

A NUMERICAL APPROACH TO THE OPTIMAL CONTROL OF THERMALLY CONVECTIVE FLOWS*

YONGCUN SONG[†], XIAOMING YUAN[‡], AND HANGRUI YUE[§]

*Dedicated to the memory of Roland Glowinski (1937–2022): a dear friend and great mentor.*¹

Abstract. The optimal control of thermally convective flows is usually modeled by an optimization problem with constraints of Boussinesq equations that consist of the Navier-Stokes equation and an advection-diffusion equation. This optimal control problem is challenging from both theoretical analysis and algorithmic design perspectives. For example, the nonlinearity and coupling of fluid flows and energy transports prevent direct applications of gradient type algorithms in practice. In this paper, we propose an efficient numerical method to solve this problem based on the operator splitting and optimization techniques. In particular, we employ the Marchuk-Yanenko method leveraged by the L^2 -projection for the time discretization of the Boussinesq equations so that the Boussinesq equations are decomposed into some easier linear equations without any difficulty in deriving the corresponding adjoint system. Consequently, at each iteration, four easy linear advection-diffusion equations and two degenerated Stokes equations at each time step are needed to be solved for computing a gradient. Then, we apply the Bercovier-Pironneau finite element method for space discretization, and design a BFGS type algorithm for solving the fully discretized optimal control problem. We look into the structure of the problem, and design a meticulous strategy to seek step sizes for the BFGS efficiently. Efficiency of the numerical approach is promisingly validated by the results of some preliminary numerical experiments.

Key words. Optimal control, thermally convective flows, Boussinesq equations, operator-splitting method, BFGS.

AMS subject classifications. 49M41, 35Q93, 35Q90, 65K05

1. Introduction. Thermally convective flows arise in many applications such as thermal insulation, cooling of fluids in channels surrounding nuclear reactor core, the circulation of liquid metals in solidifying ingot, the manufacture of crystals, and modeling, designing, and controlling energy-efficient building systems, see e.g., [2, 22, 42]. Such flows are typically modeled by the Boussinesq equations, which consist of the Navier-Stokes equation for incompressible viscous flow coupled with an advection-diffusion equation for temperature. In practice, thermally convective flows also play a crucial role in the control of different physical processes. In view of this, various controls of thermally convective flows have been widely studied from different perspectives in the past decades. For instance, the optimal control of temperature peaks along the bounding surfaces of containers of fluid flows has been studied in [27]. In [13], feedback control for thermal fluid was considered, and in [31] the optimal control of flow separation in a channel flow using temperature control was investigated. Linear quadratic regulator control and pointwise control of the Boussinesq equations were studied in [11] and [37], respectively. More related literature on the control of thermally convective flows can be referred to [1, 2, 3, 5, 6, 31, 14, 32, 33, 34, 41, 42] and references therein.

*28 November, 2022

Funding: The work of the first author was supported by the Humboldt Research Fellowship for postdoctoral researchers. The work of the second author was supported by a seed fund for basic research at The University of Hong Kong. The work of the third author was supported by the Fundamental Research Funds for the Central Universities, Nankai University (Grant Number 63221035)

[†]Chair for Dynamics, Control and Numerics, Alexander von Humboldt-Professorship, Department of Data Science, Friedrich-Alexander-Universität Erlangen-Nürnberg, 91058 Erlangen, Germany. ysong307@gmail.com.

[‡]Department of Mathematics, The University of Hong Kong, Hong Kong, China. xmyuan@hku.hk

[§]School of Mathematical Sciences, Nankai University, Tianjin, China. yuehangrui@gmail.com

¹The authors were encouraged by Roland Glowinski to consider this work when he was visiting Hong Kong in 2019.

1.1. Problem statement and motivation. The class of thermally convective flow under our consideration is modeled by the following non-dimensional Boussinesq equations:

$$(1.1) \quad \begin{cases} \frac{\partial y}{\partial t} + (y \cdot \nabla)y - \nu_1 \Delta y + \nabla p = \theta \mathbf{e}_2 & \text{in } Q, \\ \nabla \cdot y = 0 & \text{in } Q, \\ \frac{\partial \theta}{\partial t} - \nu_2 \Delta \theta + y \cdot \nabla \theta = 0 & \text{in } Q, \end{cases}$$

together with the boundary and initial conditions

$$(1.2) \quad \begin{cases} y = 0 & \text{on } \Sigma, \quad y(0) = y_0. \\ \nu_2 \frac{\partial \theta}{\partial \vec{n}} = v \chi_{\Sigma_c} & \text{on } \Sigma, \quad \theta(0) = \theta_0. \end{cases}$$

Above, $Q = \Omega \times (0, T)$ and $\Sigma = \Gamma \times (0, T)$, where $\Omega \subset \mathbb{R}^2$ is a bounded domain with Lipschitz continuous boundary $\Gamma = \partial\Omega$. The function χ_{Σ_c} is the characteristic function of $\Sigma_c = \Gamma_c \times (0, T)$ with $\Gamma_c \subset \Gamma$. The variable $y = (y_1, y_2)^T$ is the flow velocity, p is the pressure deviation from the hydrostatic and θ is a normalized temperature deviation. The coefficients $\nu_1 = \sqrt{\frac{Pr}{Ra}}$ and $\nu_2 = \frac{1}{\sqrt{RaPr}}$ are constants with Pr and Ra the Prandtl and Rayleigh numbers, respectively. The vector $\mathbf{e}_2 = (0, 1)^T$, and \vec{n} is the unit outward normal vector. Equation (1.1)-(1.2) describes thermal convection, with the flow induced by gravity and differences in temperature on the boundary Γ . For brevity, we focus on the boundary conditions (1.2) in the following discussions and other types of boundary conditions can be treated similarly.

We study the optimal control of thermally convective flows modeled by (1.1)-(1.2) via the heat flux $v \in L^2(\Sigma_c)$, which can be mathematically expressed as

$$(1.3) \quad \begin{cases} u \in L^2(\Sigma_c), \\ J(u) \leq J(v), \forall v \in L^2(\Sigma_c), \end{cases}$$

where the objective functional J is given by

$$(1.4) \quad J(v) = \frac{1}{2} \iint_Q |y - y_d|^2 dxdt + \frac{\alpha}{2} \iint_{\Sigma_c} v^2 dxdt, \forall v \in L^2(\Sigma_c),$$

the constant $\alpha > 0$ is a regularization parameter, and the state $y = y(t; v)$ obtained from the control v is the solution to the equation (1.1)-(1.2). The optimal control problem (1.3)-(1.4) aims at determining the velocity-pressure-temperature triplet (y, p, θ) — by controlling the heat flux v — such that y best matches a target velocity field y_d . Such a model plays an important role in many real applications, e.g., optimal control of semiconductor melts in zone-melting and Czochralski growth configurations [3], and control of energy efficient building systems [10]. Therefore, it is of practical significance to develop an efficient numerical method for solving (1.3)-(1.4).

In addition to the tracking-type objective functional given by (1.4), one may also be interested in the following vorticity reduction objective functional

$$(1.5) \quad J(v) = \frac{1}{2} \iint_Q |\nabla \times y(v)|^2 dxdt + \frac{\alpha}{2} \iint_{\Sigma_c} v^2 dxdt, \forall v \in L^2(\Sigma_c),$$

where $\frac{1}{2} \iint_Q |\nabla \times y(v)|^2 dxdt$ measures the vorticity of the flow. Minimizing (1.5) has significant applications in science and engineering such as control of turbulence and control of crystal growth process [2, 3]. To expose our main ideas clearly, we focus on the objective functional (1.4) and all results can be easily applied to the case of (1.5) by slight modifications.

In the equation (1.1)-(1.2), the Navier-Stokes equation and an advection-diffusion are coupled; the nonlinearity of $(y \cdot \nabla)y$ and $(y \cdot \nabla)\theta$ are coupled with the incompressibility condition $\nabla \cdot y = 0$; and the problem (1.3)-(1.4) is highly nonconvex because of the nonlinearity. The complicated structure and nonconvexity lead to enormous obstacles in solving (1.3)-(1.4). In particular, as to be shown in Theorem 2.2, computing a gradient of $J(v)$ requires solving equation (1.1)-(1.2) and the corresponding adjoint system, and thus is computationally challenging. Consequently, although gradient-type methods (e.g., gradient descent methods and conjugate gradient methods) can be conceptually applied to (1.3)-(1.4), it is rather nontrivial to implement them in practice. All these obvious difficulties explain that research on numerical study for (1.3)-(1.4) is still in its infancy.

Some easier cases of (1.3)-(1.4) have been studied in [32, 33, 34], where the state equation (1.1)-(1.2) was replaced by its stationary counterpart. In [30], robust temperature and velocity output tracking for linearized Boussinesq equations was studied. For the general nonlinear time-dependent case (1.1)-(1.2), the exact controllability was analyzed in [18], the existence of a solution and the first-order optimality condition was studied in [3, 4, 14]. In [12], local exponential stabilization of (1.1)-(1.2) was studied with control acting on a portion of the boundary. Computationally, a semi-implicit scheme was suggested in [2, 3] for the time discretization and a damped Picard iteration was proposed for solving the first-order optimality system of (1.3)-(1.4). Note that the semi-implicit time discretization scheme is only conditionally stable, and in practice, a tiny time step might be necessary to ensure numerical stability and robustness. The damped Picard iteration is essentially a gradient descent method, which usually converges slowly. In [41], a piecewise-in-time optimal control approach was proposed, which matches the velocity fields y_d at a sequence of time intervals, and the velocity tracking at each time interval is then formulated as an optimal control problem modeled by a stationary Boussinesq equation. A solution of (1.3)-(1.4) was then obtained by patching together all the solutions of local optimal control problems at each time interval. However, the piecewise-in-time optimal approach can only pursue a suboptimal control of (1.3)-(1.4). In [42], an adaptive procedure was proposed by using a proper orthogonal decomposition approach with the sequential quadratic programming method to obtain a reduced-order model for the optimal control problem (1.3) with the vorticity reduction objective functional (1.5). Note that the backward Euler method was used in [42] for the time discretization, which leads to a large-scale and computationally expensive complex nonlinear system at each time step with all components coupled together.

