Adjoint-Based Optimal Control of Time-Dependent Free Boundary Problems

JAN MARBURGER

ABSTRACT: In this paper we show a simplified optimisation approach for free boundary problems in arbitrary space dimensions. This approach is mainly based on an extended operator splitting which allows a decoupling of the domain deformation and solving the remaining partial differential equation. First we give a short introduction to free boundary problems and the problems occurring in optimisation. Then we introduce the extended operator splitting and apply it to a general minimisation subject to a time-dependent scalar-valued partial differential equation. This yields a time-discretised optimisation problem which allows us a quite simple application of adjoint-based optimisation methods. Finally, we verify this approach numerically by the optimisation of a flow problem (Navier-Stokes equation) and the final shape of a Stefan-type problem.

Keywords: Optimal control, constraint optimisation, shape optimisation, adjoint approach, evolution equation, free boundary problem, free surface flow, Stefan-type problem

1. Introduction

Optimisation of free surface problems [5] [10] often occurs in industrial applications. Some examples are the stabilisation of a liquid surface for sloshing [11] or optimising the shape of solidification processes [10]. Free surface problems are still challenging from an analytical as well as a numerical point of view. Here, the domain is an unknown of the equation system which depends on the states, e.g., a water surface is driven by the flow velocity. Since these problem are already hard to handle, the optimisation of such processes is very complex. Especially adjoint-based approaches [18] are very difficult to apply due to the state-dependent domain. Here, several assumptions and methods were derived to handle this kind of problem. For special cases it is possible to describe the free boundary by a graph [15] or introducing a level-set or phase field function [2]. Another approach is the pullback of the time- and state-dependent problem. For special cases it is possible to describe the free boundary by a graph [15] or introducing a level-set or phase field function [2]. Another approach is the pullback of the time- and state-dependent problem to a reference domain and perform all calculations in there. From an optimisation point of view, all of these methods have the disadvantage of very complex derivatives describing the variation of the domain. Note that often these derivatives are, in contrast to stationary problems, hard to interpret for time-dependent problems.

In this paper we show a simplified optimisation approach for free boundary problems in arbitrary space dimensions which bases on an extended time-discretisation of the problem. We consider the problem: Minimise $J(y, \Omega, u)$ subject to the free surface problem of finding $(y, \Omega)$ such that

\begin{align}
\partial_t y(t) + A(t)y(t) &= f(u(t)) \quad \text{in } \Omega(t) \\
B(t)y(t) &= g(u(t)) \quad \text{on } \Gamma(t) \\
y(0) &= y_0 \quad \text{in } \Omega(0) \\
C(y, \Omega) &= 0
\end{align}

holds for all $t \in (0, T]$. Here, $\Omega(t) \subset \mathbb{R}^d$ denotes the time- and state-dependent domain, $y$ a scalar-valued function, $A$ an arbitrary differential operator and $C(y, \Omega)$ a constraint function defining the free boundary. Moreover, $B$ and $g : \mathbb{R} \rightarrow \mathbb{R}$ denote appropriate boundary conditions, $u$ a control function and $f : \mathbb{R} \rightarrow \mathbb{R}$ a right hand side term depending on $u$, e.g., a localisation function. The difficulty is the dependency of the domain $\Omega(t)$ on the solution $y$ of the partial differential equation. To obtain the domain, we solve, roughly speaking, a minimisation problem for each time step $t \in (0, T]$ in order to fulfill the constraint for the free boundary. Hence, the minimisation of the cost functional $J$ would be subjected to the minimisation of the constraint function $C$ in order to solve the state equation).

In order to avoid the minimisation problem for solving the state equation, we perform a complete pullback of equation 1 to a reference domain $\hat{\Omega}$. For this, we introduce a flux function $F : \mathbb{R} \rightarrow \mathbb{R}^d$ depending on the state $y$. This function is given by a characteristic velocity (e.g., flow) or a smooth continuation of the boundary motion, cf. figure 1. Together with the transformation $\Phi$ given by

$$\partial_t \Phi = F(y \circ \Phi) \quad \text{in } \hat{\Omega} \times (0, T)$$

with $\Phi(X, 0) = X$ in $\hat{\Omega}$.

we resolve the constraint function for the domain by $\Omega(t) = \Phi(\hat{\Omega}, t)$ and hence $C(y, \Phi(\hat{\Omega})) = 0$ holds, see figure 2. Therefore, we reformulate the original minimisation problem of the cost function $J$ to: Minimise
\[ \dot{J}(\hat{y}, \Phi, u) \] subject to
\[ \begin{aligned}
\partial_t \hat{y} + \hat{A}(\Phi) \hat{y} &= f(u \circ \Phi) \quad \text{in } \hat{\Omega} \times (0, T) \\
\hat{B}(\Phi) \hat{y} &= g(u \circ \Phi) \quad \text{in } \hat{\Gamma} \times (0, T) \\
\hat{y}(0) &= \hat{y}_0 \quad \text{in } \hat{\Omega}
\end{aligned} \]
(2)
\[ \partial_t \Phi = F(\hat{y}) \quad \text{in } \hat{\Omega} \times (0, T) \]
\[ \Phi(0) = \text{Id}_{\hat{\Omega}} \quad \text{in } \hat{\Omega} \]

with \( \hat{y} := y \circ \Phi \) and \( \text{Id}_\Omega(x) := x \) as the identity map in \( \Omega \). Note that also the operators \( A \) and \( B \) change to \( \hat{A}(\Phi) \) and \( \hat{B}(\Phi) \), respectively. Now the domain is fixed but the dependency of the differential operator on \( \Phi \) yields very complex derivatives.

In the following we derive a simplified optimisation approach for free boundary problems using an extended time discretisation which is based on an operator splitting. This approach is a blend of the original (1) and transformed formulation (2) of the optimisation problem.

## 2. Formulation of the State Equation

In this section we reformulate the state equation by a time-discretisation. Particularly this is done by applying an operator splitting scheme, which allows a simplified optimisation approach later on. First, we show a first order splitting scheme for a simple convection-diffusion equation in \( \mathbb{R}^d \) and transfer the results to the state equation (1). Finally, we introduce an appropriate time-discrete Hilbert space which simplifies the application of adjoint-based optimisation.

