape (right).

Fig. 3. Optimization for five different initial shapes and fluxes (asymmetric initial condition) with Reynolds number at (a) *Re* = 0.01 and (b) *Re* = 20. Each plot window is a unit square in non-dimensional units. The left column in each sub-figure is the initial guess for the shape; the right column is the shape obtained by our optimization algorithm. Each row, indexed 1–5 by increasing occlusion, corresponds to a desired amount of flux $Q(\Omega) = C_Q$ with increasing flux towards the bottom. The area constraint in each case is $A(\Omega) = C_A = 0.4$. The rate of energy dissipation J_{scaled} and average flux Q is stated underneath each plot. The interior curves of each shape represent streamlines of the flow referenced to the moving wave frame.
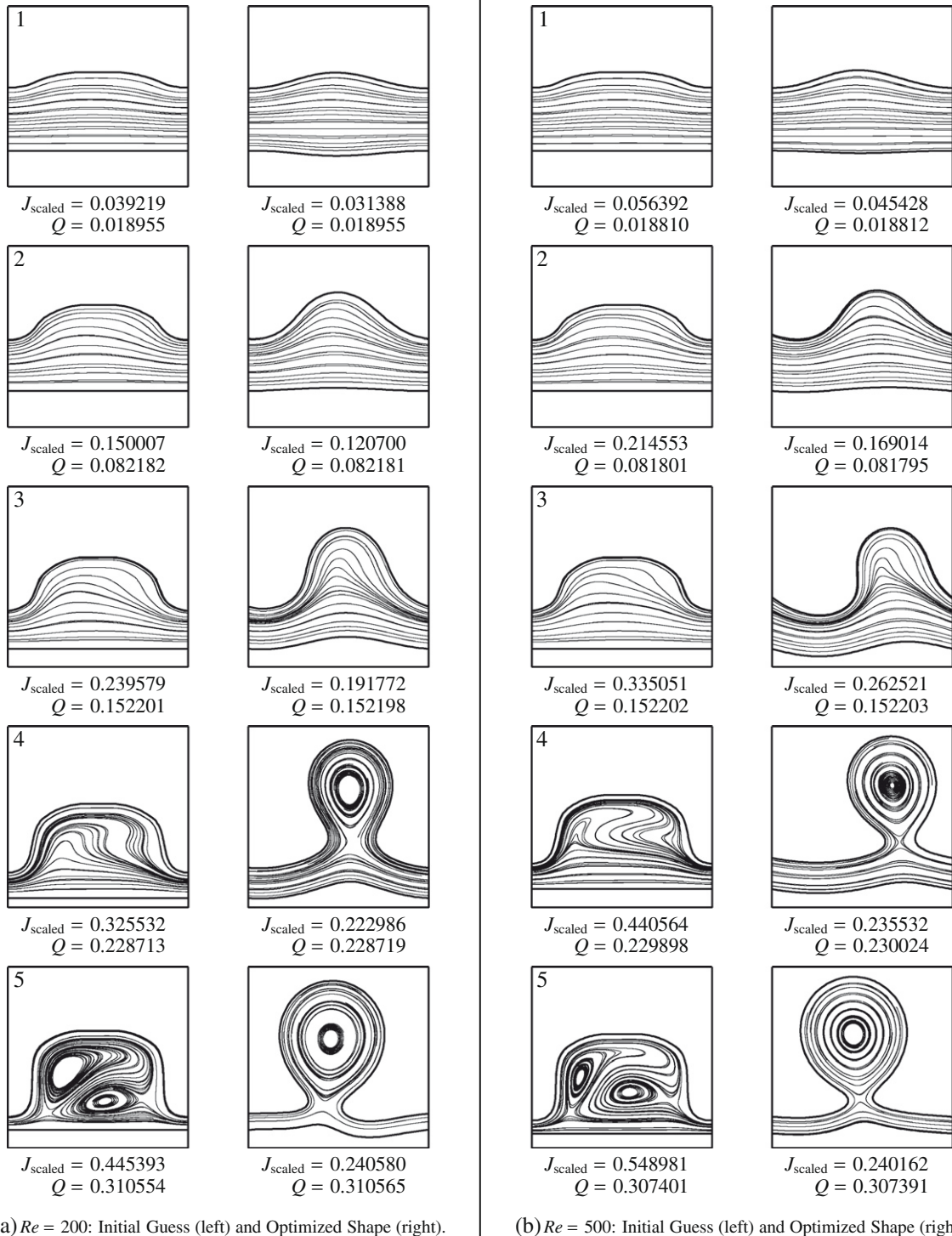
$J_{\text{scaled}} = 0.039219$
$Q = 0.018955$

$J_{\text{scaled}} = 0.031388$
$Q = 0.018955$

$J_{\text{scaled}} = 0.056392$
$Q = 0.018810$

$J_{\text{scaled}} = 0.045428$
$Q = 0.018812$

$J_{\text{scaled}} = 0.150007$
$Q = 0.082182$

$J_{\text{scaled}} = 0.120700$
$Q = 0.082181$

$J_{\text{scaled}} = 0.214553$
$Q = 0.081801$

$J_{\text{scaled}} = 0.169014$
$Q = 0.081795$

$J_{\text{scaled}} = 0.239579$
$Q = 0.152201$

$J_{\text{scaled}} = 0.191772$
$Q = 0.152198$

$J_{\text{scaled}} = 0.335051$
$Q = 0.152202$

$J_{\text{scaled}} = 0.262521$
$Q = 0.152203$

$J_{\text{scaled}} = 0.325532$
$Q = 0.228713$

$J_{\text{scaled}} = 0.222986$
$Q = 0.228719$

$J_{\text{scaled}} = 0.440564$
$Q = 0.229898$

$J_{\text{scaled}} = 0.235532$
$Q = 0.230024$

$J_{\text{scaled}} = 0.445393$
$Q = 0.310554$

$J_{\text{scaled}} = 0.240580$
$Q = 0.310565$

$J_{\text{scaled}} = 0.548981$
$Q = 0.307401$

$J_{\text{scaled}} = 0.240162$
$Q = 0.307391$

(a) $Re = 200$: Initial Guess (left) and Optimized Shape (right).   (b) $Re = 500$: Initial Guess (left) and Optimized Shape (right).

Fig. 4. Optimization for five different initial shapes and fluxes (asymmetric initial condition) with Reynolds number at (a) $Re = 200$ and (b) $Re = 500$. Same format as in Fig. 3.

and constraints are not scaled similarly, then it can adversely affect the convergence rate of our optimization algorithm [48]. Of course, we also scale δJ by $Re/50$.

For all numerical tests, we set $\xi = 0$ in (47). In our backtracking line search routine, we set a minimum acceptable step size of $\alpha_{\min} := 5 \times 10^{-6}$. In most of our numerical runs, we never achieve the minimum step size; we usually find
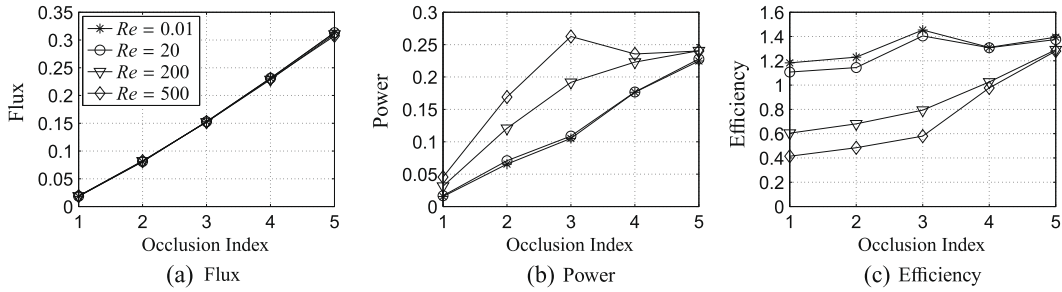
Fig. 5. Optimization results for the asymmetric case with both walls free. Plots of (a) flux Q, (b) power J_{scaled}, and (c) efficiency $\eta_{\text{eff}} := Q/J_{\text{scaled}}$ are given versus the occlusion of the channel (index = 1 means low occlusion; index = 5 means high occlusion). Four cases $Re = 0.01, 20, 200, 500$ are plotted for each occlusion index; the same legend notation is used in all three plots. The values plotted here correspond to the final shapes resulting from our optimization algorithm (see Figs. 3 and 4). In part (a), note the linear relationship between occlusion and average flux. Also note that Q is the non-dimensional flux, thus it is *independent* of the Reynolds number.
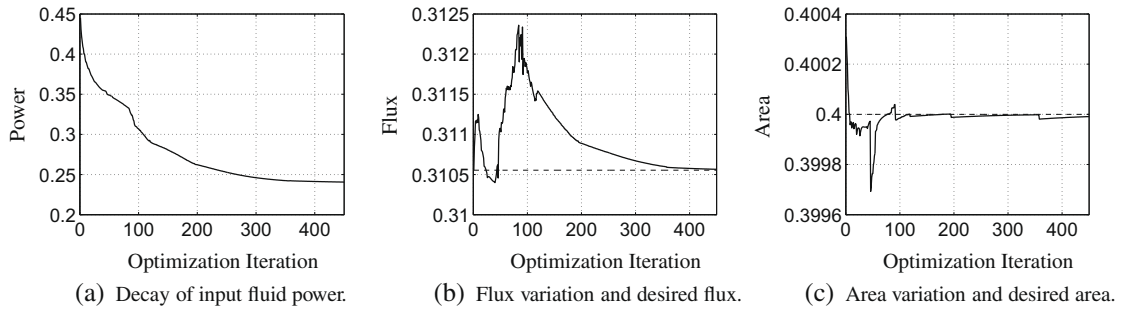

Fig. 6. Variation of relevant quantities for the case in Fig. 4(a), occlusion index 5 at $Re = 200$. Plots of (a) power J_{scaled}, (b) flux Q, and (c) area A are shown versus the optimization iteration index. In plots (b) and (c), a dashed line denotes the desired value. The input power is reduced by -46% of the initial amount. Violation of the flux and area constraints at the last iteration is $+0.0037\%$ and -0.0022% of the desired values, respectively. The 'kink' in part (a), at iteration index ≈ 85, corresponds to a sudden change in the way the shape is evolving (i.e. a better path in shape space is found to decrease the cost). Note how the merit function (46) allows for the constraints to be violated in the first 100 iterations, but eventually brings the evolving shape back to the constraint manifold.

