

Adjoint–Based Optimization of PDEs in Moving Domains

Bartosz Protas^a and Wenyuan Liao^a

^a*Department of Mathematics & Statistics, McMaster University, Hamilton, Ontario, Canada*

Abstract

In this investigation we address the problem of adjoint–based optimization of PDE systems in moving domains. As an example we consider the one–dimensional heat equation with prescribed boundary temperatures and heat fluxes. We discuss two methods of deriving an adjoint system necessary to obtain a gradient of a cost functional. In the first approach we derive the adjoint system after mapping the problem to a fixed domain, whereas in the second approach we derive the adjoint directly in the moving domain by employing methods of the noncylindrical calculus. We show that the operations of transforming the system from a variable to a fixed domain and deriving the adjoint do not commute and that, while the gradient information contained in both systems is the same, the second approach results in an adjoint problem with a simpler structure which is therefore easier to implement numerically. This approach is then used to solve a moving boundary optimization problem for our model system.

Key words: optimal control, adjoint equations, moving domains

PACS: 47.85.L-, 44.05.+e, 47.55.dr

1 Introduction

In this investigation we are interested in the computational solution of optimal control problems for a class of partial differential equations defined in variable domains. In such cases the shape of the domain is itself unknown and must be determined as a part of the solution of the direct (forward) problem. Applications of such problems are manifold and mostly include modeling and control of systems involving change of phase, such as solidification [1], and data assimilation for problems involving a free surface, such as the shallow water models in atmospheric and oceanic sciences [2]. Our own investigation is motivated by the problem of optimization of advanced welding processes in automotive manufacturing. Thus,

generally speaking, we are interested in PDE–constrained optimization problems

$$\begin{aligned} & \min_{u, \phi} j(u, \phi), \\ & \text{subject to } \mathcal{G}(u, \phi) = 0, \end{aligned} \tag{1}$$

where u and ϕ are, respectively, the state of the system and the control, $j(u, \phi)$ is the cost functional and $\mathcal{G}(u, \phi) = 0$ represents the PDE constraint. Noting that, subject to certain assumptions on \mathcal{G} , we have $u = u(\phi)$, one usually rewrites (1) in an equivalent unconstrained formulation using the reduced cost functional $\mathcal{J}(\phi) \triangleq j(u(\phi), \phi)$ (“ \triangleq ” means equality by definition) as

$$\min_{\phi} \mathcal{J}(\phi). \tag{2}$$

In our presentation hereafter we will skip the adjective “reduced”. Problem (2) can be solved using any of gradient–based approaches, such as the conjugate gradient method, or a variant of the quasi–Newton approach [3]. A central ingredient of any such technique is computation of the gradient $\nabla \mathcal{J}$ of the cost functional with respect to the control variable ϕ . For problems involving optimization of PDE systems such a gradient can be conveniently obtained in terms of the solution of a suitably defined adjoint system [4], which ensures that this gradient respects the PDE constraints. This adjoint–based approach has led to many successful applications of optimal control to problems governed by PDEs in meteorological data assimilation [5], flow control [6] and mixing [7], to mention just a few areas. However, when dealing with PDE problems defined in moving domains, one needs to address several issues which do not arise in fixed–domain problems. First of all, this domain variability must be properly accounted for in the derivation of the linear perturbation equation, the so–called “tangent linear model”, based on which the adjoint system is subsequently defined. Secondly, the domain variability must also be accounted for in the definition of the inner product defining the adjoint. Depending on how the variable domain is described mathematically, one can distinguish two general cases [8]. Denoting a time–dependent domain $\Omega(t)$ and its boundary $\partial\Omega(t)$, where $\Omega(t) \subset \mathbb{R}^d$ and d is the spatial dimension, an *implicitly* defined domain is characterized by the condition $\forall_{\mathbf{x} \in \partial\Omega(t)} F(\mathbf{x}) = f_0$ for a function $F : \mathbb{R}^d \rightarrow \mathbb{R}$ and a constant $f_0 \in \mathbb{R}$. Thus, the domain boundary is an isocontour of the function F . This formulation is natural, for instance, in problems involving a change of phase. On the other hand, *explicitly defined* domains are characterized by the condition $\forall_{\mathbf{x} \in \partial\Omega(t)} \dot{\mathbf{x}} = g(t, \mathbf{u})$, where \mathbf{u} is a vector of the dependent variables in the problem and $g(t, \mathbf{u})$ is a function describing the velocity of the domain boundary. This formulation is more natural in free–surface problems. Given an implicitly defined domain, it is usually possible to construct the corresponding explicit definition using the implicit function theorem. In this investigation we will focus on optimization of PDEs in implicitly defined variable domains; the companion problem concerning explicitly defined domains is left for the future.

The optimization problem becomes particularly interesting when the cost func-

tional $\mathcal{J}(\phi)$ is expressed in terms of the evolution of the domain boundaries. This may correspond, for example, to the situation when a part of the domain boundary should follow a prescribed trajectory. This problem is closely related to the shape optimization problem [9], where certain mathematical techniques we will use originated. As a matter of fact, the problem of optimizing the temporal evolution of a domain $\Omega(t) \subset \mathbb{R}^d$, where $t \in [0, T]$, can be equivalently regarded as the problem of optimizing the shape of a “tube” $\cup_{t \in [0, T]} \{t\} \times \Omega(t)$ embedded in an extended time–space domain $[0, T] \times \mathbb{R}^d$. However, the approach we will discuss below is preferable from the computational point of view, as it avoids the use of the boundary curvature defined with respect to the time–space coordinates.

Adjoint–based optimization of PDE systems in variable domains has received only limited attention in the literature. Control of a free–surface problem with the domain defined explicitly was considered in [2,10]. Control of a Stefan problem similar to the problem addressed here was investigated in [11], but using less general methods than the approach discussed in the present study. Adjoint–based methods were also used for optimal control of solidification fronts by Zabaras (see [12] for a review). These investigations, however, applied the equation for the domain evolution as a “soft” constraint only, i.e., it was not enforced exactly as an equality constraint, but its violations were instead penalized in the cost functional. Consequently, calculation of the cost functional gradient did not differ much from the case involving a fixed domain. Yet another approach was adopted by Hinze and Ziegenbalg in [13,14] who applied adjoint–based optimization to a two–phase Stefan problem with the interface parametrized as a graph of a function. Recently, there has also been some interest in the use of level set methods for shape and topology optimization [15]. In our investigation we are interested in a problem where the control has the form of the Neumann boundary condition on a part of the boundary. The mathematical theory for the particular case with the velocity of the boundary serving as the control variable was presented in a recent monograph [16]. In our present investigation we will employ elements of this theory.

Our goal in this study is to assess, from the computational viewpoint, two methods of calculating the cost functional gradient $\nabla \mathcal{J}$ in the presence of a PDE constraint in a variable domain. In the first approach we use a suitable mapping to transform the system from a moving to a fixed domain in which it is then optimized. We will show that the adjoint system obtained in this way has a rather awkward structure characterized by the presence of integral constraints. In the second approach we use the methods of the noncylindrical calculus [16] in order to differentiate the original PDE system with respect to evolution of the domain. We will observe that the adjoint system obtained in this way will have a simpler structure while still containing the same gradient information. Feasibility of this approach will be confirmed with computations concerning our model problem. For the sake of simplicity and in order to emphasize the generic, discretization–independent, character of problem addressed here, we will follow the “differentiate–then–discretize” approach [4] in which an expression for the gradient can be obtained using the infinite–dimensional

PDE formulation which has to be subsequently discretized.

The structure of the paper is as follows: in the next Section we introduce the one-dimensional (1D) model optimization problem that will serve as our example, in Section 3 we derive an expression for the cost functional gradient using a mapping to a fixed domain, an approach to gradient calculation in a variable domain employing the noncylindrical calculus is presented in Section 4, whereas in Section 5 we compare the two approaches, then computational examples are presented in Section 6, while conclusions and outlook are deferred to Section 7.

2 Statement of the Model Problem

Here we introduce our PDE-constrained optimization problem. The governing system has the form of a 1D heat equation defined on a variable domain $\Omega(t) \triangleq [a(t), b(t)] \subset \mathbb{R}$

$$\frac{\partial u}{\partial t} - v \frac{\partial^2 u}{\partial x^2} = 0 \quad \text{in } (0, T] \times [a(t), b(t)], \quad (3a)$$

$$\frac{\partial u}{\partial x} \Big|_{a(t)} = \phi(t), \quad \frac{\partial u}{\partial x} \Big|_{b(t)} = w \quad \text{in } (0, T], \quad (3b)$$

$$u \Big|_{a(t)} = u \Big|_{b(t)} = u_b \quad \text{in } (0, T], \quad (3c)$$

$$u \Big|_{t=0} = u_0 \quad \text{in } [a(0), b(0)], \quad (3d)$$

where $v \in \mathbb{R}^+$ is the diffusion coefficient (\mathbb{R}^+ does not include 0), $w, u_b \in \mathbb{R}$ represent the Neumann and Dirichlet boundary data and $u_0 : \Omega(0) \rightarrow \mathbb{R}$ is the initial condition. We will assume that $w \neq 0$ and $a(t) < b(t)$ for all times $t \geq 0$. The function $\phi : [0, T] \rightarrow \mathbb{R}^+$ in Neumann boundary condition (3b) is our *control*. We note that, comparing to the heat equation in a fixed domain, system (3) has more boundary conditions. The reason is that two additional relations are needed to determine the evolution of the domain given by $a(t)$ and $b(t)$. This is done by invoking the implicit function theorem in order to differentiate conditions (3c)

$$\frac{du}{dt} \Big|_{a(t)} = \frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} \frac{da}{dt} = \frac{du_b}{dt} \Big|_{a(t)} = 0 \implies V_a \triangleq \frac{da}{dt} = - \frac{\frac{\partial u}{\partial t}}{\frac{\partial u}{\partial x}} \Big|_{a(t)} \quad (4)$$

and likewise for $V_b \triangleq \frac{db}{dt}$. Relation (4) is well-defined owing to the assumptions made on the Neumann data in (3b). In the context of the heat transfer problems, conditions (3b) represent the heat flux across the boundary, whereas conditions (3c) mean that the temperature at the boundary $\partial\Omega(t)$ is constant and equal to u_b (e.g., the phase change temperature). Thus, system (3) may be regarded as a modified form of the one-phase Stefan problem [17] (the actual Stefan problem is characterized by somewhat different heat flux conditions). Finally, we emphasize that, even

though equation (3a) alone is formally linear, moving domain problem (3) is in fact *nonlinear*.