1.2. Our methodology. In this paper, we propose an efficient and relatively easy-to-implement numerical method to solve (1.3)-(1.4). We employ the discretize-then-optimize approach, i.e., we first discretize the optimal control problem (1.3)-(1.4) and then compute the gradient in a fully discrete setting. Compared with the optimize-then-discretize approach, the discretize-then-optimize approach is more advantageous in that the fully discrete state equation and the fully discrete adjoint equation are strict in duality. This fact guarantees that the computed negative gradient is a descent direction of the fully discrete optimal control problem. We refer to, e.g. [23, 43], for more discussions on the difference between the discretize-then-optimize and optimize-then-discretize approaches. To implement the discretize-then-optimize approach, we advocate combining an operator splitting scheme for time discretization and a finite element method for space discretization. Then, the Broyden-Fletcher-Goldfarb-Shanno (BFGS) quasi-Newton method is applied to solve the resulting optimization problem. Although these numerical approaches have been individually developed in different literatures, it is highly nontrivial to implement them synergically to solve the problem (1.3)-(1.4) due to its complicated structure.

Recall that when a gradient-type method is applied to solve (1.3)-(1.4), computing the gradient of the objective functional $J(v)$ requires solving the state equation (1.1)-(1.2) and the corresponding adjoint equation, which is usually challenging and computationally expensive in numerical implementation. To address this issue, we aim at developing an easily implementable time discretization scheme for (1.3)-(1.4) such that the resulting discrete gradient is cheap to compute. Our philosophy is to decompose the state equation (1.1)-(1.2) into smaller and easier parts so that the computation of a discrete gradient only requires solving several much easier subproblems. More precisely, we aim at untieing the velocity-pressure-temperature

triplet, and decoupling the nonlinearity of $(y \cdot \nabla)y$ and $(y \cdot \nabla)\theta$ from the incompressibility condition $\nabla \cdot y = 0$. For this purpose, we consider the operator splitting techniques. In the literature, it has been shown that various operator splitting methods such as the Douglas-Rachford splitting method in [16], the Peaceman-Rachford splitting method in [40], and the θ -scheme in [20, 21] are very efficient for solving various partial differential equations (PDEs) (see [22]). Their applications to optimal control problems, however, have not been well explored. It is worth noting that a straightforward application of the aforementioned operator splitting methods to the optimal control problem (1.3)–(1.4) leads to some immediate difficulties in numerical implementation. In particular, as discussed in Remark 3.3, deriving the corresponding adjoint equation is challenging due to some coupled terms at different time intervals. This issue has also been mentioned in [8], where a semi-implicit finite difference scheme was used for the time discretization of an optimal control problem modeled by the Navier-Stokes equation. The author commented “There are other more sophisticated schemes that have better accuracy and stability properties. Examples are the operator-splitting schemes” and “However, such schemes will introduce some nontrivial complications when deriving corresponding adjoint equation”. To the best of our knowledge, there has been almost no development on implementing operator splitting methods to optimal control problems including (1.3)–(1.4).

To tackle the aforementioned issues, we propose to leverage the Marchuk-Yanenko method [36] and the L^2 -projection [15] to implement the time discretization of equation (1.1)–(1.2). The Marchuk-Yanenko method is chosen because it does not lead to coupled terms at different time intervals and hence does not introduce any complication in deriving the corresponding adjoint equation. We are motivated to consider the L^2 -projection method by its popular application for unsteady incompressible Navier-Stokes equations, see [22, Chapter 7] and the survey paper [25] for a thorough discussion. Inspired by [25], we advocate the L^2 -projection with an incremental term of the pressure p (see (3.5)) to increase numerical accuracy and stability. Consequently, a scheme combining the Marchuk-Yanenko method with the incremental L^2 -projection is proposed for the time discretization of (1.1)–(1.2). The resulting scheme only needs to solve a sequence of decoupled linear time-independent equations for y , p , and θ at each time step. Computing time can thus be substantially lowered for large-scale cases. With the proposed time-discretization scheme, the gradient is relatively easier to compute when a gradient-type method is applied. More precisely, we only need to solve four linear advection-diffusion equations and two degenerated Stokes equations at each time step to compute the gradient. All these equations can be easily solved by some well-developed numerical methods in the literature, e.g., the fixed-point iterative schemes in [9] for advection-diffusion equations and the preconditioned conjugate gradient methods (e.g., [22, Chapter 3]) for degenerated Stokes equations, respectively.

With the well-designed operator splitting time-discretization scheme, the gradient of the objective functional $J(v)$ at $v \in L^2(\Sigma_c)$ is easy to compute and thus classic gradient descent methods can be applied to solve (1.3)–(1.4). However, gradient descent methods may converge slowly and inefficiently. To address this issue, we suggest using the BFGS method for solving (1.3)–(1.4). It is well known that the BFGS method may be very effective to deal with large-scale optimization problems, see, e.g. [39]. To implement the BFGS, it is crucial to seek a suitable step size for each iteration while some commonly used backtracking line search strategies are usually too expensive for the settings under our consideration, because evaluating the objective function value $J(v)$ is required repeatedly and evaluating $J(v)$ entails solving the state equation (1.1)–(1.2). To tackle this difficulty, we look into the structure of the problem (1.3)–(1.4) meticulously and propose an efficient inexact strategy for determining step sizes that requires solving only a few linear equations. Thus, the implementation of the BFGS is easy and the computation for solving (1.3)–(1.4) becomes much cheaper.

Finally, we mention that the central concern of this work is to design an efficient numerical approach to the optimal control problem (1.3)–(1.4). Other techniques developed in the literatures such as the model reduction technique [42] and the memory-saving strategy [8] can also be embedded into the implementation of our proposed numerical approach to further reduce the computational cost. Furthermore, since the Navier-Stokes equation is involved in (1.1)–(1.2), our proposed method can be directly applied to optimal control problems modeled by the Navier-Stokes equation [8, 26].

1.3. Organization. The outline of this paper is as follows. In Section 2, we present some preliminary results and derive the first-order optimality conditions for (1.3)–(1.4). In Section 3, we propose an operator splitting method for the time discretization of (1.3)–(1.4) and a finite element method for the space discretization. Then, a limited-memory BFGS method with an efficient step size strategy is proposed in Section 4 for solving the fully discretized problem of (1.3)–(1.4). Some preliminary numerical results are reported in Section 5 to validate the efficiency of our proposed numerical approach. Finally, some conclusions are drawn in Section 6.

2. Analysis of problem (1.3)–(1.4). In this section, we introduce some notations and preliminary results, which will be used in the following discussions. Then, we show the existence of optimal controls for (1.3)–(1.4) and derive the associated first-order optimality conditions.

2.1. Notations. We shall use the standard notations for the Sobolev spaces $H^m(\Omega)$ with norm $\|\cdot\|_m$ and $L^2(\Omega)$ with norm $\|\cdot\|$. Let $H_0^m(\Omega)$ be the closure of the space $C_0^\infty(\Omega)$ under the norm $\|\cdot\|_m$, where $C_0^\infty(\Omega)$ denotes the space of all infinitely differentiable functions over Ω with a compact support in Ω . Let X be a Banach space with a norm $\|\cdot\|_X$. Then, the space $L^2(0, T; X)$ consists of all measurable functions $z : (0, T) \rightarrow X$ satisfying

$$\|z\|_{L^2(0, T; X)} := \left(\int_0^T \|z(t)\|_X^2 dt \right)^{\frac{1}{2}} < +\infty.$$

Moreover, we use the following function spaces:

$$Y := \{y \in [L^2(\Omega)]^2, \nabla \cdot y = 0\}, \quad V := \{y \in [H_0^1(\Omega)]^2, \nabla \cdot y = 0\}, \quad P := L_0^2(\Omega) = \{p \in L^2(\Omega), \int_\Omega p \, dx = 0\}$$

$$W_y(0, T) := \{y_t \in L^2(0, T; V^*), y \in L^2(0, T; V)\}, \quad W_\theta(0, T) := \{\theta_t \in L^2(0, T; H^1(\Omega)^*), \theta \in L^2(0, T; H^1(\Omega))\},$$

where Z^* is the dual space of Z with Z a Banach space. To simplify the notation, we introduce the following bilinear and trilinear forms:

$$\begin{cases} a(y, \varphi) = \nu_1 \int_\Omega \nabla y \cdot \nabla \varphi \, dx, & \forall y, \varphi \in [H^1(\Omega)]^2, \\ b(q, y) = \int_\Omega q \nabla \cdot y \, dx, & q \in P, y \in [H^1(\Omega)]^2, \\ d(\theta, \psi) = \nu_2 \int_\Omega \nabla \theta \cdot \nabla \psi \, dx, & \forall \theta, \psi \in H^1(\Omega), \\ c(y, w, \varphi) = \int_\Omega (y \cdot \nabla) w \cdot \varphi \, dx, & \forall y, w, \varphi \in [H^1(\Omega)]^2, \\ e(y, \theta, \psi) = \int_\Omega (y \cdot \nabla \theta) \psi \, dx, & \forall y \in [H^1(\Omega)]^2, \theta, \psi \in H^1(\Omega). \end{cases}$$

It is easy to verify that both bilinear forms $a(y, \varphi)$ and $d(\theta, \psi)$ are continuous and coercive using some standard arguments as those in e.g., [31].