$\alpha \approx 0.1-0.01$. One cause of taking very small steps is that the Navier–Stokes and adjoint PDE's are not solved accurately enough, which can lead to an inconsistency between J and δJ (similarly for Q and δQ). In other words, this can cause the optimization routine to become 'stuck'. If we reach the minimum step size for more than two iterations, we uniformly refine the mesh to increase accuracy in solving the PDEs and continue with the optimization; this avoids becoming stuck. Another cause of taking very small steps is an inappropriate choice of the function space $\mathbb{V}$ (42). We experimented with using the space $L^2(\Gamma)$ for $\mathbb{V}$ and found that this forces the optimization to take extremely small steps to avoid developing oscillations in the shape. Recalling the discussion in Remark 4, this is a crucial instance where knowing the continuous level functional sensitivities $\delta J, \delta Q$ is able to benefit the optimization algorithm (i.e. the choice of $\mathbb{V}$).

We define an efficiency for peristaltic pumping by

$$\eta_{\text{eff}} := \frac{Q}{J_{\text{scaled}}}. \tag{69}$$

Eq. (69) is different than the standard definition $\eta_{\text{eff}} = \Delta p \cdot Q/J_{\text{scaled}}$ [26,19] because our problem does not have a net rise in pressure per wavelength of the channel (i.e. we are in the free-pumping regime).

We present a variety of applications for our optimization method. Section 7.1 shows several cases for a range of Re where the initial domain shape is not symmetric top-bottom and the top and bottom walls Γ^+ and Γ^- are free to evolve in order to reduce the cost J. Here, we find that the resulting shapes are far from symmetric. In Section 7.2 we explicitly enforce symmetry, so only the top wall moves, and we find that the input fluid power is much larger than the previous case (up to 92% higher, see Table 1). Section 7.3 is similar to the first case, except we fix the bottom wall to be horizontal and only allow the top wall to evolve. Lastly, Section 7.4 shows the optimization of a double peaked initial shape. All simulations were run until the reduction of the cost J appeared to saturate and the relative change in the power was less than 0.001% of the initial cost and the constraint violation for both flux and area was less than 0.1% of the desired values.
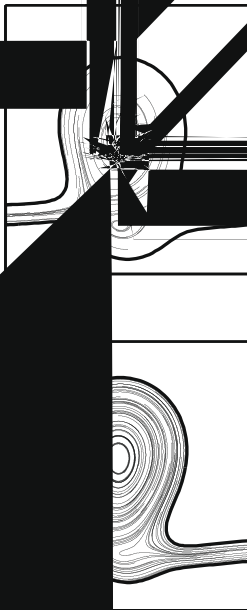
...ymmetric solutions with both walls free

... 4, we show the results of optimizing peristaltic wave shape for four different cases of Reynolds number ... 00, 500. For each Reynolds number, we chose a sequence of five initial shapes with a varying amount of ... n (i.e. the minimal spacing between the walls). We index the initial shapes according to the amount of ... 1 is the least occluded while index 5 is the most occluded. We refer to this as the occlusion (occ.) index. ... mplies more flux.

(a) Final shape from Figure 4 (a) (occ. index 5, $Re = 200$).   (b) Final shape from Figure 4 (b) (occ. index 3, $Re = 5$
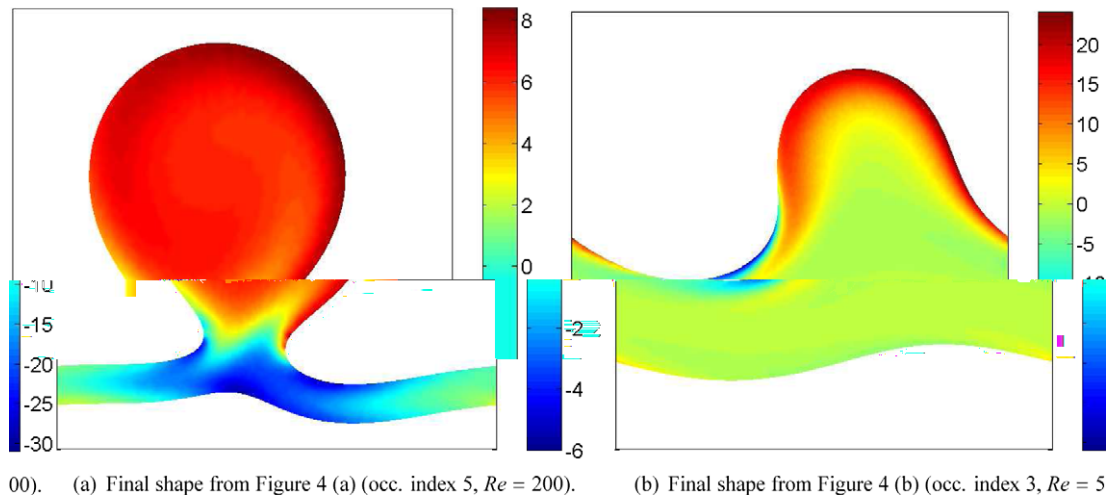
Fig. 8. Vorticity (color) plots for two final shapes from Fig. 4. Both 'waves' propagate towards the right in the fixed lab frame. In part (a), the final shape has trapped a large rotating bolus of fluid in a 'bag-like' region. Part (b) shows a medium desired flux case with vorticity concentrated at the top wall. Note that neither wave shape (i.e. the top wall) is the graph of a function.

In all cases, the desired domain area is $C_A = 0.4$ and the initial guess for the shape satisfies this. The desired flux C_Q is taken to be the average flux for the initial shape. The streamlines of the flow (plotted with respect to the moving wave frame) are also shown. Note the lack of symmetry (top/bottom) between the channel walls. This becomes very pronounced for larger fluxes (see the bottom two rows of Figs. 3 and 4) where the top wall is no longer the graph of a function. Moreover, the streamlines for the higher flux cases show the presence of two vortices (in the initial shape of the channel) which implies that the viscous dissipation J is large. Our optimization algorithm reduces J by deforming the walls so as to isolate a *single* bolus of fluid, which propagates towards the right in the fixed lab frame. It is clear that this configuration is favorable for reducing J by reducing the symmetrized form of the velocity gradient (see Fig. 7 for an illustration of the change in the flow field as the optimization progresses).
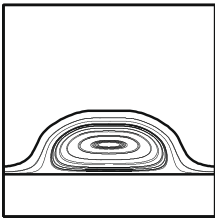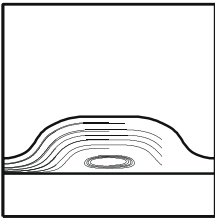
In row 3 of Fig. 3, it appears that the final shape of the walls is close to two out of phase sine waves. It has been shown in the literature (for instance, see [9]) that the average flux in peristaltic pumping for two sine waves that are perfectly out of phase is close to zero in the long wavelength regime. However, the domain shapes in our experiments do not fall under the long wavelength assumption. But it is interesting that in the 'short' wavelength regime such an arrangement can now appear as an optimal solution.

There is also an asymmetry that develops as the Reynolds number increases. For instance, note the difference in the final shape between Figs. 3 and 4 for occlusion index 3. Also, the streamlines become more asymmetric with increasing Re for all cases. However, despite the broken left/right symmetry, the final shapes for the higher flux cases (rows 4 and 5) are remarkably similar for the range of Reynolds numbers considered here. For low Reynolds number, see Fig. 3(a), the final shapes are left-right symmetric which is a consequence of the reversibility of Stokes flow. In row 5 of Fig. 3(a), the final shape is offset from the exact center. In our optimization runs, we noticed a small amount of 'sliding' of the shape in a few cases. However, this is only a superficial effect because all peristaltic pumping shapes are invariant under translation in the $\mathbf{e}_1$ direction (because of periodicity). We believe this is due to discretization error, especially since the meshes used to represent Ω are unstructured with no underlying symmetry.
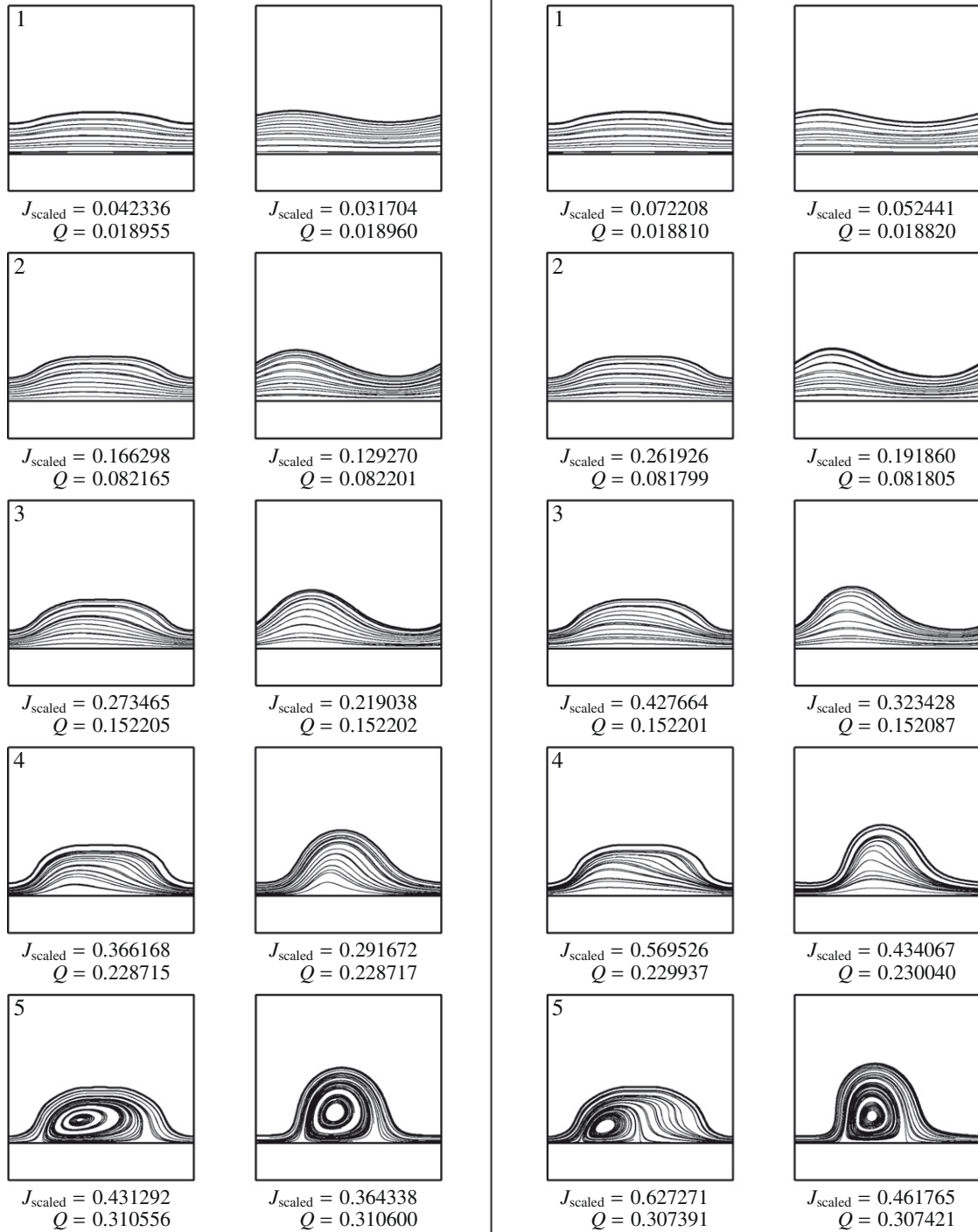
In Fig. 5, we plot the flux, input fluid power (or rate of viscous dissipation), and the pumping efficiency versus the occlusion index for all four cases of Reynolds number (referring to Figs. 3 and 4). Note that the amount of occlusion is directly proportional to the flux. The input power also increases with occlusion for low Reynolds number, but peaks at index 3 for $Re = 500$. There is also a peak in the efficiency for $Re = 0.01$ and 20 at occlusion index 3. The efficiency improves with increasing channel occlusion for higher Reynolds number.