As regards the choice of the cost functional, we are interested in expressions depending on the *evolution* of the domain and will use

$$\mathcal{J}(\phi) = \frac{1}{2} \int_0^T [b(t; \phi) - \bar{b}(t)]^2 dt, \quad (5)$$

where $\bar{b}(t)$ represents the desired trajectory of the right boundary $b(t)$. Cost functional (5) can be supplemented with a Tikhonov–type penalty term, as required for regularization [18]. Thus, our optimization problem consists in finding the Neumann boundary condition ϕ at the left boundary, so that the trajectory of the opposite boundary follows a prescribed path. We conclude by saying that, given the diffusive nature of equation (3a), this is a strongly ill–posed inverse problem. A quantitative characterization of ill–posedness of this problem could be done by examining the singular value decomposition (SVD) of the Hessian of cost functional (5) [19]. This would, however, require the computation of the second–order differentials of (5) which is outside the scope of the present paper.

A local minimizer $\hat{\phi}$ of our problem is characterized by the first–order optimality conditions which imply the vanishing of the Gâteaux differential, defined as $\mathcal{J}'(\phi; \phi') \triangleq \lim_{\varepsilon \rightarrow 0} \frac{1}{\varepsilon} [\mathcal{J}(\phi + \varepsilon \phi') - \mathcal{J}(\phi)]$, of (5)

$$\forall_{\phi'} \quad \mathcal{J}'(\hat{\phi}; \phi') = 0, \quad (6)$$

where ϕ' is an arbitrary perturbation of the control. The local minimizer $\hat{\phi}$ can be found using an iterative descent algorithm

$$\begin{aligned} \phi^{(n+1)} &= \phi^{(n)} + \mathcal{B}(\nabla \mathcal{J}(\phi^{(n)})), \quad n = 1, \dots, \\ \phi^{(1)} &= \phi_0, \end{aligned} \quad (7)$$

where n is the iteration count and ϕ_0 the initial guess for the control variable. Representation (7) is generic and specific algorithms, such as, e.g., the steepest descent, conjugate gradients, or Newton’s method, can be obtained by making an appropriate choice of the operator \mathcal{B} [3]. The cost functional gradient $\nabla \mathcal{J}(\phi)$ is extracted from the Gâteaux differential $\mathcal{J}'(\phi; \phi')$ using the Riesz theorem

$$(\nabla \mathcal{J}, \phi')_{\mathcal{X}} = \mathcal{J}'(\phi; \phi'), \quad (8)$$

where $(\cdot, \cdot)_{\mathcal{X}}$ is the inner product in the Hilbert space \mathcal{X} .

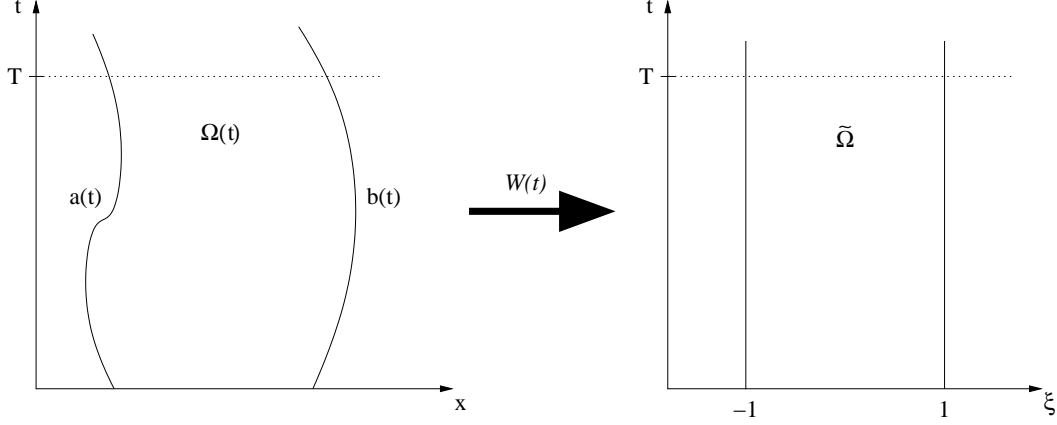


Fig. 1. Schematic transformation $W = W(t)$ of the moving domain $\Omega(t)$ to the fixed domain $\tilde{\Omega}$.

3 Gradient via Mapping to a Fixed Domain

In this Section we present an approach to computing the gradient ∇J of cost functional (5) based on a transformation of problem (3) to a fixed domain (Fig. 1). In the original system (3) the domain $\Omega(t)$ is characterized by the positions of its two endpoints $a(t)$ and $b(t)$. In order to simplify the resulting expressions, we will use instead the variables

$$L(t) \triangleq b(t) - a(t), \quad (9a)$$

$$x_0(t) \triangleq \frac{a(t) + b(t)}{2}, \quad (9b)$$

i.e., the domain width and the position of the domain centerpoint. Introducing the coordinate $\xi \in \tilde{\Omega} \triangleq [-1, 1]$, the transformation from $\Omega(t)$ to the fixed domain $\tilde{\Omega}$ is defined by the change of variables $x = x(t, \xi) = \frac{L(t)}{2}\xi + x_0(t)$ and $\tilde{u}(t, \xi) = u(t, x(t, \xi))$. Noting that the differential operators transform as $\frac{\partial u}{\partial x} = \frac{\partial \tilde{u}}{\partial \xi} \frac{2}{L}$, $\frac{\partial^2 u}{\partial x^2} = \frac{\partial^2 \tilde{u}}{\partial \xi^2} \frac{4}{L^2}$, and $\frac{\partial u}{\partial t} \Big|_x = \frac{\partial \tilde{u}}{\partial t} \Big|_{\xi(t,x)} + \frac{\partial \tilde{u}}{\partial \xi} \Big|_t \frac{d\xi}{dt}$, the transformation of system (3) to the fixed domain $\tilde{\Omega}$ yields

$$\frac{\partial \tilde{u}}{\partial t} - \frac{\partial \tilde{u}}{\partial \xi} \frac{2\dot{x}_0 + \xi \dot{L}}{L} - \frac{4\nu}{L^2} \frac{\partial^2 \tilde{u}}{\partial \xi^2} = 0 \quad \text{in } (0, T] \times [-1, 1], \quad (10a)$$

$$\frac{\partial \tilde{u}}{\partial \xi} \Big|_{-1} = \frac{L}{2} \phi, \quad \frac{\partial \tilde{u}}{\partial \xi} \Big|_1 = \frac{L}{2} w \quad \text{in } (0, T], \quad (10b)$$

$$\tilde{u} \Big|_{-1} = \tilde{u} \Big|_1 = u_b \quad \text{in } (0, T], \quad (10c)$$

$$\tilde{u} \Big|_{t=0} = \tilde{u}_0 \quad \text{in } [-1, 1], \quad (10d)$$

where the dot denotes differentiation with respect to the time. Cost functional (5) becomes

$$J(\phi) = \frac{1}{2} \int_0^T \left[x_0(t) + \frac{L(t)}{2} - \bar{b}(t) \right]^2 dt. \quad (11)$$

The Gâteaux differential of cost functional (11) obtained applying the definition formula is given by

$$J'(\phi; \phi') = \int_0^T \left(x_0 + \frac{L}{2} - \bar{b} \right) \left(x'_0 + \frac{L'}{2} \right) dt, \quad (12)$$

where $x'_0 = x'_0(\phi; \phi')$ and $L' = L'(\phi; \phi')$ are the perturbation variables which, together with the perturbation variable $\tilde{u}' = \tilde{u}'(\phi; \phi')$, are the solutions of the perturbation system

$$\begin{aligned} \frac{\partial \tilde{u}'}{\partial t} - \frac{\partial \tilde{u}'}{\partial \xi} \frac{2x_0 + \xi \dot{L}}{L} - \frac{\partial \tilde{u}'}{\partial \xi} \frac{(2x'_0 + \xi \dot{L}')L - (2x_0 + \xi \dot{L})L'}{L^2} \\ - \frac{4\nu}{L^2} \frac{\partial^2 \tilde{u}'}{\partial \xi^2} + L' \frac{8\nu}{L^3} \frac{\partial^2 \tilde{u}}{\partial \xi^2} = 0 \end{aligned} \quad \text{in } (0, T] \times [-1, 1], \quad (13a)$$

$$\left. \frac{\partial \tilde{u}'}{\partial \xi} \right|_{-1} - \frac{L'}{2} \phi = \frac{L}{2} \phi', \quad \left. \frac{\partial \tilde{u}}{\partial \xi} \right|_1 - \frac{L'}{2} w = 0 \quad \text{in } (0, T], \quad (13b)$$

$$\tilde{u}'|_{-1} = \tilde{u}'|_1 = 0 \quad \text{in } (0, T], \quad (13c)$$

$$\tilde{u}'|_{t=0} = 0 \quad \text{in } [-1, 1], \quad (13d)$$

$$x'_0|_{t=0} = 0, \quad L'|_{t=0} = 0. \quad (13e)$$

This perturbation system is obtained replacing ϕ in (10) with $\phi + \varepsilon\phi'$ and representing the solutions of (10) as

$$\begin{aligned} \tilde{u}(\phi + \varepsilon\phi') &= \tilde{u}(\phi) + \varepsilon\tilde{u}'(\phi; \phi') + O(\varepsilon^2), \\ L(\phi + \varepsilon\phi') &= L(\phi) + \varepsilon L'(\phi; \phi') + O(\varepsilon^2), \\ x_0(\phi + \varepsilon\phi') &= x_0(\phi) + \varepsilon x'_0(\phi; \phi') + O(\varepsilon^2). \end{aligned}$$

The resulting system is then linearized and simplified using (10). In expression (12) for the Gâteaux differential the perturbation ϕ' appears implicitly in the state perturbation variables $L' = L'(\phi; \phi')$ and $x'_0 = x'_0(\phi; \phi')$, hence at this stage we cannot use (8) to extract the cost functional gradient directly from (12). This will, however, become possible after (12) is transformed using suitably defined adjoint variables.