2.2. Existence of optimal controls. With these notations, the variational formulation of the state equation reads as follows: find $(y, p, \theta) \in L^2(0, T; [H_0^1(\Omega)]^2) \times L^2(0, T; P) \times L^2(0, T; H^1(\Omega))$ such that for a.e. $t \in (0, T)$,

$$(2.1) \quad \begin{cases} (y_t, \varphi) + a(y(t), \varphi) + c(y(t), y(t), \varphi) - b(p(t), \varphi) = (\theta(t) e_2, \varphi) & \forall \varphi \in [H_0^1(\Omega)]^2, \\ b(q, y(t)) = 0 & \forall q \in P, \\ (\theta_t, \psi) + d(\theta(t), \psi) + e(y(t), \theta(t), \psi) = (v(t) \chi_{\Gamma_c}, \psi)_\Gamma & \forall \psi \in H^1(\Omega), \\ y(x, 0) = y_0, \quad \theta(x, 0) = \theta_0 & x \in \Omega. \end{cases}$$

Here and in what follows, the notation (\cdot, \cdot) stands for the canonical L^2 -inner product. Moreover, if we consider the subspace $V \subset [H_0^1(\Omega)]^2$, then the variational formulation (2.1) can be reformulated as: find $(y, \theta) \in L^2(0, T; V) \times L^2(0, T; H^1(\Omega))$ such that for a.e. $t \in (0, T)$

$$\begin{cases} (y_t, \varphi) + a(y(t), \varphi) + c(y(t), y(t), \varphi) = (\theta(t)\mathbf{e}_2, \varphi) & \forall \varphi \in V, \\ (\theta_t, \psi) + d(\theta(t), \psi) + e(y(t), \theta(t), \psi) = (v(t)\chi_{\Gamma_c}, \psi)_\Gamma & \forall \psi \in H^1(\Omega), \\ y(x, 0) = y_0, \quad \theta(x, 0) = \theta_0 & x \in \Omega. \end{cases}$$

For the state equation (1.1)–(1.2), we have the following result on the existence and uniqueness of the solution. We refer to, such as [19], for the details of the proof.

THEOREM 2.1. *Suppose that $y_0 \in Y$ satisfies $y_0|_\Gamma = 0$, $\theta_0 \in L^2(\Omega)$ and $v \in L^2(0, T; H^{-1/2}(\Gamma))$. Then, the state equation (1.1)–(1.2) admits a unique weak solution $(y, p, \theta) \in W_y(0, T) \times P \times W_\theta(0, T)$.*

Using the result of Theorem 2.1 and considering the continuous embeddings:

$$W_y(0, T) \hookrightarrow C(0, T; Y),$$

it can be shown that the objective functional (1.4) is well-defined. The existence of a solution to the optimal control problem (1.3)–(1.4) can be proved following the lines in [35]. We thus omit the proof here for succinctness, and refer to [1, 4, 14] for the details. Since problem (1.3)–(1.4) under investigation is non-convex, the uniqueness of the solution cannot be guaranteed and only a local minimizer can be pursued in general cases.

2.3. First-order optimality conditions. In this subsection, we use a formal perturbation analysis, which has been well developed in [35], to derive the first-order optimality condition of the optimal control problem (1.3)–(1.4). Let $DJ(v)$ be the first-order derivative of J at $v \in L^2(\Sigma_c)$. If $u \in L^2(\Sigma_c)$ is a local optimal control to (1.3)–(1.4), then the optimality condition corresponding to the optimal control problem (1.3)–(1.4) reads as

$$DJ(u) = 0.$$

To specify $DJ(v)$, let us consider $v \in L^2(\Sigma_c)$ and a small perturbation $\delta v \in L^2(\Sigma_c)$, we then have

$$(2.2) \quad \delta J(v) = (DJ(v), \delta v) = \iint_Q (y - y_d)\delta y \, dxdt + \alpha \iint_{\Sigma_c} v\delta v \, dxdt,$$

where δy is the solution of

$$\begin{cases} \frac{\partial \delta y}{\partial t} + (y \cdot \nabla)\delta y + (\delta y \cdot \nabla)y - \nu_1 \Delta \delta y + \nabla \delta p = \delta \theta \mathbf{e}_2 & \text{in } \Omega \times (0, T), \\ \nabla \cdot \delta y = 0 & \text{in } \Omega \times (0, T), \\ \delta y = 0 & \text{on } \Gamma \times (0, T), \\ \delta y(0) = 0, \end{cases}$$

and $\delta \theta$ satisfies

$$\begin{cases} \frac{\partial \delta \theta}{\partial t} - \nu_2 \Delta \delta \theta + y \cdot \nabla \delta \theta + \delta y \cdot \nabla \theta = 0 & \text{in } \Omega \times (0, T), \\ \nu_2 \frac{\partial \delta \theta}{\partial \bar{n}} = \delta v \chi_{\Sigma_c} & \text{on } \Gamma \times (0, T), \\ \delta \theta(0) = 0. \end{cases}$$

For any $z \in [L^2(Q)]^2$ and $\xi, \zeta \in L^2(Q)$, we have

$$(2.3) \quad \iint_Q \left(\frac{\partial \delta y}{\partial t} + (y \cdot \nabla) \delta y + (\delta y \cdot \nabla) y - \nu_1 \Delta \delta y + \nabla \delta p \right) \cdot z dx dt = \iint_Q \delta \theta \mathbf{e}_2 \cdot z dx dt,$$

$$(2.4) \quad \iint_Q \nabla \cdot \delta y \xi dx dt = 0,$$

$$(2.5) \quad \iint_Q \left(\frac{\partial \delta \theta}{\partial t} - \nu_2 \Delta \delta \theta + y \cdot \nabla \delta \theta + \delta y \cdot \nabla \theta \right) \zeta dx dt = 0.$$

Applying Green's formula to (2.3)–(2.5), we obtain

$$(2.6) \quad \begin{aligned} & \delta y(T)z(T) - \delta y(0)z(0) + \iint_Q \left(-\frac{\partial z}{\partial t} - \nu_1 \Delta z + (\nabla y)^T z - (y \cdot \nabla) z \right) \cdot \delta y dx dt \\ & - \iint_Q (\nabla \cdot z) \delta p dx dt + \iint_\Sigma \frac{\partial \delta p}{\partial \bar{n}} z dx dt = \iint_Q \delta \theta \mathbf{e}_2 \cdot z dx dt, \end{aligned}$$

$$(2.7) \quad - \iint_Q \nabla \xi \cdot \delta y dx dt = 0,$$

$$(2.8) \quad \begin{aligned} & \delta \theta(T)\zeta(T) - \delta \theta(0)\zeta(0) + \iint_Q \left(-\frac{\partial \zeta}{\partial t} - \nu_2 \Delta \zeta - y \cdot \nabla \zeta \right) \delta \theta dx dt \\ & + \iint_Q \zeta \nabla \theta \cdot \delta y dx dt = \nu_2 \iint_\Sigma \left(\frac{\partial \delta \theta}{\partial \bar{n}} \zeta - \frac{\partial \zeta}{\partial \bar{n}} \delta \theta \right) dx dt. \end{aligned}$$

Summing up (2.6)–(2.8) shows that

$$(2.9) \quad \begin{aligned} & \delta y(T)z(T) - \delta y(0)z(0) + \delta \theta(T)\zeta(T) - \delta \theta(0)\zeta(0) \\ & + \iint_Q \left(-\frac{\partial z}{\partial t} - \nu_1 \Delta z + (\nabla y)^T z - (y \cdot \nabla) z - \nabla \xi \right) \cdot \delta y dx dt \\ & + \iint_Q \zeta \nabla \theta \cdot \delta y dx dt + \iint_Q \left(-\frac{\partial \zeta}{\partial t} - \nu_2 \Delta \zeta - y \cdot \nabla \zeta \right) \delta \theta dx dt \\ & - \iint_Q (\nabla \cdot z) \delta p dx dt + \iint_\Sigma \frac{\partial \delta p}{\partial \bar{n}} z dx dt \\ & = \iint_Q \delta \theta \mathbf{e}_2 \cdot z dx dt + \nu_2 \iint_\Sigma \left(\frac{\partial \delta \theta}{\partial \bar{n}} \zeta - \frac{\partial \zeta}{\partial \bar{n}} \delta \theta \right) dx dt. \end{aligned}$$

If we let z and ζ satisfy

$$\begin{aligned} -\frac{\partial z}{\partial t} - \nu_1 \Delta z + (\nabla y)^T z - (y \cdot \nabla) z + \nabla \xi + \zeta \nabla \theta &= y - y_d, \\ -\nabla \cdot z &= 0, \\ -\frac{\partial \zeta}{\partial t} - \nu_2 \Delta \zeta - y \cdot \nabla \zeta &= z \cdot \mathbf{e}_2, \end{aligned}$$

together with the boundary and initial conditions

$$\begin{aligned} z &= 0 \text{ on } \Sigma, \quad \nu_2 \frac{\partial \zeta}{\partial \bar{n}} = 0 \text{ on } \Sigma, \\ z(T) &= 0, \quad \zeta(T) = 0, \end{aligned}$$

Then, equality (2.9) reduces to

$$\iint_Q (y - y_d) \cdot \delta y \, dxdt = \iint_\Sigma \zeta \delta v \chi_{\Sigma_c} \, dxdt.$$

It follows from (2.2) that

$$\delta J(v) = \iint_{\Sigma_c} (\zeta + \alpha v) \delta v \, dxdt,$$

which implies that

$$DJ(v) = \zeta|_{\Sigma_c} + \alpha v.$$

From the above discussions, we obtain the following results.