### 2.1. Basics of Operator Splitting

In the following we use a Yanenko splitting [7] which is illustrated by a simple convection-diffusion equation in \( \mathbb{R}^d \)
\[ \partial_t y + v \cdot \nabla y - \Delta y = f \quad \text{in } \mathbb{R}^d \times (0, T) \]
with \( y(0) = y_0 \) in \( \mathbb{R}^d \)
where \( y : \mathbb{R}^d \rightarrow \mathbb{R} \) and \( v : \mathbb{R}^d \rightarrow \mathbb{R}^d \) denotes a smooth vector field. Note that the velocity field can also depend on \( y \). This problem is divided into two subproblems. For the first time interval \([0, \tau]\) the Yanenko splitting reads
\[ \partial_t y^* + v \cdot \nabla y^* = 0 \quad \text{in } \mathbb{R}^d \times (0, \tau) \]
\[ y^*(0) = y_0 \quad \text{in } \mathbb{R}^d \]
\[ \partial_t y - \Delta y = f \quad \text{in } \mathbb{R}^d \times (0, \tau) \]
\[ y(0) = y^*(\tau) \quad \text{in } \mathbb{R}^d \]
The resulting subproblems can be solved by different methods. In particular, we consider the convection part from a Lagrangian viewpoint, cf. [9], which yields
\[ \partial_t \hat{y}^* = 0 \quad \text{in } \mathbb{R}^d \times (0, \tau) \]
\[ \hat{y}^*(0) = y_0 \quad \text{in } \mathbb{R}^d \]
\[ \partial_t \Phi = v \circ \Phi \quad \text{in } \mathbb{R}^d \times (0, \tau) \]
\[ \Phi(X, 0) = X \quad \text{in } \mathbb{R}^d \]
for \( \hat{y}^* := y^* \circ \Phi \). Thus the solution \( y^* \) at time \( \tau \) is given by
\[ y^*(\Phi(X, \tau), \tau) = \hat{y}^*(X, \tau) = y_0(X) \]
and hence \( y^*(\tau) = y_0(\Phi^{-1}(\tau)) \) where \( \Phi^{-1} \) denotes the inverse of \( \Phi \). The diffusion part is written as

\[
\partial_t y - \Delta y = f \quad \text{in} \, \Phi(\mathbb{R}^d, \tau) \times (0, \tau]
\]

\[
y(0) = y_0(\Phi^{-1}(\tau)) \quad \text{in} \, \Phi(\mathbb{R}^d, \tau)
\]

Note that \( \Phi(\mathbb{R}^d, \tau) = \mathbb{R}^d \) but the metric, needed for example for spatial operators, is induced by the transformation \( \Phi \). Applying this scheme iteratively, we solve the convection-diffusion equation for all time intervals. Finally, a time-discretisation is performed. Since the Yanenko splitting is of order \( O(\tau) \), a first order scheme is sufficient.

\[
\Phi^{(n)} = \text{Id}_{\Omega^{(n)}} + \tau F^{(n)}
\]

\[
\frac{1}{\tau} \left( y^{(n+1)} - y^{(n)} \circ (\Phi^{(n)})^{-1} \right) - \Delta y^{(n+1)} = f(t^{(n+1)})
\]

with \( \Omega^{(n+1)} = \Phi^{(n)}(\Omega^{(n)}) \). This approach, i.e. solving the convection part by a Lagrangian viewpoint, is the main principle of particle methods, cf. [13], which will be used for the numerical results later on.

**Remark 1.** Due to the definition, the continuous transformation \( \Phi \), given by (2), is a diffeomorphism, cf. [13]. For small time steps also the time-discrete transformations \( \Phi^{(n)} \) are diffeomorphisms. This implies that \( \det(\nabla \Phi^{(n)}) > 0 \) is always satisfied.

### 2.2. Time Discretisation of the State Equation

Now we apply the above splitting scheme to the state equation (1). For this, we extend (1) by a convection part given by a smooth continuation of the boundary motion. Particularly, we obtain

\[
\partial_t y(t) + F(y(t)) \cdot \nabla y(t) - F(y(t)) \cdot \nabla y(t) + A(t) y(t) = f(t)
\]

\[
B(t) y(t) = g(t)
\]

\[
y(0) = y_0
\]

for all \( t \in (0, T) \). Here the flux function \( F : \mathbb{R} \rightarrow \mathbb{R}^d \) defines the deformation of the entire domain, cf. equation (2) and figure 1. Applying the splitting scheme of the previous section to this equation we obtain

\[
\Phi^{(n)} = \text{Id}_{\Omega^{(n)}} + \tau F^{(n)}
\]

\[
\frac{1}{\tau} \left( y^{(n+1)} - y^{(n)} \circ (\Phi^{(n)})^{-1} \right) - \Delta y^{(n+1)} = f(t^{(n+1)})
\]

with \( \Omega^{(n+1)} := \Omega(t^{(n+1)}) = \Phi^{(n)}(\Omega^{(n)}) \). Note that the transformation, which generates the new domain, is solved explicitly. For this reason it is sufficient to treat the artificial convection term \( F(y) \cdot \nabla y \) also explicitly in the above equation. Moreover, the transformation \( \Phi^{(n)} \) depends on \( y^{(n)} \) only. Hence we treat \( \Phi^{(n)} \) as function of \( y^{(n)} \) given by

\[
\Phi^{(n)}[y] := \text{Id}_{\Omega^{(n)}} + \tau F(y^{(n)})
\]

in the following. To consider weak formulations later on, we introduce the function space

\[
V := \prod_{n=0}^N V^{(n)}
\]

where \( V^{(n)} \) denotes the spatial space at time \( t^{(n)} \), for instance \( V^{(n)} := H^1(\Omega^{(n)}) \). These spaces are built recursively, that is,

\[
V^{(0)} \xrightarrow{\Phi^{(0)}} V^{(1)} \xrightarrow{\Phi^{(1)}} V^{(2)} \xrightarrow{\Phi^{(2)}} \ldots
\]
and depend on the data, e.g. right hand side or boundary conditions. If $V^{(n)}$ are Hilbert spaces, we define the corresponding inner product by

$$\langle x, y \rangle_{V^{(n)}} := \sum_{n=0}^{N_t} \tau \langle x^{(n)}, y^{(n)} \rangle_{V^{(n)}}$$

for $x, y \in V$. Note that for small time steps every transformation is a diffeomorphism which maps from $\Omega^{(n)}$ to $\Omega^{(n+1)}$ as state in the above remark. Therefore, the domain does not become singular and hence if $V^{(n)}$ is a Hilbert space then also $V^{(n+1)}$ is a Hilbert space. Consequently, (6) is an inner product and hence also $V$ is a Hilbert space.

3. Optimal Control Problem

Let the state space $V$ and the space of Lagrange multipliers $Z$ be Hilbert spaces which are defined analogous to (5). Let the space of control functions $U$ be a Hilbert space. Moreover, we introduce the space

$$H := \prod_{n=0}^{N_t} H^{(n)} \quad \text{with} \quad H^{(n)} := L^2(\Omega^{(n)}).$$

The index $\Gamma$ denotes the corresponding function space on the boundary, e.g. $H^{(\Gamma)} = L^2(\partial \Omega^{(n)})$.

3.1. Weak Formulation. We generalise equation (3) in a weak sense using the function spaces defined above as

$$\frac{1}{\tau} (y^{(n+1)} - y^{(n)} \circ (\Phi^{(n)}(y)))^{-1} - (F(y^{(n)}) : \nabla y^{(n)}) \circ (\Phi^{(n)}(y))^{-1} + A^{(n+1)} y^{(n+1)} = B^{(n+1)} u$$

in $(V^{(n+1)})^*$. The operator $A^{(n)} : V^{(n)} \rightarrow (V^{(n)})^*$ denotes the weak counterpart of $A$ and $B$ used in equation (3) and $B^{(n)} \in L(U; (V^{(n)})^*)$ of $f$ and $g$. For example, we obtain for the Laplacian $A^{(n)} y^{(n)} := \Delta y^{(n)}$ with Neumann boundary, $f(u) = 0$ and $g(u) = u$

$$\langle A^{(n)} y^{(n)}, \lambda \rangle_{(V^{(n)})^*} := - \int_{\Omega^{(n)}} \nabla y^{(n)} \cdot \nabla \lambda \, dx$$

for $\lambda \in V^{(n)}$.