We begin by introducing the adjoint variables $\tilde{u}^* : [0, T] \times \tilde{\Omega} \rightarrow \mathbb{R}$, $\tilde{a}^* : [0, T] \rightarrow \mathbb{R}$ and $\tilde{b}^* : [0, T] \rightarrow \mathbb{R}$. Then we multiply (13a) by \tilde{u}^* and integrate over time and space, multiply (13b) by \tilde{a}^* and $-\tilde{b}^*$ and integrate over time, and add all the terms

together which gives

$$\begin{aligned}
I_1 = & \int_0^T \int_{-1}^1 \left[\frac{\partial \tilde{u}'}{\partial t} - \frac{\partial \tilde{u}'}{\partial \xi} \frac{2\dot{x}_0 + \xi \dot{L}}{L} - \frac{\partial \tilde{u}}{\partial \xi} \frac{(2\dot{x}'_0 + \xi \dot{L}')L - (2\dot{x}_0 + \xi \dot{L})L'}{L^2} \right. \\
& \left. - \frac{4\nu}{L^2} \frac{\partial^2 \tilde{u}'}{\partial \xi^2} + L' \frac{8\nu}{L^3} \frac{\partial^2 \tilde{u}}{\partial \xi^2} \right] \tilde{u}^* d\xi dt + \int_0^T \left(\frac{\partial \tilde{u}'}{\partial \xi} \Big|_{-1} - \frac{L'}{2} \phi - \frac{L}{2} \phi' \right) \tilde{a}^* dt \\
& - \int_0^T \left(\frac{\partial \tilde{u}}{\partial \xi} \Big|_1 - \frac{L'}{2} w \right) \tilde{b}^* dt = 0.
\end{aligned} \quad (14)$$

Performing integration by parts with respect to t and ξ , and using (13c) we obtain

$$\begin{aligned}
I_1 = & \int_0^T \int_{-1}^1 \tilde{u}' \left[-\frac{\partial \tilde{u}^*}{\partial t} + \frac{\dot{L}}{L} \tilde{u}^* + \frac{\partial \tilde{u}^*}{\partial \xi} \frac{2\dot{x}_0 + \xi \dot{L}}{L} - \frac{4\nu}{L^2} \frac{\partial^2 \tilde{u}^*}{\partial \xi^2} \right] d\xi dt \\
& + \int_{-1}^1 \tilde{u}' \tilde{u}^* \Big|_{t=0}^{t=T} d\xi - \int_0^T \frac{\partial \tilde{u}}{\partial \xi} \Big|_1 \left(\frac{4\nu}{L^2} \tilde{u}^* \Big|_1 + \tilde{b}^* \right) - \frac{\partial \tilde{u}}{\partial \xi} \Big|_{-1} \left(\frac{4\nu}{L^2} \tilde{u}^* \Big|_{-1} + \tilde{a}^* \right) dt \\
& + \int_0^T L' \left\{ \int_{-1}^1 \left[\frac{d}{dt} \left(\frac{\xi}{L} \frac{\partial \tilde{u}}{\partial \xi} \tilde{u}^* \right) + \frac{2\dot{x}_0 + \xi \dot{L}}{L^2} \frac{\partial \tilde{u}}{\partial \xi} \tilde{u}^* + \frac{8\nu}{L^3} \frac{\partial^2 \tilde{u}}{\partial \xi^2} \tilde{u}^* \right] d\xi - \frac{\phi}{2} \tilde{a}^* + \frac{w}{2} \tilde{b}^* \right\} dt \\
& + \int_0^T \dot{x}'_0 \left[\int_{-1}^1 \frac{d}{dt} \left(\frac{2}{L} \frac{\partial \tilde{u}}{\partial \xi} \tilde{u}^* \right) d\xi \right] dt + \int_0^T \frac{L}{2} \tilde{a}^* \phi' dt \\
& - \left[\frac{L'}{L} \int_{-1}^1 \frac{\xi}{L} \frac{\partial \tilde{u}}{\partial \xi} \tilde{u}^* d\xi \right]_{t=0}^{t=T} - \left[\frac{2\dot{x}'_0}{L} \int_{-1}^1 \frac{\partial \tilde{u}}{\partial \xi} \tilde{u}^* d\xi \right]_{t=0}^{t=T}.
\end{aligned} \quad (15)$$

Defining now the adjoint system as follows

$$-\frac{\partial \tilde{u}^*}{\partial t} + \frac{\dot{L}}{L} \tilde{u}^* + \frac{\partial \tilde{u}^*}{\partial \xi} \frac{2\dot{x}_0 + \xi \dot{L}}{L} - \frac{4\nu}{L^2} \frac{\partial^2 \tilde{u}^*}{\partial \xi^2} = 0 \quad \text{in } (0, T] \times [-1, 1], \quad (16a)$$

$$\tilde{u}^* \Big|_{-1} = -\frac{L^2}{4\nu} \tilde{a}^*, \quad \tilde{u}^* \Big|_1 = -\frac{L^2}{4\nu} \tilde{b}^* \quad \text{in } (0, T], \quad (16b)$$

$$\begin{aligned}
\int_{-1}^1 \left[\frac{d}{dt} \left(\frac{\xi}{L} \frac{\partial \tilde{u}}{\partial \xi} \tilde{u}^* \right) + \frac{2\dot{x}_0 + \xi \dot{L}}{L^2} \frac{\partial \tilde{u}}{\partial \xi} \tilde{u}^* + \frac{8\nu}{L^3} \frac{\partial^2 \tilde{u}}{\partial \xi^2} \tilde{u}^* \right] d\xi - \frac{\phi}{2} \tilde{a}^* + \frac{w}{2} \tilde{b}^* = \\
= \frac{1}{2} \left(x_0 + \frac{L}{2} - \bar{b} \right) \quad \text{in } (0, T], \quad (16c)
\end{aligned}$$

$$\int_{-1}^1 \frac{d}{dt} \left(\frac{2}{L} \frac{\partial \tilde{u}}{\partial \xi} \tilde{u}^* \right) d\xi = x_0 + \frac{L}{2} - \bar{b} \quad \text{in } (0, T], \quad (16d)$$

$$\tilde{u}^* \Big|_{t=T} = 0 \quad \text{in } [-1, 1], \quad (16e)$$

$$\tilde{a}^* \Big|_{t=T} = 0, \quad \tilde{b}^* \Big|_{t=T} = 0 \quad (16f)$$

and using (13d)–(13e) reduces (15) to

$$\int_0^T \frac{L}{2} \tilde{a}^* \phi' dt = \int_0^T \left(x_0 + \frac{L}{2} - \bar{b} \right) \left(\dot{x}'_0 + \frac{L'}{2} \right) dt = \mathcal{J}'(\phi; \phi'), \quad (17)$$

from which we can extract the L_2 cost functional gradient as

$$\nabla j = \frac{L}{2} \tilde{a}^* \quad \text{in } [0, T]. \quad (18)$$

We conclude this Section by commenting on the structure of adjoint system (16). We note that, in addition to satisfying evolution equation (16a), the adjoint variable \tilde{u}^* also has to satisfy two evolutionary integral constraints (16c) and (16d). These constraints can be accommodated by adjusting the adjoint variables \tilde{a}^* and \tilde{b}^* in boundary conditions (16b). However, the presence of these nonlocal integral constraints can severely complicate numerical treatment of problem (16). These difficulties will likely be aggravated further for problems formulated in higher spatial dimensions.

4 Gradient via Noncylindrical Calculus

In this Section we derive an expression for the gradient of cost functional (5) which will directly account for the variability of the domain in which the PDE constraint (3) is defined. Differentiation of solutions of PDEs defined in variable domains is made possible by the use of the “noncylindrical calculus” [16] whose main results are reviewed below, while the reader is referred to the original source for further details. Regarding $\Omega_0 \triangleq \Omega(0)$ as a reference domain, we introduce the flowmap $\mathcal{T} = \mathcal{T}(t)$ to parametrize the evolution of the domain as $\Omega(t) = \mathcal{T}(t)\Omega_0$. We can now define a noncylindrical set (a “tube”) as

$$Q \triangleq \bigcup_{t \in [0, T]} \{t\} \times \Omega(t) \quad (19)$$

which represent the evolution of the domain $\Omega(t)$ in the space–time “coordinates”. The appellation “noncylindrical” refers to the fact that in such coordinates Q forms a distorted tube, rather than a straight cylinder (Fig. 1, left vs. right schematic). In the present case the domain evolution clearly depends on the control ϕ , hence $Q = Q(\phi)$. Furthermore, as is evident from the structure of system (3), one may not in general perturb the solution u by perturbing the control ϕ *without* modifying the shape of the tube Q . Consequently, every dependent variable depends on the control variable ϕ through the domain evolution $Q(\phi)$, i.e., $u = u(Q(\phi))$, $a = a(Q(\phi))$ and $b = b(Q(\phi))$. This fact has important consequences for how differentials of the state variables are calculated with respect to the control ϕ : by the chain rule, differentiation is performed first with respect to the evolution Q of the domain which is in turn differentiated with respect to the control ϕ . This can be represented *schematically*

as

$$\frac{Du(Q(\phi))}{D\phi} \cdot \phi' = \underbrace{\frac{Du(Q(\phi))}{DQ}}_{\text{“outer” differential}} \cdot \underbrace{\left(\frac{DQ(\phi)}{D\phi} \cdot \phi'\right)}_{\text{“inner” differential}}, \quad (20)$$

where $\frac{D}{D\phi}$ corresponds to a Gâteaux or Fréchet differential with respect to ϕ and $\cdot X$ means that the differential is calculated in the direction X . Computation of the outer differential requires the use of the noncylindrical calculus [16] which provides the framework for differentiation of solutions of PDEs with respect to domain evolution. The central ansatz here is that the domain evolution Q can be parametrized using a velocity field V defined on a larger (“hold-all”) domain $D \subset \mathbb{R}$, such that $V|_{\partial D} = 0$ and $\bar{\Omega}(t) \subset D$, and differentiation is then carried out with respect to this velocity field. A differential with respect to domain evolution parametrized by V is calculated in the direction W which, as the “inner” differential in (20), is in turn expressed as a differential of V with respect to ϕ computed in the direction ϕ' . We emphasize that the differentials of the state variables are characterized by a PDE system obtained via linearization of system (3). Since our model problem is formulated in a 1D domain, many of the following results could be reduced to a more explicit form, however, we choose to state them in a more generic form admitting a straightforward generalization to problems in higher dimensions.