THEOREM 2.2. *If $u \in L^2(\Sigma_c)$ is an optimal control of (1.3)–(1.4), then it satisfies the following optimality system*

$$(2.10) \quad DJ(u) = \alpha u + \zeta|_{\Sigma_c} = 0,$$

with ζ obtained from the solutions of the following fully coupled systems:

$$(2.11) \quad \begin{cases} \frac{\partial y}{\partial t} + (y \cdot \nabla)y - \nu_1 \Delta y + \nabla p = \theta \mathbf{e}_2 & \text{in } \Omega \times (0, T), \\ \nabla \cdot y = 0 & \text{in } \Omega \times (0, T), \\ y = 0 & \text{on } \Gamma \times (0, T), \\ y(0) = y_0, \end{cases}$$

$$(2.12) \quad \begin{cases} \frac{\partial \theta}{\partial t} - \nu_2 \Delta \theta + y \cdot \nabla \theta = 0 & \text{in } \Omega \times (0, T), \\ \nu_2 \frac{\partial \theta}{\partial \bar{n}} = u \chi_{\Sigma_c} & \text{on } \Gamma \times (0, T), \\ \theta(0) = \theta_0, \end{cases}$$

and

$$(2.13) \quad \begin{cases} -\frac{\partial z}{\partial t} - \nu_1 \Delta z + (\nabla y)^T z - (y \cdot \nabla)z - \nabla \xi + \zeta \nabla \theta = y - y_d & \text{in } \Omega \times (0, T), \\ -\nabla \cdot z = 0 & \text{in } \Omega \times (0, T), \\ z = 0 & \text{on } \Gamma \times (0, T), \\ z(T) = 0, \end{cases}$$

$$(2.14) \quad \begin{cases} -\frac{\partial \zeta}{\partial t} - \nu_2 \Delta \zeta - y \cdot \nabla \zeta = z \cdot \mathbf{e}_2 & \text{in } \Omega \times (0, T), \\ \nu_2 \frac{\partial \zeta}{\partial \bar{n}} = 0 & \text{on } \Gamma \times (0, T), \\ \zeta(T) = 0. \end{cases}$$

Here, equations (2.13) and (2.14) are the adjoint equations corresponding to velocity and temperature, respectively. Equations (2.11)–(2.12) are the state equations defined in (1.1)–(1.2).

REMARK 2.3. *The optimality systems for the vorticity reduction case (1.5) are the same as those given by (2.10)–(2.14) but with $y - y_d$ in (2.13) replaced by $\nabla \times (\nabla \times y)$.*

To compute the gradient $DJ(v)$ for any given $v \in L^2(\Sigma_c)$, it follows from Theorem 2.2 that one has to solve the state system (2.11)–(2.12) and the adjoint system (2.13)–(2.14). Both the state and adjoint systems consist of coupled PDEs and they are high-dimensional. Hence, computing the gradient $DJ(v)$ is challenging and computationally expensive, and some sophisticated techniques are required to reduce the computational cost.

3. Space and time discretizations. In this section, we first discuss the L^2 -projection and the Marchuk-Yanenko splitting for the time discretization of the optimal control problem (1.3)–(1.4). Then, the first-order optimality conditions are derived for the time-discretized optimal control problem. Finally, a finite element method is presented for the space discretization to obtain a fully discretized formulation of the optimal control problem (1.3)–(1.4).

3.1. An operator splitting method for the time discretization. Assuming that T is finite, for any given positive integer N , let $\Delta t = T/N$ be the time discretization step size and $t_n = n\Delta t$, $\forall n = 1, \dots, N$. Then, we approximate U by $U^{\Delta t} = [L^2(\Gamma_c)]^N$ and v by $\mathbf{v} = (\mathbf{v}^n)_{n=1}^N \in U^{\Delta t}$.

We can define the scalar product on $U^{\Delta t}$ as

$$(\mathbf{u}, \mathbf{v})_{U^{\Delta t}} = \Delta t \sum_{n=1}^N \int_{\Gamma_c} \mathbf{u}^n \mathbf{v}^n dx, \quad \forall \mathbf{u}, \mathbf{v} \in U^{\Delta t}.$$

Then the time-discretized formulation of problem (1.3)–(1.4) reads as

$$(3.1) \quad \begin{cases} \mathbf{u} \in U^{\Delta t}, \\ J^{\Delta t}(\mathbf{u}) \leq J^{\Delta t}(\mathbf{v}), \quad \forall \mathbf{v} \in U^{\Delta t}, \end{cases}$$

with the time-discretized objective functional

$$J^{\Delta t}(\mathbf{v}) = \frac{1}{2} \Delta t \sum_{n=1}^N \int_{\Omega} |\mathbf{y}^n - y_d^n|^2 dx + \frac{\alpha}{2} \Delta t \sum_{n=1}^N \int_{\Gamma_c} |\mathbf{v}^n|^2 dx,$$

where the time-discretized target function $y_d^n = y_d(t_n)$, $\forall n = 1, \dots, N$, and $(\mathbf{y}^n)_{n=1}^N$ are given from $(\mathbf{v}^n)_{n=1}^N$ by the solution of the following time-discretized state equations:

$$(3.2) \quad \mathbf{y}^0 = y_0, \quad \theta^0 = \theta_0, \quad \mathbf{p}^0 = 0,$$

for $n = 1, \dots, N$, solve

$$(3.3) \quad \begin{cases} \frac{\tilde{\mathbf{y}}^n - \mathbf{y}^{n-1}}{\Delta t} - \nu_1 \Delta \tilde{\mathbf{y}}^n + (\mathbf{y}^{n-1} \cdot \nabla) \tilde{\mathbf{y}}^n + \nabla \mathbf{p}^{n-1} = \theta^n \mathbf{e}_2 & \text{in } \Omega, \\ \tilde{\mathbf{y}}^n = 0 & \text{on } \Gamma, \end{cases}$$

$$(3.4) \quad \begin{cases} \frac{\theta^n - \theta^{n-1}}{\Delta t} - \nu_2 \Delta \theta^n + \mathbf{y}^{n-1} \cdot \nabla \theta^n = 0 & \text{in } \Omega, \\ \nu_2 \frac{\partial \theta^n}{\partial \vec{n}} = \mathbf{v}^n \chi_{\Gamma_c} & \text{on } \Gamma, \end{cases}$$

and

$$(3.5) \quad \begin{cases} \frac{\mathbf{y}^n - \tilde{\mathbf{y}}^n}{\Delta t} + \nabla(\mathbf{p}^n - \mathbf{p}^{n-1}) = 0 & \text{in } \Omega, \\ \nabla \cdot \mathbf{y}^n = 0 & \text{in } \Omega, \\ \mathbf{y}^n \cdot \vec{n} = 0 & \text{on } \Gamma. \end{cases}$$

Using the above operator splitting scheme (3.3)–(3.5), we decouple the nonlinearity and the incompressibility in the state equation (1.1)–(1.2), and meanwhile treat the temperature variable θ separately. As a result, we only need to handle three rather simple linear equations (3.3)–(3.5) at each time step.

REMARK 3.1. Note that the Neumann boundary condition $\mathbf{y}^n \cdot \vec{n} = 0$ is used to guarantee the well-posedness of (3.5). Alternatively, one may consider replacing (3.5) by the following equation with Dirichlet boundary condition:

$$(3.6) \quad \begin{cases} \frac{\mathbf{y}^n - \tilde{\mathbf{y}}^n}{\Delta t} + \nabla(\mathbf{p}^n - \mathbf{p}^{n-1}) = 0 & \text{in } \Omega, \\ \nabla \cdot \mathbf{y}^n = 0 & \text{in } \Omega, \\ \mathbf{y}^n = 0 & \text{on } \Gamma. \end{cases}$$

As mentioned in [22, Chapter 7], problem (3.6) is generally not well-posed since the boundary condition $\mathbf{y}^n = 0$ is too demanding for a solution which does not have the $H^1(\Omega)$ -regularity. Despite this fact, the finite element discretized analogue of (3.5) (see (3.31)) is well-posed. Moreover, compared with (3.3)–(3.5), numerical experiments show that more accurate approximate solutions can be always obtained by the scheme (3.3), (3.4) and (3.6).

3.2. First-order optimality conditions for (3.1)–(3.5). In this subsection, we derive the first-order optimality conditions for the time-discretized optimal control problem (3.1)–(3.5) using perturbation analysis as what we have done for the continuous case.

Let $\mathbf{u} \in U^{\Delta t}$ be an optimal control of (3.1)–(3.5) and $DJ^{\Delta t}(\mathbf{v})$ be the first-order differential of the functional $J^{\Delta t}$ at $\mathbf{v} \in \mathcal{U}^{\Delta t}$. Then, the following first-order optimality condition holds

$$DJ^{\Delta t}(\mathbf{u}) = 0.$$

To calculate $DJ^{\Delta t}(\mathbf{v})$, we first observe that

$$(3.7) \quad \delta J^{\Delta t}(\mathbf{v}) = (DJ^{\Delta t}(\mathbf{v}), \delta \mathbf{v}) = \Delta t \sum_{n=1}^N \int_{\Omega} (\mathbf{y}^n - \mathbf{y}_d^n) \cdot \delta \mathbf{y}^n dx + \alpha \Delta t \sum_{n=1}^N \int_{\Gamma_c} \mathbf{v}^n \delta \mathbf{v}^n dx,$$

and $(\delta \mathbf{y}^n)_{n=1}^N$, $(\delta \boldsymbol{\theta}^n)_{n=1}^N$ and $(\delta \mathbf{p}^n)_{n=1}^N$ satisfy the perturbed time-discretized state equation: given $\delta \mathbf{y}^0 = 0$, $\delta \boldsymbol{\theta}^0 = 0$, and $\delta \mathbf{p}^0 = 0$, then for $n = 1, \dots, N$,