3.2. Minimisation Problem. We consider the minimisation problem

$$(P) \quad \min_{(y,u) \in V \times U} J(y,u) \quad \text{subject to} \quad e(y,u) = 0$$

where $J : V \times U \rightarrow \mathbb{R}_+^1$ is a cost functional and $e : V \times U \rightarrow Z^*$ is determined by (7), i.e.

$$\langle e(y,u), \lambda \rangle_{Z^*} := \sum_{n=0}^{N_t-1} \left[ \langle y^{(n)}, \lambda^{(n)} \rangle_{H^{(n)}} + \tau \langle A^{(n)} y^{(n)}, \lambda^{(n)} \rangle_{(V^{(n)})^*} - \tau \langle B^{(n)} u, \lambda^{(n)} \rangle_{(V^{(n)})^*} \right]$$

$$- \sum_{n=0}^{N_t-1} \left[ \langle y^{(n)} \circ (\Phi^{(n)}(y)), \lambda^{(n+1)} \rangle_{H^{(n+1)}} + \tau \langle F(y^{(n)}) : \nabla y^{(n)} \circ (\Phi^{(n)}(y)), \lambda^{(n+1)} \rangle_{H^{(n+1)}} \right]$$

Note that the constraint function $e$ does not depend on the domains $\Omega^{(n)}$ explicitly. The condition $\Omega^{(n+1)} = \Phi^{(n)}(\Omega^{(n)})$, needed to establish the space $V^{(n+1)}$, is given implicitly by the definition of $\Phi^{(n)}$.

Remark 2. On the one hand, the missing condition $\Omega^{(n+1)} = \Phi^{(n)}(\Omega^{(n)})$ in the above constraint function yields an underdetermined system. On the other hand, the solution of the approach introduced in section 2 satisfies the constraint function. This solution is used to establish the optimality condition and hence we assume the domains to be known. In standard approaches, fixing the domain would yield no information about the variation of the domain or the quantity describing it, e.g. a transformation. Here, this information is not fully lost as we still obtain information about domain variations by the push forward terms in the constraint and cost function.
The above minimisation problem is the time-discrete counterpart to the minimisation of $J(y, \Omega, u)$ stated in the very beginning. Here, all spatial dependencies, e.g. integration domains or evaluation positions, in the cost functional are replaced by the corresponding transformation $\Phi^{(n)}[y]$. More details about the reformulation of the cost functional can be found in section 4 or [13].

To find a minimum of (P) we use the Lagrangian multiplier theorem, that is, we determine the critical points of the Lagrange functional $L : V \times U \times Z \to \mathbb{R}$ defined by

$$L(y, u, \lambda) := J(y, u) + \langle e(y, u), \lambda \rangle_{Z^*, Z}$$

Then the Karush-Kuhn-Tucker system reads

$$\langle \partial_y L(y, u, \lambda), \psi \rangle_{V^*, V} = 0, \quad \langle \partial_u L(y, u, \lambda), \psi_u \rangle_{U^*, U} = 0 \quad \text{and} \quad \langle \partial_\lambda L(y, u, \lambda), \psi_\lambda \rangle_{Z^*, Z} = 0$$

for all $\psi \in V$, $\psi_u \in U$ and $\psi_\lambda \in Z$. For more detail we refer to [18]. The partial derivatives in $V$ and $Z$ are interpreted as

$$\langle \partial_y L(y, u, \lambda), \psi \rangle_{V^*, V} = \sum_{n=0}^{N_v} \int_{\Omega} \partial_y (\psi^{(n)}) (y, u, \lambda) \psi d\omega(x)$$

The derivatives with respect to the control $u$ and the states $y^{(n)}$ are straightforward except for the push forward terms, whose derivatives are derived in the following.

**Lemma 3.1.** Let $\Omega \subset \mathbb{R}^d$ and $\Phi : \Omega \to \mathbb{R}^d$ be a smooth diffeomorphism. Then for $d = 1, 2, 3$

$$\text{div}(\text{det}(\nabla \Phi) \nabla \Phi^{-T}) = 0$$

holds.

**Proof.** See [13], p. 27. □

**Lemma 3.2.** Let $\Omega \subset \mathbb{R}^d$ and $\Phi$ be a smooth diffeomorphism. Moreover, let $y : \Omega \to \mathbb{R}$ and $\lambda : \Phi(\Omega) \to \mathbb{R}$ be sufficiently smooth. Then

$$\frac{\partial}{\partial \psi} \left( \int_{\Omega} y (\lambda \circ \Phi) \text{det}(\nabla \Phi) \, dx \right)[\psi] = \int_{\partial \Omega} y (\lambda \circ \Phi) \text{det}(\nabla \Phi) \nabla \Phi^{-T} n \cdot \psi d\omega(x)$$

$$- \int_{\Omega} \text{det}(\nabla \Phi) \nabla \Phi^{-T} \nabla y \lambda \circ \Phi \cdot \psi \, dx$$

holds for $\psi : \Omega \to \mathbb{R}^d$.

**Proof.**

$$\frac{\partial}{\partial \psi} \left( \int_{\Omega} y \lambda \circ \Phi \text{det}(\nabla \Phi) \, dx \right)[\psi] = \int_{\Omega} y (D \lambda \circ \Phi) \psi \text{det}(\nabla \Phi) + y \lambda \circ \Phi \text{det}(\nabla \Phi) \nabla \Phi^{-T} : \nabla \psi \, dx$$

$$= \int_{\Omega} y (D \lambda \circ \Phi) \psi \text{det}(\nabla \Phi) - \text{det}(\nabla \Phi) \nabla \Phi^{-T} \nabla y \lambda \circ \Phi \cdot \psi - y \text{div} \left( \text{det}(\nabla \Phi) \nabla \Phi^{-T} \lambda \circ \Phi \cdot \psi \right)$$

$$- y \text{det}(\nabla \Phi) \nabla \Phi^{-T} \nabla y \lambda \circ \Phi \cdot \psi \, dx + \int_{\partial \Omega} y (\lambda \circ \Phi) \text{det}(\nabla \Phi) \nabla \Phi^{-T} n \cdot \psi d\omega(x)$$

by using integration by parts. Applying lemma [18] we obtain

$$= \int_{\partial \Omega} y (\lambda \circ \Phi) \text{det}(\nabla \Phi) \nabla \Phi^{-T} n \cdot \psi d\omega(x) - \int_{\Omega} \text{det}(\nabla \Phi) \nabla \Phi^{-T} \nabla y \lambda \circ \Phi \cdot \psi \, dx$$