As a representative sample, we give more information in Figs. 6 and 7 about the optimization process for the case of (occ. index 5, $Re = 200$) described in Fig. 4(a). Fig. 6 shows the change in power, flux, and area with iteration and Fig. 7 shows several frames depicting the evolution of the peristaltic channel shape during the optimization. From Fig. 6 we see that the input fluid power decays as the optimization progresses (reduced by -46%), while the constraints are at first violated and then eventually return to their desired values. In Fig. 7, we see how the algorithm acts to isolate the flow into an asymmetric 'bag-like' region. In a biological application, there will certainly be some penalty associated with deforming the channel walls, e.g. energy expended in muscle contraction. There is no penalty for wall deformation in our optimization criteria. However, it is interesting to note the kinds of optimal shapes favored when wall deformations are not costly to perform.

In Fig. 8, we plot the vorticity field (in color) for the final shapes of (occ. index 5, $Re = 200$) and (occ. index 3, $Re = 500$) depicted in Fig. 4. Both 'waves' propagate towards the right in the fixed lab frame. Part (a) of the figure clearly shows a large

...gion of the fluid (i.e. a bolus) trapped in a 'bag-like' region. The vorticity in the bag is approximately uniform and is ...ounter-clockwise (positive) which is consistent with the motion of the traveling wave. A region of negative vortic-...ent in the 'tube' region. In part (b), the desired flux is less and the optimized shape has a less extreme wave-shape.

$J_{\text{scaled}} = 0.042336$
$Q = 0.018955$

$J_{\text{scaled}} = 0.031704$
$Q = 0.018960$

$J_{\text{scaled}} = 0.072208$
$Q = 0.018810$

$J_{\text{scaled}} = 0.052441$
$Q = 0.018820$

$J_{\text{scaled}} = 0.166298$
$Q = 0.082165$

$J_{\text{scaled}} = 0.129270$
$Q = 0.082201$

$J_{\text{scaled}} = 0.261926$
$Q = 0.081799$

$J_{\text{scaled}} = 0.191860$
$Q = 0.081805$

$J_{\text{scaled}} = 0.273465$
$Q = 0.152205$

$J_{\text{scaled}} = 0.219038$
$Q = 0.152202$

$J_{\text{scaled}} = 0.427664$
$Q = 0.152201$

$J_{\text{scaled}} = 0.323428$
$Q = 0.152087$

$J_{\text{scaled}} = 0.366168$
$Q = 0.228715$

$J_{\text{scaled}} = 0.291672$
$Q = 0.228717$

$J_{\text{scaled}} = 0.569526$
$Q = 0.229937$

$J_{\text{scaled}} = 0.434067$
$Q = 0.230040$

$J_{\text{scaled}} = 0.431292$
$Q = 0.310556$

$J_{\text{scaled}} = 0.364338$
$Q = 0.310600$

$J_{\text{scaled}} = 0.627271$
$Q = 0.307391$

$J_{\text{scaled}} = 0.461765$
$Q = 0.307421$

(a) $Re = 200$: Initial Guess (left) and Optimized Shape (right).   (b) $Re = 500$: Initial Guess (left) and Optimized Shape (right).

Fig. 10. Optimization for five different initial shapes and fluxes (symmetry top/down is explicitly enforced) with Reynolds number at (a) $Re = 200$ and (b) $Re = 500$. Same format as in Fig. 9. The desired flux $Q(\Omega) = C_Q$ for each occlusion index and Reynolds number is the same as for each occlusion index and Reynolds number in Fig. 4.

Here, the vorticity (both positive and negative) is concentrated on the top wall; the vorticity in the lower region is close to zero. For both shapes, the top wall cannot be represented as the graph of a function. In most of the peristaltic pumping literature, the traveling wave is assumed to be the graph of a function which translates at a constant speed. This is not the case in our framework because we have treated the traveling wave shape differently and the walls are inextensible.
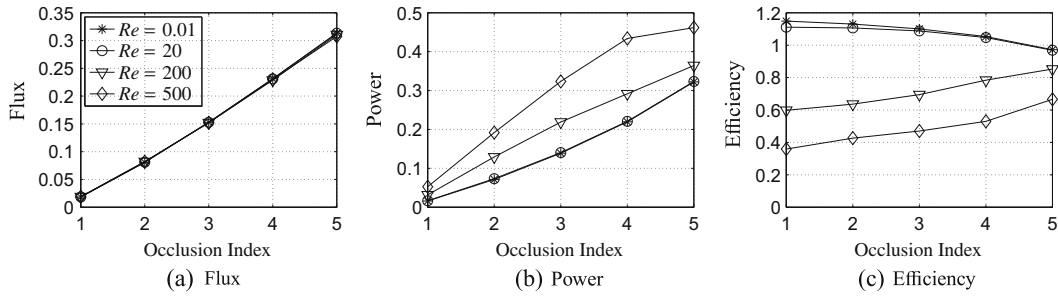
Fig. 11. Optimization results when top/down symmetry is explicitly enforced. Thus, only the top wall is free to deform. Plots of (a) flux Q, (b) power J_{scaled}, and (c) efficiency $\eta_{\text{eff}} := Q/J_{\text{scaled}}$ are given versus the occlusion of the channel (index = 1 means low occlusion; index = 5 means high occlusion). Four cases $Re = 0.01, 20, 200, 500$ are plotted for each occlusion index; the same legend notation is used in all three plots. The values plotted here correspond to the final shapes resulting from our optimization algorithm (see Figs. 9 and 10).
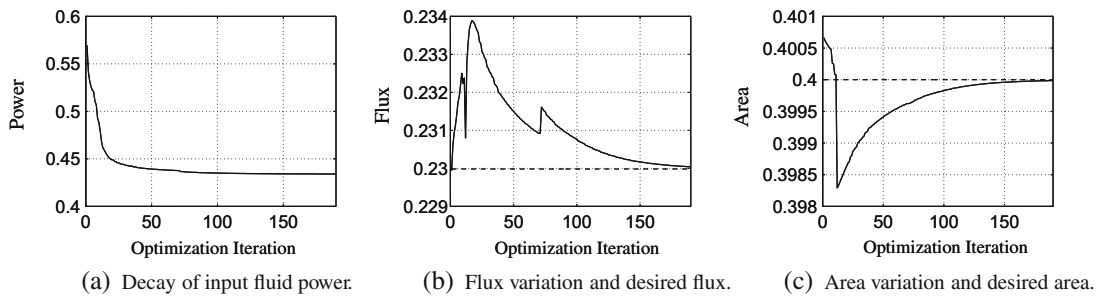
Fig. 12. Variation of relevant quantities for the case in Fig. 10(b), occlusion index 4 at $Re = 500$. Plots of (a) power J_{scaled}, (b) flux Q, and (c) area A are shown versus the optimization iteration index. In plots (b) and (c), a dashed line denotes the desired value. The input power is reduced by -23.8% of the initial amount. Violation of the flux and area constraints at the last iteration is $+0.0242\%$ and -0.0035% of the desired values, respectively. Note the jumps in the flux at approximate iteration indices 12 and 72. This corresponds to globally re-meshing the fluid domain.

7.2. Up-down symmetric solutions with both walls free

For this case, we only optimize the top-half of the channel. Specifically, we restrict Γ^- (here, the x-axis) to be a line-of-symmetry with boundary condition given by

$$\begin{aligned}
\mathbf{u} \cdot v &= 0, \quad \text{on } \Gamma^-, \\
\tau \cdot \sigma v &= 0, \quad \text{on } \Gamma^-,
\end{aligned} \tag{70}$$

which means that $u_2 = 0$ and $\partial u_1/\partial y$ where $\mathbf{u} = (u_1, u_2)$. Hence, the top wall shape is left free to be optimized, while the bottom wall is made flat/horizontal. Eq. (70) is used in the Navier–Stokes system (8) and the adjoint systems (23) and (32).