$$(3.8) \quad \begin{cases} \frac{\delta \tilde{\mathbf{y}}^n - \delta \mathbf{y}^{n-1}}{\Delta t} - \nu_1 \Delta \delta \tilde{\mathbf{y}}^n + (\mathbf{y}^{n-1} \cdot \nabla) \delta \tilde{\mathbf{y}}^n + (\delta \mathbf{y}^{n-1} \cdot \nabla) \tilde{\mathbf{y}}^n + \nabla \delta \mathbf{p}^{n-1} = \delta \boldsymbol{\theta}^n \mathbf{e}_2 & \text{in } \Omega, \\ \delta \tilde{\mathbf{y}}^n = 0, & \text{on } \Gamma, \end{cases}$$

$$(3.9) \quad \begin{cases} \frac{\delta \boldsymbol{\theta}^n - \delta \boldsymbol{\theta}^{n-1}}{\Delta t} - \nu_2 \Delta \delta \boldsymbol{\theta}^n + \delta \mathbf{y}^{n-1} \cdot \nabla \boldsymbol{\theta}^n + \mathbf{y}^{n-1} \cdot \nabla \delta \boldsymbol{\theta}^n = 0 & \text{in } \Omega, \\ \nu_2 \frac{\partial \delta \boldsymbol{\theta}^n}{\partial \vec{n}} = \delta \mathbf{v}^n \chi_{\Gamma_c} & \text{on } \Gamma. \end{cases}$$

$$(3.10) \quad \begin{cases} \frac{\delta \mathbf{y}^n - \delta \tilde{\mathbf{y}}^n}{\Delta t} + \nabla(\delta \mathbf{p}^n - \delta \mathbf{p}^{n-1}) = 0 & \text{in } \Omega, \\ \nabla \cdot \delta \mathbf{y}^n = 0 & \text{in } \Omega, \\ \delta \mathbf{y}^n \cdot \vec{n} = 0 & \text{on } \Gamma. \end{cases}$$

Let us choose variables $(\mathbf{z}^n)_{n=1}^{N+1}$, $(\tilde{\mathbf{z}}^n)_{n=1}^{N+1}$, $(\boldsymbol{\xi}^n)_{n=1}^N$ and $(\boldsymbol{\zeta}^n)_{n=1}^{N+1}$, where \mathbf{z}^n , $\tilde{\mathbf{z}}^n$, $\boldsymbol{\xi}^n$ and $\boldsymbol{\zeta}^n$ are smooth functions of x . Multiplying both sides of the first equation in (3.8), (3.9) by $\tilde{\mathbf{z}}^n$, $\boldsymbol{\zeta}^n$, and the first two equations in (3.10) by \mathbf{z}^n and $\boldsymbol{\xi}^n$, respectively. Then integrating continuously over Ω and discretely over

(0, T), we obtain

$$(3.11) \quad \Delta t \sum_{n=1}^N \int_{\Omega} \left(\frac{\delta \tilde{\mathbf{y}}^n - \delta \mathbf{y}^{n-1}}{\Delta t} - \nu_1 \Delta \delta \tilde{\mathbf{y}}^n + (\mathbf{y}^{n-1} \cdot \nabla) \delta \tilde{\mathbf{y}}^n + (\delta \mathbf{y}^{n-1} \cdot \nabla) \tilde{\mathbf{y}}^n + \nabla \delta \mathbf{p}^{n-1} \right) \cdot \tilde{\mathbf{z}}^n dx = \Delta t \sum_{n=1}^N \int_{\Omega} \delta \theta^n \mathbf{e}_2 \cdot \tilde{\mathbf{z}}^n dx,$$

$$(3.12) \quad \Delta t \sum_{n=1}^N \int_{\Omega} \left(\frac{\delta \theta^n - \delta \theta^{n-1}}{\Delta t} - \nu_2 \Delta \delta \theta^n + \delta \mathbf{y}^{n-1} \cdot \nabla \theta^n + \mathbf{y}^{n-1} \cdot \nabla \delta \theta^n \right) \zeta^n dx = 0,$$

$$(3.13) \quad \Delta t \sum_{n=1}^N \int_{\Omega} \left(\frac{\delta \mathbf{y}^n - \delta \tilde{\mathbf{y}}^n}{\Delta t} + \nabla (\delta \mathbf{p}^n - \delta \mathbf{p}^{n-1}) \right) \cdot \mathbf{z}^n dx = 0,$$

$$(3.14) \quad \Delta t \sum_{n=1}^N \int_{\Omega} \nabla \cdot \delta \mathbf{y}^n \xi^n dx = 0.$$

Since $\delta \mathbf{y}^0 = 0$ and $\delta \theta^0 = 0$, it is easy to verify that

$$(3.15) \quad \begin{aligned} & \Delta t \sum_{n=1}^N \int_{\Omega} \frac{\delta \tilde{\mathbf{y}}^n - \delta \mathbf{y}^{n-1}}{\Delta t} \cdot \tilde{\mathbf{z}}^n + \frac{\delta \mathbf{y}^n - \delta \tilde{\mathbf{y}}^n}{\Delta t} \cdot \mathbf{z}^n dx \\ &= \Delta t \sum_{n=1}^N \int_{\Omega} \frac{\tilde{\mathbf{z}}^n - \mathbf{z}^n}{\Delta t} \delta \tilde{\mathbf{y}}^n dx + \Delta t \sum_{n=1}^N \int_{\Omega} \frac{\mathbf{z}^n - \tilde{\mathbf{z}}^{n+1}}{\Delta t} \delta \mathbf{y}^n dx + \int_{\Omega} \tilde{\mathbf{z}}^{N+1} \delta \mathbf{y}^N dx, \end{aligned}$$

and

$$(3.16) \quad \Delta t \sum_{n=1}^N \int_{\Omega} \frac{\delta \theta^n - \delta \theta^{n-1}}{\Delta t} \zeta^n dx = \Delta t \sum_{n=1}^N \int_{\Omega} \frac{\zeta^n - \zeta^{n+1}}{\Delta t} \delta \theta^n dx + \int_{\Omega} \zeta^{N+1} \delta \theta^N dx.$$

Using Green's formula and taking the boundary conditions (3.8), (3.9) and (3.10) into account, we have

$$(3.17) \quad \begin{aligned} & \Delta t \sum_{n=1}^N \int_{\Omega} ((\mathbf{y}^{n-1} \cdot \nabla) \delta \tilde{\mathbf{y}}^n + (\delta \mathbf{y}^{n-1} \cdot \nabla) \tilde{\mathbf{y}}^n) \cdot \tilde{\mathbf{z}}^n dx \\ &= \Delta t \sum_{n=1}^N \int_{\Omega} (\tilde{\mathbf{y}}^n)^T \tilde{\mathbf{z}}^n \cdot \delta \mathbf{y}^{n-1} - (\mathbf{y}^{n-1} \cdot \nabla) \tilde{\mathbf{z}}^n \cdot \delta \tilde{\mathbf{y}}^n - \nabla \cdot \mathbf{y}^{n-1} \tilde{\mathbf{z}}^n \cdot \delta \tilde{\mathbf{y}}^n dx \\ &= \Delta t \sum_{n=0}^{N-1} \int_{\Omega} (\tilde{\mathbf{y}}^{n+1})^T \tilde{\mathbf{z}}^{n+1} \cdot \delta \mathbf{y}^n dx - \Delta t \sum_{n=1}^N \int_{\Omega} ((\mathbf{y}^{n-1} \cdot \nabla) \tilde{\mathbf{z}}^n + \nabla \cdot \mathbf{y}^{n-1} \tilde{\mathbf{z}}^n) \cdot \delta \tilde{\mathbf{y}}^n dx \\ &= \Delta t \sum_{n=1}^N \int_{\Omega} (\tilde{\mathbf{y}}^{n+1})^T \tilde{\mathbf{z}}^{n+1} \cdot \delta \mathbf{y}^n - ((\mathbf{y}^{n-1} \cdot \nabla) \tilde{\mathbf{z}}^n + \nabla \cdot \mathbf{y}^{n-1} \tilde{\mathbf{z}}^n) \cdot \delta \tilde{\mathbf{y}}^n dx - \Delta t \int_{\Omega} (\tilde{\mathbf{y}}^{N+1})^T \tilde{\mathbf{z}}^{N+1} \cdot \delta \mathbf{y}^N dx, \end{aligned}$$

where the last equality follows from $\delta \mathbf{y}^0 = 0$ and $\tilde{\mathbf{y}}^{N+1}$ is an auxiliary variable to simplify the derivation of $\tilde{\mathbf{z}}^{N+1}$.