by using $(D \lambda \circ \Phi) \psi = (\nabla \lambda \circ \Phi) \cdot \psi$ and $\text{div}(aB) = a \text{div}(B) + B(\nabla a)$ for a scalar-valued function $a$ and a matrix-valued function $B$. □
**Theorem 3.3.** Let \( \varphi : \Omega^{(n)} \to \mathbb{R} \) and \( \lambda : \Omega^{(n+1)} \to \mathbb{R} \) be sufficiently smooth. Moreover, \( F : y \mapsto F(y) \) denotes a vector field depending on \( y^{(n)} \), \( \Phi^{(n)}(y) \) is given by \( \{ \} \) and \( \Omega^{(n+1)} = \Phi^{(n)}(y)(\Omega^{(n)}) \). Then

\[
D_{y^{(n)}} \left( \int_{\Omega^{(n+1)}} (\varphi(\Phi^{(n)}(y))^{-1}) \lambda \, dx \right) \bigg|_{y^{(n)}} = \int_{\partial \Omega^{(n)}} \! \! \! \tau \varphi(\lambda \circ \Phi^{(n)}(y)) DF(y^{(n)}) \psi \cdot n \, dx \\
- \int_{\Omega^{(n)}} \! \! \! \tau(\lambda \circ \Phi^{(n)}(y)) \nabla \varphi \cdot DF(y^{(n)}) \psi \, d\omega(x) + O(\tau^2)
\]

holds for \( \psi : \Omega^{(n)} \to \mathbb{R} \).

**Proof.** Due to definition we define

\[
\int_{\Omega^{(n+1)}} (\varphi(\Phi^{(n)}(y))^{-1}) \lambda \, dx = \int_{\Omega^{(n)}} \varphi(\lambda \circ \Phi^{(n)}(y)) \det(\nabla \Phi^{(n)}(y)) \, dx =: K(\Phi^{(n)}(y))
\]

The chain rule yields the variation with respect to \( y^{(n)} \) as

\[
D_{y^{(n)}} K(\Phi^{(n)}(y))[\psi_y] = DK(\Phi^{(n)}(y))(\partial_{y^{(n)}} \Phi^{(n)}(y))[\psi_y],
\]

Using the definition of \( \Phi^{(n)}(y) \) yields

\[
\partial_{y^{(n)}} \Phi^{(n)}(y)[\psi_y] = \tau DF(y^{(n)})[\psi_y]
\]

and applying lemma 5.2 gives

\[
\partial_{y^{(n)}} K[\psi_y] = \int_{\partial \Omega^{(n)}} \varphi(\lambda \circ \Phi^{(n)}(y)) \det(\nabla \Phi^{(n)}(y)) \nabla \Phi^{(n)}(y)^{-T} n \cdot \tau DF(y^{(n)}) \psi_y \, dx \\
- \int_{\Omega^{(n)}} \det(\nabla \varphi) \nabla \varphi^{-T} \nabla \varphi(\lambda \circ \Phi^{(n)}(y)) \cdot \tau DF(y^{(n)}) \psi_y \, d\omega(x)
\]

Since the determinant and inverse of a matrix satisfies

\[
det(I + \varepsilon A) = 1 + \varepsilon \text{Tr}(A) + O(\varepsilon^2) \quad \text{and} \quad (I + \varepsilon A)^{-1} = I - \varepsilon A + O(\varepsilon^2),
\]

respectively, for small \( \varepsilon > 0 \), we obtain

\[
det(\nabla \Phi^{(n)}(y)) = det(I + \tau F(y^{(n)})) = 1 + \tau \text{div}(F(y^{(n)})) + O(\tau^2)
\]

and

\[
(\nabla \Phi^{(n)}(y))^{-1} = (I + \tau \nabla F(y^{(n)}))^{-1} = I - \tau \nabla F(y^{(n)}) + O(\tau^2)
\]

for small time steps \( \tau > 0 \). Therefore, we get

\[
\partial_{y^{(n)}} K[\psi_y] = \int_{\partial \Omega^{(n)}} \varphi(\lambda \circ \Phi^{(n)}(y))(1 + \tau \text{div}(F(y^{(n)})))^T n \cdot \tau DF(y^{(n)}) \psi_y \, dx \\
- \int_{\Omega^{(n)}} (1 + \tau \text{div}(F(y^{(n)})))^T \nabla \varphi(\lambda \circ \Phi^{(n)}(y)) \cdot \tau DF(y^{(n)}) \psi_y \, d\omega(x) + O(\tau^2)
\]

which finally yields

\[
\partial_{y^{(n)}} K[\psi_y] = \int_{\partial \Omega^{(n)}} \tau \varphi(\lambda \circ \Phi^{(n)}(y)) DF(y^{(n)}) \psi_y \cdot n \, dx \\
- \int_{\Omega^{(n)}} \tau(\lambda \circ \Phi^{(n)}(y)) \nabla \varphi \cdot DF(y^{(n)}) \psi_y \, d\omega(x) + O(\tau^2)
\]

\( \square \)

**Remark 3.** The above theorem holds for arbitrary flux functions \( F \), i.e. \( F : V^{(n)} \to W \) for an appropriate Banach space \( W \).
Furthermore, we obtain for the final time step 

\[ (\partial\varphi(\Phi^{(n)}[y])^{-1})[\psi_\varphi], \lambda \}_{H^{(n+1)}} = (\lambda \circ \Phi^{(n)}[y], \psi_\varphi)_{H^{(n)}} + \tau(\lambda \circ \Phi^{(n)}[y] F^{(n)}(y^{(n)}) \cdot \nabla y^{(n)}, \psi_\varphi)_{H^{(n)}} + O(\tau^2) \]

holds for \( \psi_\varphi \in V^{(n)} \) and

\[ (\partial_y(\Phi^{(n)}[y])^{-1})[\psi_y], \lambda \}_{H^{(n+1)}} = (\varphi(\lambda \circ \Phi^{(n)}[y]) F^{(n)}(y^{(n)}) \cdot n, \psi_y)_{H^{(n)}} - \tau((\lambda \circ \Phi^{(n)}[y]) \nabla \varphi \cdot F^{(n)}(y^{(n)}), \psi_y)_{H^{(n)}} + O(\tau^2) \]

for \( \psi_y \in V^{(n)} \).

Proof. The first part is given by

\[
(\partial\varphi(\Phi^{(n)}[y])^{-1})[\psi_\varphi], \lambda \}_{H^{(n+1)}} = \int_{\Omega^{(n+1)}} (\psi_\varphi(\Phi^{(n)}[y])^{-1}) \lambda \, dx
\]

\[
= \int_{\Omega^{(n)}} \psi_\varphi(\lambda \circ \Phi^{(n)}[y]) \det(\nabla\Phi^{(n)}[y]) \, dx = \int_{\Omega^{(n)}} \psi_\varphi(\lambda \circ \Phi^{(n)}[y])(1 + \tau \text{div}(F(y^{(n)}))) \, dx + O(\tau^2)
\]

which yields the assumption with \text{div}(F(y^{(n)})) = F^{(n)}(y^{(n)}) \cdot \nabla y^{(n)}.

The second part is a direct consequence of theorem \[\ref{thm:main}\] by using the fact that \( F : \mathbb{R} \to \mathbb{R}^d \).

