Riccati-feedback Control of a Two-dimensional Two-phase Stefan Problem

Björn Baran* Peter Benner† Jens Saak‡

*Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstr. 1, 39106 Magdeburg.
Email: baran@mpi-magdeburg.mpg.de, ORCID: 0000-0001-6570-3653
†Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstr. 1, 39106 Magdeburg.
Email: benner@mpi-magdeburg.mpg.de, ORCID: 0000-0003-3362-4103
‡Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstr. 1, 39106 Magdeburg.
Email: saak@mpi-magdeburg.mpg.de, ORCID: 0000-0001-5567-9637

Abstract: We discuss the feedback control problem for a two-dimensional two-phase Stefan problem. In our approach, we use a sharp interface representation in combination with mesh-movement to track the interface position. To attain a feedback control, we apply the linear-quadratic regulator approach to a suitable linearization of the problem. We address details regarding the discretization and the interface representation therein. Further, we document the matrix assembly to generate a non-autonomous generalized differential Riccati equation. To numerically solve the Riccati equation, we use low-rank factored and matrix-valued versions of the non-autonomous backward differentiation formulas, which incorporate implicit index reduction techniques. For the numerical simulation of the feedback controlled Stefan problem, we use a time-adaptive fractional-step-theta scheme.

We provide the implementations for the developed methods and test these in several numerical experiments. With these experiments we show that our feedback control approach is applicable to the Stefan control problem and makes this large-scale problem computable. Also, we discuss the influence of several controller design parameters, such as the choice of inputs and outputs.

Keywords: two-phase Stefan problem, closed-loop, feedback, differential Riccati equation, non-autonomous, boundary control

AMS subject classifications: 35R35, 49N10, 65F45, 93A15, 93B52, 93C10

Novelty statement: We propose a linear-quadratic regulator approach for the feedback stabilization of a two-dimensional two-phase Stefan problem, where the control target is to steer the interface position. This closed-loop control problem goes beyond existing open and closed-loop control approaches for one-dimensional or one-phase Stefan problems. Further, this is the first time the linear-quadratic regulator approach is applied to this type of problem with a moving interface or inner boundary. Our approach can handle the non-linearities and differential-algebraic structures induced by the Stefan problem as well as time-dependent matrices that are present in a non-autonomous differential Riccati equation. We use a new non-autonomous backward differentiation formula method to numerically solve this Riccati equation and compute feedback controls, which successfully stabilize the interface position.

1 Introduction

The solidification and melting of materials is an active and intensively studied field with numerous applications. This phase-change problem can be modeled by a non-linear PDE and is often called Stefan problem after J. Stefan who describes it in his works [Ste89, Ste91, Ste90]. In a certain domain, the temperature of the material is either below, above, or equal to the specific melting temperature of the material. Accordingly, the domain is split into a solid and a liquid phase, which are separated by an interface or inner boundary. J. Stefan formulates what is now called Stefan condition in [Ste89], which couples the time-derivative of the interface position with the jump of the temperature gradient along the interface. On the other hand, J. Stefan was not the first to consider this type of problem. G. Lamé and B. P. Clapeyron were concerned with this in an earlier work as well [LC31], such that it is also called Lamé–Clapeyron problem.

Several books address the Stefan problem, e.g., [Rub71, NCM11, Gup18, KK20b]. An extensive historical survey of the Stefan problem can be found in the book by L. I. Rubenšteĭn [Rub71, Introduction: §1]. Early works on the Stefan problem usually consider the one-dimensional case, while higher dimensional cases where only studied decades after the original publication.

The aim of this manuscript is to compute and apply feedback control to the two-dimensional two-phase Stefan problem. While this problem has been studied in combination with open-loop controls several times, see e.g. [Zie08, Ber10, ANS14, ANS15, BBHS18] and the reference therein, only recently, also closed-loop, i.e. feedback control, for the Stefan problem has been discussed in [KK19, KK20a, KK20b]. However, these works address only the one-dimensional case. On the one hand, the novelty of our work lies in the consideration of feedback control for the two-phase Stefan problem in two spatial dimensions in contrast to the one-phase or one-dimensional Stefan problem. On the other hand, we apply the linear-quadratic regulator (LQR) approach to the Stefan problem, which goes beyond the types of problems that have been studied in connection with this approach.

In particular, the application of LQR requires the treatment of the non-linearities, the differential-algebraic nature, and the time-varying character of the Stefan problem. In this manuscript, we address the details on how to transform it into a linear ordinary differential equation, apply the LQR approach to compute feedback controls, and then use these feedback controls to stabilize the interface position in the non-linear differential-algebraic problem. In order to do this, we linearize the Stefan problem around a reference trajectory and assemble a generalized differential Riccati equation (DRE). For the derivation of the LQR problem and the resulting DRE we refer to [Rei72, BG91]. An extensive study on DREs and their numerous applications can be found in [AKFIJ03]. Since the Stefan condition is an algebraic equation, which is coupled to the Stefan problem, the DRE we are considering results from a differential-algebraic equation (DAE). Details on generalized DREs can be found in [KM90a] and for DREs resulting from DAEs, see [KM90b].

To solve the Stefan problem numerically and assemble the DRE, we discretize in space using the finite element method (FEM) by applying the software FEniCS [LWH12]. This results in a large-scale matrix-valued DRE with time-dependent coefficients, denoted as a non-autonomous DRE. While we observe in numerical experiments that, usually, its solution can be well approximated by a low-rank factorization, this is proven theoretically only for the autonomous DRE in [Sti18b]. Well known methods, which use the low-rank structure of the numerical solution, are splitting schemes [Sti15a, Sti15b, Sti18a, OPW18, MOPP18], Rosenbrock and Peer methods [Men12, LMS15, Lan17, BL18] as well as the backward differentiation formulas (BDF) [BM04, Men12, LMS15, BM18]. Krylov subspace methods [BBH21, KM20, KS20, GHJK18] and exponential integrators [LZL20] for DREs have been developed recently as well. In [Men12, BM18] Rosenbrock and BDF methods and in [LMS15, Lan17] also Peer methods are studied for a non-autonomous DRE where the mass matrix is constant. In extension of this, splitting schemes and BDF methods are developed in [BBSS21] for non-autonomous DREs with a time-varying mass matrix. In contrast to the BDF methods, the splitting schemes require that the coefficients can be decomposed into a time-dependent scalar function times a constant matrix. The non-autonomous DREs we consider go beyond this case. We use mesh movement techniques to track the interface, and, as a consequence, the single matrix entries change very differently, possibly for all matrices. Thus, the BDF methods are the most promising method for the non-autonomous DRE resulting from the Stefan problem and we apply the non-autonomous BDF method from [BBSS21].

Structure of the Manuscript In this manuscript, we derive and apply several feedback controls to the two-dimensional two-phase Stefan problem. For this, we state the equations that model the Stefan problem and the mesh movement that we use to track the interface position (Section 2). Then, we linearize the resulting system and semi-discretize it in space (Section 3). With the resulting time-dependent matrices we formulate a non-autonomous DRE resulting from the LQR approach (Section 4). We solve this DRE with the non-autonomous BDF method and use the resulting feedback gain matrices to compute a feedback control (also Section 4). We apply the described methods in several numerical experiments for different parameter settings and specify where all our codes and data are available (Section 5).

Notation In most equations, we omit the time-dependence (t), the spatial dependence (x), or the combination of both (t, x). This is supposed to improve the readability of the equations.

2 Two-dimensional Two-phase Stefan Problem

In this section, we define the equations describing the Stefan problem, its boundary conditions and initial values. This involves equations characterizing the temperature and interface movement. Our goal is to use a Riccati feedback control approach to control the interface position. For this, we choose a sharp interface representation in the formulation of the Stefan problem, opposed to, e.g., the level set representation of the interface in [Ber10]. As in [BPS10, BPS13], we extend the interface movement to the whole domain. This ensures the mesh regularity for the semi-discretized Stefan problem that is described in Section 3.

At each $t \in [0, t_{\text{end}}]$, the domain is $\Omega(t) \subset \mathbb{R}^2$. One instance is illustrated in Figure 1. We split the domain $\Omega(t)$ into the two regions corresponding to the two phases. These are the region where the material is in its solid phase $\Omega_s(t)$ and accordingly, the region $\Omega_l(t)$ related to the liquid phase. The two phases are separated by the interface $\Gamma_{\text{int}}(t)$. This inner phase-boundary can move such that its position is time-dependent. Thus, also the two phases $\Omega_s(t)$ and $\Omega_l(t)$ are time-dependent and, as a consequence, so is the whole domain $\Omega(t)$ and its boundary regions. The boundary of $\Omega(t)$ is separated into $\Gamma_u(t)$, $\Gamma_{\text{cool}}(t)$ and $\Gamma_{\text{N}}(t)$ as depicted in Figure 1. Note that the outer shape of $\Omega(t)$ is constant for the realization chosen in this manuscript. Thus, the time-dependence of $\Omega(t)$ is not absolutely necessary even though its sub-domains are time-dependent. However, we keep the time-dependence in our notation in order to not restrict our methods to this case.

We follow the definition of the Stefan problem from [BBHS18]. However, here we use it in a more compact form, i.e. omitting the couplings with the Navier-Stokes equations and the interface graph formulation. While the Navier-Stokes equations alone add additional algebraic constraints, making the DAE harder to classify, both the Navier-Stokes equations and the interface graph formulation add more nonlinearities to the problem. In order to develop the general numerical strategy, we first want to study the feedback control problem for this simplified setting without these couplings, before delving into the additional technical challenges of the full problem formulation.