In a similar way, we can obtain

$$\begin{aligned}
& \Delta t \sum_{n=1}^N \int_{\Omega} \left(\delta \mathbf{y}^{n-1} \cdot \nabla \boldsymbol{\theta}^n + \mathbf{y}^{n-1} \cdot \nabla \delta \boldsymbol{\theta}^n \right) \zeta^n dx \\
&= \Delta t \sum_{n=1}^N \int_{\Omega} \zeta^n \nabla \boldsymbol{\theta}^n \cdot \delta \mathbf{y}^{n-1} - \mathbf{y}^{n-1} \cdot \nabla \zeta^n \delta \boldsymbol{\theta}^n - (\nabla \cdot \mathbf{y}^{n-1}) \zeta^n \delta \boldsymbol{\theta}^n dx \\
(3.18) \quad &= \Delta t \sum_{n=1}^{N-1} \int_{\Omega} \zeta^{n+1} \nabla \boldsymbol{\theta}^{n+1} \cdot \delta \mathbf{y}^n dx - \Delta t \sum_{n=1}^N \int_{\Omega} (\mathbf{y}^{n-1} \cdot \nabla \zeta^n + \nabla \cdot \mathbf{y}^{n-1} \zeta^n) \delta \boldsymbol{\theta}^n dx \\
&= \Delta t \sum_{n=1}^N \int_{\Omega} \zeta^{n+1} \nabla \boldsymbol{\theta}^{n+1} \cdot \delta \mathbf{y}^n - (\mathbf{y}^{n-1} \cdot \nabla \zeta^n + \nabla \cdot \mathbf{y}^{n-1} \zeta^n) \delta \boldsymbol{\theta}^n dx - \int_{\Omega} \zeta^{N+1} \nabla \boldsymbol{\theta}^{N+1} \cdot \delta \mathbf{y}^N dx
\end{aligned}$$

and

$$\begin{aligned}
& \Delta t \sum_{n=1}^N \int_{\Omega} \nabla \delta \mathbf{p}^{n-1} \cdot \tilde{\mathbf{z}}^n + \nabla \delta (\mathbf{p}^n - \mathbf{p}^{n-1}) \cdot \mathbf{z}^n dx \\
&= \Delta t \sum_{n=0}^{N-1} \int_{\Omega} \nabla \cdot (\mathbf{z}^n + \tilde{\mathbf{z}}^{n+1} - \mathbf{z}^{n+1}) \delta \mathbf{p}^n dx + \int_{\Omega} \nabla \cdot \mathbf{z}^N \delta \mathbf{p}^N dx \\
(3.19) \quad & - \Delta t \sum_{n=0}^{N-1} \int_{\Gamma} \tilde{\mathbf{n}} \cdot (\mathbf{z}^n + \tilde{\mathbf{z}}^{n+1} - \mathbf{z}^{n+1}) \delta \mathbf{p}^n dx - \int_{\Gamma} \tilde{\mathbf{n}} \cdot \mathbf{z}^N \delta \mathbf{p}^N dx \\
&= \Delta t \sum_{n=1}^N \int_{\Omega} \nabla \cdot (\mathbf{z}^n + \tilde{\mathbf{z}}^{n+1} - \mathbf{z}^{n+1}) \delta \mathbf{p}^n dx - \int_{\Omega} \nabla \cdot (\tilde{\mathbf{z}}^{N+1} - \mathbf{z}^{N+1}) \delta \mathbf{p}^N dx \\
& - \Delta t \sum_{n=1}^N \int_{\Gamma} \tilde{\mathbf{n}} \cdot (\mathbf{z}^n + \tilde{\mathbf{z}}^{n+1} - \mathbf{z}^{n+1}) \delta \mathbf{p}^n dx + \int_{\Gamma} \tilde{\mathbf{n}} \cdot (\tilde{\mathbf{z}}^{N+1} - \mathbf{z}^{N+1}) \delta \mathbf{p}^N dx.
\end{aligned}$$

Substituting (3.15)–(3.19) into (3.11)–(3.14) and summing them up, and then applying Green's formula to the rest of terms, we obtain

$$\begin{aligned}
& \Delta t \sum_{n=1}^N \int_{\Omega} \left(\frac{\mathbf{z}^n - \tilde{\mathbf{z}}^{n+1}}{\Delta t} \cdot \delta \mathbf{y}^n - \nabla \boldsymbol{\xi}^n \cdot \delta \mathbf{y}^n + (\tilde{\mathbf{y}}^{n+1})^T \tilde{\mathbf{z}}^{n+1} \cdot \delta \mathbf{y}^n + \zeta^{n+1} \nabla \boldsymbol{\theta}^{n+1} \cdot \delta \mathbf{y}^n \right. \\
& + \frac{\tilde{\mathbf{z}}^n - \mathbf{z}^n}{\Delta t} \cdot \delta \tilde{\mathbf{y}}^n - \nu_1 \Delta \tilde{\mathbf{z}}^n \cdot \delta \tilde{\mathbf{y}}^n - ((\mathbf{y}^{n-1} \cdot \nabla) \tilde{\mathbf{z}}^n + \nabla \cdot \mathbf{y}^{n-1} \tilde{\mathbf{z}}^n) \cdot \delta \tilde{\mathbf{y}}^n \\
& + \frac{\zeta^n - \zeta^{n+1}}{\Delta t} \delta \boldsymbol{\theta}^n - \nu_2 \Delta \zeta^n \delta \boldsymbol{\theta}^n - (\mathbf{y}^{n-1} \cdot \nabla \zeta^n + \nabla \cdot \mathbf{y}^{n-1} \zeta^n) \delta \boldsymbol{\theta}^n + \nabla \cdot (\mathbf{z}^n + \tilde{\mathbf{z}}^{n+1} - \mathbf{z}^{n+1}) \delta \mathbf{p}^n \Big) dx \\
(3.20) \quad & + \Delta t \int_{\Omega} \left(\tilde{\mathbf{z}}^{N+1} \cdot \delta \mathbf{y}^N + \zeta^{N+1} \delta \boldsymbol{\theta}^N - (\tilde{\mathbf{y}}^{N+1})^T \tilde{\mathbf{z}}^{N+1} \cdot \delta \mathbf{y}^N - \zeta^{N+1} \nabla \boldsymbol{\theta}^{N+1} \cdot \delta \mathbf{y}^N - \nabla \cdot (\tilde{\mathbf{z}}^{N+1} - \mathbf{z}^{N+1}) \delta \mathbf{p}^N \right) dx \\
&= \Delta t \sum_{n=1}^N \left(\int_{\Omega} \tilde{\mathbf{z}}^n \cdot \mathbf{e}_2 \delta \boldsymbol{\theta}^n dx + \nu_1 \int_{\Gamma} \frac{\partial \delta \tilde{\mathbf{y}}^n}{\partial \tilde{\mathbf{n}}} \tilde{\mathbf{z}}^n - \frac{\partial \tilde{\mathbf{z}}^n}{\partial \tilde{\mathbf{n}}} \cdot \delta \tilde{\mathbf{y}}^n dx + \nu_2 \int_{\Gamma} \frac{\partial \delta \boldsymbol{\theta}^n}{\partial \tilde{\mathbf{n}}} \zeta^n - \frac{\partial \zeta^n}{\partial \tilde{\mathbf{n}}} \cdot \delta \boldsymbol{\theta}^n dx \right) \\
& + \Delta t \sum_{n=1}^N \int_{\Omega} \tilde{\mathbf{n}} \cdot (\mathbf{z}^n + \tilde{\mathbf{z}}^{n+1} - \mathbf{z}^{n+1}) \delta \mathbf{p}^n dx - \int_{\Omega} \tilde{\mathbf{n}} \cdot (\tilde{\mathbf{z}}^{N+1} - \mathbf{z}^{N+1}) \delta \mathbf{p}^N dx.
\end{aligned}$$

Let $(\tilde{\mathbf{z}}^n)_{n=1}^{N+1}$, $(\mathbf{z}^n)_{n=1}^{N+1}$ and $(\zeta^n)_{n=1}^{N+1}$ satisfy the following equations

$$(3.21) \quad \tilde{\mathbf{z}}^{N+1} = 0, \quad \mathbf{z}^{N+1} = 0 \quad \text{and} \quad \zeta^{N+1} = 0,$$

for $n = N, N - 1, \dots, 2, 1$

$$(3.22) \quad \begin{cases} \frac{\mathbf{z}^n - \tilde{\mathbf{z}}^{n+1}}{\Delta t} - \nabla \boldsymbol{\zeta}^n + (\tilde{\mathbf{y}}^{n+1})^T \tilde{\mathbf{z}}^{n+1} + \boldsymbol{\zeta}^{n+1} \nabla \boldsymbol{\theta}^{n+1} = \mathbf{y}^n - \mathbf{y}_d^n & \text{in } \Omega, \\ \nabla \cdot (\mathbf{z}^n + \tilde{\mathbf{z}}^{n+1} - \mathbf{z}^{n+1}) = 0 & \text{in } \Omega, \\ \mathbf{z}^n \cdot \vec{n} = 0 & \text{on } \Gamma, \end{cases}$$

$$(3.23) \quad \begin{cases} \frac{\tilde{\mathbf{z}}^n - \mathbf{z}^n}{\Delta t} - \nu_1 \Delta \tilde{\mathbf{z}}^n - ((\mathbf{y}^{n-1} \cdot \nabla) \tilde{\mathbf{z}}^n + \nabla \cdot \mathbf{y}^{n-1} \tilde{\mathbf{z}}^n) = 0 & \text{in } \Omega, \\ \tilde{\mathbf{z}}^n = 0 & \text{on } \Gamma \end{cases}$$

and

$$(3.24) \quad \begin{cases} \frac{\boldsymbol{\zeta}^n - \boldsymbol{\zeta}^{n+1}}{\Delta t} - \nu_2 \Delta \boldsymbol{\zeta}^n - (\mathbf{y}^{n-1} \cdot \nabla \boldsymbol{\zeta}^n + \nabla \cdot \mathbf{y}^{n-1} \boldsymbol{\zeta}^n) = \tilde{\mathbf{z}}^n \cdot \mathbf{e}_2 & \text{in } \Omega, \\ \nu_2 \frac{\partial \boldsymbol{\zeta}^n}{\partial \vec{n}} = 0 & \text{on } \Gamma. \end{cases}$$

Then, equality (3.20) reduces to

$$\Delta t \sum_{n=1}^N \int_{\Omega} (\mathbf{y}^n - \mathbf{y}_d^n) \delta \mathbf{y}^n dx = \Delta t \sum_{n=1}^N \int_{\Gamma} \boldsymbol{\zeta}^n \delta \mathbf{v}^n \chi_{\Gamma_c} dx.$$

Taking (3.7) into account, we obtain

$$\delta J^{\Delta t}(\mathbf{v}) = (DJ^{\Delta t}(\mathbf{v}), \delta \mathbf{v}) = \Delta t \sum_{n=1}^N \int_{\Gamma_c} (\boldsymbol{\zeta}^n + \alpha \mathbf{v}^n) \delta \mathbf{v}^n dx,$$

which implies that

$$(3.25) \quad DJ^{\Delta t}(\mathbf{v}) = (\boldsymbol{\zeta}^n|_{\Gamma_c} + \alpha \mathbf{v}^n)_{n=1}^N.$$

We summarize the first-order optimality conditions for the time-discretized problem (3.1)-(3.5) as follows:

THEOREM 3.2. *If $\mathbf{u} = \{\mathbf{u}^n\}_{n=1}^N \in U^{\Delta t}$ is an optimal control of the time-discretized problem (3.1)-(3.5), then the following optimality conditions hold*

$$\alpha \mathbf{u}^n + \boldsymbol{\zeta}^n|_{\Gamma_c} = 0, \quad \forall n = 1, 2, \dots, N,$$

where $(\boldsymbol{\zeta}^n)_{n=1}^N$ is the successive solution of the time-discretized state equations (3.2)–(3.5) and the corresponding adjoint equations (3.21)–(3.24).