3.3. Adjoint System. We apply the above theorems to the optimal control problem (P). The derivative with respect to the control function \( u \) is given by

\[
(\partial u L, \psi_u)_{U^*, U} = (\partial u J(y, u), \psi_u)_{U^*, U} - \sum_{n=1}^{N_t} \tau (\langle B^{(n)} \rangle^* \lambda^{(n)}, \psi_u)_{U^*, U}
\]

for all \( \psi_u \in U \). The first variation of the Lagrange functional with respect to \( y^{(n)} \) is

\[
(\partial y^{(n)} L, \psi_y)_{V^{(n)}^*, V^{(n)}} = (\partial y^{(n)} J(y, u), \psi_y)_{V^{(n)}^*, V^{(n)}} + (\lambda^{(n)}, \psi_y)_{H^{(n)}} + \tau(\partial A(y^{(n)})) \psi_y, \lambda^{(n)}\}_{V^{(n)}^*, V^{(n)}}
\]

\[
- \left( (\lambda^{(n+1))}_{V^{(n)}_y} + \tau(\partial_y^{(n)} F^{(n)}(y^{(n)})) \cdot \nabla y^{(n)}, \psi_y)_{H^{(n)}} + \tau(\partial y^{(n)} \lambda^{(n+1)} \circ \Phi^{(n)}[y]) F^{(n)}(y^{(n)}) \cdot n, \psi_y)_{H^{(n)}},
\]

\[
- \tau(\partial y^{(n)} \lambda^{(n+1)} \circ \Phi^{(n)}[y]) F^{(n)}(y^{(n)}) \cdot n, \psi_y)_{H^{(n)}} + O(\tau^2)
\]

for all \( \psi_y \in V^{(n)} \) in the corresponding time step \( n = 1 \ldots N_t - 1 \) by using corollary \[\ref{cor:3.4}\]. Only the inner products in the \{ \} brackets are a result of the implicit variation with respect to the domain. Terms, which are handled explicitly in the time-discretisation, e.g. \( \tau F(y) \cdot \nabla y \), have variations of order \( \tau^2 \) which are directly included in \( O(\tau^2) \). All remaining terms are due to the variation of the partial differential equation as usual. We simplify the above result for \( \partial y^{(n)} L \) as

\[
(\partial y^{(n)} L, \psi_y)_{V^{(n)}^*, V^{(n)}} = (\partial y^{(n)} J(y, u), \psi_y)_{V^{(n)}^*, V^{(n)}} + (\lambda^{(n)}, \psi_y)_{H^{(n)}} + \tau(\partial A(y^{(n)})) \psi_y, \lambda^{(n)}\}_{V^{(n)}^*, V^{(n)}} - \left( (\lambda^{(n+1))}_{V^{(n)}_y} + \tau(\partial_y^{(n)} F^{(n)}(y^{(n)})) \cdot \nabla y^{(n)}, \psi_y)_{H^{(n)}} - \tau(\text{div}(\psi_y F^{(n)}(y^{(n)})) \lambda^{(n+1)} \circ \Phi^{(n)}[y]) F^{(n)}(y^{(n)}) \cdot n, \psi_y)_{H^{(n)}},
\]

Furthermore, we obtain for the final time step \( N_t \)

\[
(\partial y^{(N_t)} L, \psi_y)_{V^{(N_t)}^*, V^{(N_t)}} = (\partial y^{(N_t)} J(y, u), \psi_y)_{V^{(N_t)}^*, V^{(N_t)}} + (\lambda^{(N_t)}, \psi_y)_{H^{(N_t)}} + \tau(\partial A(y^{(N_t)})) \psi_y, \lambda^{(N_t)}\}_{V^{(N_t)}^*, V^{(N_t)}} + \tau(\partial_y^{(N_t)} F^{(n)}(y^{(n)})) \cdot n, \psi_y)_{H^{(n)}} + O(\tau^2)
\]
All terms of order $\tau$ are a consequence of the time-implicit scheme we chose for the discretisation and they disappear in explicit schemes. Since we only consider small time steps $\tau \ll 1$, we neglect them in the following for the last time step. Using the fact that
\[
\langle \text{div} (\psi_y F(y^{(n)})), \lambda \rangle_{H^{(n)}} = \langle \lambda F(y^{(n)}) \cdot n, \psi_y \rangle_{H^{(n)}} - \langle \nabla \lambda \cdot F(y^{(n)}), \psi_y \rangle_{H^{(n)}}
\]
hold, we can, roughly, identify the adjoint equation as
\[
\frac{1}{\tau} (\lambda^{(n)} - \lambda^{(n+1)} \circ \Phi^{(n)}(y)) + DA(y^{(n)}) \lambda^{(n)} + \frac{1}{\tau} \partial_y (\lambda^{(n+1)} \circ \Phi^{(n)}(y)) \cdot F(y^{(n)}) + C^{(n)}(\lambda^{(n+1)} \circ \Phi^{(n)}(y)) = \langle \nabla \lambda \cdot F(y^{(n)}), \psi_y \rangle_{H^{(n)}}
\]
in $(V^{(n)})^*$ and hence an equation in $\Omega^{(n)}$. Here $\mathcal{C}^{(n)} : V^{(n)} \to (V^{(n)})^*$ is given by
\[
\langle \mathcal{C}^{(n)}(\lambda^{(n+1)} \circ \Phi^{(n)}(y)), \psi \rangle_{(V^{(n)})^*, V^{(n)}} := \langle \nabla (\lambda^{(n+1)} \circ \Phi^{(n)}(y)) (y^{(n)} F'(y^{(n)}) + F(y^{(n)})) \cdot n, \psi \rangle_{H^{(n)}}
\]
and represents the boundary values. It is, among others, a consequence of the implicit domain variation. The gradient of the reduced cost functional $\hat{J}(u) := J(y(u), u)$ is given by
\[
\nabla \hat{J}(u) := R_U^{-1} \left( \frac{1}{\tau} \partial_y J(y, u) - \sum_{n=1}^{N_t} (B^{(n)})^* \lambda^{(n)} \right)
\]
where $R_U$ denotes the Riesz isomorphism $R_U : U \to U^*$.

3.4. Conclusion. The optimisation approach described in this section is very close to the numerical implementation of free boundary problems. The basic principle is illustrated in figure 3. Here, the state (or forward) equation yields the domains $\Omega^{(n)}$ recursively by the transformation $\Phi^{(n)}(y)$. Consequently, the adjoint (or backward) equation uses the same domains, starting from $\Omega^{(N_t)}$ to $\Omega^{(0)}$. Particularly, the push forward term of the discrete time derivative in the forward problem changes to a pullback term in the adjoint equation.

Instead of deriving the variation of each domain $\Omega^{(n)}$ directly, it is handled by the variation of the push forward terms, e.g. in the time-derivative, with respect to $\Phi^{(n)}(y)$ and hence to $y$. This procedure allows a easy derivation of the adjoint equation as only a few simple (explicit) terms are needed to obtain information about the shape dependency. Moreover, we showed above that terms, handled by an explicit time-discretisation, do not yield a contribution to the domain variation as the derivatives are mainly of order $\tau^2$. The next section shows some numerical examples applying the above method.