Domain evolution is parametrized using the velocity (speed) method [16] which associates a velocity field V to the flowmap \mathcal{T} , so that

$$\begin{cases} \frac{\partial \mathcal{T}(t, x)}{\partial t} = V(t, \mathcal{T}(t, x)), & t \in (0, T], \\ \mathcal{T}(0, x) = x, & \text{in } \bar{\Omega}(0). \end{cases} \quad (21)$$

Thus, the evolution of a domain $\Omega(t)$ can be regarded as driven by the velocity field V , i.e., $\Omega = \Omega(V)$. We can now introduce the *transverse map* $\tilde{\mathcal{T}}$

$$\tilde{\mathcal{T}} : \bar{\Omega}(t) \rightarrow \bar{\Omega}(t, \rho) \triangleq \bar{\Omega}(V + \rho W), \quad (22)$$

which represents modifications of a domain evolution $\Omega(t)$ due to the perturbation ρW applied to the velocity field V . It can be shown [16] that the flow associated with a transverse map (22) is given in terms of a *transverse velocity field* $\tilde{Z}(t, \rho)$ as

$$\begin{cases} \frac{d\tilde{\mathcal{T}}(t, \rho)}{d\rho} = \tilde{Z}(t, \rho), & \rho > 0 \\ \tilde{\mathcal{T}}(t, 0) = x, & \text{in } \bar{\Omega}(t). \end{cases} \quad (23)$$

Derivatives with respect to the pseudo-time ρ will be needed at $\rho = 0$ only, hence we can define $Z \triangleq \tilde{Z}|_{\rho=0}$. It is known [16] that evolution in time of the transverse

velocity field Z is governed by the following initial–value problem

$$\begin{cases} \frac{\partial Z}{\partial t} + [Z, V] = W & \text{in } (0, T] \times D, \\ Z|_{t=0} = 0 & \text{in } D, \end{cases} \quad (24)$$

where $[Z, V] \triangleq \frac{\partial Z}{\partial x}V - \frac{\partial V}{\partial x}Z$ denotes the Lie bracket of the pair (Z, V) and, for convenience, the transverse field Z is defined in the hold–all domain D . Transverse system (24) describes how the domain evolution is perturbed if the velocity V is perturbed in the direction W . The velocity fields V and W , as well as system (24), are defined at every point in D , however, from the governing system one can usually obtain explicitly the value of V at the boundary $\partial\Omega(t)$ only [cf. (4)]. Then the field V in the interior of the domain $\Omega(t)$ can be reconstructed as an *extension* of its boundary values (“traces”), i.e., as $V = \text{Ext}(V_a, V_b)$, so that $V_a = V|_a$ and $V_b = V|_b$. Likewise, the perturbation field W can be obtained as an extension of the boundary values of the differential of V , i.e., $W = \text{Ext}(W_a, W_b)$. The fact that the extension fields are not explicitly, or even uniquely, defined is not a problem, since in the subsequent developments we will only need a trace of system (24) on the domain boundary. Now our goal is to determine how functions defined on moving domains can be differentiated with respect to the domain evolution thus parametrized by V . This is addressed by the following two definitions:

Definition 1 *Given a function $f^\rho : \rho \rightarrow f(V + \rho W) \circ \tilde{T}(t, \rho)$, where \circ denotes a composition of maps, the noncylindrical (“tube”) material derivative $\dot{f}(V; W)$ of $f(V)$ at V and in the direction W is given by*

$$\dot{f}(V; W) \triangleq \frac{d}{d\rho} f^\rho \Big|_{\rho=0}. \quad (25)$$

□

Definition 2 *The noncylindrical (“tube”) shape derivative $f'(V; W)$ of $f(V)$ calculated at V and in the direction W is given by*

$$f'(V; W) \triangleq \dot{f}(V; W) - \frac{\partial f}{\partial x}(V)Z. \quad (26)$$

□

We thus have two types of the noncylindrical (tube) derivative: the material and the shape derivative (denoted with a dot and an apostrophe, respectively). One usually starts by computing the former using (25) and then obtains the latter using transformation (26). We are now in the position to calculate the complete Fréchet differential of a state variable in our problem, for instance, the quantity $u = u(V(\phi))$. In

accordance with scheme (20) and using formulas (25) and (26) we obtain

$$\begin{aligned}
u'(V(\phi); \phi') &= \dot{u}(V(\phi); \phi') - \frac{\partial u}{\partial x}(V(\phi))Z \\
&= \frac{d}{d\varepsilon} [u(V(\phi + \varepsilon\phi'))]_{\varepsilon=0} - \frac{\partial u}{\partial x}(V(\phi))Z \\
&= \left[\dot{u} \left(\frac{d}{d\varepsilon} V(\phi + \varepsilon\phi') \right) \right]_{\varepsilon=0} - \frac{\partial u}{\partial x}(V(\phi))Z \\
&= \dot{u}(V(\phi); V'(\phi; \phi')) - \frac{\partial u}{\partial x}(V(\phi))Z.
\end{aligned} \tag{27}$$

The functions $\dot{u}(V(\phi); W)$ and $u'(V(\phi); W)$ represent, respectively, the material and shape Fréchet differentials of $u(V(\phi))$ computed in the direction W which, by the chain rule, is in turn given by the Fréchet differential of the velocity field V with respect to the control ϕ , i.e., $W = V'(\phi; \phi')$. Each of the material and shape differentials appearing in (27) satisfies a PDE together with appropriate initial and boundary conditions. We remark that for our further developments it is more convenient to use the equations for the shape, rather than the material, differentials. The reason is that equations for shape differentials, as well as their adjoints, have structure more similar to the original governing equations than equation for material differentials do. This can be quite important from the implementation point of view, as it simplifies development of the code for the adjoint problem based on an existing code for the governing system. We can now proceed to calculate the differential of the cost functional and derive equations for perturbations of the dependent (state) variables.

We begin by calculating the Gâteaux differential of functional (5). This functional depends on the control ϕ through the shape of the tube Q only, so that in view of the above discussion it can be rewritten as

$$\mathcal{J}(\phi) = \bar{\mathcal{J}}(V(\phi)). \tag{28}$$

Now using the chain rule, the Gâteaux differential of $\mathcal{J}(\phi)$ can be expressed as

$$\mathcal{J}'(\phi; \phi') = \frac{d}{d\varepsilon} [\bar{\mathcal{J}}(V(\phi + \varepsilon\phi'))]_{\varepsilon=0} = \bar{\mathcal{J}}'(V(\phi); \frac{d}{d\varepsilon} V(\phi + \varepsilon\phi')|_{\varepsilon=0}) = \bar{\mathcal{J}}'(V(\phi); V'(\phi; \phi')), \tag{29}$$

where $\bar{\mathcal{J}}'(V; W)$ is the noncylindrical differential of $\bar{\mathcal{J}}(V)$ with respect to domain evolution evaluated in the direction $W = V'(\phi; \phi')$. The differential $\bar{\mathcal{J}}'(V; W)$ can be calculated as follows

$$\bar{\mathcal{J}}'(V, W) = \frac{d}{d\rho} \frac{1}{2} \int_0^T [b(V + \rho W) - \bar{b}]^2 dt \Big|_{\rho=0} = \int_0^T [b(V) - \bar{b}]Z dt, \tag{30}$$

where, in view of (23), we used $\frac{d}{d\rho} b(V + \rho W)|_{\rho=0} = \frac{d}{d\rho} \mathcal{T}(V + \rho W)|_{\rho=0} = Z$ and $Z = Z(V; W)$ satisfies system (24). The field W appearing on the right-hand side

(RHS) in (24) is obtained as $W = \text{Ext}(W_a, W_b)$, where, by (26),

$$W_a = V'(\phi; \phi')|_a = \dot{V}(\phi; \phi')|_a - \frac{\partial V}{\partial x} \Big|_a Z|_a, \quad (31a)$$

$$W_b = V'(\phi; \phi')|_b = \dot{V}(\phi; \phi')|_b - \frac{\partial V}{\partial x} \Big|_b Z|_b. \quad (31b)$$

Using (4) and carrying out the differentiation we obtain

$$\begin{aligned} \dot{V}(\phi; \phi')|_a &= \frac{d}{d\varepsilon} V_a(\phi + \varepsilon \phi')|_{\varepsilon=0} = - \frac{\frac{\partial u'}{\partial t} + V_a \frac{\partial u'}{\partial x} + Z \frac{\partial}{\partial x} \left(\frac{\partial u}{\partial t} + V_a \frac{\partial u}{\partial x} \right)}{\frac{\partial u}{\partial x}} \\ &= - \frac{\frac{du'}{dt} + Z \frac{\partial}{\partial x} \left(\frac{du}{dt} \right)}{\frac{\partial u}{\partial x}}, \end{aligned} \quad (32)$$

where all the partial derivatives are evaluated at $x = a(t)$ and $\dot{V}|_b$ can be calculated in an analogous manner.

Employing the methods of the noncylindrical calculus outlined above and described in detail in [16] we obtain the perturbation system characterizing the shape differential $u'(V(\phi); \phi')$ as follows

$$\frac{\partial u'}{\partial t} - v \frac{\partial^2 u'}{\partial x^2} = 0 \quad \text{in } (0, T] \times [a(t), b(t)], \quad (33a)$$

$$\frac{\partial u'}{\partial x} \Big|_{a(t)} = - \frac{\partial^2 u}{\partial x^2} \Big|_{a(t)} Z|_{a(t)} + \phi' \quad \text{in } (0, T], \quad (33b)$$

$$\frac{\partial u'}{\partial x} \Big|_{b(t)} = - \frac{\partial^2 u}{\partial x^2} \Big|_{b(t)} Z|_{b(t)}$$

$$\frac{dZ}{dt} = \frac{\partial Z}{\partial t} + \frac{\partial Z}{\partial x} V = W + \frac{\partial V}{\partial x} Z = \dot{V}(\phi; \phi') \quad \text{in } (0, T] \times D, \quad (33c)$$

$$u'|_{t=0} = 0 \quad \text{in } [a(0), b(0)], \quad (33d)$$

$$Z|_{t=0} = 0 \quad \text{in } D, \quad (33e)$$

where we simplified transverse equation (24), so that its RHS can now be expressed in terms of (32) as $\dot{V} = \text{Ext}(\dot{V}_a, \dot{V}_b)$. We remark that such simplification of (24) may not be possible in higher spatial dimensions. At this stage we still cannot use (8) to extract the cost functional gradient ∇J from (30), because the control perturbation ϕ' is not factored out, but is instead buried in perturbation system (33). This will, however, become possible after (30) is transformed using suitably defined adjoint variables.