Figs. 9 and 10 (in the format of Figs. 3 and 4) show our results when symmetry is explicitly imposed. Note only the top-half of the domain is shown but the power J_{scaled}, flux Q, and area A is given for the entire domain. The desired domain area for all cases is again $C_A = 0.4$ which the initial shape satisfies. For each occlusion index and Reynolds number, the desired flux C_Q is set to be the same as for the corresponding occlusion index and Reynolds number in Figs. 3 and 4. This was done in order to directly compare the results of this section with Section 7.1.

The optimal power achieved by the optimization routine when symmetry is enforced is higher than in the previous section (asymmetric case). Computing the normalized difference $(J_{\text{sym}} - J_{\text{asym}})/J_{\text{asym}}$ for occlusion index 1 and 2 says that the symmetric case is $\approx 1\%$ to 15% larger than the asymmetric case. The difference for indices 3, 4, and 5 ranges from $\approx 14\%$ to 92% above the asymmetric case (see Table 1). The reduction in power is substantial for the asymmetric case given the less constrained nature of the optimization in Section 7.1 (i.e. both walls are independent of each other). The power cost is especially expensive in the symmetric case for the largest desired flux because of the presence of two vortices instead of one in the asymmetric case.

In Fig. 11, we plot the flux, input fluid power (or rate of viscous dissipation), and the pumping efficiency versus the occlusion index for all four cases of Reynolds number (referring to Figs. 9 and 10). Flux increases linearly with channel occlusion just as in Fig. 5. But there is no peak in the input fluid power, only a general increase with increasing occlusion. There is also no peak in the efficiency as before. Here, η_{eff} decreases (increases) with increasing occlusion when $Re = 0.01, 20$ ($Re = 200, 500$).

(a) $Re = 0.01$: Initial Guess (left) and Optimized Shape (right).   (b) $Re = 20$: Initial Guess (left) and Optimized Shape (right).
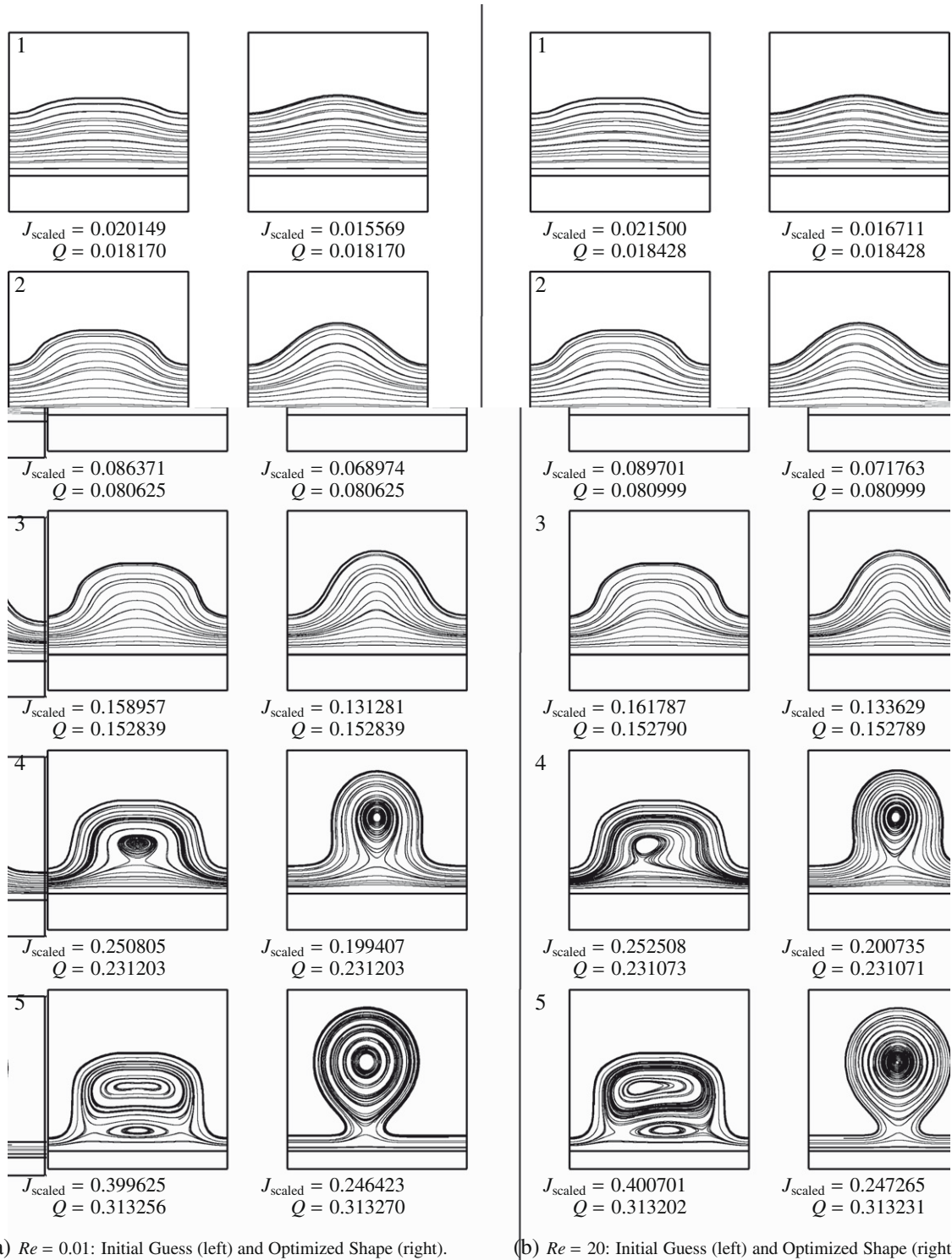
Fig. 13. Optimization for five different initial shapes and fluxes (asymmetric initial condition, bottom wall constrained to be flat) with Reynolds number at (a) $Re = 0.01$ and (b) $Re = 20$. Same format as in Fig. 3. The area constraint is $A(\Omega) = C_A = 0.4$. The desired fluxes $Q(\Omega) = C_Q$ are exactly the same as in Fig. 3. In part (a), occlusion index 5, the final shape is shifted over from the center. This is allowable because any peristaltic pumping shape is invariant under translation (by periodicity).

Fig. 12 shows the change in power, flux, and area with iteration for the case of (occ. index 4, $Re = 500$) described in Fig. 10(b). Our algorithm decreases the input fluid power, relative to the starting point, by -23.8%. The input fluid power
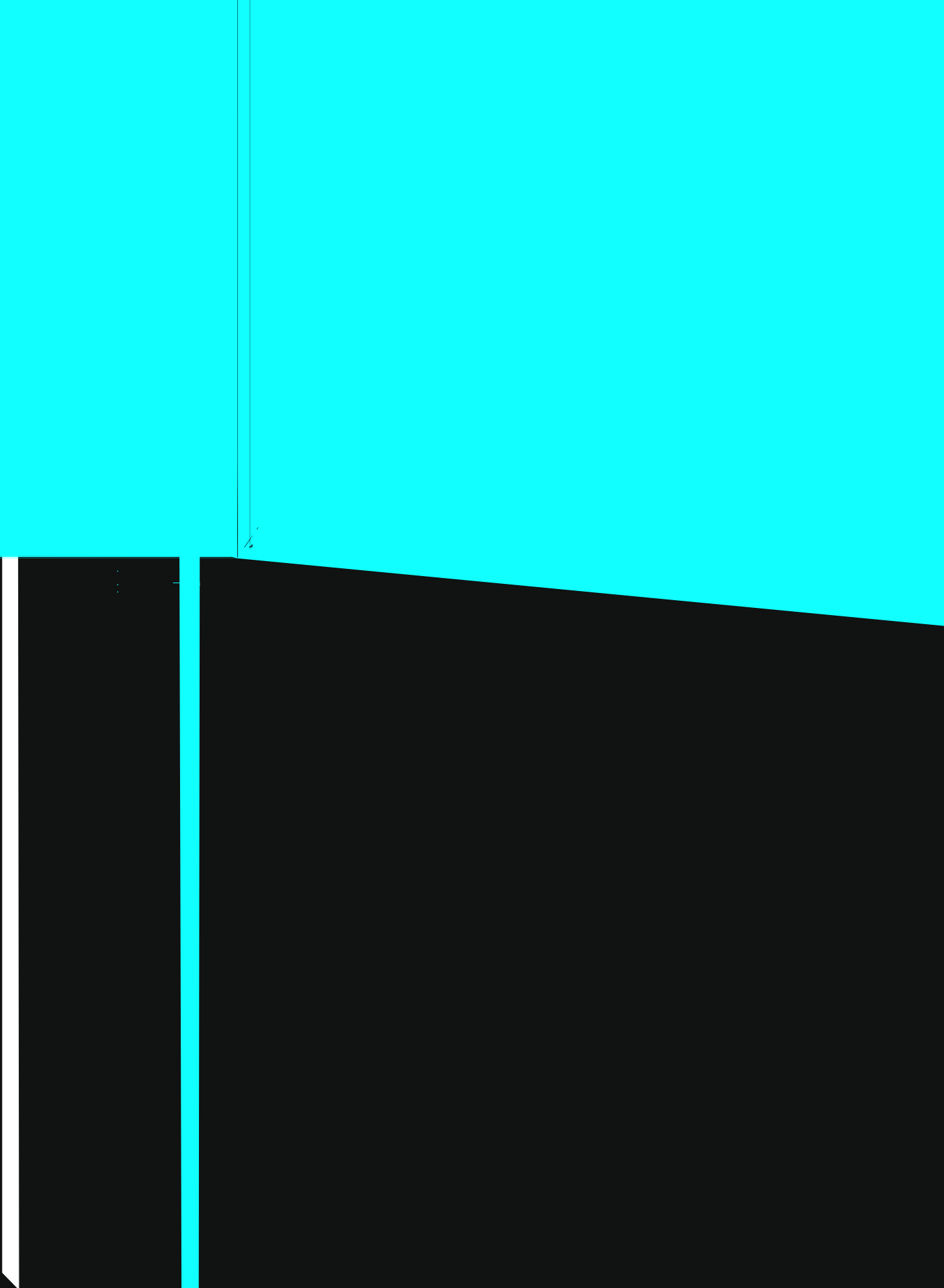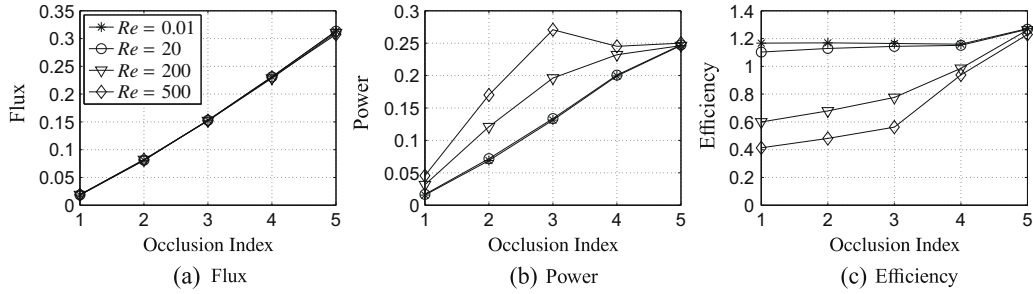
Fig. 15. Optimization results for the asymmetric case with bottom wall constrained to be flat. Plots of (a) flux Q, (b) power J_{scaled}, and (c) efficiency $\eta_{eff} := Q/J_{scaled}$ are given versus the occlusion of the channel (index = 1 means low occlusion; index = 5 means high occlusion). Four cases $Re = 0.01, 20, 200, 500$ are plotted for each occlusion index; the same legend notation is used in all three plots. Note the linear relationship between occlusion and average flux and recall that Q is non-dimensional, thus it does not depend on Re.
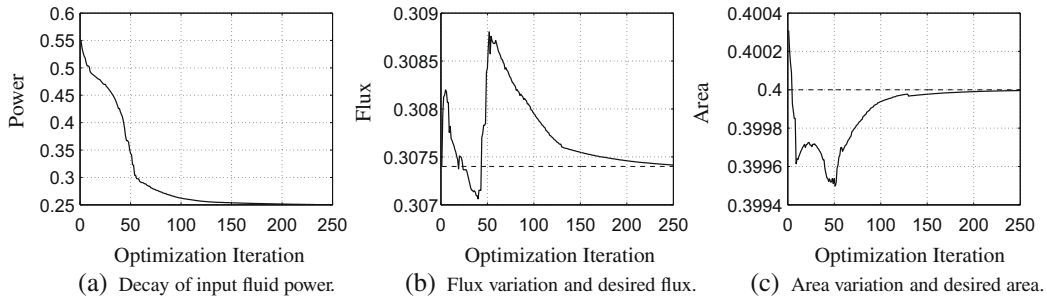