To compute a gradient $DJ^{\Delta t}(\mathbf{v})$, we only need to solve four linear advection-diffusion equations: (3.3), (3.4), (3.23) and (3.24), and two degenerated Stokes equations: (3.5) and (3.22), at each time step. All these equations can be easily solved by some well-developed numerical methods in the literature. For instance, a fixed-point iterative process is proposed in [9] for solving advection-diffusion problems, and a preconditioned conjugate gradient method is suggested in [22] to solve degenerated Stokes equations. More details can be found in [22] and references therein.

REMARK 3.3. *Following [22], implementing the Peaceman-Rachford splitting method [40] to (1.1)-(1.2) yields the following time-discretized equations (with $0 < \beta < 1$, $0 < \gamma < 1$ and $\beta + \gamma = 1$): $\mathbf{y}^0 = \mathbf{y}_0$, $\boldsymbol{\theta}^0 = \boldsymbol{\theta}_0$, then for $n = 1, \dots, N$, solve*

$$(3.26) \quad \begin{cases} \frac{\tilde{\mathbf{y}}^n - \mathbf{y}^{n-1}}{\Delta t/2} - \beta \nu_1 \Delta \tilde{\mathbf{y}}^n + \nabla \tilde{\mathbf{p}}^n = \boldsymbol{\theta}^{n-1} \mathbf{e}_2 + \gamma \nu_1 \Delta \mathbf{y}^{n-1} - (\mathbf{y}^{n-1} \cdot \nabla) \mathbf{y}^{n-1} & \text{in } \Omega, \\ \nabla \cdot \tilde{\mathbf{y}}^n = 0 & \text{in } \Omega, \\ \tilde{\mathbf{y}}^n = 0 & \text{on } \Gamma, \end{cases}$$

$$(3.27) \quad \begin{cases} \frac{\mathbf{y}^n - \tilde{\mathbf{y}}^n}{\Delta t/2} - \gamma\nu_1\Delta\mathbf{y}^n + (\mathbf{y}^n \cdot \nabla)\mathbf{y}^n = \boldsymbol{\theta}^{n-1}\mathbf{e}_2 + \beta\nu_1\Delta\tilde{\mathbf{y}}^n - \nabla\tilde{\mathbf{p}}^n & \text{in } \Omega, \\ \nabla \cdot \mathbf{y}^n = 0 & \text{in } \Omega, \\ \mathbf{y}^n \cdot \vec{n} = 0 & \text{on } \Gamma, \end{cases}$$

and

$$(3.28) \quad \begin{cases} \frac{\boldsymbol{\theta}^n - \boldsymbol{\theta}^{n-1}}{\Delta t} - \nu_2\Delta\boldsymbol{\theta}^n + \mathbf{y}^n \cdot \nabla\boldsymbol{\theta}^n = 0 & \text{in } \Omega, \\ \nu_2\frac{\partial\boldsymbol{\theta}^n}{\partial\vec{n}} = \mathbf{v}^n\chi_{\Gamma_c} & \text{on } \Gamma. \end{cases}$$

We can see that solving the resulting subproblems (3.26)-(3.28) is not difficult. However, in a similar way as what we have done for deriving (3.25), we found that it is challenging to obtain the adjoint equations associated with the time-discretized equations (3.26)-(3.28). A particular reason is that some terms arise in different time intervals (e.g., $\beta\nu_1\Delta\tilde{\mathbf{y}}^n$ and $\nabla\tilde{\mathbf{p}}^n$ arise in both $[t^{n-1}, t^{n-1} + \Delta t/2]$ and $[t^{n-1} + \Delta t/2, t^n]$), which makes their adjoint variables difficult to be determined. The same concerns apply also to the Douglas-Rachford splitting method [16] and the θ -scheme [20, 21]. By contrast, there is no coupled term between different time intervals in our proposed scheme (3.3)-(3.5), and thus no particular difficulty is introduced in deriving the corresponding adjoint equation as shown in (3.7)-(3.25).

Using the notations introduced in Section 2.1, it is easy to show that the variational formulations of the time-discretized state equations (3.2)-(3.5) read as, for $n = 1, 2, \dots, N$, find $(\mathbf{y}^n, \tilde{\mathbf{y}}^n, \mathbf{p}^n, \boldsymbol{\theta}^n) \in [H^1(\Omega)]^2 \times [H_0^1(\Omega)]^2 \times P \times H^1(\Omega)$, such that

$$\begin{cases} \left(\frac{\tilde{\mathbf{y}}^n - \mathbf{y}^{n-1}}{\Delta t}, \varphi \right) - b(\mathbf{p}^{n-1}, \varphi) + a(\tilde{\mathbf{y}}^n, \varphi) + c(\mathbf{y}^{n-1}, \tilde{\mathbf{y}}^n, \varphi) = (\boldsymbol{\theta}^n \mathbf{e}_2, \varphi), \quad \forall \varphi \in [H_0^1(\Omega)]^2, \\ \left(\frac{\boldsymbol{\theta}^n - \boldsymbol{\theta}^{n-1}}{\Delta t}, \psi \right) + d(\boldsymbol{\theta}^n, \psi) + e(\mathbf{y}^{n-1}, \boldsymbol{\theta}^n, \psi) = (\mathbf{v}^n \chi_{\Gamma_c}, \psi)_\Gamma, \quad \forall \psi \in H^1(\Omega), \\ \left(\frac{\mathbf{y}^n - \tilde{\mathbf{y}}^n}{\Delta t}, \varphi \right) - b(\mathbf{p}^n - \mathbf{p}^{n-1}, \varphi) = 0, \quad \forall \varphi \in [H^1(\Omega)]^2, \\ b(q, \mathbf{y}^n) = 0, \quad \forall q \in P, \\ \mathbf{y}^0 = \mathbf{y}_0, \quad \boldsymbol{\theta}^0 = \boldsymbol{\theta}_0. \end{cases}$$

Similarly, the variational formulations of the adjoint equations (3.21)-(3.24) are then given by: for $n = N, N-1, \dots, 1$, find $(\mathbf{z}^n, \tilde{\mathbf{z}}^n, \boldsymbol{\xi}^n, \boldsymbol{\zeta}^n) \in [H^1(\Omega)]^2 \times [H_0^1(\Omega)]^2 \times P \times H^1(\Omega)$, such that

$$\begin{cases} \left(\frac{\mathbf{z}^n - \tilde{\mathbf{z}}^{n+1}}{\Delta t}, \varphi \right) + b(\boldsymbol{\xi}^n, \varphi) + c(\varphi, \tilde{\mathbf{y}}^{n+1}, \tilde{\mathbf{z}}^{n+1}) \\ \quad + e(\varphi, \boldsymbol{\theta}^{n+1}, \boldsymbol{\zeta}^{n+1}) = (\mathbf{y}^n - \mathbf{y}_d^n, \varphi), \quad \forall \varphi \in [H^1(\Omega)]^2, \\ -b(q, \mathbf{z}^n + \tilde{\mathbf{z}}^{n+1} - \mathbf{z}^{n+1}) = 0, \quad \forall q \in P, \\ \left(\frac{\tilde{\mathbf{z}}^n - \mathbf{z}^n}{\Delta t}, \varphi \right) + a(\tilde{\mathbf{z}}^n, \varphi) + c(\mathbf{y}^{n-1}, \varphi, \tilde{\mathbf{z}}^n) = 0, \quad \forall \varphi \in [H_0^1(\Omega)]^2, \\ \left(\frac{\boldsymbol{\zeta}^n - \boldsymbol{\zeta}^{n+1}}{\Delta t}, \psi \right) + d(\boldsymbol{\zeta}^n, \psi) + e(\mathbf{y}^{n-1}, \psi, \boldsymbol{\zeta}^n) = (\tilde{\mathbf{z}}^n \cdot \mathbf{e}_2, \psi), \quad \forall \psi \in H^1(\Omega), \\ \tilde{\mathbf{z}}^{N+1} = 0, \quad \mathbf{z}^{N+1} = 0, \quad \boldsymbol{\zeta}^{N+1} = 0. \end{cases}$$

3.3. Space discretization. For the space discretization of problem (3.1)-(3.5), we employ the Bercovier-Pironneau finite element pair [7] (a.k.a P_1 - P_1 iso P_2 finite element). More concretely, the velocity \mathbf{y} and the pressure p are approximated by conforming linear finite elements with the mesh sizes h and $2h$, respectively.

For the approximation of the temperature θ and the control v , we use the same finite element space as the one used for the velocity y .