We begin by introducing the adjoint variables $u^* : Q \rightarrow \mathbb{R}$ and $Z^* : [0, T] \times D \rightarrow \mathbb{R}$. We then multiply (33a) by u^* and integrate over the tube Q , multiply (33c) by Z^*

and integrate over $[0, T] \times D$, and add the two expressions together

$$I_2 = \int_0^T \int_{a(t)}^{b(t)} \left(\frac{\partial u'}{\partial t} - v \frac{\partial^2 u'}{\partial x^2} \right) u^* dx dt + \int_0^T \int_D \left(\frac{dZ}{dt} - \dot{V} \right) Z^* dx dt = 0. \quad (34)$$

As regards the adjoint transverse field Z^* , we will assume for it the following representation $Z^* = \gamma_{a,b}^*(N\zeta^*)$, where $\gamma_{a,b}^*$ is the adjoint of the trace operator $\gamma_{a,b}$ [16], $N = 1$ for $x = b$ and $N = -1$ for $x = a$. The trace operator $\gamma_{a,b}$ assigns to every function defined on the domain $\bar{\Omega}$ its boundary values at a and b , i.e., for $f : \bar{\Omega} \rightarrow \mathbb{R}$ we have $\gamma_{a,b}(f) = \{f|_a, f|_b\}$. The adjoint variable ζ^* is therefore supported at $x = a(t)$ and $x = b(t)$ only (denoted ζ_a^* and ζ_b^* , respectively). As a result, the second term in (34) simplifies as follows

$$\begin{aligned} \int_0^T \int_D \left(\frac{dZ}{dt} - \dot{V} \right) Z^* dx dt &= \int_0^T \int_D \left(\frac{dZ}{dt} - \dot{V} \right) \gamma_{a,b}^*(N\zeta^*) dx dt = \\ &= \int_0^T \int_D \gamma_{a,b} \left(\frac{dZ}{dt} - \dot{V} \right) N\zeta^* dx dt = \int_0^T \left[\left(\frac{dZ}{dt} - \dot{V} \right) \zeta^* \right]_{x=a}^{x=b} dt. \end{aligned} \quad (35)$$

Using (34), (35) and (32), performing integration by parts with respect to t and x , and then using (33b) we obtain

$$\begin{aligned} I_2 &= \int_0^T \int_{a(t)}^{b(t)} u' \left(\frac{\partial u^*}{\partial t} - v \frac{\partial^2 u^*}{\partial x^2} \right) dx dt + \left[\int_{a(t)}^{b(t)} u' u^* dx \right]_{t=0}^{t=T} \\ &\quad - \int_0^T \left\{ u' \left[u^* V - v \frac{\partial u^*}{\partial x} + \frac{d}{dt} \left(\frac{\zeta^*}{\frac{\partial u}{\partial x}} \right) \right] \right\} \Big|_{a(t)}^{b(t)} dt + \int_0^T v u^* |_{a(t)} \phi' dt \\ &\quad - \int_0^T \left[Z \left(\frac{d\zeta^*}{dt} - v \frac{\partial^2 u^*}{\partial x^2} - \frac{\partial}{\partial x} \left(\frac{du}{dx} \right) \zeta^* \right) \right]_{a(t)}^{b(t)} dt \\ &\quad + \left[\left(Z - \frac{u'}{\frac{\partial u}{\partial x}} \right) \zeta^* \right]_{a(t)}^{b(t)} \Big|_{t=0}^{t=T}. \end{aligned} \quad (36)$$

We remark that, since $\Omega(t)$ is time-dependent, when integrating by parts with respect to time the first term in (34) we had to use the Reynolds transport theorem

$$\int_{a(t)}^{b(t)} \frac{\partial g}{\partial t} dx = \frac{d}{dt} \int_{a(t)}^{b(t)} g dx - [gV]_{a(t)}^{b(t)},$$

where $g : Q \rightarrow \mathbb{R}$ is an arbitrary function. Defining now the adjoint system as fol-

lows

$$\frac{\partial u^*}{\partial t} - \mathbf{v} \frac{\partial^2 u^*}{\partial x^2} = 0 \quad \text{in } (0, T] \times [a(t), b(t)], \quad (37a)$$

$$u^*|_{a(t)} V_a - \mathbf{v} \frac{\partial u^*}{\partial x} \Big|_{a(t)} = -\frac{d}{dt} \left(\frac{\zeta_a^*}{\frac{\partial u}{\partial x} \Big|_{a(t)}} \right) \quad \text{in } (0, T], \quad (37b)$$

$$u^*|_{b(t)} V_b - \mathbf{v} \frac{\partial u^*}{\partial x} \Big|_{b(t)} = -\frac{d}{dt} \left(\frac{\zeta_b^*}{\frac{\partial u}{\partial x} \Big|_{b(t)}} \right)$$

$$\frac{d\zeta_a^*}{dt} - \zeta_a^* \frac{\frac{\partial}{\partial x} \left(\frac{du}{dt} \right)}{\frac{\partial u}{\partial x}} \Big|_{a(t)} = \mathbf{v} \frac{\partial^2 u}{\partial x^2} \Big|_{a(t)} u^*|_{a(t)} \quad \text{in } (0, T], \quad (37c)$$

$$\frac{d\zeta_b^*}{dt} - \zeta_b^* \frac{\frac{\partial}{\partial x} \left(\frac{du}{dt} \right)}{\frac{\partial u}{\partial x}} \Big|_{b(t)} = \mathbf{v} \frac{\partial^2 u}{\partial x^2} \Big|_{b(t)} u^*|_{b(t)} + [b(\phi) - \bar{b}] \quad \text{in } (0, T], \quad (37d)$$

$$u^*|_{t=T} = 0 \quad \text{in } [a(0), b(0)], \quad (37e)$$

$$\zeta_a^*|_{t=T} = \zeta_b^*|_{t=T} = 0 \quad (37f)$$

and using (33d)–(33e) reduces (36) to

$$\int_0^T \mathbf{v} u^*|_{a(t)} \phi' dt = \int_0^T [b(\phi) - \bar{b}] Z dt = \mathcal{J}'(\phi; \phi'), \quad (38)$$

from which we can extract the L_2 cost functional gradient as

$$\nabla \mathcal{J} = \mathbf{v} u^*|_{a(t)} \quad \text{in } [0, T]. \quad (39)$$

We notice that the left-hand sides of adjoint transverse equations (37c) and (37d) are, up to the factors, respectively, $-\frac{\partial u}{\partial x} \Big|_a$ and $-\frac{\partial u}{\partial x} \Big|_b$, equal to the expressions on the RHS in boundary conditions (37b). Therefore, boundary conditions (37b) can be combined with equations (3a), (37c)–(37d) to entirely eliminate the adjoint transverse variables ζ_a^* and ζ_b^* , so that adjoint system (37) can take a simpler form

$$\frac{\partial u^*}{\partial t} - \mathbf{v} \frac{\partial^2 u^*}{\partial x^2} = 0 \quad \text{in } (0, T] \times [a(t), b(t)], \quad (40a)$$

$$\mathbf{v} \frac{\partial u^*}{\partial x} \Big|_{a(t)} = 0 \quad \text{in } (0, T] \quad (40b)$$

$$\mathbf{v} \frac{\partial u^*}{\partial x} \Big|_{b(t)} = \frac{[b(\phi) - \bar{b}]}{\frac{\partial u}{\partial x} \Big|_{b(t)}} \quad \text{in } (0, T], \quad (40c)$$

$$u^*|_{t=T} = 0 \quad \text{in } [a(0), b(0)]. \quad (40d)$$

Finally, we remark that the same form of adjoint system (40) and expression for the gradient (39) is obtained treating our problem as a shape optimization problem in an extended time–space domain $[0, T] \times \mathbb{R}$ and employing standard methods of shape differentiation [20].

5 Discussion

In this Section we compare the two approaches to gradient computation introduced in Sections 3 and 4. First we note that the two adjoint systems (16) and (40) provide equivalent gradient information (18) and (39) which is due to the fact that they both correspond to the Gâteaux differential of the same cost functional (5) with respect to the same control variable ϕ . However, the structure of these adjoint systems is in fact quite different. System (16) is defined on a fixed, time–independent, domain $\tilde{\Omega}$, and the evolution of the adjoint variables is constrained by two nonlocal conditions (16c) and (16d). On the other hand, system (40) is defined on a time–dependent, albeit predetermined, domain $\Omega(t)$ and does not involve any nonlocal constraints. From the point of view of numerical solution, it is more convenient to work with a problem defined in a fixed domain such as $\tilde{\Omega}$. Therefore, using transformation (9) and defining $\check{u}^*(t, \xi) \triangleq u^*(t, x(t, \xi))$ we can now transform system (40) to the fixed domain $\tilde{\Omega}$ which yields

$$\frac{\partial \check{u}^*}{\partial t} + \frac{\partial \check{u}^*}{\partial \xi} \frac{2\dot{x}_0 + \xi \dot{L}}{L} - \frac{4\nu}{L^2} \frac{\partial^2 \check{u}^*}{\partial \xi^2} = 0 \quad \text{in } (0, T] \times [-1, 1], \quad (41a)$$

$$\frac{2\nu}{L} \frac{\partial \check{u}^*}{\partial \xi} \Big|_{-1} = 0 \quad \text{in } (0, T] \quad (41b)$$

$$\frac{2\nu}{L} \frac{\partial \check{u}^*}{\partial \xi} \Big|_1 = \frac{[b(\phi) - \bar{b}]}{\frac{2}{L} \frac{\partial \bar{u}}{\partial \xi} \Big|_1} \quad \text{in } (0, T] \quad (41c)$$

$$\check{u}^* \Big|_{t=T} = 0 \quad \text{in } [-1, 1]. \quad (41d)$$

We emphasize that the structure of this system is different, and arguably simpler, than the structure of system (16) obtained by first transforming problem (3) to a fixed domain and then deriving the adjoint. This observation illustrates the fact that the operations of transforming a problem to a fixed domain and deriving the adjoint do not, in general, commute. Since both adjoint problems contain the same gradient information, in our computations in Section 6 we use the one with the simpler structure, namely, (41). The solutions of the direct, perturbation and adjoint problems in the variable domain are computed by first transforming (3), (33) and (40) to the fixed domain, solving the resulting fixed–domain systems, and then transforming the solutions back to the variable domain. We stress that this is done for convenience only and is independent of how the adjoint system is actually derived.