We denote the temperature as $\Theta(t)$ and model it with the partial differential Equation (1):

$$\dot{\Theta} - \Upsilon \cdot \nabla\Theta - \alpha\Delta\Theta = 0, \qquad \text{on } (0, t_{\text{end}}] \times \Omega, \tag{1a}$$
$$\partial_{\boldsymbol{n}}\Theta = u, \qquad \text{on } (0, t_{\text{end}}] \times \Gamma_u, \tag{1b}$$
$$\Theta = \Theta_{\text{cool}}, \qquad \text{on } (0, t_{\text{end}}] \times \Gamma_{\text{cool}}, \tag{1c}$$
$$\Theta = \Theta_{\text{melt}}, \qquad \text{on } (0, t_{\text{end}}] \times \Gamma_{\text{int}}, \tag{1d}$$
$$\partial_{\boldsymbol{n}}\Theta = 0, \qquad \text{on } (0, t_{\text{end}}] \times \Gamma_{\text{N}}, \tag{1e}$$
$$\Theta(0) = \Theta_0, \qquad \text{on } \Omega. \tag{1f}$$

In Equation (1b), we apply the control $u(t)$ as a Neumann condition on the control boundary $\Gamma_u(t)$. The Equations (1c) and (1d) describe the Dirichlet conditions on the cooling boundary $\Gamma_{\text{cool}}(t)$ and the interface $\Gamma_{\text{int}}(t)$ with the constants Θ_{cool} and Θ_{melt}, respectively. Equation (1f) presents the initial condition with the initial temperature distribution Θ_0. The heat conductivities

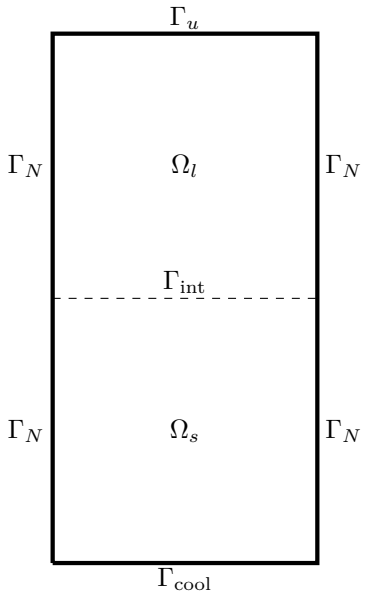

Figure 1: One Instance of the Domain $\Omega(t) \subset \mathbb{R}^2$ of the Stefan Problem.

in the solid phase k_s and in the liquid phase k_l are collected in $\alpha(\Theta(t))$:

$$\alpha = \begin{cases} k_s, & \text{on } \Omega_s, \\ k_l, & \text{on } \Omega_l. \end{cases}$$

In Equation (1a), the temperature is coupled with the extended interface movement $\Upsilon(\nabla\Theta(t))$. For every $t \in (0, t_{\text{end}}]$, we model $\Upsilon(\nabla\Theta(t))$ with a system of algebraic equations, in the sense that they do not contain any time-derivatives:

$$\Delta\Upsilon = 0, \qquad \text{on } \Omega, \tag{2a}$$
$$\Upsilon = \left(\frac{1}{\ell}[k(\nabla\Theta(t))]_l^s\right) \cdot \boldsymbol{n}_{\text{int}}, \qquad \text{on } \Gamma_{\text{int}}, \tag{2b}$$
$$\Upsilon = 0, \qquad \text{on } \Gamma_{\text{cool}} \cup \Gamma_u, \tag{2c}$$
$$\Upsilon \cdot \boldsymbol{n} = 0, \qquad \text{on } \Gamma_{\text{N}}, \tag{2d}$$
$$\Upsilon(0) = 0, \qquad \text{on } \Omega, \tag{2e}$$

On the interface, $\Upsilon\big|_{\Gamma_{\text{int}}}(\nabla\Theta(t)) = \Upsilon_{\text{int}}(\nabla\Theta(t))$ is the interface movement in normal direction, where $\boldsymbol{n}_{\text{int}}(t)$ is the unit normal vector pointing from $\Omega_s(t)$ to $\Omega_l(t)$. $\Upsilon_{\text{int}}(\nabla\Theta(t))$ is coupled to $\Theta(t)$ through the Stefan condition (2b). Here, ℓ is the latent heat constant and

$$[k(\nabla\Theta)]_l^s = k_s \partial_{\boldsymbol{n}_{\text{int}}}\Theta\big|_{\Omega_s} - k_l \partial_{-\boldsymbol{n}_{\text{int}}}\Theta\big|_{\Omega_l} \tag{3}$$

is the jump of the temperature gradient along $\Gamma_{\text{int}}(t)$. Equations (2c) and (2d) ensure that the outer boundaries of $\Omega(t)$ do not move such that the outer shape of the domain does not change.

We use $\Upsilon(\nabla\Theta(t))$ in Section 3 for the mesh movement. The Stefan problem, which is described by the system of Equations (1) and (2), is a non-linear system of DAEs on a time-varying domain. Due to the coupling of $\Theta(t)$ and $\Upsilon_{\text{int}}(\nabla\Theta(t))$ in (2b) and the two temperature-dependent phases in the definition of $\alpha(\Theta(t))$, both, $\Upsilon_{\text{int}}(\nabla\Theta(t))$ and $\alpha(\Theta(t))$, depend on $\Theta(t)$. Thus, the terms $\Upsilon(\nabla\Theta(t)) \cdot \nabla\Theta(t)$ and $\alpha(\Theta(t))\Delta\Theta(t)$ in Equation (1a) are nonlinear.

3 Discretization and Linearization

In order to apply the LQR approach in Section 4, we need to formulate the Stefan problem in a standard state-space format, which is linear and semi-discretized in space. Thus, we describe how to transform the coupled DAE system of Equations (1) and (2) into

$$\begin{aligned} \mathcal{M}\dot{x}^h &= \mathcal{A}x^h + \hat{\mathcal{B}}u^h, \\ y^h &= \mathcal{C}x^h, \end{aligned} \tag{4}$$

in this section. To generate the square matrices $\mathcal{A}(t), \mathcal{M}(t) \in \mathbb{R}^{n\times n}$, the input matrix $\hat{\mathcal{B}}(t) \in \mathbb{R}^{n\times m}$, and output matrix $\mathcal{C}(t) \in \mathbb{R}^{r\times n}$, we spatially discretize and linearize the Stefan problem. Further, we take special care on how the boundary conditions (1d) and (2b) are treated in the definition of the matrices for Equation (4) since they are of particular importance for our feedback control problem.

For the spatial discretization, we use the finite element method (FEM) on a mesh of triangular cells $\mathscr{Q}^h(t) = \{Q(t)\}$ that changes over time, driven by the movement of the interface. The interface $\Gamma_{\text{int}}(t)$ itself is represented explicitly and sharply through facets that are aligned with $\Gamma_{\text{int}}(t)$. In order to track $\Gamma_{\text{int}}(t)$ with the mesh, we use the semi-discrete extended interface movement $\Upsilon^h(\nabla\Theta^h(t))$ to adapt the mesh inside the whole domain, as described in [BPS10, BPS13]. In this way, we prevent mesh tangling and too strong deformations. See [BBHS18] for a detailed description of our implementation.

In order to define the linearization of the Stefan problem, we require a trajectory generated by applying an open-loop control approach to the nonlinear problem. For this, we take the open-loop control approach from [BBHS18]. This (desired) reference trajectory contains the semi-discrete reference solutions $\tilde{\Theta}^h(t)$, $\tilde{\Upsilon}^h(\nabla\tilde{\Theta}^h(t))$, $\Gamma_{\text{int,ref}}(t)$, and the heat conductivities $\tilde{\alpha}(t)$, depending on the reference trajectory.

Combining these two techniques, we derive the linearized, semi-discrete version of Equation (1a)

$$\dot{\Theta}^h - \Upsilon^h \cdot \nabla\tilde{\Theta}^h - \tilde{\alpha}\Delta\Theta^h = 0, \qquad \text{on } (0, t_{\text{end}}] \times \Omega, \tag{5}$$

by using the semi-discrete states $(\Theta^h(t), \Upsilon^h(\nabla\Theta^h(t)))$ and, in particular, by replacing $\Theta^h(t)$ in the convection term by $\tilde{\Theta}^h(t)$ and $\alpha(t)$, which depends on $\Theta^h(t)$, by the reference heat conductivity $\tilde{\alpha}(t)$:

In order to get the matrices for Equation (4), we pose the semi-discrete variational formulations of Equation (5) together with its boundary conditions and Equation (2):

$$\begin{aligned} 0 &= \int_\Omega \dot{\Theta}^h \cdot \mathsf{v}^h \mathrm{d}x - \int_\Omega \Upsilon^h \cdot \nabla\tilde{\Theta}^h \cdot \mathsf{v}^h \mathrm{d}x + \int_\Omega \alpha\nabla\Theta^h \cdot \nabla\mathsf{v}^h \mathrm{d}x - \int_{\Gamma_u} k_l u^h \cdot \mathsf{v}^h \mathrm{d}s, \\ 0 &= -\int_\Omega \nabla\Upsilon^h \cdot \nabla\hat{\mathsf{v}}^h \mathrm{d}x + \int_{\Gamma_{\text{int}}} \Upsilon^h \cdot \hat{\mathsf{v}}^h \mathrm{d}s - \int_{\Gamma_{\text{int}}} \frac{1}{\ell}[k(\nabla\Theta^h)]_l^s \cdot \boldsymbol{n}_{\text{int}} \cdot \hat{\mathsf{v}}^h \mathrm{d}s. \end{aligned}$$

We denote the semi-discrete test functions with $\mathsf{v}^h(t)$ and $\hat{\mathsf{v}}^h(t)$ and reformulate these variational formulations into a matrix-based form:

$$\begin{aligned} \dot{\Theta}^h &= A_{\Theta\Theta}\Theta^h + A_{\Theta\Upsilon}\Upsilon^h + B_\Theta u^h, && \text{on } (0, t_{\text{end}}] \times \Omega, \\ 0 &= A_{\Upsilon\Theta}\Theta^h + A_{\Upsilon\Upsilon}\Upsilon^h, && \text{on } (0, t_{\text{end}}] \times \Omega. \end{aligned}$$

Here, the coefficient matrices are defined via the inner products

$$\begin{aligned}
\langle M_\Theta \Theta^h, \mathsf{v}^h \rangle &= \int_\Omega \Theta^h \cdot \mathsf{v}^h \,\mathrm{d}x, \\
\langle A_{\Theta\Theta} \Theta^h, \mathsf{v}^h \rangle &= -\int_\Omega \alpha \nabla \Theta^h \cdot \nabla \mathsf{v}^h \,\mathrm{d}x, \\
\langle A_{\Theta\Upsilon} \Upsilon^h, \mathsf{v}^h \rangle &= \int_\Omega \Upsilon^h \cdot \nabla \tilde{\Theta}^h \cdot \mathsf{v}^h \,\mathrm{d}x, \\
\langle A_{\Upsilon\Upsilon} \Upsilon^h, \hat{\mathsf{v}}^h \rangle &= \int_\Omega \nabla \Upsilon^h \cdot \nabla \hat{\mathsf{v}}^h \mathrm{d}x - \int_{\Gamma_{\mathrm{int}}} \Upsilon^h \cdot \hat{\mathsf{v}}^h \mathrm{d}s \\
\langle A_{\Upsilon\Theta} \Theta^h, \hat{\mathsf{v}}^h \rangle &= \int_{\Gamma_{\mathrm{int}}} \frac{1}{\ell} [k(\nabla \Theta^h)]_l^s \cdot \boldsymbol{n}_{\mathrm{int}} \cdot \hat{\mathsf{v}}^h \mathrm{d}s, \\
\langle B_\Theta u^h, \mathsf{v}^h \rangle &= \int_{\Gamma_u} k_l u^h \cdot \mathsf{v}^h \mathrm{d}s.
\end{aligned} \tag{6}$$