4. Numerical Examples

In this section we verify the optimisation approach shown in the previous sections numerically. The first test case involve the Navier-Stokes equation with a free surface, c.f. [1], where the transformation is given in a natural way, that is, the flow velocity is used. The second example is a Stefan-type problem where the final shape of a melting or solidification process is optimised. Here, the transformation is not given by a flow velocity but an artificial motion, similar to an ALE [3] method.
4.1. Navier-Stokes equation. In this example, we optimise the filling of an open liquid tank with small obstacles shown in figure 4. In particular, we search an inflow profile such that for a given outflow profile the free surface remains as still as possible. This yields the optimisation problem: Minimise

\[ J(u, \Omega) := \frac{1}{2} \int_0^T \int_{\Gamma_f} |u(x, t)|^2 \, d\omega(x) \, dt \]

subject to the Navier-Stokes equation

\begin{align*}
\partial_t u + u \cdot \nabla u - \nu \Delta u &= -\nabla p \quad \text{in } \Omega_t \times (0, T) \\
\text{div} u &= 0 \quad \text{in } \Omega_t \times (0, T) \\
uu &= 0 \quad \text{on } \Gamma_w \times (0, T) \\
u &= u_o \quad \text{on } \Gamma_o \times (0, T) \\
\nabla p \cdot n &= c \quad \text{on } \Gamma_i \times (0, T) \\
\nabla u \cdot n &= 0 \quad \text{on } \Gamma_f \times (0, T) \\
p &= 0 \quad \text{on } \Gamma_l \times (0, T) \\
u(0) &= 0 \quad \text{in } \Omega_0.
\end{align*}

The domain \( \Omega_t \) depends on the velocity profile \( u \) at the free surface \( \Gamma_f \). In particular, the motion of \( \Gamma_f \) is given by \( u|_{\Gamma_f} \). For convenience we use (9e) as inflow condition. Here, \( c \) denotes the control variable depending on space and time, i.e. \( c : \Gamma_i \times [0, T] \to \mathbb{R} \). From a physical point of view, \( \nabla p \cdot n \) is proportional to the inflow flux in normal direction. The condition for the free surface, (9f) and (9g), is described in more detail in e.g. [8].

4.1.1. Time Discretisation. For the time-discretisation a Chorin projection \[4, 12\] is used. In particular, we choose the domain transformation \( \Phi^{(n)} : \Omega^{(n)} \to \Omega^{(n+1)} \) as

\[ \Phi^{(n)}\{u\} := \text{Id}_{\Omega^{(n)}} + \tau u^{(n)} \]

Then, the time discretisation of (9) reads

\begin{align*}
\frac{1}{\tau} (u^{(n+1)} - (u^{(n)} \circ (\Phi^{(n)}\{u\})^{-1})) - \nu \Delta u^{(n+1)} &= -\nabla p^{(n+1)} \quad \text{in } \Omega^{(n+1)} \\
\Delta p^{(n+1)} &= \frac{1}{\sigma} \text{div} u^{(n+1)} \quad \text{in } \Omega^{(n+1)} \\
uu^{(n+1)} &= 0 \quad \text{on } \Gamma_w \\
u^{(n+1)} &= u_o \quad \text{on } \Gamma_o \\
p^{(n+1)} &= 0 \quad \text{on } \Gamma_f^{(n+1)} \\
\nabla p^{(n+1)} \cdot n &= c^{(n+1)} \quad \text{on } \Gamma_i \\
u^{(0)} &= 0 \quad \text{in } \Omega^{(0)}.
\end{align*}
where $\sigma > 0$ denotes a regularisation parameter. All boundaries not mentioned above are set to uniform Neumann condition, i.e. $\nabla p \cdot n = 0$ or $\nabla u \cdot n = 0$, which is a consequence of Chorin’s projection. The cost function is discretised straightforward

$$J(u, c) := \frac{1}{2} \sum_{n=1}^{N_t-1} \tau \int_{\Gamma_f(n)} |u^{(n)}|^2 \, dw(x).$$

4.1.2. Adjoint system. Using the previous optimisation approach, we obtain the identification of the adjoint system as

$$\begin{aligned}
\frac{1}{\tau} \left( \lambda_u^{(n)} - (\lambda_u^{(n+1)} \circ \Phi^{(n)}(u)) \right) - \nu \Delta \lambda_u^{(n)} &= \nabla \lambda_p^{(n)} - (\lambda_u^{(n+1)} \circ \Phi^{(n)}(u)) \cdot \nabla u^{(n)} \quad \text{in } \Omega^{(n)}
- \sigma \Delta \lambda_p^{(n)} &= \text{div} \lambda_u^{(n)} \quad \text{in } \Omega^{(n)}
\nu \nabla \lambda_u^{(n)} \cdot n &= -u^{(n)} - (\lambda_u^{(n+1)} \circ \Phi^{(n)}(u)) \cdot n \quad \text{on } \Gamma_f^{(n)}
\lambda_p^{(n)} &= 0 \quad \text{on } \Gamma_f^{(n)}
\nu \nabla \lambda_p^{(n)} \cdot n &= -\lambda_u^{(n)} \cdot n \quad \text{on } \Gamma_i
\lambda_u^{(n)} &= 0 \quad \text{on } \Gamma_i
\sigma \nabla \lambda_p \cdot n &= 0 \quad \text{on } \Gamma_{w,o}
\lambda_u^{(N_t)} &= 0 \quad \text{in } \Omega^{(N_t)}.
\end{aligned}$$

The gradient of the reduced cost functional $\hat{J}(c) := J(u(c), c)$ is identified by

$$\nabla \hat{J}(c) := -\sigma \lambda_p|_{\Gamma_f}.$$  

A detailed derivation of the adjoint system can be found in [13]. Note that only the pullback terms on the right hand side, e.g. $(\lambda_u^{(n+1)} \circ \Phi^{(n)}(u)) \cdot \nabla u^{(n)}$, are a result of the domain variation.

4.1.3. Numerical Results. For the numerical implementation we use a meshless particle method, cf. [13]. These methods are superior to mesh-based methods like finite elements as no connectivity information is used and hence no expensive remeshing is needed if the positions of the supporting points change. Particularly, we use the finite pointset method, cf. [17]. The optimisation process is performed by a second order BFGS method with Armijo rule, cf. [10].

The final time is set to $T = 2.5$, the step size $\tau = 0.005$. The domain has a width and height of 5.0. Moreover, the viscosity is set to $\nu = 10$ and $\sigma = 0.005$. The outflow velocity is given by a parabolic profile with $u_{\text{max}} = 3.0$ in order to avoid singularities at the corners.