As regards the computational cost, in the 1D case it will be marginally smaller for

system (41) than for system (16) which is due to the simpler mathematical structure and fewer dependent variables in system (41). Because of the same reason, implementation of system (41) is going to be more straightforward than in the case of system (16). It appears that the computational advantages of the formulation based on the adjoint system derived in the variable domain are going to be even more significant in the case of the spatial dimension larger than one, since in such situations the computational cost of the approach based on a mapping of the variable domain to a fixed domain will be increased by the cost of determining this mapping (transformation W in Fig. 1). Indeed, while in the two-dimensional (2D) case such a transformation could possibly be found using conformal mapping techniques, no such tools seem to be readily available in the three-dimensional (3D) case. In contrast, such limitations do not exist in the approach in which the adjoint system is derived in the variable domain, and in fact in [28] we apply this technique to a complicated problem in 3D.

6 Numerical Examples

In this Section we provide numerical examples illustrating the concepts introduced above. For the sake of clarity, we will use a rather simple approach to the numerical solution of problems (10) and (41) combining a spectral collocation discretization in space with an explicit Euler discretization in time [21]. Given the smoothness of solutions and the high accuracy of the spectral discretization, accurate results can be obtained already with very few grid points in space, and the time step restriction due to the explicit time discretization can be easily accommodated. To fix attention, we consider the following values of the parameters: $\nu = 10^{-3}$, $T = 300$, $w = -2$ with the initial condition $\tilde{u}_0(\xi) = -\xi^2 + 1$. Unless stated otherwise, the results presented below were obtained using $N = 10$ grid points in space and the time step $\Delta t = 10^{-2}$. Analysis of the consistency of the gradient calculation indicates that already at this spatial resolution the truncation errors become comparable to round-off errors, so that using finer spatial resolution is unnecessary. An evolution of the solution u of problem (3) in space and time corresponding to the control $\phi(t) = \phi_0(t) \triangleq 0.25 + 1.75 \left(1 - \frac{t}{T}\right)$ is shown in Fig. 2a. The control ϕ_0 will also serve as the initial guess for iterations (7). In our model optimization problem the target trajectory of the right boundary $\bar{b}(t)$ is given by a tabulated function (see Fig. 4d below). Cost functional (5) is augmented with a Tikhonov-type regularization term, i.e., $\mathcal{J}_1(\phi) = \mathcal{J}(\phi) + \mathcal{J}_0(\phi)$, where $\mathcal{J}_0(\phi) \triangleq l \int_0^T (\phi - \phi_0)^2 dt$ and we choose $l = 10^{-7}$. The solutions u' and u^* of the perturbation and adjoint problems (33) and (40) corresponding to the initial guess ϕ_0 are shown in Figs. 2b and 2c. In Fig. 2d we show the direct and adjoint transverse fields Z and ζ^* at the boundaries $a(t)$ and $b(t)$. While the adjoint transverse field ζ^* does not appear in final form (40) of the adjoint system, we show it here for completeness. To fix attention, in perturbation problem (33) the control perturbation was taken in the form $\phi'(t) = A \sin(2\pi \frac{t}{T})$,

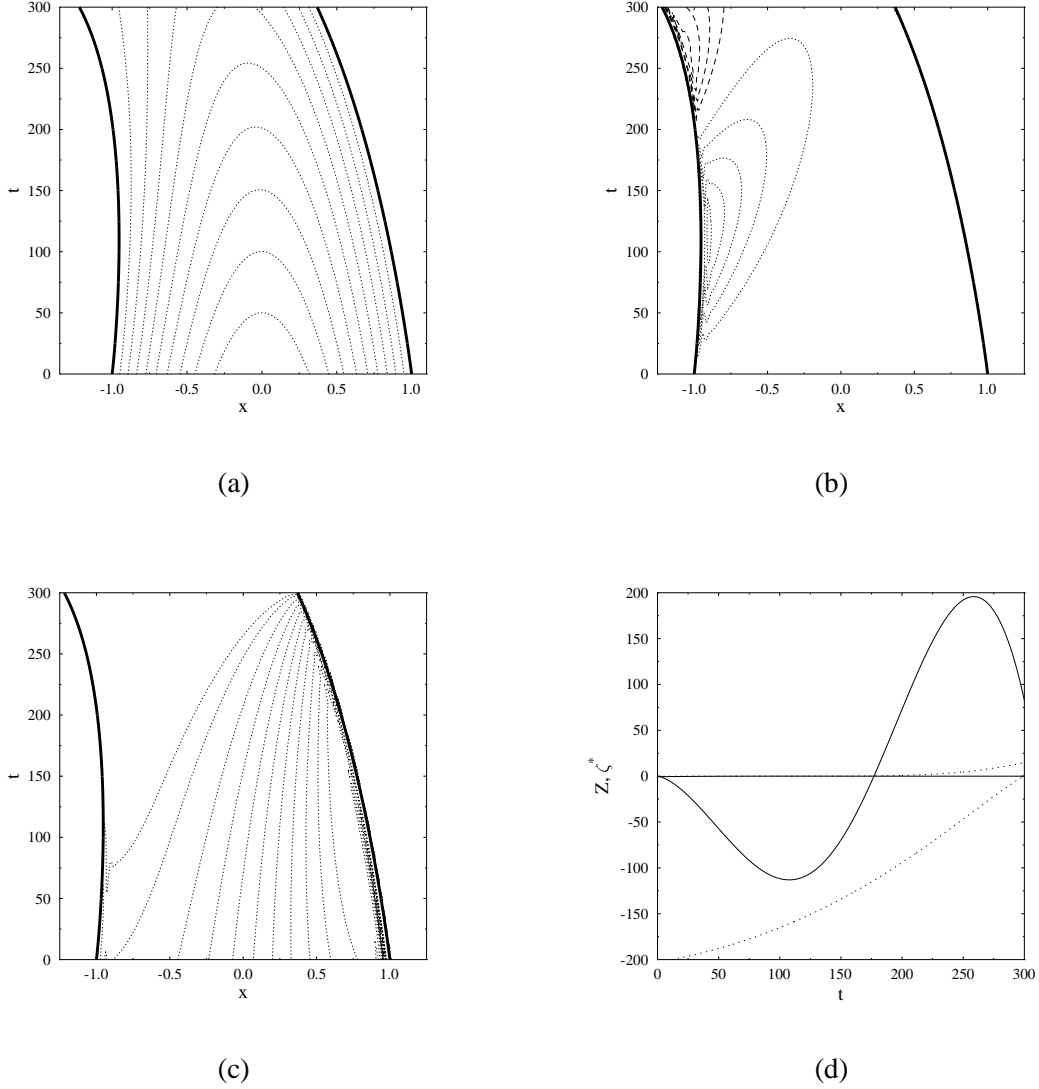


Fig. 2. Space–time evolution of (a) solution u of model system (3), (b) solution u' of perturbation system (33) and (c) solution u^* of adjoint problem (37); dotted lines represent positive values and dashed lines represent negative values of the different solutions, Fig. (d) shows the direct transverse fields Z (vanishing at $t = 0$) and the adjoint transverse fields ζ^* (vanishing at $t = T$) which are parts of the solutions of problems (33) and (37); the transverse fields correspond to (solid line) $x = a(t)$ and (dotted line) $x = b(t)$.

where $A > 0$ is an arbitrary constant. We note that solutions of both the perturbation and adjoint problems evolve in the domain $\Omega(t)$ obtained as a part of the solution of the original problem (3). As expected, the solution u' of perturbation problem (33) is concentrated close to the left boundary $x = a(t)$, whereas the solution u^* of adjoint problem (37) is concentrated close to the right boundary $x = b(t)$.

Next we proceed to analyze the consistency of the gradient ∇J obtained using system (41). A standard test [22] consists in computing the Gâteaux differential (i.e.,