With these definitions, we can formulate the semi-discrete linearized Stefan problem in the format of Equation (4):

$$\begin{aligned}
\begin{bmatrix} M_\Theta & 0 \\ 0 & 0 \end{bmatrix} \frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} \Theta^h \\ \Upsilon^h \end{bmatrix} &= \begin{bmatrix} A_{\Theta\Theta} & A_{\Theta\Upsilon} \\ A_{\Upsilon\Theta} & A_{\Upsilon\Upsilon} \end{bmatrix} \begin{bmatrix} \Theta^h \\ \Upsilon^h \end{bmatrix} + \begin{bmatrix} B_\Theta \\ 0 \end{bmatrix} \mathbf{u}^h, \\
\mathbf{y}^h &= \begin{bmatrix} C_\Theta & 0 \end{bmatrix} \begin{bmatrix} \Theta^h \\ \Upsilon^h \end{bmatrix}.
\end{aligned} \tag{7}$$

Several choices are plausible for the output matrix $C_\Theta(t)$. We will introduce some in Section 5. With the zero-blocks in Equation (7), the DAE structure is clearly visible. Since, at every time instance t, $A_{\Upsilon\Upsilon}(t)$ represents the Poisson operator with a Dirichlet boundary part, it is always non-singular. Further, the mass matrix with respect to the temperature $M_\Theta(t)$ is symmetric positive definite and, in particular, always non-singular as well. Thus, Equation (7) is a DAE in semi-explicit form of differential index 1 (see e.g. [KM06]) and we can apply the implicit index-reduction techniques from [FRM08] for this type of DAEs.

Next, we discuss the treatment of the boundary conditions in the FEM matrices in Equations (6) and (7). This is of special importance for our objective to use these matrices to compute a feedback control for the Stefan problem in Section 4. The subject of attention in our feedback control problem is the position of $\Gamma_{\mathrm{int}}(t)$, which is directly linked to the boundary conditions (1d) and (2b) since they are defined on $\Gamma_{\mathrm{int}}(t)$. Equations (1d) and (2b) are also important for the coupling of Equations (1) and (2). This coupling is of particular importance for the feedback control problem since the interface movement is defined in the Stefan condition (2b) and the control is applied in Equation (1b). One common approach to handle Dirichlet boundary conditions, such as Equations (1d) and (2b), is to remove the rows and columns corresponding to the degrees of freedom (DOFs) at the related boundary regions from the FEM matrices. However, these DOFs are related to the coupling of $\Theta^h(t)$ and $\Upsilon^h(\nabla\Theta^h(t))$, and, thus, the coupling of the temperature and the interface movement. By removing these DOFs, the matrices would lose important information, which is necessary for the computation of a feedback control. Thus, the corresponding DOFs on $\Gamma_{\mathrm{int}}(t)$ are to be present in the FEM matrices. In order to treat the boundary conditions (1d) and (2b) appropriately for our feedback control problem we, from now on, consider the difference states

$$\begin{aligned}
\Theta_\Delta^h &= \Theta^h - \tilde{\Theta}^h, \\
\Upsilon_\Delta^h &= \Upsilon^h - \tilde{\Upsilon}^h.
\end{aligned}$$

We formulate our feedback control problem in terms of $\Theta^h_\Delta(t)$ and $\Upsilon^h_\Delta(\nabla\Theta^h(t))$. As a result, the desired state, which we stabilize with a feedback control, is the all-zero-state. The semi-discrete version of the Dirichlet condition in Equation (1d) in terms of the difference state $\Theta^h_\Delta(t)$ is equivalent to

$$0 = \Theta_{\Delta,\text{melt}} - \Theta^h_\Delta, \qquad \text{on } (0, t_\text{end}] \times \Gamma_\text{int}. \tag{8}$$

Further, both terms $\Theta_{\Delta,\text{melt}}(t) = \Theta_\text{melt}(t) - \tilde{\Theta}_\text{melt}(t) = 0$ and $\Theta^h_\Delta(t)\big|_{\Gamma_\text{int}(t)} = 0$ in Equation (8) are constant. Thus, also the time derivative equals zero, $\dot{\Theta}^h_\Delta(t) = 0$, and we add this equation to Equation (8). As a result, we can formulate the modified condition (9) to replace Equation (1d) in terms of the difference state for the FEM matrices, yielding

$$\dot{\Theta}^h_\Delta = \Theta_{\Delta,\text{melt}} - \Theta^h_\Delta, \qquad \text{on } (0, t_\text{end}] \times \Gamma_\text{int}. \tag{9}$$

To incorporate Equation (9) into the matrices, we denote $\Theta^h_{\Delta,\Gamma}(t) = \Theta^h_\Delta(t)\big|_{\Gamma_\text{int}(t)}$ and assume we can split

$$\Theta^h_\Delta = \begin{bmatrix} \bar{\Theta}^h_\Delta \\ \Theta^h_{\Delta,\Gamma} \end{bmatrix}.$$

In order to ensure the Dirichlet condition on $\Gamma_\text{int}(t)$, we add Equation (9) to the related block matrices, which then read

$$\tilde{M}_\Theta = \begin{bmatrix} M_\Theta & 0 \\ 0 & I \end{bmatrix}, \; \tilde{A}_{\Theta\Theta} = \begin{bmatrix} A_{\Theta\Theta} & A_{\Theta\Theta_\Gamma} \\ 0 & -I \end{bmatrix}, \; \tilde{A}_{\Theta\Upsilon} = \begin{bmatrix} A_{\Theta\Upsilon} \\ 0 \end{bmatrix}, \; \tilde{A}_{\Upsilon\Theta} = \begin{bmatrix} A_{\Upsilon\Theta} & A_{\Upsilon\Theta_\Gamma} \end{bmatrix}.$$

With this, the conditions in Equations (1b) and (2b) can be incorporated explicitly into the definition of $A_{\Upsilon\Theta}(t)$ and $B_\Theta(t)$. Regarding the remaining boundary conditions, the Neumann condition in Equation (1e) does not need any extra attention, since it is incorporated automatically. The Dirichlet boundary conditions in Equations (1c), (2c) and (2d) can be handled by the common approach to remove the rows and columns corresponding to the DOFs at the related boundary regions from the matrices.

Equation (7) with Equation (9) incorporated reads

$$\begin{aligned} \begin{bmatrix} \tilde{M}_\Theta & 0 \\ 0 & 0 \end{bmatrix} \frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} \Theta^h_\Delta \\ \Upsilon^h_\Delta \end{bmatrix} &= \begin{bmatrix} \tilde{A}_{\Theta\Theta} & \tilde{A}_{\Theta\Upsilon} \\ \tilde{A}_{\Upsilon\Theta} & A_{\Upsilon\Upsilon} \end{bmatrix} \begin{bmatrix} \Theta^h_\Delta \\ \Upsilon^h_\Delta \end{bmatrix} + \begin{bmatrix} B_\Theta \\ 0 \end{bmatrix} \mathbf{u}^h, \\ \mathbf{y}^h &= \begin{bmatrix} C_\Theta & 0 \end{bmatrix} \begin{bmatrix} \Theta^h_\Delta \\ \Upsilon^h_\Delta \end{bmatrix}. \end{aligned} \tag{10}$$

Notably, this modification preserves the DAE structure and index of Equation (7). Consequently, we follow [FRM08] and apply the Schur complement to remove the algebraic conditions. This yields the new system matrices for the Stefan problem

$$\begin{aligned} \mathcal{M}(t) &= \tilde{M}_\Theta(t), \\ \mathcal{A}(t) &= \tilde{A}_{\Theta\Theta}(t) - \tilde{A}_{\Theta\Upsilon}(t) A^{-1}_{\Upsilon\Upsilon}(t) \tilde{A}_{\Upsilon\Theta}(t), \\ \hat{\mathcal{B}}(t) &= B_\Theta(t), \\ \mathcal{C}(t) &= C_\Theta(t). \end{aligned} \tag{11}$$

With the matrices from Equation (11), we can transform the Stefan problem into the formulation of Equation (4) and are able to apply the LQR approach for the computation of a feedback control. In order to have a computationally efficient method, we keep the sparse structure of the matrices and never compute the generally dense Schur complement for $\mathcal{A}(t)$ explicitly. Instead we apply

$\mathcal{A}(t)$ implicitly and use the matrices from Equation (10). For details, see [FRM08], or the numerical implementation in [SKB21].

In Section 4, we describe the related numerical methods. For this, we further discretize the Stefan problem in time. For the simulation forward in time, we use the reference time-steps

$$\begin{aligned} 0 = t_0 < t_1 < \ldots < t_{n_t-1} < t_{n_t} = t_{\text{end}}, \\ \mathcal{T}_{\text{fwd}}^{\text{ref}} = \{t_0, t_1, \ldots, t_{n_t-1}, t_{n_t}\}. \end{aligned} \tag{12}$$

In Section 4, we solve differential Riccati equations backwards in time. For this we use the same time-steps $\hat{t}_k = t_{n_t-k}$ in reversed order

$$\begin{aligned} t_{\text{end}} = \hat{t}_0 > \hat{t}_1 > \ldots > \hat{t}_{n_t-1} > \hat{t}_{n_t} = t_0, \\ \mathcal{T}_{\text{bwd}} = \{\hat{t}_0, \hat{t}_1, \ldots, \hat{t}_{n_t-1}, \hat{t}_{n_t}\}. \end{aligned} \tag{13}$$

We use the same time-steps because the matrices from Equation (11), which form the coefficients of the differential Riccati equations, are assembled during the forward simulation. When we apply a time-adaptive method in the forward simulation, additional time-steps can be added to $\mathcal{T}_{\text{fwd}}^{\text{ref}}$.

4 Non-autonomous Linear-Quadratic Regulator

In this section, we formulate the Riccati-feedback approach for the Stefan problem. We focus on the non-autonomous character of this problem, which is induced by the moving interface and the consequently changing sub-domains. This results in time-dependent coefficients in Equation (4).

In order to derive a feedback-based stabilization of the Stefan problem, we use the LQR approach (e.g., [Son98]). We use this approach because it is well studied for related types of problems, e.g., convection diffusion equations [Wei16], and demonstrates promising performance for these.