Fig. 16. Variation of relevant quantities for the case in Fig. 14(b), occlusion index 5 at $Re = 500$. Plots of (a) power J_{scaled}, (b) flux Q, and (c) area A are shown versus the optimization iteration index. In plots (b) and (c), a dashed line denotes the desired value. The input power is reduced by −54.4% of the initial amount. Violation of the flux and area constraints at the last iteration is +0.0047% and −0.0010% of the desired values, respectively. Note the 'kink' regions in the power at approximate iteration indices 10 and 55, which correlate with the relatively large violations of the flux constraint (also at iteration indices 10 and 55). Immediately after the kinks, the power decrease slows down and the optimization evolves the shape in order to improve the constraint satisfaction. The small jumps in the constraints correspond to re-meshing the fluid domain.

7.3. Up-down asymmetric solutions with bottom wall flat

This section is the same as Section 7.1, except that the bottom wall is constrained to be flat/horizontal. This is similar to many industrial pumps where rollers traverse one side of a tube. Note: there is no line-of-symmetry here. Figs. 13 and 14 show our results for a range of desired fluxes and Re (similar to Figs. 3 and 4). The desired domain area for all cases is again $C_A = 0.4$ which the initial shape satisfies. The desired fluxes correspond to those in Figs. 3 and 4. This was done in order to directly compare the results of this section with Section 7.1.

The optimization in this section is slightly more constrained than in Section 7.1 because the bottom wall is forced to be flat. Thus, the optimal shapes in this case have cost value J slightly higher than when both walls are allowed to change (approximately 0.3–25% higher, see Table 1). Other than this, the final shapes are very similar to Figs. 3 and 4. The largest difference can be seen for occlusion index 2 and 3.

In Fig. 15, we plot the flux, input fluid power (or rate of viscous dissipation), and the pumping efficiency versus the occlusion index for all four cases of Reynolds number (referring to Figs. 13 and 14). The plots are similar to Fig. 5, e.g. the power peaks at index 3 for $Re = 500$. However, the efficiency does not peak for low Re. In fact, the efficiency is almost constant for $Re = 0.01$ and 20. But the efficiency does improve markedly, with increasing occlusion index, for $Re = 200$ and 500.

Fig. 16 shows the change in power, flux, and area with iteration for the case of (occ. index 5, $Re = 500$) described in Fig. 14(b). Our algorithm decreases the input fluid power by −54.4%. The general behavior is similar to Figs. 6 and 12.

7.4. Optimization of shape with "double peaked" initial data

Our last example considers an initial shape with two peaks within one wavelength of the channel. The optimization criteria is the same and we allow both walls to deform. The desired flux C_Q and volume C_A is taken to be the value given by the initial domain shape. Fig. 17 shows several frames depicting the evolution of the peristaltic channel shape during the optimization. The initial shape has a large amount of viscous dissipation as can be seen by the rapidly curving streamlines. The algorithm acts to reduce the input fluid power (viscous dissipation) by removing the two peaks and making the channel more uniform in thickness. The final shape is similar to the out of phase sine wave shape in Fig. 3(a) (see occlusion index
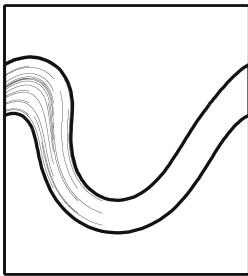
3). During optimization the left-right symmetry is broken, which is most likely due to having an unstructured, non-symmetric mesh. Nevertheless, the final shape has left-right symmetry because the flow regime is low Reynolds number $Re = 0.01$. There is a slight shift of the final shape from being perfectly centered. Again, this is only superficial because peristaltic pumping profiles are invariant under translation (by periodicity).

In Fig. 18, we see the change in power, flux, and area with optimization iteration. The input fluid power decays as the optimization progresses (reduced by -81.5%). The constraints become violated during different parts of the optimization, but eventually return to their desired values.
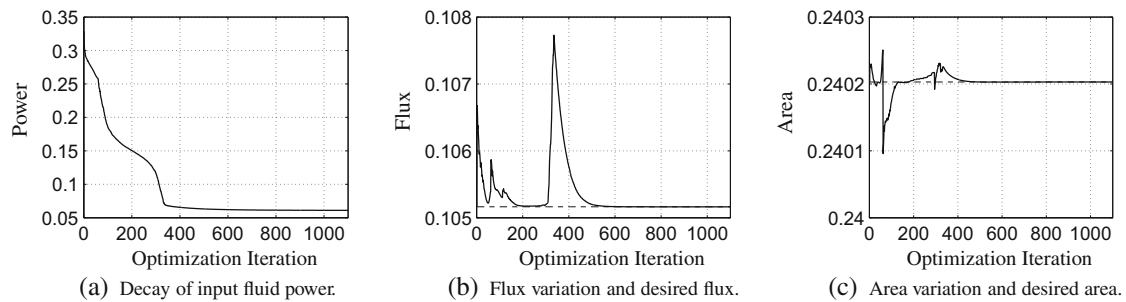
(a) Decay of input fluid power. (b) Flux variation and desired flux. (c) Area variation and desired area.

Fig. 18. Variation of relevant quantities for the case in Fig. 17, $Re = 0.01$. Plots of (a) power J_{scaled}, (b) flux Q, and (c) area A are shown versus the optimization iteration index. In plots (b) and (c), a dashed line denotes the desired value. The input power is reduced by -81.5% of the initial amount. Violation of the flux and area constraints at the last iteration is -0.0002% and 0.0000% of the desired values, respectively.

8. Conclusion

We have presented a variational method for optimizing the shape of a peristaltic traveling wave for pumping. This method is potentially useful for understanding certain biological processes, such as mucus transport in the lungs and embryonic transport in the reproductive system, and optimizing industrial pumps that operate by peristalsis.

8.1. Direct extensions of the 2D Case

Our method can be easily extended to include other models of fluid response. For instance, a Generalized Newtonian model that has shear-thinning or shear-thickening can be included in our framework without much effort, and would be relevant for examining mucus transport in the lungs. A fully viscoelastic model, such as Oldroyd-B, would be relevant for investigating peristaltic pumping of polymeric fluids; this situation was simulated in [25], where the traveling wave was defined to be a sine wave. However, optimizing with an Oldroyd-B model would require significant changes in the PDE solvers.

Other extensions could include penalizing large deformations of the channel walls (recall the bag-like regions in rows 4 and 5 of Figs. 3 and 4) by adding a penalty term (e.g. $\int_\Gamma \kappa^2$ or $\int_\Gamma 1$) to the optimization cost function. One can also add an elastic wall model that is coupled to the interior fluid, thus changing the boundary conditions of the governing PDE equations.

In addition to optimizing the shape, one can also allow for local stretching and contraction of the channel walls (we treat the channel walls as inextensible in this paper). In other words, one can treat the velocity boundary condition as an extra control variable (or function) to be optimized. Of course, one would need to put suitable constraints on the amount of in-wall deformation. Another interesting extension would be to look at a non-periodic channel with an open end. Optimization in this setting would lead to an optimal control problem for the time-varying shape of the channel walls.

8.2. Optimization in 3D

The first step for investigating peristaltic pumping in 3D would be to assume an axisymmetric shape, which is relevant to many applications. For this, cylindrical coordinates would be an obvious choice in posing the problem. And while the sensitivities would certainly change and require a different derivation, the fundamental quantities would remain the same, i.e. curvature and surface stresses on the 1D curve representing the axisymmetric shape would still appear in the linear forms representing the sensitivities. However, the channel walls could no longer be assumed to be inextensible, in which case an elastic wall model should be used. In this case, it would still be relevant to look at static channel shapes with respect to the wave frame. Here, one could use the elastic wall model to infer the velocity boundary condition for a *given* shape of the domain and wave speed. In addition, another energy term could be added to the cost functional that depends on the elastic wall energy to investigate the trade-off between viscous dissipation and wall deformation.