Figure [9] and [10] show the results for the uncontrolled and optimised case, respectively. Here the colour represents the velocity magnitude. The uncontrolled case, that is, no inflow is given, yields a high velocity at the free boundary and therefore a large deformation of it, as expected. The resulting behaviour of the free surface is an effect of the high viscosity of the fluid. The controlled case forms out a straight flow, as good as possible for the given geometry, from the inflow to the outflow, which does not affect the free surface strongly. Hence, the surface velocity is much smaller than for the uncontrolled case, cf. figure [9]. Note that due to the setting, the desired value of the velocity $u = 0$ at the free surface is not reachable and hence $\min J = 0$ cannot be expected. The corresponding evaluation of the cost functional and gradient norm is shown in figure [8]. The gradient norm shows a strong decrease in the first iterations, then it becomes flatter with order 1.04 to 1.25. Note that this is not surprisingly as the problem is highly non convex. On the one hand, the Navier-Stokes equation yields a complicated constraint, on the other hand, the adaptation can cause problems for the optimisation process.
4.2. **Stefan-type Problem.** The second example deals with a Stefan-type problem [19]. These problems describe phase transitions, e.g., from liquid to solid, see figure 5. Particularly, we solve the heat equation

\[
\begin{align*}
\partial_t \vartheta - \Delta \vartheta &= f \quad \text{in } \Omega_t \times (0, T) \\
\vartheta &= \vartheta_m \quad \text{on } \Gamma_t \times (0, T) \\
\vartheta(0) &= \vartheta_0 \quad \text{in } \Omega_0 \\
V_n &= \beta \nabla \vartheta \cdot n
\end{align*}
\]

(10)

with \( \vartheta_m \in \mathbb{R} \) as the boundary temperature (melting point) and \( \vartheta_0 \) as the initial temperature distribution. \( V_n \) denotes the velocity of the boundary in normal direction and hence also defines \( \Omega_t \). Moreover, \( \beta \in \mathbb{R} \) denotes a material constant. Note that the sign of \( \beta \) indicates whether \( \Omega_t \) is a liquid or solid phase.

For the minimisation we start with an arbitrary domain \( \Omega_0 \) with boundary \( \Gamma_0 \) and want to achieve a desired shape with boundary \( \Gamma_d \) at final time \( T \). In the following, we choose an ellipse for \( \Gamma_d \). This desired boundary shape is parametrised in order to formulate an optimisation problem. The easiest way of a parametrisation is the implicit definition of an ellipse given by

\[
\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1
\]

(11)

in two dimensions. Here, \( a \) and \( b \) denote the stretching in the \( x \)- and \( y \)-direction, respectively. Hence, the domain \( \Omega_T \) has the boundary \( \Gamma_d \) if (11) is satisfied for all points on the boundary \( \Gamma_T \). The minimisation problem can therefore be interpreted as: Find a heat source \( f \) such that

\[
\frac{1}{2} \int_{\Gamma_T} \left( \frac{x^2}{a^2} + \frac{y^2}{b^2} - 1 \right)^2 \, d\omega(x)
\]

(12)

is minimal subject to (10). The integral is minimal, in particular zero, if all boundary points satisfy (11). For all other settings, i.e., a measurable amount of points do not lay on the desired boundary, the integral is greater than zero.

4.2.1. **Time-discretisation.** For convenience, we replace the Dirichlet boundary condition of equation (10) by a Robin condition, i.e.,

\[
\vartheta = \vartheta_m \quad \rightarrow \quad \vartheta + \kappa \nabla \vartheta \cdot n = \vartheta_m
\]

on \( \Gamma_t \) where \( \kappa > 0 \) denotes a small regularisation parameter depending on the discretisation. Since we are dealing with a problem without convection and we only know the motion of the boundary, the choice of the transformation is not obvious. In particular, we choose a smooth continuation of the boundary velocity by solving a Laplace equation similar to the ALE method, cf. [3].

\[
-\Delta u = 0 \quad \text{in } \Omega_t \\
u + \kappa \nabla u \cdot n = \beta (\nabla \vartheta \cdot n)n \quad \text{on } \Gamma_f
\]

where we use a regularisation of the Dirichlet boundary condition as before.

We obtain the time discrete system by first solving the velocity equation

\[
-\Delta u^{(n)} = 0 \quad \text{in } \Omega^{(n)}
\]

\[
u^{(n)} + \kappa \nabla u^{(n)} \cdot n = \frac{\beta}{\kappa} (\vartheta_m - \vartheta^{(n)})n \quad \text{on } \Gamma^{(n)}
\]

then the transformation

\[
\Phi^{(n)} = \text{Id}_{\Omega^{(n)}} + \tau u^{(n)}
\]

in \( \Omega^{(n)} \)

and finally the heat equation

\[
\frac{1}{\tau} (\vartheta^{(n+1)} - (\vartheta^{(n)} \circ (\Phi^{(n)})^{-1})) - \Delta \vartheta^{(n+1)} = (u^{(n)} \cdot \nabla \vartheta^{(n)}) \circ (\Phi^{(n)})^{-1} + \chi^{(n+1)} \quad \text{in } \Omega^{(n+1)}
\]

\[
\vartheta^{(n+1)} + \kappa \nabla \vartheta^{(n+1)} \cdot n = \vartheta_m \quad \text{on } \Gamma^{(n+1)}
\]

with the initial condition

\[
\vartheta^{(0)} = \vartheta_0 \quad \text{in } \Omega^{(0)}.
\]
where $\chi$ denotes a spatial localisation function depending on $x$ only. Note that we use the Robin condition of the heat equation to replace $\nabla \vartheta \cdot n$ in the velocity equation and add a convection term to the right hand side of the heat equation due to the artificial transformation of the domain.

The discrete cost functional reads

$$J(u, c) := \frac{1}{2} \int_{\Gamma(N_t)} \left( (\Phi^{(N_t)}(u))^T E (\Phi^{(N_t)}(u)) - 1 \right)^{2\alpha} d\omega(x)$$

with

$$E = \begin{pmatrix} a^{-2} & 0 \\ 0 & b^{-2} \end{pmatrix}$$

which corresponds to [12]. Note that we use the power of $2\alpha$ in the cost functional to be able to consider also $L^p$ norms.

4.2.2. Adjoint system. Using the previous optimisation approach, we identify the adjoint equation and gradient as

$$-\Delta \lambda^{(n)}_u = (\lambda^{(n+1)}_\vartheta \circ \Phi^{(n)}(u)) \nabla \vartheta^{(n)}$$

in $\Omega^{(n)}$

$$\lambda^{(n)}_u + \kappa \nabla \lambda^{(n)}_u \cdot n = 0$$

on $\Gamma^{(n)}$

$$\frac{1}{\tau} \big( \lambda^{(n)}_\vartheta - \lambda^{(n+1)}_\vartheta \circ \Phi^{(n)}(u) \big) - \Delta \lambda^{(n)}_\vartheta = -u^{(n)} \cdot \nabla (\lambda^{(n+1)}_\vartheta \circ \Phi^{(n)}(u))$$

in $\Omega^{(n)}$

$$\lambda^{(n)}_\vartheta + \kappa \nabla \lambda^{(n)}_\vartheta \cdot n = \kappa (\lambda^{(n+1)}_\vartheta \circ \Phi^{(n)}(u)) u^{(n)} \cdot n - \frac{\beta}{\kappa} \lambda^{(n)}_u \cdot n$$

on $\Gamma^{(n)}$

for $n = 1, \ldots, N_t - 1$. For $n = N_t$ we get

$$-\Delta \lambda^{(N_t)}_u = 0$$

in $\Omega^{(N_t)}$

$$\lambda^{(N_t)}_u + \kappa \nabla \lambda^{(N_t)}_u \cdot n = -2\alpha \kappa \left( (\Phi^{(N_t)}(u))^T E (\Phi^{(N_t)}(u)) - 1 \right)^{2\alpha-1} E (\Phi^{(N_t)}(u))$$

on $\Gamma^{(N_t)}$

and

$$\lambda^{(N_t)}_\vartheta - \tau \Delta \lambda^{(N_t)}_\vartheta = 0$$

in $\Omega^{(N_t)}$

$$\lambda^{(N_t)}_\vartheta + \kappa \nabla \lambda^{(N_t)}_\vartheta \cdot n = -\frac{\beta}{\kappa} \lambda^{(N_t)}_u \cdot n$$

on $\Gamma^{(N_t)}$.