To formulate the control problem, we define a quadratic cost functional $J(y^h, u_\Delta^h)$, tracking the deviation of the output from the desired output as well as penalizing the control costs with a weight factor $0 < \lambda \in \mathbb{R}$. The cost functional is thus defined as

$$J(y^h, u_\Delta^h) = \frac{1}{2} \int_0^{t_{\text{end}}} \left\| y^h - y_d^h \right\|^2 + \lambda \left\| u_\Delta^h \right\|^2 \, \mathrm{d}t. \tag{14}$$

We minimize this cost functional subject to the linear time-varying system (4). With the matrices from Equation (11) the LQR problem reads

$$\begin{aligned} &\min_{u_\Delta^h} J(y^h, u_\Delta^h) \\ &\text{subject to} \\ &\qquad \mathcal{M}\dot{x}_\Delta^h = \mathcal{A}x_\Delta^h + \hat{\mathcal{B}}u_\Delta^h, \\ &\qquad y^h = \mathcal{C}x_\Delta^h. \end{aligned} \tag{15}$$

The unique solution to the LQR problem (15) is (see, e.g., [Meh91])

$$u_\Delta^h = -\mathcal{K}x_\Delta^h, \tag{16}$$

where the feedback gain matrix

$$\mathcal{K} = \frac{1}{\lambda}\hat{\mathcal{B}}^\mathsf{T}\mathbf{X}\mathcal{M} = \frac{1}{\sqrt{\lambda}}\mathcal{B}^\mathsf{T}\mathbf{X}\mathcal{M} \tag{17}$$

requires the solution $\mathbf{X}(t) \in \mathbb{R}^{n\times n}$ of a differential Riccati equation (DRE). For the Stefan problem, this is the large-scale matrix-valued non-autonomous generalized DRE

$$-\frac{\mathrm{d}}{\mathrm{d}t}(\mathcal{M}^\mathsf{T}\mathbf{X}\mathcal{M}) = \mathcal{C}^\mathsf{T}\mathcal{C} + \mathcal{A}^\mathsf{T}\mathbf{X}\mathcal{M} + \mathcal{M}^\mathsf{T}\mathbf{X}\mathcal{A} - \mathcal{M}^\mathsf{T}\mathbf{X}\mathcal{B}\mathcal{B}^\mathsf{T}\mathbf{X}\mathcal{M}. \tag{18}$$

All coefficients of Equation (18) can be non-autonomous (but we skip the (t)-dependency for better readability). The coefficients of the DRE, at each time instance t, are the matrices $\mathcal{A}(t), \mathcal{M}(t)$,

Algorithm 1: Non-autonomous low-rank factor BDF method of order $\wp$

Input: $\mathcal{A}(t), \mathcal{M}(t), \dot{\mathcal{M}}(t), \mathcal{B}(t), \mathcal{C}(t), \lambda, \mathscr{T}_{\text{bwd}}, \wp, L_0, \ldots, L_{\wp-1}, D_0, \ldots, D_{\wp-1}$
Output: $\mathcal{K}_k, k = 1, \ldots, n_t$

1 **for** $k = \wp, \ldots, n_t$ **do**
2 $\quad \mathcal{A}_k = \tau_k \beta \mathcal{A}(\hat{t}_k) - \frac{1}{2}\mathcal{M}(\hat{t}_k)$
3 $\quad \mathcal{M}_k = \mathcal{M}(\hat{t}_k)$
4 $\quad \mathcal{B}_k = \sqrt{\tau\beta}\mathcal{B}(\hat{t}_k)$
5 $\quad \mathcal{C}_k^\mathsf{T} = \left[\mathcal{C}(\hat{t}_k)^\mathsf{T}, \mathcal{M}_k^\mathsf{T} L_{k-1}, \ldots, \mathcal{M}_k^\mathsf{T} L_{k-\wp}\right]$
6 $\quad \mathcal{S}_k = \begin{bmatrix} \tau\beta I_r & & & \\ & -\alpha_1 D_{k-1} & & \\ & & \ddots & \\ & & & -\alpha_\wp D_{k-\wp} \end{bmatrix}$
7 $\quad$ solve ARE (20) for L_k and D_k
8 $\quad \mathcal{K}_{n_t-k} = \frac{1}{\sqrt{\lambda}}\mathcal{B}(\hat{t}_k)^\mathsf{T} L_k D_k L_k^\mathsf{T} \mathcal{M}_k$
9 **end**

$\dot{\mathcal{M}}(t) \in \mathbb{R}^{n\times n}$, $\mathcal{B}(t) \in \mathbb{R}^{n\times m}$, and $\mathcal{C}(t) \in \mathbb{R}^{r\times n}$ from Equation (11), where the input matrix $\mathcal{B}(t) = \frac{1}{\sqrt{\lambda}}\hat{\mathcal{B}}(t)$ is scaled. In order to solve the DRE, the time-derivative requires special treatment due to the time-dependent mass matrix $\mathcal{M}(t)$. The left-hand side in Equation (18) can be computed applying the chain rule,

$$-\frac{\mathrm{d}}{\mathrm{d}t}(\mathcal{M}^\mathsf{T}\mathbf{X}\mathcal{M}) = -\dot{\mathcal{M}}^\mathsf{T}\mathbf{X}\mathcal{M} - \mathcal{M}^\mathsf{T}\dot{\mathbf{X}}\mathcal{M} - \mathcal{M}^\mathsf{T}\mathbf{X}\dot{\mathcal{M}}.$$

We subtract the two terms containing $\dot{\mathcal{M}}(t)$ from Equation (18) to obtain a DRE with the time-derivative of $\mathcal{M}(t)$ moved to the right-hand side:

$$-\mathcal{M}^\mathsf{T}\dot{\mathbf{X}}\mathcal{M} = \mathcal{C}^\mathsf{T}\mathcal{C} + (\dot{\mathcal{M}} + \mathcal{A})^\mathsf{T}\mathbf{X}\mathcal{M} + \mathcal{M}^\mathsf{T}\mathbf{X}(\dot{\mathcal{M}} + \mathcal{A}) - \mathcal{M}^\mathsf{T}\mathbf{X}\mathcal{B}\mathcal{B}^\mathsf{T}\mathbf{X}\mathcal{M}. \tag{19}$$

Besides the challenges arising from solving a large-scale matrix-valued DRE, the time-dependent matrices, especially the presence of $\dot{\mathcal{M}}(t)$, impose additional difficulties, which we address in this section.

To numerically compute the feedback gain matrix $\mathcal{K}_k = \mathcal{K}(t_k)$ and the feedback control $u_k = u_\Delta^h(t_k)$ for $t_k \in \mathscr{T}_{\text{bwd}}$, the solution of the DRE (19) is required. We use efficient low-rank methods for the computation of the numerical solution of the DRE, $\mathbf{X}_k = \mathbf{X}(t_k)$, since we assume that $\mathbf{X}_k$ has a low (numerical) rank motivated by [Sti18b]. For this, we use only a small number of inputs and outputs in our experiments, i.e. $m, r \ll n$. As introduced in [LMS15], which is motivated by [BLT09], $\mathbf{X}_k$ can, thus, be approximated to high accuracy by the decomposition

$$\mathbf{X}_k \approx L_k D_k L_k^\mathsf{T},$$

where the matrices $L_k \in \mathbb{R}^{n\times s}$ and $D_k \in \mathbb{R}^{s\times s}$ have rank $s \ll n$. We use the low-rank non-autonomous backward differentiation formula (BDF) (see Algorithm 1) to solve the non-autonomous DRE (19). This method is described in greater detail in [BBSS21, Section 2]. Compared to an open-loop control problem, the DRE replaces the adjoint equations in the LQR setting. Consequently, we solve the DRE backwards in time and have the time-steps $\mathscr{T}_{\text{bwd}}$ (see Equation (13)) in reversed order as an input to Algorithm 1. In each step of the BDF method, we solve an algebraic Riccati equation (ARE)

$$\begin{aligned} 0 = {} & \mathcal{C}_k^\mathsf{T}\mathcal{S}_k\mathcal{C}_k + (\dot{\mathcal{M}}_k + \mathcal{A}_k)^\mathsf{T}\mathbf{X}_k\mathcal{M}_k + \mathcal{M}_k^\mathsf{T}\mathbf{X}_k(\dot{\mathcal{M}}_k + \mathcal{A}_k) \\ & - \mathcal{M}_k^\mathsf{T}\mathbf{X}_k\mathcal{B}_k\mathcal{B}_k^\mathsf{T}\mathbf{X}_k\mathcal{M}_k \end{aligned} \tag{20}$$

for the low-rank solution $\mathbf{X}_k \approx L_k D_k L_k^\mathsf{T}$. The matrices in Equation (20) are constructed in Lines 2 to 6. The coefficients α_j and β are taken from the literature, e.g. [AP98]. The BDF

Algorithm 2: LQR for the Stefan problem

1 solve open-loop control problem to get the reference trajectory and $\mathcal{A}(t)$, $\mathcal{M}(t)$, $\dot{\mathcal{M}}(t)$, $\mathcal{B}(t)$, $\mathcal{C}(t)$ of the linearized problem
2 solve DRE (19) with Algorithm 1 to get $\mathcal{K}_k, k = 1, \ldots, n_t$
3 apply $\mathcal{K}_k$ in a forward simulation of the Stefan problem

method is a multistep method. Thus, to start a BDF method of order $\wp > 1$, the terminal values $\mathbf{X}_0, \ldots, \mathbf{X}_{\wp-1}$ are required with sufficient accuracy to obtain the desired order of convergence. These values can be computed with sufficiently small time-steps of the order $\wp - 1$ method. This procedure is repeated recursively for the order $\wp - 1$ method to compute terminal values for its start. For details see [BBSS21, Algorithm 2], or the implementation in [SKB21, mess_bdf_dre.m]. This method requires an additional input parameter that we set to $n_\wp = 10$ as in [BBSS21].

Algorithm 1 is embedded in the open source software package M-M.E.S.S. 2.1 [SKB21], where it benefits from efficient solvers for the ARE (20).

To have an overview of the three steps for the Riccati-feedback stabilization of the Stefan problem, we collect the overall procedure in Algorithm 2. The method we choose for the forward simulations of the Stefan problem in Lines 1 and 3 is a fractional-step-theta scheme and for solving the DRE in Line 2 the non-autonomous BDF methods, which we described above. The time discretization $\mathscr{T}_{\text{fwd}}^{\text{ref}}$, defined in Equation (12), is used in all three steps. In Line 2, it is used backwards in time ($\mathscr{T}_{\text{bwd}}$, Equation (13)) and, for the second forward solve in Line 3, additional time steps can be added adaptively. We use the time-adaptivity to prevent numerical instabilities that can occur with feedback controls that have very large variation, as demonstrated in [BBSS21]. The specific time-adaptivity is tailored to include the relative change of the feedback control. A detailed description of this time-adaptive fractional-step-theta scheme with a relative control-based indicator function is presented in [BBSS21, Section 3].