We believe that with axisymmetry one would still see non-obvious optimal shapes resulting from our algorithm. It is conceivable that a torus-like region (analogous to the 'bag' region depicted in the higher flux cases) could be an optimal shape if the enclosed volume per wavelength is large enough. But other shapes, like those in the symmetry enforced case (see Section 7.2) may also be likely.

For general 3D peristalsis, the time-independent assumption (in the wave frame) may still be relevant for moderate Reynolds number. The most critical aspect will be dealing properly with the geometry. For instance, the curvature computation given in Section 6.3 will *not* work when approximating the surface by a piecewise linear triangulation (i.e. the L^2 projection of curvature does not converge for 2D surfaces in 3D; see [75]). In order to get a convergent method, higher order geometric elements (curved triangles) must be used. In contrast, the L^2 projection for the stress does generalize to the 3D case, and the error analysis would simplify when using curved triangles to approximate the surface. Lastly, optimizing general 3D peristal-

sis will be more computationally intense and require adequate meshing tools to deal with the deforming domain as the optimization algorithm progresses [76].

Acknowledgments

Appendix A. Differential geometry

We collect here some elementary results from differential geometry. Let Γ be a closed surface, or curve, and let ∇_Γ be the surface (or tangential) gradient. The subscript Γ in ∇_Γ refers to the surface on which the surface gradient is being computed. The Laplace–Beltrami operator is given by $\Delta_\Gamma := \nabla_\Gamma \cdot \nabla_\Gamma$. In the case of a 1D curve, $\nabla_\Gamma = \tau \partial_\tau$ (i.e. the surface gradient is the oriented unit tangent vector times the derivative with respect to arc-length) and $\Delta_\Gamma = \partial_\tau^2$.

We make repeated use of the identity:

$$\mathbf{I} = \boldsymbol{v}\boldsymbol{v}^\dagger + \boldsymbol{\tau}\boldsymbol{\tau}^\dagger = \boldsymbol{v} \otimes \boldsymbol{v} + \boldsymbol{\tau} \otimes \boldsymbol{\tau}, \tag{71}$$

where $\boldsymbol{v}$ is the outer unit normal to Γ. We also make repeated use of the following decomposition of the gradient and divergence operators:

$$\nabla() = \boldsymbol{v}^\dagger(\boldsymbol{v} \cdot \nabla)() + \nabla_\Gamma(), \tag{72}$$
$$\nabla \cdot () = \boldsymbol{v}^\dagger \cdot [(\boldsymbol{v} \cdot \nabla)()] + \nabla_\Gamma \cdot (). \tag{73}$$

Next, let $\mathbf{X}$ be a vector parameterization of the surface (or curve) Γ. Then $\nabla_\Gamma \mathbf{X}$ is a projection matrix onto the tangent space of Γ, i.e. $\nabla_\Gamma \mathbf{X} = \mathbf{I} - \boldsymbol{v} \otimes \boldsymbol{v}$. We also have the following symmetric relations:

$$\nabla_\Gamma \mathbf{X} = [\nabla_\Gamma \mathbf{X}]^\dagger, \tag{74}$$
$$\nabla_\Gamma \boldsymbol{v} = [\nabla_\Gamma \boldsymbol{v}]^\dagger. \tag{75}$$

Furthermore, the Laplace–Beltrami (or 'surface' Laplacian) of the parameterization is the total curvature vector:

$$-\Delta_\Gamma \mathbf{X} := -\nabla_\Gamma \cdot [\nabla_\Gamma \mathbf{X}] = \kappa \boldsymbol{v}, \tag{76}$$

where κ is the total scalar curvature [77] and is positive when Γ is convex. The following 'integration by parts' formulas involving curvature are useful:

$$\int_\Gamma \nabla_\Gamma \omega = \int_\Gamma \omega \kappa \boldsymbol{v}, \qquad \int_\Gamma \nabla_\Gamma \boldsymbol{\varphi} = \int_\Gamma \boldsymbol{\varphi} \otimes \boldsymbol{v}\kappa, \qquad \int_\Gamma \nabla_\Gamma \cdot \boldsymbol{\varphi} = \int_\Gamma \boldsymbol{\varphi} \cdot \boldsymbol{v}\kappa, \tag{77}$$

for all smooth scalar ω and vector $\boldsymbol{\varphi}$.

Lastly, when Γ is perturbed by a vector field $\mathbf{V}$, we have that the material derivative of the unit outward normal and tangent vectors is given by:

$$\dot{\boldsymbol{v}} = -[\nabla_\Gamma \mathbf{V}]^\dagger \boldsymbol{v}, \tag{78}$$
$$\dot{\boldsymbol{\tau}} = (\boldsymbol{v}^\dagger[\nabla_\Gamma \mathbf{V}]\boldsymbol{\tau})\boldsymbol{v}, \tag{79}$$

where $\boldsymbol{\tau} := \boldsymbol{v}^\perp = (-v_2, v_1)$ is the oriented tangent vector.

Appendix B. Some identities

Let $\mathbf{u}$, $\mathbf{r}$ be smooth vector fields defined on a smooth domain Ω (with $\Gamma = \partial\Omega$) and suppose $\mathbf{u} = \mathbf{r} = \boldsymbol{\tau}$ on Γ and $\nabla \cdot \mathbf{u} = \nabla \cdot \mathbf{r} = 0$. The first identity we have is:

$$0 = \nabla \cdot \mathbf{u}|_\Gamma = [(\boldsymbol{v} \cdot \nabla)\mathbf{u}] \cdot \boldsymbol{v} + \nabla_\Gamma \cdot \mathbf{u} = [(\boldsymbol{v} \cdot \nabla)\mathbf{u}] \cdot \boldsymbol{v}, \tag{80}$$

because $\nabla_\Gamma \cdot \mathbf{u} = \pm\nabla_\Gamma \cdot \boldsymbol{\tau} = 0$ on Γ. Next, we have

$$\begin{aligned}(D(\mathbf{u})\boldsymbol{v}) \cdot (D(\mathbf{r})\boldsymbol{v}) &= \boldsymbol{v}^\dagger D(\mathbf{u})D(\mathbf{r})\boldsymbol{v}, \\ &= (\boldsymbol{v}^\dagger D(\mathbf{u})\boldsymbol{v})(\boldsymbol{v}^\dagger D(\mathbf{r})\boldsymbol{v}) + (\boldsymbol{v} \cdot D(\mathbf{u})\boldsymbol{\tau})(\boldsymbol{\tau} \cdot D(\mathbf{r})\boldsymbol{v}), \\ &= (\boldsymbol{v} \cdot D(\mathbf{u})\boldsymbol{\tau})(\boldsymbol{\tau} \cdot D(\mathbf{r})\boldsymbol{v}),\end{aligned} \tag{81}$$

by expanding with (71), and using (80).

The next relation involves $D(\cdot)$ (recall $D(\mathbf{u}) := (\nabla\mathbf{u} + (\nabla\mathbf{u})^\dagger)$):

$$D(\mathbf{u}) : D(\mathbf{r})|_\Gamma = [(D(\mathbf{u})\nu)\nu^\dagger + (D(\mathbf{u})\tau)\tau^\dagger] : D(\mathbf{r}),$$
$$= (D(\mathbf{u})\nu) \cdot (D(\mathbf{r})\nu) + (D(\mathbf{u})\tau) \cdot (D(\mathbf{r})\tau),$$

where we used (71). Concentrating on $(D(\mathbf{u})\tau) \cdot (D(\mathbf{r})\tau)$, we see that

$$(D(\mathbf{u})\tau) \cdot (D(\mathbf{r})\tau)|_\Gamma = (\nu^\dagger D(\mathbf{u})\tau)(\nu^\dagger D(\mathbf{r})\tau) + (\tau^\dagger D(\mathbf{u})\tau)(\tau^\dagger D(\mathbf{r})\tau),$$
$$= (\nu^\dagger D(\mathbf{u})\tau)(\nu^\dagger D(\mathbf{r})\tau) + [(\partial_\tau \mathbf{u}) \cdot \tau][(\partial_\tau \mathbf{r}) \cdot \tau],$$
$$= (\nu^\dagger D(\mathbf{u})\tau)(\nu^\dagger D(\mathbf{r})\tau),$$

again using (71) and the fact that $\mathbf{u}$ and $\mathbf{r}$ are tangential on Γ. Therefore,

$$D(\mathbf{u}) : D(\mathbf{r})|_\Gamma = (D(\mathbf{u})\nu) \cdot (D(\mathbf{r})\nu) + (\nu^\dagger D(\mathbf{u})\tau)(\nu^\dagger D(\mathbf{r})\tau),$$
$$= 2(D(\mathbf{u})\nu) \cdot (D(\mathbf{r})\nu) = 2(\tau \cdot D(\mathbf{u})\nu) \cdot (\tau \cdot D(\mathbf{r})\nu), \tag{82}$$

by the relation (81) and symmetry of $D(\cdot)$.

Lastly, we have the following relation involving the stress tensor $\boldsymbol{\sigma}$ (recall (2)):

$$\boldsymbol{\sigma} : \nabla \mathbf{r} = \frac{1}{2Re}[D(\mathbf{u}) : D(\mathbf{r})],$$
$$= \frac{1}{Re}(\tau \cdot D(\mathbf{u})\nu) \cdot (\tau \cdot D(\mathbf{r})\nu), \tag{83}$$
$$= (\tau \cdot D(\mathbf{r})\nu)(\tau \cdot \boldsymbol{\sigma}\nu),$$

because both $\mathbf{u}$ and $\mathbf{r}$ satisfy (80). Note that (81)–(83) are also true when $\mathbf{r}$ is replaced by $\mathbf{u}$.