The gradient of the reduced cost functional reads

$$\nabla \tilde{J}(c)^{(n)} := - \int_{\Omega^{(n)}} \chi(x) \lambda^{(n)}_\vartheta(x) dx.$$ 

Again, a more detailed derivation of the adjoint equations can be found in [13].

4.2.3. Numerical Results. The spatial discretisation is similar to the previous test case, performed by a meshless method with adaptation. Again, the BFGS method with Armijo rule is used for the optimisation. The setting is $\alpha = 2, \beta = -1, \kappa = 0.01$ and $\vartheta_m = 0$. Moreover, we set the time step size $\tau = 0.01$ and the final time to $T = 0.3$. The localisation function $\chi$ is given by

$$\chi(x) c^{(n)} := \sum_{i=1}^{N_x} \sum_{j=1}^{N_x} c^{(n)}_{ij} \exp \left( -25 \left( (x - x_{ij})^2 + (y - y_{ij})^2 \right) \right)$$

for the supporting points $-1, -0.6, -0.2, 0.2, 0.6, 1.0$, i.e. $x_{11} = -1, y_{11} = -1$ till $x_{66} = 1, y_{66} = 1$. Furthermore, the initial domain is given by a square with edge length 0.6 and the desired shape is given by a circle of radius 1 which yields $a = 1.0$ and $b = 1.0$. Note that this example does not base on a physical setting as we only want to show the feasibility of our method.

The results are shown in figure [14] where the colour represents the absolute value of the source induced by the control function, i.e. $\chi c$. Moreover, the black line denotes the desired shape at final time. The optimised case shows a small deformation in the first time steps, which becomes larger close to the final time. The shape at final time is very close to the desired one. The convergence, shown in figure [15] is
approximately of order 1.5 and no Armijo step size reduction is needed. Furthermore, the plateaus in the cost functional and gradient norm are due to a reset of the BFGS matrix, implemented for stability reasons. Note that we neglect these plateaus for the consideration of the convergence order.

5. Conclusion

In this paper we presented a simple approach for the optimisation of free boundary problems which is very close to their numerical implementation. In particular, we applied an extended time-discretisation based on an operator splitting. For this a decoupling of the deformation of the domain and the solution of the remaining partial differential equation was performed. With this approach, adjoint-based optimisation can be applied easily. Since we do not perform an explicit derivative with respect to the domain, this information is partly obtained by the variation of the push forward terms needed for the discrete time derivative. The numerical results based on the Navier-Stokes equation and a Stefan-type problem showed that this optimisation approach yields good results and is very easy to implement numerically.

References

[1] J.T. Beale. The initial value problem for the Navier-Stokes equations with a free surface. Communications on Pure and Applied Mathematics, 34:359–392, 1981.
[2] M.K. Bernauer and R. Herzog. Optimal control of the classical two-phase Stefan problem in level set formulation. Preprint TU Chemnitz, 2010.
[3] H. Braess and P. Wriggers. Arbitrary Lagrangian Eulerian finite element analysis of free surface flow. Computer Methods in Applied Mechanics and Engineering, 190(1-2):95–109, 2000.
[4] A. Chorin. Numerical solution of the Navier-Stokes equations. J. Math. Comp., 22:745–762, 1968.
[5] P. Colli, C. Verdi, and A. Visintin. Free boundary problems. Theory and applications. International Series of Numerical Mathematics. Birkhäuser Verlag, 2004.
[6] P. Deuflhard. Newton methods for nonlinear problems: Affine invariance and adaptive algorithms. Springer series in computational mathematics. Springer, 2004.
[7] I. Faragó. Splitting methods and their application to the abstract Cauchy problems. Numerical Analysis and Its Applications, pages 35–45, 2005.
[8] J.H. Ferziger and M. Perić. Numerische Strömungsmechanik. Springer, 2008.
[9] E. Godlewski and P.A. Raviart. Numerical approximation of hyperbolic systems of conservation laws. Springer Verlag, 1996.
[10] H. Hinze and S. Ziegenbalg. Control of the free boundary in a two-phase Stefan problem with flow driven by convection. Proceedings in Applied Mathematics and Mechanics, 2007.
[11] R.A. Ibrahim. Liquid sloshing dynamics: Theory and applications. Cambridge University Press, 2005.
[12] J. Kuhnert, M. Schäfer, and R. Gerstenberger. Meshfree numerical scheme for time dependent problems in continuum mechanics. Preprint available at Fraunhofer ITWM.
[13] J. Marburger. Optimal control based on meshfree approximations. PhD thesis, TU Kaiserslautern, 2011.
[14] S. Mas-Gallic and P. A. Raviart. A particle method for first-order symmetric systems. Numerische Mathematik, 51:323–352, 1987.
[15] S. Repke. Adjoint-based optimization approaches for stationary free surface flows. PhD thesis, TU Kaiserslautern, 2011.
[16] W. Shyy, H.S. Udaykumar, M.M. Rao, and R.W. Smith. Computational fluid dynamics with moving boundaries. Dover Books on Engineering Series. Dover Publications, 2007.
[17] S. Tiwari and J. Kuhnert. Grid free method for solving the Poisson equation. Berichte des Fraunhofer ITWM, 25, 2001.
[18] F. Troeltzsch. Optimale Steuerung partieller Differentialgleichungen. Vieweg Verlag, 1st edition, 2005.
[19] A. Visintin. Introduction to Stefan-type problems. Handbook of Differential Equations: Evolutionary Equations, 4:377–484, 2008.

(*) FRAUNHOFER-INSTITUT FÜR TECHNO- UND WIRTSCHAFTSMATHEMATIK, FRAUNHOFER-PLATZ 1, D-67663 KAISERSLAUTERN, E-MAIL: JAN@NIT-SERVICE.DE
Figure 6. Filling of a liquid tank at time 0, 1.25, 2.5 for the uncontrolled case.

Figure 7. Filling of a liquid tank at time 0, 1.25, 2.5 for the optimised case.

Figure 8. Cost functional and relative gradient norm.

Figure 9. Mean surface velocity $\|u\|_{L^2(\Gamma)}$ over time for the uncontrolled and controlled case.
Figure 10. Optimal solution for the Stefan problem with $a = 1.0$, $b = 1.0$. The time steps are equally spaced from $t = 0$ to $T = 0.3$. The colour represents the magnitude of the source and the black line the desired shape at final time.

Figure 11. Cost functional and gradient norm for $a = 1.0$, $b = 1.0$. (a) Cost functional. (b) Relative gradient norm.