We provide the references for the codes of the methods above in Section 5 and showcase the behavior of Algorithms 1 and 2 by means of several numerical experiments.

5 Numerical Experiments

With several numerical experiments, we demonstrate that the LQR approach is applicable to this type of control problem, i.e., the Stefan problem, regardless of several numerical challenges and approximations that arise similar to other types of problems, where this approach is applied successfully. The Stefan problem is non-linear, which we address with a linearization. Further, the coefficients of the corresponding DRE are large-scale matrices. Thus, we use a low-rank representation of the DRE solution to make it feasible in regard of the computational cost and the memory requirements. Additionally, the Stefan problem is a DAE as well as, e.g., the Navier-Stokes equations, for which the LQR approach is applied successfully in [Wei16]. To handle the DAE structure, we use an implicit index reduction technique.

In addition, for the Stefan problem, the matrices are time-dependent and we approximate the time-derivative of the mass matrix by centered differences. Consequently, we use the non-autonomous BDF method to solve the DRE and compute the feedback gain matrices. Then, at intermediate time-steps that are generated by the time-adaptive fractional-step-theta scheme, we interpolate the feedback gain matrices. An alternative is to interpolate the DRE solution with the corresponding low-rank factors and compute the feedback gain matrices at intermediate time-steps as in Line 7 of Algorithm 1. However, this is more expensive and not necessary in our setting. All these techniques can introduce additional approximation errors. Still, the computed feedback controls can steer the interface back to the desired trajectory successfully in our experiments, similar to previously studied problem types.

With several numerical experiments, we illustrate the performance of the feedback stabilization, which is computed with Algorithm 2. The open-loop computations in Line 1 are reported in [BBHS18] and the assessment of the runtime performance of different methods for the solution of the DRE in Line 2 is left to [BBSS21]. Instead, we focus on the behavior of the feedback

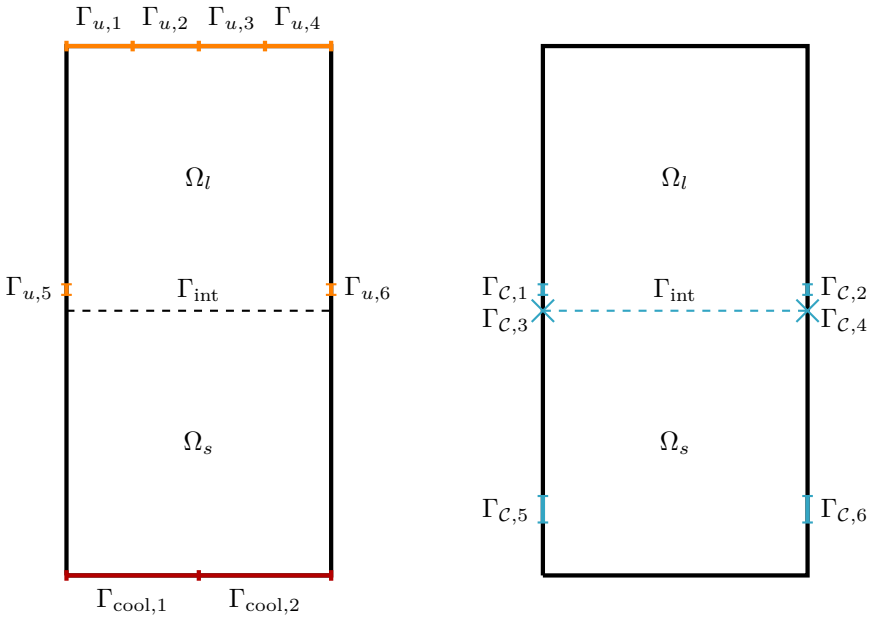

Figure 2: Input, Perturbation (left), and Output Areas (right)

t_{end}	k_s	k_l	Θ_{cool}	Θ_{melt}	ℓ	Θ_0	τ_k
1	6	10	-1	0	10	$\Theta_{\text{cool}} \cdot (1 - 2 \cdot x_2)$	$\in [10^{-4}, 2.5 \cdot 10^{-3}]$

Table 1: Stefan Problem Model Parameters

stabilization in Line 3. This behavior is strongly influenced by the weight factor λ in the cost functional as well as different choices of inputs and outputs, as the experiments demonstrate. In order to asses the robustness of our computed feedback controls, we are interested in uncertainties in the cooling, i.e., we investigate different perturbations $\varphi(t)$ to the Dirichlet boundary condition at $\Gamma_{\text{cool}}(t)$.

We depict several boundary regions for perturbations ($\Gamma_{\text{cool}}(t)$) as well as inputs ($\Gamma_u(t)$) and outputs ($\Gamma_{\mathcal{C}}(t)$) in Figure 2. As for $\Omega(t)$, the boundary regions for inputs $\Gamma_{u,1}(t), \ldots, \Gamma_{u,4}(t)$ are constant in this representation of the domain. However, we do not restrict ourselves to this case and, thus, keep the time-dependence in our notation.

The particular domain $\Omega(t)$, we choose for the experiments, is a rectangle $[0, 0.5] \times [0, 1]$ with the initial interface position at height 0.5 as a horizontal line. For the discretization, we choose a mesh of triangles with 3 899 vertices and 401 time-steps. This, using standard P1 elements in FEniCS, results in 3 899 DOFs for $\Theta(t)$ and 7 798 DOFs for $\Upsilon(\nabla\Theta(t))$. After the removal of the DOFs corresponding to Dirichlet boundary conditions (see Section 3), the size of the matrices is $n = 11\,429$. The model parameters are listed in Table 1.

Code Availability

All codes and data to reproduce the presented results are available at [Bar21]. The non-autonomous BDF method is incorporated in the software package M-M.E.S.S. 2.1 [SKB21]

5.1 Experiment 1

For this first experiment, we focus on the influence of weight parameters and outputs on the performance of the feedback control. The desired interface trajectory, which we intend to stabilize with the feedback control, is a flat horizontal line moving from its initial height at 0.5 downward by 0.004. Since we assume the Stefan problem to be asymptotically stable, the interface returns to the desired trajectory after a perturbation without a feedback control as well. However, this

is not achieved within the time horizon. With the feedback control, we intend to prevent the interface deviation to a certain extend and to steer the interface back in a much shorter time after a perturbation occurs.

The interface deviation is caused by the perturbation $\varphi(t)$, which is generated randomly in the form of three scalar values in the range of $[-\Theta_{\text{cool}}, \Theta_{\text{cool}}]$. These are applied for a period of four time-steps of $\mathscr{T}_{\text{fwd}}^{\text{ref}}$ at the times 0.1, 0.3, and 0.5 on $\Gamma_{\text{cool}}(t) = \Gamma_{\text{cool},1}(t) \cup \Gamma_{\text{cool},2}(t)$:

$$\Theta = \Theta_{\text{cool}} + \varphi, \qquad \text{on } (0, t_{\text{end}}] \times \Gamma_{\text{cool}}.$$

The trajectory of $\varphi(t)$ is pictured in the top part of Figure 3. The scaling of the y-axis is relative to Θ_{cool}.

Here, for the LQR problem, we use a single input $u_{\mathcal{K}}(t)$ with the same value on $\Gamma_{u,1}(t) \cup \ldots \cup \Gamma_{u,4}(t)$, i.e. $\mathcal{B}(t) \in \mathbb{R}^{n\times 1}$. The top part of Figure 3 shows the control for three different LQR designs, resulting from three different combinations of weights and outputs:

$$\begin{aligned} u_1 &: \quad \lambda = 10^{-4}, \quad && 2 \text{ outputs: } \Gamma_{\mathcal{C},3}, \Gamma_{\mathcal{C},4}, \\ u_2 &: \quad \lambda = 10^{-6}, \quad && 2 \text{ outputs: } \Gamma_{\mathcal{C},3}, \Gamma_{\mathcal{C},4}, \\ u_3 &: \quad \lambda = 1.6 \cdot 10^{-2}, \quad && 7 \text{ outputs: } \Gamma_{\mathcal{C},1}, \ldots, \Gamma_{\mathcal{C},6}, \Gamma_{\text{int}}. \end{aligned}$$

The outputs on $\Gamma_{\mathcal{C},1}(t)$, $\Gamma_{\mathcal{C},2}(t)$, $\Gamma_{\mathcal{C},5}(t)$, and $\Gamma_{\mathcal{C},6}(t)$ measure averaged temperatures on the corresponding interval, while $\Gamma_{\mathcal{C},3}(t)$ and $\Gamma_{\mathcal{C},4}(t)$ represent point measurements of the temperature. The output at $\Gamma_{\text{int}}(t)$ monitors the difference of the interface movement to the desired movement

$$\langle \mathcal{C}_{\text{int}}, \mathsf{v}^h \rangle = \int_{\Gamma_{\text{int}}} \left(\frac{1}{\ell} [k_s (\nabla \Theta_\Delta)_s - k_l (\nabla \Theta_\Delta)_l] \right) \cdot \boldsymbol{n}_{\text{int}} \cdot \mathsf{v}^h \mathrm{d}s$$

and is defined via the jump term of the Stefan condition (Equation (2b)). This output alone is not suited for an effective control-design. It would generate a large output, and thus an active feedback-response, only while the perturbation is actively driving the interface away from the desired trajectory. However, it can not detect a difference in the position of the interface. Consequently, it would not steer the interface back but would keep it on a "parallel trajectory".

Thus, in the first two LQR designs, the controls $u_1(t)$ and $u_2(t)$ are based on two outputs that measure the temperature at the desired interface position on the boundary, such that $\mathcal{C}(t) \in \mathbb{R}^{2\times n}$, while $u_3(t)$, in the third setting, uses seven outputs, i.e. $\mathcal{C}(t) \in \mathbb{R}^{7\times n}$. With this, we compute the feedback gain matrices according to Section 4 with Algorithm 1 and set $\wp = 1$. Then, we simulate the closed-loop system with these feedback gain matrices together with the perturbations. The resulting feedback controls $u_1(t)$, $u_2(t)$, and $u_3(t)$ are displayed in the top part of Figure 3 and the interface positions in the middle part of Figure 3. The interface positions are relative to the desired interface position at the point $x^*(t)$ with the largest deviation on the interface:

$$\begin{aligned} \Gamma_{\text{int},\Delta}(t, x^*) &= \Gamma_{\text{int,ref}}(t, x^*) - \Gamma_{\text{int}}(t, x^*), \\ x^* &= \operatorname*{argmax}_{x_1 \in [0,0.5]} \left| \Gamma_{\text{int,ref}}(t, x_1) - \Gamma_{\text{int}}(t, x_1) \right|. \end{aligned}$$

The feedback control $u_1(t)$ is most active shortly after the perturbation starts and it steers the interface back to the desired position in much reduced time, as expected from theory. Again as expected, a smaller weight factor λ allows the control $u_2(t)$ to react with a larger magnitude input while the perturbation is active. It is, therefore, able to stop the interface from deviating earlier and drives it back even faster.