Appendix C. Integration by parts formulae

Let $\mathbf{u}$, $\mathbf{v}$, $\mathbf{w}$ be smooth vector fields. Then, we have:

$$\int_\Omega [(\mathbf{u} \cdot \nabla)\mathbf{v}] \cdot \mathbf{w} = \int_\Omega [u_i w_j (\partial_i v_j)],$$
$$= -\int_\Omega [v_j w_j (\partial_i u_i) + u_i v_j (\partial_i w_j)] + \int_{\partial\Omega} [(u_i n_i)(v_j w_j)], \tag{84}$$
$$= -\int_\Omega [(\nabla \cdot \mathbf{u})(\mathbf{v} \cdot \mathbf{w}) + [(\mathbf{u} \cdot \nabla)\mathbf{w}] \cdot \mathbf{v}] + \int_{\partial\Omega} (\mathbf{u} \cdot \nu)(\mathbf{v} \cdot \mathbf{w}).$$

Eq. (84) gives the following special case. Suppose $\mathbf{u}$ is the velocity component of the solution of (8) and $\mathbf{u}'$ is the velocity component of the solution of the shape derivative PDE (19). Then, by (84), we have

$$\int_\Omega [(\mathbf{u} \cdot \nabla)\mathbf{u}] \cdot \mathbf{u}' = -\int_\Omega [(\nabla \cdot \mathbf{u})(\mathbf{u} \cdot \mathbf{u}') + [(\mathbf{u} \cdot \nabla)\mathbf{u}'] \cdot \mathbf{u}] + \int_{\partial\Omega} (\mathbf{u} \cdot \nu)(\mathbf{u} \cdot \mathbf{u}'),$$
$$= -\int_\Omega [(\mathbf{u} \cdot \nabla)\mathbf{u}'] \cdot \mathbf{u}, \tag{85}$$

where we used the divergence free constraint and the fact that $\mathbf{u} \cdot \nu = 0$. We also have

$$\int_\Omega [(\mathbf{u}' \cdot \nabla)\mathbf{u}] \cdot \mathbf{u} = -\int_\Omega [(\nabla \cdot \mathbf{u}')(\mathbf{u} \cdot \mathbf{u}) + [(\mathbf{u}' \cdot \nabla)\mathbf{u}] \cdot \mathbf{u}] + \int_{\partial\Omega} (\mathbf{u}' \cdot \nu)(\mathbf{u} \cdot \mathbf{u}),$$
$$= -\int_\Omega [(\mathbf{u}' \cdot \nabla)\mathbf{u}] \cdot \mathbf{u} + \int_{\partial\Omega} (\mathbf{u}' \cdot \nu), \tag{86}$$
$$= -\int_\Omega [(\mathbf{u}' \cdot \nabla)\mathbf{u}] \cdot \mathbf{u} + \int_\Omega \nabla \cdot \mathbf{u}' = -\int_\Omega [(\mathbf{u}' \cdot \nabla)\mathbf{u}] \cdot \mathbf{u},$$

which implies that

$$\int_\Omega [(\mathbf{u}' \cdot \nabla)\mathbf{u}] \cdot \mathbf{u} = 0. \tag{87}$$

References

[1] F. Kiil, The Function of the Ureter and Renal Pelvis: Pressurer Recordings and Radiographic Studies of the Normal and Diseased Upper Urinary Tract of Man, W.B. Saunders Co., Philadelphia, 1957.
[2] S. Boyarsky, Surgical physiology of the renal pelvis and ureter, Monogr. Surg. Sci. 16 (1964) 173–213.
[3] D.D. Chiras, Human Biology, Jones and Bartlett Publishers, 2005.
[4] L.M. Srivastava, V.P. Srivastava, Peristaltic transport of blood: Casson model – ii, Journal of Biomechanics 17 (11) (1984) 821–829.
[5] O. Eytan, A.J. Jaffa, J. Har-Toov, E. Dalach, D. Elad, Dynamics of the intrauterine fluid–wall interface, Annals of Biomedical Engineering 27 (1999) 372–379. doi:10.1114/1.181. URL: <http://www.ingentaconnect.com/content/klu/abme/1999/00000027/00000003/00483214>.

[6] L.J. Fauci, R. Dillon, Biofluidmechanics of reproduction, Annual Review of Fluid Mechanics 38(1) (2006) 371–394. Availabel from: http://arjournals.annualreviews.org/doi/pdf/10.1146/annurev.fluid.37.061903.175725, doi:10.1146/annurev.fluid.37.061903.175725. URL: <http://arjournals.annualreviews.org/doi/abs/10.1146/annurev.fluid.37.061903.175725>.

[7] W. Mitzner, Airway smooth muscle: The appendix of the lung, American Journal of Respiratory and Critical Care Medicine 169(7) (2004) 787–790. Availabel from: http://ajrccm.atsjournals.org/cgi/reprint/169/7/787.pdf, doi:10.1164/rccm.200312-1636PP. URL: <http://ajrccm.atsjournals.org>.

[8] E.C. Jesudason, N.P. Smith, M.G. Connell, D.G. Spiller, M.R.H. White, D.G. Fernig, P.D. Losty, Peristauncoupled from hypoplastic lung growth, American Journal of Physiology – Lung Cellular and Molecular Physiology 291 (4) (2006) L559–L565. Available from: http://ajplung.physiology.org/cgi/reprint/291/4/L559.pdf, doi:10.1152/ajplung.00498.2005. URL: <http://ajplung.physiology.org/cgi/content/abstract/291/4/L559>.

[9] C. Pozrikidis, A study of peristaltic flow, Journal of Fluid Mechanics 180 (1987) 515–527, doi:10.1017/S0022112087001939.

[10] A.M. Provost, W.H. Schwarz, A theoretical study of viscous effects in peristaltic pumping, Journal of Fluid Mechanics 279 (1994) 177–195, doi:10.1017/S0022112094003873.

[11] M. Mishra, A.R. Rao, Peristaltic transport of a newtonian fluid in an asymmetric channel, Zeitschrift Angewandte Mathematik und Physik 54 (2003) 532–550.

[12] D. Tang, S. Rankin, Numerical and asymptotic solutions for peristaltic motion of nonlinear viscous flows with elastic free boundaries, SIAM Journal on Scientific Computing 14(6) (1993) 1300–1319. doi:10.1137/0914077. URL: <http://link.aip.org/link/?SCE/14/1300/1>.

[13] M.H. Haroun, Effect of wall compliance on peristaltic transport of a newtonian fluid in an asymmetric channel, Mathematical Problems in Engineering 2006 (2006) 19, doi:10.1155/MPE/2006/61475.

[14] J.C. Misra, S.K. Pandey, Peristaltic transport in a tapered tube, Mathematical and Computer Modelling 22 (8) (1995) 137–151. doi:10.1016/0895-7177(95)00162-U. URL: <http://www.sciencedirect.com/science/article/B6V0V-3YCMGP7-/2/7305048438a8c9542f2870536d0c3e00>.

[15] G. Taylor, Analysis of the swimming of microscopic organisms, Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences 209 (1099) (1951) 447–461. URL: <http://www.jstor.org/stable/98828>.

[16] J.E. Gregory, G.A. Bentley, The peristaltic reflex in the isolated guinea-pig ileum during drug-induced spasm of the longitudinal muscle, Australian Journal of Experimental Biology and Medical Science 46 (1) (1968) 1–16.

[17] F.C.P. Yin, Y.C. Fung, Comparison of theory and experiment in peristaltic transport, Journal of Fluid Mechanics 47 (1) (1971) 93–112, doi:10.1017/S0022112071000958.

[18] S.L. Weinberg, E.C. Eckstein, A.H. Shapiro, An experimental study of peristaltic pumping, Journal of Fluid Mechanics 49 (3) (1971) 461–479, doi:10.1017/S0022112071002209.

[19] G. Böhme, R. Friedrich, Peristaltic flow of viscoelastic liquids, Journal of Fluid Mechanics 128 (1983) 109–122.

[20] E.F. Elshehawey, Z.M. Gharsseldien, Peristaltic transport of three-layered flow with variable viscosity, Applied Mathematics and Computation 153(2) (2004) 417–432. doi:10.1016/S0096-3003(03)00642-8. URL: <http://www.sciencedirect.com/science/article/B6TY8-497HG66-2/2/f525c45c0a4b367ade88703e087233c7>.

[21] M.H. Haroun, Effect of deborah number and phase difference on peristaltic transport of a third-order fluid in an asymmetric channel, Communications in Nonlinear Science and Numerical Simulation 12(8) (2007) 1464–1480. doi:10.1016/j.cnsns.2006.03.002. URL: http://www.sciencedirect.com/science/article/B6X3D-4JT3S44-1/2/41d2fb2ae1d163d400560c939dcd767a.

[22] T. Hayat, N. Alvi, N. Ali, Peristaltic mechanism of a maxwell fluid in an asymmetric channel, Nonlinear Analysis: Real World Applications 9(4) (2008) 1474–1490. doi:10.1016/j.nonrwa.2007.03.013. URL: <http://www.sciencedirect.com/science/article/B6W7S-4NDDM24-2/2/31fbefdc5aa9bd84ec8b2fbc15d45fe3>.

[23] T. Hayat, N. Ali, Effect of variable viscosity on the peristaltic transport of a newtonian fluid in an asymmetric channel, Applied Mathematical Modelling 32(5) (2008) 761–774. doi:10.1016/j.apm.2007.02.010. URL: <http://www.sciencedirect.com/science/article/B6TYC-4N4J%33M-2/2/b48e1e7c3023c84087e5b8030c9d5780>.

[24] T. Hayat, N. Ali, S. Asghar, Peristaltic motion of a burger's fluid in a planar channel, Applied Mathematics and Computation 186(1) (2007) 309–329. doi:10.1016/j.amc.2006.07.098. URL: <http://www.sciencedirect.com/science/article/B6TY8-4KVXHFY-7/2/8ce0bcb7b54372d19022471c951148ac>.

[25] J. Teran, L. Fauci, M. Shelley, Peristaltic pumping and irreversibility of a Stokesian viscoelastic fluid, Physics of Fluids 20(7) (2008).

[26] N. Liron, On peristaltic flow and its efficiency, Bulletin of Mathematical Biology 38(6) (1976) 573–596. doi:10.1016/S0092-8240(76)80001-8. URL: <http://www.sciencedirect.com/science/article/B6WC7-4GP25JF-1/2/24530852a4f8045e8b397d865575920a>.