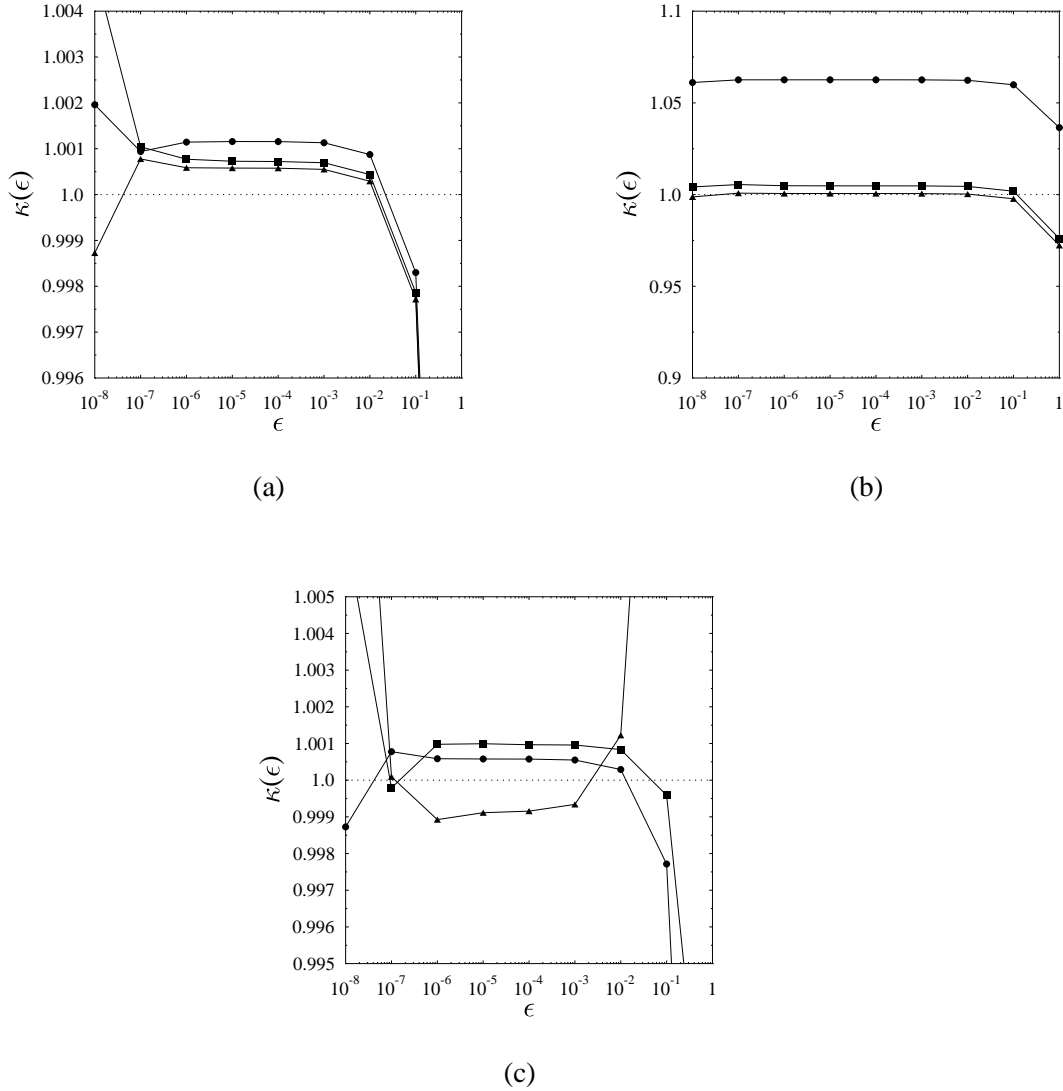


Fig. 3. Measure of the error $\kappa(\epsilon)$ in determination of the cost functional gradient using adjoint system (41) for (a) different time steps [(circles) $\Delta t = 5.0 \cdot 10^{-2}$, (squares) $\Delta t = 2.0 \cdot 10^{-2}$, and (triangles) $\Delta t = 1.0 \cdot 10^{-2}$] with a fixed perturbation $m = 1$ and spatial resolution $N = 10$, (b) different spatial resolutions [(circles) $N = 6$, (squares) $N = 8$, and (triangles) $N = 10$] with a fixed perturbation $m = 1$ and time step $\Delta t = 1.0 \cdot 10^{-2}$, and (c) different perturbations [(circles) $m = 1$, (squares) $m = 3$, and (triangles) $m = 15$] with a fixed spatial resolution $N = 10$ and time step $\Delta t = 1.0 \cdot 10^{-2}$.

the directional derivative) of the cost functional $\mathcal{J}(\phi)$ in some arbitrary direction ϕ' using relation (38) and comparing it to the result obtained with a forward finite-difference formula. Thus, deviation of the quantity $\kappa(\epsilon) \triangleq \frac{\mathcal{J}(\phi + \epsilon\phi') - \mathcal{J}(\phi)}{\epsilon(\nabla\mathcal{J}, \phi')}$ from unity is a measure of the error. In order to focus on the gradient computed using the adjoint system, in the cost functional here we do not include the Tikhonov regularization term. Moreover, in order to exclude the interpolation errors, in these tests

we also set $\bar{b} = 1.0$. In Fig. 3 we show the behavior of $\kappa(\varepsilon)$ corresponding to $\phi = \phi_0$ and different time steps Δt (Fig. 3a), different numbers of grid points in the spatial discretization N (Fig. 3b) and different perturbations $\phi'(t) = \sin(m2\pi\frac{t}{T})$ (Fig. 3c), with the remaining parameters held fixed. As expected, all three Figures reveal an increase of the error for large values of ε , which is due to the truncations errors, and also for very small values of ε , which is due to the subtractive cancellation (round-off) errors. In Fig. 3a and 3b we observe that, as the temporal and spatial discretizations are refined, $\kappa(\varepsilon)$ approaches the unity for intermediate values of ε . In Fig. 3c we remark that, while the error increases for larger frequencies of the perturbation $\phi'(t)$, in all cases shown it remains small and close to 0.1%. We emphasize that, since we are using the “differentiate–then–discretize” rather than “discretize–then–differentiate” approach, the gradient should not be expected to be accurate up to the machine precision [4]. Finally, we remark that the range of ε where the values of $\kappa(\varepsilon)$ are close to the unity spans between four and five orders of magnitude.

Finally, we move on to discuss the results of optimization. As regards descent algorithm (7), we choose the Polak–Ribiere version of the conjugate gradient method [3]. This is a popular approach to solution of unconstrained optimization problems and we refer the reader to [23] for an analysis of its convergence properties and to [24] for a discussion of some new developments. We observe that repeated solution of problems (3) and (40) followed in the optimization process by an update of the boundary conditions in (3) may exhibit the tendency to amplify the high–frequency noise. Motivated by the work of Jameson [25,26], and also by our earlier investigations [27], we get around this difficulty by requiring that the gradients belong to the Sobolev space $H^1(0, T)$ with the norm defined as $\|z\|_{H^1} \triangleq \int_0^T z^2 + \gamma^2 (\frac{\partial z}{\partial t})^2 dt$ which ensures smoothness of the descent directions (γ is an adjustable parameter which, unless otherwise stated, is fixed as $\gamma = 1$). The Sobolev gradient $\nabla^{H^1} j$ is obtained using (8) with $\mathcal{X} = H^1(0, T)$, (38) and the definition of the H^1 inner product associated with the norm $\|\cdot\|_{H^1}$ as

$$\int_0^T \nabla^{H^1} j \phi' + \gamma^2 \frac{\partial \nabla^{H^1} j}{\partial t} \frac{\partial \phi'}{\partial t} dt = \int_0^T \nu u^*|_{a(t)} \phi' dt \quad (42)$$

from which, after performing integration by parts and assuming homogeneous Dirichlet boundary conditions for $\nabla^{H^1} j$ at $t = 0$ and $t = T$, we obtain the smoothed gradient as a solution of the following Helmholtz problem

$$\begin{aligned} \left(1 - \gamma^2 \frac{\partial^2}{\partial t^2}\right) \nabla^{H^1} j &= \nu u^*|_{a(t)}, \\ \nabla^{H^1} j|_{t=0} &= \nabla^{H^1} j|_{t=T} = 0. \end{aligned} \quad (43)$$

In Fig. 4a we compare the original L_2 gradient ∇j and the smoothed H^1 gradient $\nabla^{H^1} j$ obtained at the fifth iteration. In order to make the smoothing effect of (43) more visible, the Sobolev gradient shown in Fig. 4a was computed using $\gamma = 100$.

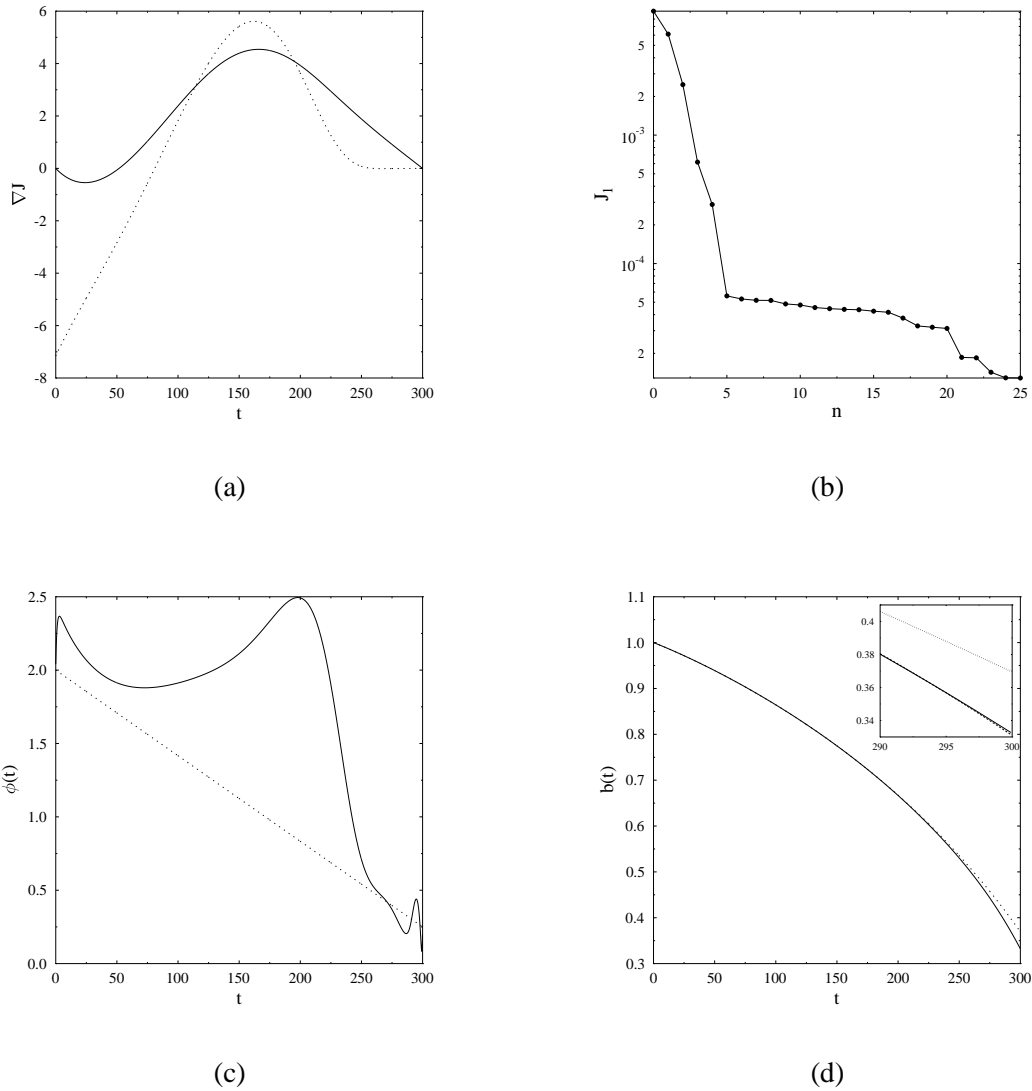


Fig. 4. (a) Comparison of (dotted line) the L_2 gradient ∇J and (solid line) the Sobolev gradient $\nabla^{H^1} J$ (vertical scale in the figure is arbitrary), (b) decrease of the cost functional J with the number n of iterations, (c) (dotted line) the initial guess for the control ϕ_0 and (solid line) the optimal control $\hat{\phi}$ as a function of time t and (d) (dotted line) the initial trajectory $b(\phi_0)$, (dashed line) target trajectory \bar{b} and (solid line) optimal trajectory $b(\hat{\phi})$ of the right boundary as a function of time t ; the inset magnifies the differences between the three trajectories close to the end of the time window.