It should be mentioned that smaller weights reduce the contribution of the control cost term in the cost functional (14). This term can also be interpreted as a regularization term. Consequently, smaller weights decrease the regularity of the feedback control problem. We note that this results in larger computational cost for the solution of the DRE due to slower convergence of the internal iterative solvers used in the BDF method, but refer to [BBSS21, Section 4.2] for a more detailed discussion of this issue.

The weights for the LQR settings of $u_1(t)$ and $u_2(t)$ are not directly comparable to the setting for $u_3(t)$, since this is based on different outputs. The outputs at $\Gamma_{\mathcal{C},5}(t)$ and $\Gamma_{\mathcal{C},6}(t)$ allow it to

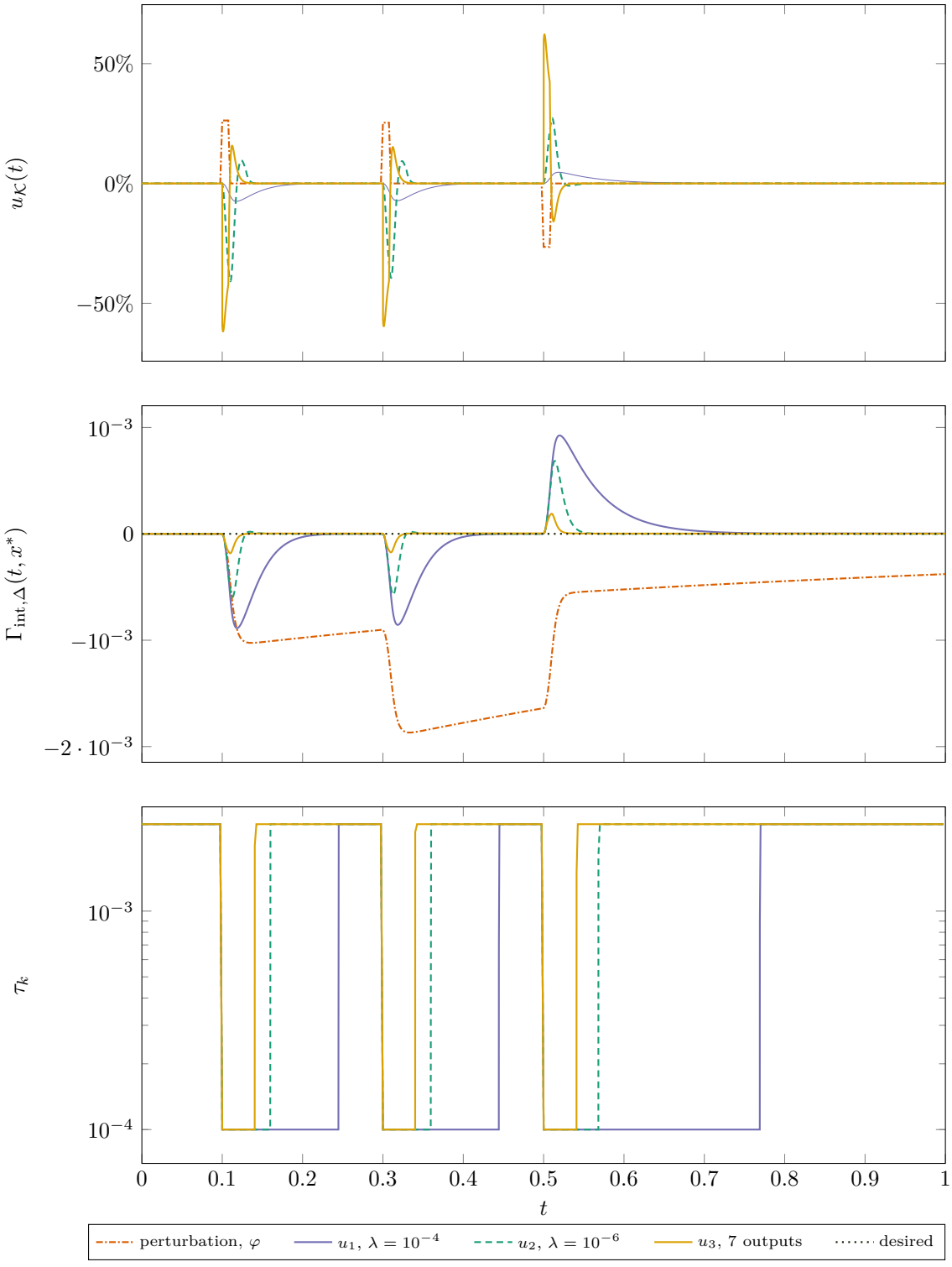

Figure 3: Perturbation and feedback (top), relative interface position (middle), and time-step sizes (bottom) for different weights and outputs (Experiment 1)

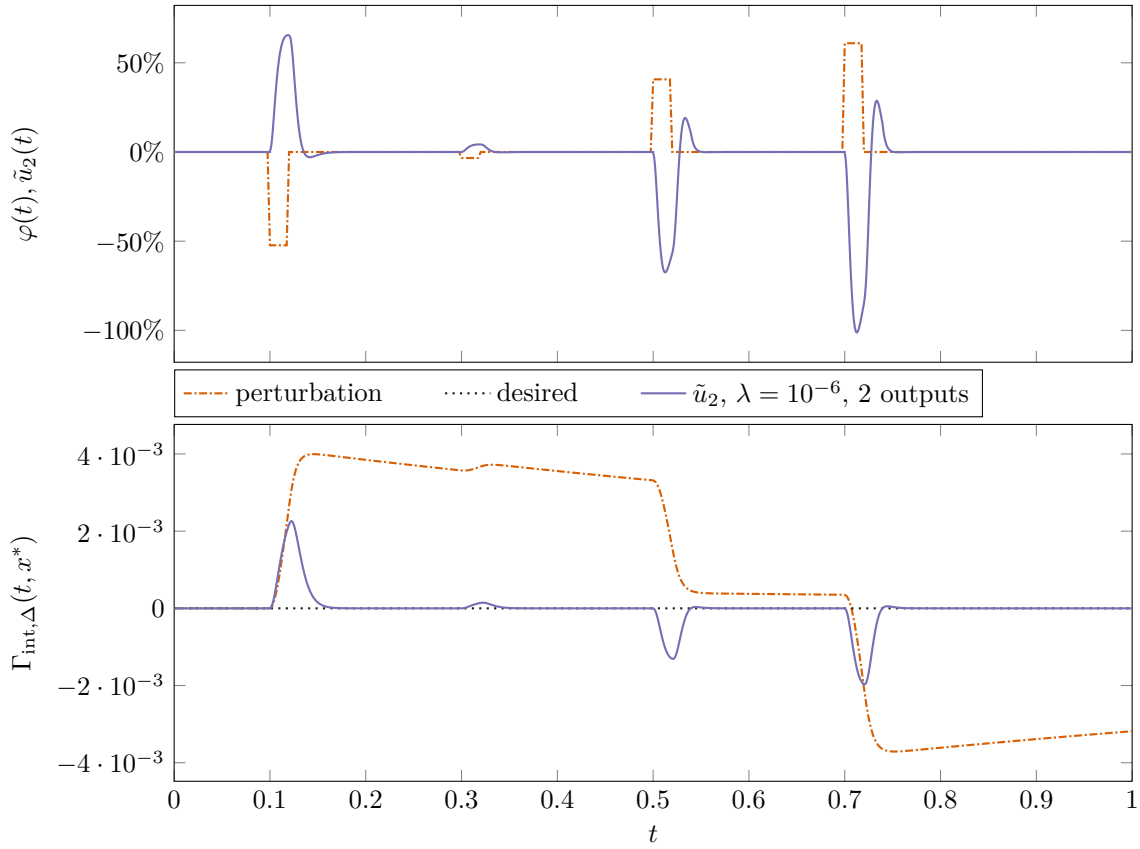

Figure 4: Perturbation and feedback (top) and relative interface position (bottom) for an interface moving upwards (Experiment 2)

detect the temperature perturbation earlier and the output that monitors the movement of $\Gamma_{\text{int}}(t)$ can observe the deviation of the interface earlier. Thus, $u_3(t)$ is most active immediately after the perturbation starts and moves the interface back much faster.

Note, that we use a time-adaptive fractional-step-theta scheme with a relative feedback control-based indicator function to simulate the Stefan problem together with a feedback control. For this example, the bottom part of Figure 3 shows the adaptive time-step sizes. While the feedback control is inactive, like at the beginning or end of the simulation, the method chooses the largest time-step size of $\tau_k = 0.0025$. As soon as the feedback control is active, the method chooses the minimal time-step size of $\tau_k = 10^{-4}$. This prevents numerical instabilities that can occur with very small λ and thus very active feedback controls as is demonstrated in [BBSS21, Section 4.2].

Experiment 1 demonstrates the influence of the chosen weight parameter λ in the cost functional (Equation (14)). Smaller weight parameters increase the impact of the deviation term in the cost functional indirectly by decreasing the relevance of the control costs which are measured by the second term and scaled by λ. Further, the choice and number of outputs has a significant impact on the performance of the feedback control.

The next experiment demonstrates that our method works as well with increased time-dependence of the data in the non-autonomous DRE.

5.2 Experiment 2

In our second experiment, the desired interface trajectory is, again, a flat horizontal line moving from its initial height at 0.5, this time, upward by 0.1. This is a longer distance interface movement than in Experiment 1 on the same time horizon and, in turn, results in a stronger time-dependence of the matrices for the DRE. The investigated LQR design is chosen as in the second case in Experiment 1.

That means, we use the same weight ($\lambda = 10^{-6}$) and outputs as for $u_2(t)$ above and call the corresponding feedback control $\tilde{u}_2(t)$. This time we use four randomly generated perturbations applied at the times $0.1, 0.3, 0.5$, and 0.7, which, together with $\tilde{u}_2(t)$, can be found in the top part of Figure 4. Again, the perturbations cause the interface to deviate from the desired trajectory. The feedback control $\tilde{u}_2(t)$ behaves similarly to the previous experiment. It stops the interface from deviating and drives it back to the desired position, as expected from the theory. Additionally, the time-adaptivity behaves analogously to Experiment 1.