[27] S. Takabatake, K. Ayukawa, A. Mori, Peristaltic pumping in circular cylindrical tubes: a numerical study of fluid transport and its efficiency, Journal of Fluid Mechanics 193 (-1) (1988) 267–283, doi:10.1017/S0022112088002149.

[28] P. Hariharan, V. Seshadri, R.K. Banerjee, Peristaltic transport of non-newtonian fluid in a diverging tube with different wave forms, Mathematical and Computer Modelling 48(7–8) (2008) 998–1017. doi:10.1016/j.mcm.2007.10.018. URL: <http://www.sciencedirect.com/science/article/B6V0V-4RBYFXD-8/2/fdba9d132a8a2cfbd87a8041e0f7168f>.

[29] O. Pironneau, Optimal Shape Design for Elliptic Systems, Springer Series in Computational Physics, Springer-Verlag, New York, NY, 1984.

[30] M.C. Delfour, J.-P. Zolésio, Shapes and Geometries: Analysis, Differential Calculus, and Optimization, Advances in Design and Control, vol. 4, SIAM, 2001.

[31] B. Mohammadi, O. Pironneau, Applied Shape Optimization for Fluids Numerical Mathematics and Scientific Computation, The Clarendon Press, Oxford University Press, New York, NY, 2001.

[32] B. Kawohl, O. Pironneau, L. Tartar, J.-P. Zolésio, Optimal Shape Design, Lecture Notes in Mathematics, vol. 1740, Springer, Tróia, Portugal, 1998.

[33] M. Moubachir, J.-P. Zolésio, Moving Shape Analysis and Control: Applications to Fluid Structure Interactions, Pure and Applied Mathematics, vol. 277, Chapman and Hall/CRC, 2006.

[34] J. Haslinger, R.A.E. Mäkinen, Introduction to Shape Optimization: Theory, Approximation, and Computation, Advances in Design and Control, vol. 7, SIAM, 2003.

[35] M. Burger, A framework for the construction of level set methods for shape optimization and reconstruction, Interfaces and Free Boundaries 5 (2002) 301–329.

[36] G. Dogan, P. Morin, R.H. Nochetto, M. Verani, Discrete gradient flows for shape optimization and applications, Computer Methods in Applied Mechanics and Engineering 196 (2007) 3898–3914.

[37] M. Hintermüller, W. Ring, A second order shape optimization approach for image segmentation, SIAM Journal on Applied Mathematics 64(2) (2004) 442–467. doi:10.1137/S0036139902403901. URL: <http://link.aip.org/link/?SMM/64/442/1>.

[38] G. Allaire, F. Jouve, A.-M. Toader, Structural optimization using sensitivity analysis and a level-set method, Journal of Computational Physics 194 (2004) 363–393.

[39] J.A. Sethian, A. Wiegmann, Structural boundary design via level set and immersed interface methods, Journal of Computational Physics 163 (2) (2000) 489–528. doi:http://dx.doi.org/10.1006/jcph.2000.6581.

[40] S.J. Osher, F. Santosa, Level set methods for optimization problems involving geometry and constraints. I: Frequencies of a two-density inhomogeneous drum, Journal of Computational Physics 171 (1) (2001) 272–288. doi:http://dx.doi.org/10.1006/jcph.2001.6789.

[41] T. Belytschko, S.P. Xiao, C. Parimi, Topology optimization with implicit functions and regularization, International Journal for Numerical Methods in Engineering 57 (2003) 1177–1196.

[42] A. Novruzi, J.R. Roche, Newton's method in shape optimisation: a three-dimensional case, BIT Numerical Mathematics 40 (1) (2000) 102–120.

[43] G. Rus, R. Gallego, Hypersingular shape sensitivity boundary integral equation for crack identification under harmonic elastodynamic excitation, Computer Methods in Applied Mechanics and Engineering 196 (25–28) (2007) 2596–2618, doi:10.1016/j.cma.2006.12.004.

[44] A. Henrot, G. Villemin, An optimum design problem in magnetostatics, Mathematical Modelling and Numerical Analysis 36 (2) (2002) 223–239, doi:10.1051/m2an:2002012.

[45] J. Sokolowski, J.-P. Zolésio, Introduction to Shape Optimization, Springer Series in Computational Mathematics, Springer-Verlag, 1992.
[46] G. Fourestey, M. Moubachir, Solving inverse problems involving the Navier–Stokes equations discretized by a Lagrange–Galerkin method, Computer Methods in Applied Mechanics and Engineering 194 (2005) 877–906.
[47] K. Ito, K. Kunisch, Lagrange Multiplier Approach to Variational Problems and Applications, Advances in Design and Control, SIAM, 2008.
[48] J. Nocedal, S.J. Wright, Numerical Optimization, 2nd ed., Springer Series in Operations Research, Springer, 2006.
[49] L.C. Evans, Partial Differential Equations, American Mathematical Society, Providence, Rhode Island, 1998.
[50] R.A. Adams, J.J.F. Fournier, Sobolev Spaces, Pure and Applied Mathematics Series, second ed., 140, Elsevier, 2003.
[51] F. Brezzi, M. Fortin, Mixed and Hybrid Finite Element Methods, Springer-Verlag, New York, NY, 1991.
[52] V. Girault, P.A. Raviart, Finite Element Methods for Navier–Stokes Equations: Theory and Algorithms, Springer-Verlag, Berlin, 1986.
[53] M. Giaquinta, S. Hildebrandt, Calculus of Variations I: The Lagrangian Formalism, Grundlehren der mathematischen Wissenschaften, vol. 310, Springer-Verlag, 1996.
[54] H.C. Elman, D.J. Silvester, A.J. Wathen, Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics Numerical Mathematics and Scientific Computation, Oxford Science Publications, 2005.
[55] D. Braess, Finite Elements: Theory, Fast Solvers, and Applications in Solid Mechanics, second ed., Cambridge University Press, 2001.
[56] S.C. Brenner, L.R. Scott, The Mathematical Theory of Finite Element Methods, second ed., Springer, New York, NY, 2002.
[57] M. Křížek, Higher order global accuracy of a weighted averaged gradient of the courant elements on irregular meshes, in: M. Křížek, P. Neittaanmäki, R. Stenberg (Eds.), Finite Element Methods: Fifty Years of the Courant Element, Lecture Notes in Pure and Applied Mathematics, vol. 164, Marcel Dekker, Inc., 1993, pp. 267–276.
[58] M. Křížek, P. Neittaanmäki, Finite Element Approximation of Variational Problems and Applications, Pitman Monographs and Surveys in Pure and Applied Mathematics, John Wiley and Sons, 1990.
[59] O.C. Zienkiewicz, J.Z. Zhu, The superconvergent patch recovery and a posteriori error estimators. Part 1: The recovery technique, International Journal of Numerical Methods in Engineering 33 (1992) 1331–1364.
[60] M. Lenoir, Optimal isoparametric finite elements and error estimates for domains involving curved boundaries, SIAM Journal of Numerical Analysis 23 (3) (1986) 562–580.
[61] H. Sohr, The Navier–Stokes Equations: An Elementary Functional Analytic Approach, Birkhäuser Advanced Texts, Birkhäuser Verlag, 2001.
[62] G.F. Carey, Some further properties of the superconvergent flux projection, Communications in Numerical Methods in Engineering 18 (2002) 241–250.
[63] A.I. Pehlivanov, R.D. Lazarov, G.F. Carey, S.S. Chow, Superconvergence analysis of approximate boundary-flux calculations, Numerische Mathematik 63 (1992) 483–501.
[64] J.T. Oden, H.J. Brauchli, On the calculation of consistent stress distributions in finite element approximations, International Journal for Numerical Methods in Engineering 3 (3) (1971) 317–325.
[65] E. Hinton, J.S. Campbell, Local and global smoothing of discontinuous finite element functions using a least squares method, International Journal for Numerical Methods in Engineering 8 (3) (1974) 461480.
[66] B. Cockburn, J. Gopalakrishnan, H. Wang, Locally conservative fluxes for the continuous Galerkin method, SIAM Journal of Numerical Analysis 45 (4) (2007) 1742–1776.
[67] V. Girault, P.A. Raviart, Finite Element Approximation of the Navier–Stokes Equations, Lecture Notes in Mathematics, Springer-Verlag, 1979.
[68] R. Verfürth, Finite element approximation of incompressible Navier–Stokes equations with slip boundary condition, Numerische Mathematik 50 (6) (1987) 697–721. doi:http://dx.doi.org/10.1007/BF01398380.
[69] R. Duran, R.H. Nochetto, J. Wang, Sharp maximum norm error estimates for finite element approximations of the stokes problem in 2d, Mathematics of Computation 51(184) (1988) 491–506. URL: <http://www.jstor.org/stable/2008760>.
[70] A. Bonito, R.H. Nochetto, M.S. Pauletti, Geometrically consistent mesh modification, in preparation.
[71] S.W. Walker, A hybrid variational-level set approach to handle topological changes, Master's Thesis, University of Maryland at College Park, May 2007.
[72] J.M. Escobar, G. Montero, R. Montenegro, Rodríguez, An algebraic method for smoothing surface triangulations on a local parametric space, International Journal for Numerical Methods in Engineering 66(4) (2006) 740–760. http://dx.doi.org/10.1002/nme.1584.
[73] P.M. Knupp, Algebraic mesh quality metrics, SIAM Journal of Scientific Computing 23 (2001) 193–218.
[74] J.R. Shewchuk, Triangle: engineering a 2D quality mesh generator and Delaunay triangulator, in: M.C. Lin, D. Manocha (Eds.), Applied Computational Geometry: Towards Geometric Engineering, Lecture Notes in Computer Science, vol. 1148, Springer-Verlag, 1996, pp. 203–222. from the First ACM Workshop on Applied Computational Geometry.
[75] C.-J. Heine, Isoparametric finite element approximation of curvature on hypersurfaces, Journal of Computational Mathematics, submitted for publication.
[76] R.H. Nochetto, S.W. Walker, A hybrid variational front tracking-level set mesh generator for problems exhibiting large deformations and topological changes, Journal of Computational Physics, submitted for publication.
[77] M.P. do Carmo, Differential Geometry of Curves and Surfaces, Prentice Hall, Upper Saddle River, New Jersey, 1976.