In Fig. 4b we present the cost functional J as a function of the iteration count n . We note a steady decrease of J by almost three orders of magnitude during 25 iterations. In Fig. 4c we show the initial guess $\phi_0(t)$ for the control together with the optimal control $\hat{\phi}(t)$ determined by the algorithm after 25 iterations, whereas in Fig. 4d we show the corresponding trajectories of the right boundary, i.e., $b(\phi_0)$ and $b(\hat{\phi})$. In Fig. 4d we also include the target trajectory \bar{b} (see inset). Comparison

of rather modest modifications of b (Fig. 4d) with quite significant modifications of the corresponding control ϕ (Fig. 4c) confirms a strongly ill-posed character of the model problem investigated here,

7 Conclusions

In this investigation we addressed the problem of adjoint-based optimization of PDE systems defined in variable domains. We showed that transformation of the PDE system from a variable to a fixed domain and derivation of the adjoint do not commute. This means that, depending on the order of these operations, different forms of the adjoint system may be obtained, even though they will contain the same gradient information. In this sense, this problem is similar to the problem studied in [27] where we showed that, in general, deriving adjoints does not commute with applying differential and integral operators. In such situations the choice of the approach should be informed by the computational properties of the resulting adjoint system. Problems such as (3), in which the actual PDE is formally linear and the nonlinearity arises through variability of the domain, are said to possess a *geometric* nonlinearity. Thus, transformation of such problem to a fixed domain can be regarded as replacing this geometric nonlinearity with an algebraic one. For the model problem considered here, adjoint system (40) derived in the variable domain using methods of the noncylindrical calculus clearly has structure more amenable to numerical implementation than adjoint system (16) derived in a fixed domain. Furthermore, the approach in which the adjoint is derived in a fixed domain could be more problematic due to difficulties in finding a transformation to the fixed domain which can be significant, especially in a higher spatial dimension. Our numerical results illustrate how gradients obtained with the adjoint system derived on a variable domain can be used to solve a simple optimization problem. We reiterate that, while the different PDE systems were solved numerically using a transformation to the fixed domain, this was done for convenience only and was independent of how the adjoint system was actually derived. To the best of our knowledge, the present investigation is the first actual computational study involving an adjoint system derived using methods of the noncylindrical calculus. Work is underway employing such techniques in the study of optimization of more complicated problems occurring in industrial welding. An adjoint system characterizing sensitivity of such a problem described by equations representing the conservation of mass, momentum and energy with a change of phase is derived in [28] using the approach presented in Section 4. Computational results concerning this investigation will be presented in [29].

Another issue relevant to computational optimization of PDE systems in variable domains in the “discretize-then-optimize” framework is the potential for the use of automatic differentiation (AD). The AD approach uses transformations of the original source code for the governing system to generate a code implementing the

perturbation (tangent linear) problem and the associated adjoint problem [30,31]. It is well known that for the success of automatic differentiation, the original code must meet certain criteria as regards organization, clarity of structure, interfaces, etc. While we are not aware of any systematic studies concerning this issue, we expect that application of AD to free boundary problems might not be straightforward, since problems with geometric nonlinearities are often implemented using operators and instructions which are difficult to differentiate (e.g., `if` and `goto`). On the other hand, approaches involving mapping to a fixed domain in which such geometric nonlinearities are replaced with algebraic ones can be more amenable to AD.

A natural generalization of the results presented here would be the development of an approach to compute, for the given cost functional, the Hessian containing the second-derivative information. In addition to quantifying the degree of ill-posedness of an optimization problem, this could also make it possible to apply Newton's method instead of gradient approach (7) to find the minimizer $\hat{\phi}$. The Hessian of the cost functional can be determined by solving a family of second-order adjoint problems [32]. Derivation and implementation of such a second-order adjoint system for the present problem is an interesting question and we intend to address it in our future research.

Acknowledgments

The first author wishes to thank Dr. Oleg Volkov for many interesting comments concerning this work and, in particular, for deriving the simplified form (40) of the adjoint system. The authors also gratefully acknowledge generous support from the Natural Sciences and Engineering Research Council of Canada (the Collaborative Research and Development Program), the General Motors of Canada, Ltd., and the McMaster Centre for Advanced Manufacturing and Materials.

References

- [1] G. Z. Yang and N. Zabaras, "The adjoint method for an inverse design problem in the directional solidification of binary alloys", *Journal of Computational Physics*, **140**, 432–452, 1998.
- [2] I. Yu. Gejadze and G. J. M. Copeland, "Open Boundary Control Problem for Navier-Stokes Equations Including a Free Surface: Adjoint Sensitivity Analysis", *Computers & Mathematics with Applications* **52**, 1243–1268, (2006).
- [3] J. Nocedal and S. Wright, "Numerical Optimization", Springer, (2002).
- [4] M. D. Gunzburger, "Perspectives in flow control and optimization", SIAM, Philadelphia, (2003).
- [5] E. Kalnay, "Atmospheric Modeling, Data Assimilation and Predictability", Cambridge University Press, (2003).
- [6] T. R. Bewley, "Flow control: new challenges for a new Renaissance", *Progress in Aerospace Sciences* **37**, 21-58, (2001).
- [7] G. Mathew, I. Mezić, S. Grivopoulos, U. Vaidya and L. Petzold, "Optimal Control of Mixing in Stokes Fluid Flows", *Journal of Fluid Mechanics* (in press), (2007).
- [8] S. Osher and R. Fedkiw, "Level Set Methods and Dynamic Implicit Surfaces", Springer (2002).
- [9] J. Sokolowski and J.-P. Zolésio, "Introduction to shape optimization: shape sensitivity analysis", Springer, (1992).
- [10] I. Yu. Gejadze, G. J. M. Copeland and I. M. Navon, "Open Boundary Control Problem for Navier–Stokes Equations including a free surface: Data Assimilation", *Computers & Mathematics with Applications* **52**, 1269–1288, (2006).
- [11] A. El Badia and F. Moutazaim, "A one–phase inverse Stefan problem", *Inverse Problems*, **15**, 1507–1522, (1999).
- [12] N. Zabaras, "Adjoint methods for inverse free convection problems with application to solidification processes", *Computational Methods for Optimal Design and Control (Birkhauser Series in Progress in Systems and Control Theory)*, (Eds. J. Borggaard, E. Cliff, S. Schreck and J. Burns), Birkhauser, 391–426, (1998).
- [13] M. Hinze and S. Ziegenbalg, "Optimal control of the free boundary in a two–phase Stefan problem", *Journal of Computational Physics* **223**, 657–684 (2007).
- [14] M. Hinze and S. Ziegenbalg, "Optimal control of the free boundary in a two-phase Stefan problem with flow driven by convection", *Z. Angew. Math. Mech.* **87**, 430–448 (2007).
- [15] S. Y. Wang, K. M. Lim, B. C. Khoo and M. Y. Wang, "An extended level set method for shape and topology optimization", *Journal of Computational Physics*, **221**, 395–421, (2007).

- [16] M. Moubachir and J.-P. Zolésio, “Moving Shape Analysis and Control — Applications to Fluid Structure Interactions”, Chapman & Hall, (2006).
- [17] N. L. Gol’dman, “Inverse Stefan Problem”, Kluwer, (1997).
- [18] C. R. Vogel, “Computational Methods for Inverse Problems”, SIAM, (2002).
- [19] P. C. Hansen, “Rank-Deficient and Discrete Ill-Posed Problems. Numerical Aspects of Linear Inversion”, SIAM, (1998).
- [20] O. Volkov, private communication.
- [21] L. N. Trefethen, “Spectral Methods in MATLAB”, SIAM, (2000).
- [22] C. Homescu, I. M. Navon and Z. Li, “Suppression of vortex shedding for flow around a circular cylinder using optimal control”, *Int. J. Numer. Meth. Fluids* **38**, 43–69, (2002).
- [23] J. C. Gilbert and J. Nocedal, “Global Convergence Properties of Conjugate Gradient Methods for Optimization”, *SIAM Journal on Optimization* **2**, 21–42, (1992).
- [24] W. W. Hager and H. Zhang, “Algorithm 851: CG DESCENT, a Conjugate Gradient Method with Guaranteed Descent”, *ACM Transactions on Mathematical Software* **32**, 113–137, (2006).
- [25] A. Jameson, “Aerodynamic Design via Control Theory”, *Journal of Scientific Computing* **3**, 233–260, (1989).
- [26] S. Kim, K. Hosseini, K. Leoviriyakit and A. Jameson, “Enhancement of Adjoint Design Methods via Optimization of Adjoint Parameters”, 43rd AIAA Aerospace Sciences Meeting and Exhibit, January 10-13, 2005, Reno, NV, *AIAA Paper* 05-0448, (2005).
- [27] B. Protas, T. Bewley and G. Hagen, “A comprehensive framework for the regularization of adjoint analysis in multiscale PDE systems”, *Journal of Computational Physics*, **195** (1), 49-89, 2004.
- [28] B. Protas, W. Liao and D. W. Glander, “A Framework for Gradient-Based Optimization of Welding Processes — Mathematical Characterization of Sensitivity to Control Parameters”, (preprint), (2006).
- [29] W. Liao, B. Protas and D. W. Glander, “Computational study of adjoint-based optimization of MIG welding”, (in preparation), (2007).
- [30] R. Giering and T. Kaminski, “Recipes for Adjoint Code Construction”, *ACM Transactions On Mathematical Software* **24**, 437–474, (1998).
- [31] A. Griewank, “Evaluating Derivatives: Principles and Techniques of Automatic Differentiation”, SIAM, (2000).
- [32] F. X. Le Dimet, I. M. Navon and D. N. Daescu, “Second Order Information in Data Assimilation”, *Monthly Weather Review* **130**, 629–648, (2002).