The experiment showcases the performance of our method with strongly time-varying coefficients in the DRE. We implemented more experiments with desired interface trajectories that move considerably. Our feedback control showed the same performance for all of them. This demonstrates that our solver can also cope with the stronger time dependence without difficulties.

5.3 Experiment 3

Our third experiment revisits the basic task as in Experiment 1. This time, we generate two different perturbations $\varphi_1(t), \varphi_2(t) \in [-\Theta_{\text{cool}}, \Theta_{\text{cool}}]$, acting on the left ($\Gamma_{\text{cool},1}(t)$) and right ($\Gamma_{\text{cool},2}(t)$) part of the Dirichlet boundary at the bottom of $\Omega(t)$ (see Figure 2), with three random values each, which are applied at the times $0.1, 0.4325$, and 0.765. They are displayed in the top part of Figure 5. We apply these two perturbation functions to

$$\begin{aligned} \Theta &= \Theta_{\text{cool}} + \varphi_1, \qquad \text{on } (0, t_{\text{end}}] \times \Gamma_{\text{cool},1}, \\ \Theta &= \Theta_{\text{cool}} + \varphi_2, \qquad \text{on } (0, t_{\text{end}}] \times \Gamma_{\text{cool},2}. \end{aligned}$$

These perturbations will not only move the interface away from the desired position but also add some curvature to the deviated interface. To stabilize this interface position, we use six individual inputs at $\Gamma_{u,1}(t), \dots, \Gamma_{u,6}(t)$ ($\mathcal{B}(t) \in \mathbb{R}^{n \times 6}$) and the same two outputs at $\Gamma_{\mathcal{C},3}(t)$, $\Gamma_{\mathcal{C},4}(t)$ as in the previous experiments ($\mathcal{C}(t) \in \mathbb{R}^{2 \times n}$). For the weight factor in the cost functional we use $\lambda = 10^{-9}$. Again, this weight factor is not directly comparable to the previous experiments since we use a different combination of inputs and outputs.

The resulting perturbed and feedback controlled interfaces are displayed in the bottom part of Figure 5 at the time points $t \in [0.465, 0.56, 0.798, 1]$. With the first two perturbations, the interface is pushed downwards and assumes a distinct curvature. Right after the second perturbation ($t = 0.465$), the feedback controlled interface is already essentially back to the desired interface, but still has undesired curvature. Well before the third perturbation, at $t = 0.56$, the feedback controlled interface is almost flat again, as the desired interface. At the two endpoints of the interface, where the outputs are located, the controlled interface is directly at the desired position. The remaining curvature can not be measured by the cost functional and is, thus, not actively removed. For the given setup, this is the expected behavior of the feedback control. It is similar after the first perturbation. The third perturbation moves the interface upwards above the desired position ($t = 0.798$). At the end of the time interval ($t = 1$), the feedback controlled interface is again almost flat and back to the desired position at the ends while the uncontrolled interface is still far away.

This experiment shows that the feedback control can move the perturbed interface back to the desired trajectory and additionally control the curvature of the interface, to a certain extent. With the chosen input and output setting, the position of the interface is corrected as fast as in the previous experiments. The curvature correction requires some more time but is still performed during the considered time interval up to the level that the cost functional can measure with the given two outputs.

In all three experiments, we are able to stabilize the desired interface position with the proposed feedback control approach despite the different challenges that are posed by the Stefan problem.

6 Conclusions

In this work, we address the challenging task to derive, compute, and apply a feedback control for the two-dimensional two-phase Stefan problem. The particular challenges for the Stefan problem

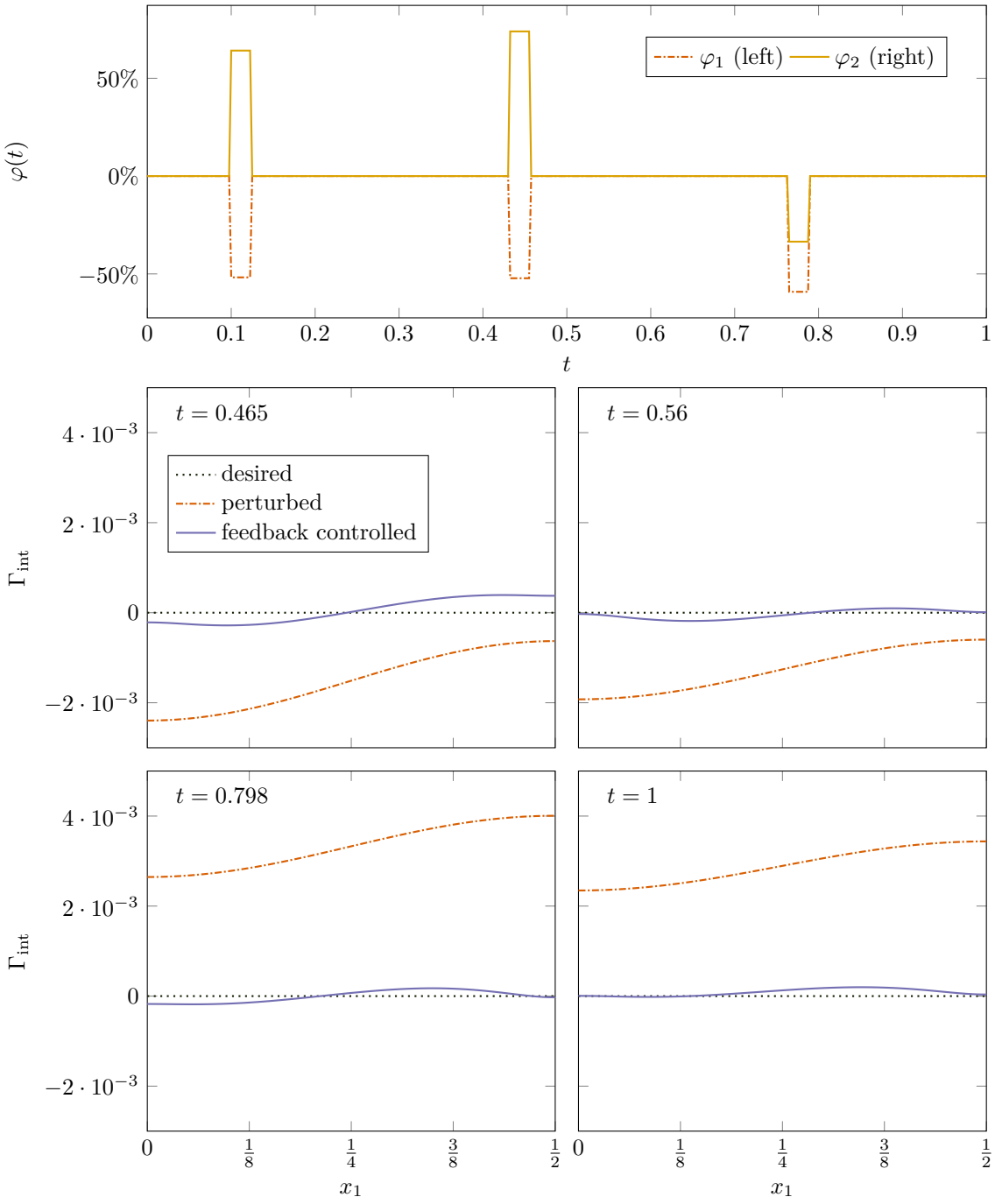

Figure 5: Two different perturbations (top), and perturbed and feedback controlled relative interface positions (bottom) (Experiment 3)

lie in the non-linearities, the DAE structure, as well as the moving inner boundary. Additionally, the applied feedback control approach results in a generalized DRE with time-dependent large-scale matrices. We address this task with a sharp interface representation and mesh movement techniques. Resulting from this coupling of the Stefan problem with mesh movement, we have a detailed description of the linearization, discretization and matrix assembly. Regarding this, we particularly elaborate on how Dirichlet boundary conditions can be treated and display the DAE structure in the resulting matrices.

To obtain a feedback control for the Stefan problem, we apply the LQR approach and treat large-scale non-autonomous DREs with a corresponding low-rank BDF method. More specifically, we also include the time-dependent mass matrix and its derivative. The feedback control resulting from this approach is applied in a forward simulation of the closed-loop Stefan problem where we handle the numerical difficulties that arise with a time-adaptive fractional-step-theta scheme specifically adapted to this process.

Through several numerical experiments, we demonstrate how effectively our methods and the resulting feedback controls perform. The performance of the feedback controls strongly depends on the choice of control parameters like the weight factor in the cost functional and the selected inputs and outputs. The outputs are particularly important since in our formulation of the Stefan problem, the interface position to be controlled, is not explicitly available as an output. Thus, the outputs need to indicate the interface deviation reliably.

Our methods include several stages of approximations, like the linearization, implicit index reduction, and low-rank representation of the DRE solution in an iterative solver. Like this, we are able to make the numerical solution of this large-scale problem feasible while the feedback controls still perform as known from problem types, that are already well studied in this regard.

Future research could investigate other promising methods for the non-autonomous DRE like, e.g., splitting schemes. Additionally, a quantitative error analysis is still to be done in future work.

References

[AKFIJ03] H. Abou-Kandil, G. Freiling, V. Ionescu, and G. Jank. *Matrix Riccati Equations in Control and Systems Theory*. Systems & Control: Foundations & Applications. Birkhäuser, Basel, Switzerland, 2003. doi:10.1007/978-3-0348-8081-7.

[ANS14] H. Antil, R. H. Nochetto, and P. Sodré. Optimal control of a free boundary problem: Analysis with second-order sufficient conditions. *SIAM J. Control Optim.*, 52(5):2771–2799, 2014. doi:10.1137/120893306.

[ANS15] H. Antil, R. H. Nochetto, and P. Sodré. Optimal control of a free boundary problem with surface tension effects: A priori error analysis. *SIAM J. Numer. Anal.*, 53(5):2279–2306, 2015. doi:10.1137/140958360.

[AP98] U. M. Ascher and L. R. Petzold. *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. SIAM, Philadelphia, 1998. doi:10.1137/1.9781611971392.

[Bar21] B. Baran. Linear Quadratic Regulator Computation for a Stefan Problem using M.-M.E.S.S. and FEniCS, April 2021. doi:10.5281/zenodo.4671886.

[BBH21] M. Behr, P. Benner, and J. Heiland. Invariant Galerkin trial spaces and Davison-Maki methods for the numerical solution of differential Riccati equations. *Applied Mathematics and Computation*, 410:126401, 2021. doi:10.1016/j.amc.2021.126401.

[BBHS18] B. Baran, P. Benner, J. Heiland, and J. Saak. Optimal control of a Stefan problem fully coupled with incompressible Navier–Stokes equations and mesh movement. *Analele Stiintifice ale Universitatii Ovidius Constanta: Seria Matematica*, XXVI(2):11–40, August 2018. doi:10.2478/auom-2018-0016.

[BBSS21] B. Baran, P. Benner, J. Saak, and T. Stillfjord. Numerical methods for closed-loop systems with non-autonomous data. e-print arXiv:X, arXiv, 2021. math.NA. URL: https://arxiv.org/X.

[Ber10] M. Bernauer. *Motion Planning for the Two-Phase Stefan Problem in Level Set Formulation*. PhD thesis, Technische Universität Chemnitz, Chemnitz, Germany, 2010. URL: http://nbn-resolving.de/urn:nbn:de:bsz:ch1-qucosa-63654.

[BG91] R. R. Bitmead and M. Gevers. *Riccati Difference and Differential Equations: Convergence, Monotonicity and Stability*, pages 263–291. Communications and Control Engineering. Springer Berlin Heidelberg, Berlin, Heidelberg, 1991. doi:10.1007/978-3-642-58223-3_10.

[BL18] P. Benner and N. Lang. Peer methods for the solution of large-scale differential matrix eqautions. e-print arXiv:1804.08524, arXiv, July 2018. math.NA. URL: https://arxiv.org/pdf/1804.08524.pdf.

[BLT09] P. Benner, R.-C. Li, and N. Truhar. On the ADI method for Sylvester equations. *J. Comput. Appl. Math.*, 233(4):1035–1045, 2009. doi:10.1016/j.cam.2009.08.108.

[BM04] P. Benner and H. Mena. BDF methods for large-scale differential Riccati equations. In B. De Moor, B. Motmans, J. Willems, P. Van Dooren, and V. Blondel, editors, *Proc. 16th Intl. Symp. Mathematical Theory of Network and Systems, MTNS 2004*, 2004.

[BM18] P. Benner and H. Mena. Numerical solution of the infinite-dimensional LQR-problem and the associated differential Riccati equations. *J. Numer. Math.*, 26(1):1–20, March 2018. published online May 2016. doi:10.1515/jnma-2016-1039.

[BPS10] E. Bänsch, J. Paul, and A. Schmidt. An ALE FEM for solid-liquid phase transitions with free melt surface. Technical report, Zentrum für Technomathematik, University of Bremen, 2010. URL: http://www.math.uni-bremen.de/zetem/cms/media.php/262/report1007.pdf.

[BPS13] E. Bänsch, J. Paul, and A. Schmidt. An ALE finite element method for a coupled Stefan problem and Navier–Stokes equations with free capillary surface. *Internat. J. Numer. Methods Fluids*, 71(10):1282–1296, 2013. doi:10.1002/fld.3711.

[FRM08] F. Freitas, J. Rommes, and N. Martins. Gramian-based reduction method applied to large sparse power system descriptor models. *IEEE Trans. Power Syst.*, 23(3):1258–1270, August 2008. doi:10.1109/TPWRS.2008.926693.

[GHJK18] Y. Güldoğan, M. Hached, K. Jbilou, and M. Kurulay. Low rank approximate solutions to large-scale differential matrix Riccati equations. *Applicationes Mathematicae*, 45(2):233–254, 2018. doi:10.4064/am2355-1-2018.

[Gup18] S. C. Gupta. *The Classical Stefan Problem (Second Edition)*. Elsevier, Amsterdam, 2018. doi:10.1016/B978-0-444-63581-5.09985-1.

[KK19] S. Koga and M. Krstic. Control of two-phase Stefan problem via single boundary heat input. In *Proceedings of the IEEE Conference on Decision and Control*, pages 2914–2919, 2019. doi:10.1109/CDC.2018.8619638.

[KK20a] S. Koga and M. Krstic. Single-boundary control of the two-phase Stefan system. *Systems Control Lett.*, 135:104573, 2020. doi:10.1016/j.sysconle.2019.104573.

[KK20b] S. Koga and M. Krstic. *Two-Phase Stefan Problem*, pages 139–157. Springer International Publishing, Cham, 2020. doi:10.1007/978-3-030-58490-0_5.

[KM90a] P. Kunkel and V. Mehrmann. Numerical solution of differential algebraic Riccati equations. *Linear Algebra Appl.*, 137/138:39–66, 1990. doi:10.1016/0024-3795(90)90126-W.

[KM90b] P. Kunkel and V. Mehrmann. Numerical solution of Riccati differential algebraic equations. In M. A. Kaashoek et al, editor, *Proceedings of the International Symposium on the Mathematical Theory of Networks and Systems, Amsterdam, Netherlands, June 1989*, pages 479–487, Basel, 1990. Birkhäuser.

[KM06] P. Kunkel and V. Mehrmann. *Differential-Algebraic Equations: Analysis and Numerical Solution*. Textbooks in Mathematics. EMS Publishing House, Zürich, Switzerland, 2006.

[KM20] A. Koskela and H. Mena. Analysis of Krylov subspace approximation to large-scale differential Riccati equations. *Electron. Trans. Numer. Anal.*, 52:431–454, 2020. doi:10.1553/etna_vol52s431.

[KS20] G. Kirsten and V. Simoncini. Order reduction methods for solving large-scale differential matrix Riccati equations. *SIAM J. Sci. Comput.*, 42(4):A2182–A2205, 2020. doi:10.1137/19M1264217.

[Lan17] N. Lang. *Numerical Methods for Large-Scale Linear Time-Varying Control Systems and related Differential Matrix Equations*. Dissertation, Technische Universität Chemnitz, Germany, June 2017. Logos-Verlag, Berlin, ISBN 978-3-8325-4700-4. URL: https://www.logos-verlag.de/cgi-bin/buch/isbn/4700.

[LC31] G. Lamé and B. P. Clapeyron. Mémoire sur la solidification par refroidissement d'un globe liquide. In *Annales Chimie Physique*, volume 47, pages 250–256, 1831.

[LMS15] N. Lang, H. Mena, and J. Saak. On the benefits of the LDL^T factorization for large-scale differential matrix equation solvers. *Linear Algebra Appl.*, 480:44–71, 2015. doi:10.1016/j.laa.2015.04.006.

[LWH12] A. Logg, G. N. Wells, and J. Hake. *DOLFIN: a C++/Python Finite Element Library*, chapter 10. Springer, 2012. doi:10.1145/1731022.1731030.

[LZL20] D. Li, X. Zhang, and R. Liu. Exponential integrators for large-scale stiff Riccati differential equations. *J. Comput. Appl. Math.*, 389:113360, 2020. doi:10.1016/j.cam.2020.113360.

[Meh91] V. Mehrmann. *The Autonomous Linear Quadratic Control Problem, Theory and Numerical Solution*. Number 163 in Lecture Notes in Control and Information Sciences. Heidelberg, July 1991.

[Men12] H. Mena. *Numerical Solution of Differential Riccati Equations Arising in Optimal Control Problems for Parabolic Partial Differential Equations*. Unidad de Publicaciones de la Facultad de Ciencias, Quito-Ecuador, first edition, May 2012. Available as ISBN: 978-9978-383-09-4.

[MOPP18] H. Mena, A. Ostermann, L.-M. Pfurtscheller, and C. Piazzola. Numerical low-rank approximation of matrix differential equations. *J. Comput. Appl. Math.*, 340:602–614, 2018. doi:10.1016/j.cam.2018.01.035.

[NCM11] M. Niezgodka, A. Crowley, and A. M. Meirmanov. *The Stefan Problem*. De Gruyter, 2011. doi:10.1515/9783110846720.245.

[OPW18] A. Ostermann, C. Piazzola, and H. Walach. Convergence of a low-rank Lie–Trotter splitting for stiff matrix differential equations. e-print arXiv:1803.10473, arXiv, March 2018. math.NA. URL: https://arxiv.org/abs/1803.10473.

[Rei72] W. T. Reid. *Riccati Differential Equations*, volume 86 of *Mathematics in Science and Engineering*. Academic Press, New York, 1972.

[Rub71] L. I. Rubenšteĭn. *The Stefan problem*, volume 27 of *Translations of Mathematical Monographs*. American Mathematical Society, Providence, R.I., 1971. Translated from the Russian by A. D. Solomon. doi:10.1090/mmono/027.

[SKB21] J. Saak, M. Köhler, and P. Benner. M-M.E.S.S.-2.1 – The Matrix Equations Sparse Solvers library, April 2021. see also: https://www.mpi-magdeburg.mpg.de/projects/mess. doi:10.5281/zenodo.4719688.

[Son98] E. D. Sontag. *Mathematical Control Theory*. Texts in Applied Mathematics. Springer-Verlag, New York, NY, 2nd edition, 1998. doi:10.1007/978-1-4612-0577-7.

[Ste89] J. Stefan. Über einige Probleme der Theorie der Wärmeleitung. *Sitzungber., Wien, Akad. Mat. Natur*, 98:473–484, 1889.

[Ste90] J. Stefan. Über die Theorie der Eisbildung. *Monatshefte für Mathematik*, 1(1):1–6, 1890.

[Ste91] J. Stefan. Über die Theorie der Eisbildung, insbesondere über die Eisbildung im Polarmeere. *Annalen der Physik und Chemie*, 42:269–286, 1891.

[Sti15a] T. Stillfjord. Low-rank second-order splitting of large-scale differential Riccati equations. *IEEE Trans. Autom. Control*, 60(10):2791–2796, 2015. doi:10.1109/TAC.2015.2398889.

[Sti15b] T. Stillfjord. *Splitting schemes for nonlinear parabolic problems*. PhD thesis, Lund University, 2015. URL: https://lup.lub.lu.se/search/ws/files/3905802/5277358.pdf.

[Sti18a] T. Stillfjord. Adaptive high-order splitting schemes for large-scale differential Riccati equations. *Numer. Algorithms*, 78:1129–1151, 2018. doi:10.1007/s11075-017-0416-8.

[Sti18b] T. Stillfjord. Singular value decay of operator-valued differential Lyapunov and Riccati equations. *SIAM J. Control Optim.*, 56:3598–3618, 2018. doi:10.1137/18M1178815.

[Wei16] H. K. Weichelt. *Numerical Aspects of Flow Stabilization by Riccati Feedback*. Dissertation, Otto-von-Guericke-Universität, Magdeburg, Germany, January 2016. URL: http://nbn-resolving.de/urn:nbn:de:gbv:ma9:1-8693.

[Zie08] S. Ziegenbalg. *Kontrolle freier Ränder bei der Erstarrung von Kristallschmelzen*. PhD thesis, Technische Universität Dresden, Dresden, Germany, 2008. In German. URL: http://nbn-resolving.de/urn:nbn:de:bsz:14-ds-1212521184972-55836.