For simplicity, we suppose from now on that Ω is a polygonal domain of \mathbb{R}^2 (or has been approximated by a family of such domains). Let \mathcal{T}_H be a classical triangulation of Ω , with H the largest length of the edges of the triangles of \mathcal{T}_H . From \mathcal{T}_H we construct \mathcal{T}_h with $h = H/2$ by joining the mid-points of the edges of the triangles of \mathcal{T}_H . We consider three finite element spaces V_h , V_{0h} and P_H defined by

$$\begin{aligned} V_h &= \{\varphi_h | \varphi_h \in C^0(\bar{\Omega}); \varphi_h|_{\mathbb{T}} \in \mathcal{P}_1, \forall \mathbb{T} \in \mathcal{T}_h\}, \\ V_{0h} &= \{\varphi_h | \varphi_h \in V_h, \varphi_h|_{\Gamma} = 0\} := V_h \cap H_0^1(\Omega), \\ P_H &= \{q_H | q_H \in C^0(\bar{\Omega}); q_H|_{\mathbb{T}} \in \mathcal{P}_1, \forall \mathbb{T} \in \mathcal{T}_H; \int_{\Omega} q_H dx = 0\}, \end{aligned}$$

with \mathcal{P}_1 the space of the polynomials of two variables of degree ≤ 1 .

With the above finite element spaces, the fully discretized formulation of problem (1.3)-(1.4) reads as

$$(3.29) \quad \begin{cases} \mathbf{u}_h \in [V_h]^N, \\ J_h^{\Delta t}(\mathbf{u}_h) \leq J_h^{\Delta t}(\mathbf{v}_h), \quad \forall \mathbf{v}_h \in [V_h]^N, \end{cases}$$

with the time-discretized objective functional

$$(3.30) \quad J_h^{\Delta t}(\mathbf{v}_h) = \frac{1}{2} \Delta t \sum_{n=1}^N \int_{\Omega} |\mathbf{y}_h^n - y_{dh}^n|^2 dx + \frac{\alpha}{2} \Delta t \sum_{n=1}^N \int_{\Gamma_c} |v_h^n|^2 dx,$$

where $(\mathbf{y}_h^n)_{n=1}^N$ are given from $(\mathbf{v}_h^n)_{n=1}^N$ by the solution of the following fully discretized state equations: for $n = 1, 2, \dots, N$, find $(\mathbf{y}_h^n, \tilde{\mathbf{y}}_h^n, \mathbf{p}_h^n, \boldsymbol{\theta}_h^n) \in [V_h]^2 \times [V_{0h}]^2 \times P_H \times V_h$, such that

$$(3.31) \quad \begin{cases} \left(\frac{\tilde{\mathbf{y}}_h^n - \mathbf{y}_h^{n-1}}{\Delta t}, \varphi_h \right) - b(\mathbf{p}_H^{n-1}, \varphi_h) + a(\tilde{\mathbf{y}}_h^n, \varphi_h) + c(\mathbf{y}_h^{n-1}, \tilde{\mathbf{y}}_h^n, \varphi_h) = (\boldsymbol{\theta}_h^n \mathbf{e}_2, \varphi_h), \quad \forall \varphi_h \in [V_{0h}]^2, \\ \left(\frac{\boldsymbol{\theta}_h^n - \boldsymbol{\theta}_h^{n-1}}{\Delta t}, \psi_h \right) + d(\boldsymbol{\theta}_h^n, \psi) + e(\mathbf{y}_h^{n-1}, \boldsymbol{\theta}_h^n, \psi_h) = (\mathbf{v}_h^n \chi_{\Gamma_c}, \psi_h)_{\Gamma}, \quad \forall \psi_h \in V_h, \\ \left(\frac{\mathbf{y}_h^n - \tilde{\mathbf{y}}_h^n}{\Delta t}, \varphi_h \right) - b(\mathbf{p}_H^n - \mathbf{p}_H^{n-1}, \varphi_h) = 0, \quad \forall \varphi_h \in [V_h]^2, \\ b(q_H, \mathbf{y}_h^n) = 0, \quad \forall q_H \in P_H, \\ \mathbf{y}_h^0 = y_{0h}, \quad \boldsymbol{\theta}_h^0 = \theta_{0h}. \end{cases}$$

REMARK 3.4. *The fully discretized state equations corresponding to the scheme (3.3), (3.4) and (3.6) read as: for $n = 1, 2, \dots, N$, find $(\mathbf{y}_h^n, \tilde{\mathbf{y}}_h^n, \mathbf{p}_h^n, \boldsymbol{\theta}_h^n) \in [V_{0h}]^2 \times [V_{0h}]^2 \times P_H \times V_h$, such that*

$$(3.32) \quad \begin{cases} \left(\frac{\tilde{\mathbf{y}}_h^n - \mathbf{y}_h^{n-1}}{\Delta t}, \varphi_h \right) - b(\mathbf{p}_H^{n-1}, \varphi_h) + a(\tilde{\mathbf{y}}_h^n, \varphi_h) + c(\mathbf{y}_h^{n-1}, \tilde{\mathbf{y}}_h^n, \varphi_h) = (\boldsymbol{\theta}_h^n \mathbf{e}_2, \varphi_h), \quad \forall \varphi_h \in [V_{0h}]^2, \\ \left(\frac{\boldsymbol{\theta}_h^n - \boldsymbol{\theta}_h^{n-1}}{\Delta t}, \psi_h \right) + d(\boldsymbol{\theta}_h^n, \psi) + e(\mathbf{y}_h^{n-1}, \boldsymbol{\theta}_h^n, \psi_h) = (\mathbf{v}_h^n \chi_{\Gamma_c}, \psi_h)_{\Gamma}, \quad \forall \psi_h \in V_h, \\ \left(\frac{\mathbf{y}_h^n - \tilde{\mathbf{y}}_h^n}{\Delta t}, \varphi_h \right) - b(\mathbf{p}_H^n - \mathbf{p}_H^{n-1}, \varphi_h) = 0, \quad \forall \varphi_h \in [V_{0h}]^2, \\ b(q_H, \mathbf{y}_h^n) = 0, \quad \forall q_H \in P_H, \\ \mathbf{y}_h^0 = y_{0h}, \quad \boldsymbol{\theta}_h^0 = \theta_{0h}. \end{cases}$$

In a similar way as what we have done in Section 3, it is easy to derive the first-order optimality conditions for the fully discretized optimal control problem (3.29)-(3.30) constrained by (3.31) or (3.32). For succinctness, we omit the details.

4. A BFGS method for problem (3.29)-(3.31). Quasi-Newton methods are milestones in the development of numerical optimization, and they are particularly efficient for large-scale problems. The common idea of quasi-Newton methods is approximating the Hessian or its inverse with only gradient information. Thanks to its nice properties and excellent performance, the BFGS method is the most representative and widely-used quasi-Newton algorithm, see, e.g. [39]. In this section, we shall discuss the application of the BFGS method to the solution of the fully discretized optimal control problem (3.29)-(3.31), and propose an easily implementable algorithm tailored to the structure of (3.29)-(3.31). For ease of notations, the subscripts h in all variables are dropped.

4.1. A generic BFGS method. Conceptually, implementing the BFGS to problem (3.29)-(3.31), we readily obtain Algorithm 4.1.

Algorithm 4.1 A generic BFGS method for the solution of problem (3.29)-(3.31)

- 1: Given an initial point \mathbf{u}_0 , an initial inverse Hessian approximation \mathcal{H}_0 .
- 2: Compute $\mathbf{g}_0 = DJ_h^{\Delta t}(\mathbf{u}_0)$; If $\mathbf{g}_0 = 0$, then $\mathbf{u} = \mathbf{u}_0$, else
- 3: **while** not converged **do**
- 4: Compute the search direction $\mathbf{d}_k = -\mathcal{H}_k \mathbf{g}_k$ and a step size ρ_k .
- 5: Update $\mathbf{u}_{k+1} = \mathbf{u}_k + \rho_k \mathbf{d}_k$ and $\mathbf{g}_{k+1} = DJ_h^{\Delta t}(\mathbf{u}_{k+1})$.
- 6: Compute $\delta \mathbf{u}_k = \mathbf{u}_{k+1} - \mathbf{u}_k$ and $\delta \mathbf{g}_k = \mathbf{g}_{k+1} - \mathbf{g}_k$.
- 7: Update \mathcal{H}_{k+1} via

$$\mathcal{H}_{k+1} = V_k^T \mathcal{H}_k V_k + \eta_k \delta \mathbf{u}_k \delta \mathbf{u}_k^T,$$

$$\text{where } V_k = I - \eta_k \delta \mathbf{g}_k \delta \mathbf{u}_k^T, \text{ and } \eta_k = \frac{1}{\delta \mathbf{g}_k^T \delta \mathbf{u}_k}.$$

- 8: **end while**
-

The direct application of the BFGS method to problem (3.29)-(3.31), however, is not practically implementable. In particular, it is necessary to specify the strategies for determining an appropriate step size ρ_k and a practical inverse Hessian approximation \mathcal{H}_k at each iteration. Recall that the discretized problem (3.29)-(3.31) is high-dimensional and the resulting Hessian is usually dense. It is thus necessary to find efficient strategies for determining the step size ρ_k and the inverse Hessian approximation \mathcal{H}_k to implement the BFGS method or its variants. Because of the huge dimensionality and thus the demanding requirement of memory in computation, we focus on the classic limited-memory BFGS (L-BFGS) method in [38] and elaborate on these issues in the following part of this section.

4.2. Computation of the step size ρ_k . A crucial step to implement Algorithm 4.1 is the computation of an appropriate step size ρ_k . That is, finding ρ to substantially reduce the univariate function

$$P_k(\rho) := J_h^{\Delta t}(\mathbf{u}_k + \rho \mathbf{d}_k),$$

where $\mathbf{u}_k, \mathbf{d}_k \in [V_h]^N$ are known. To this end, a natural idea is to employ some line search strategies, such as the backtracking strategy based on the Armijo–Goldstein condition or the Wolf condition, see e.g., [39]. It is worth noting that these line search strategies require the evaluation of $J_h^{\Delta t}(\mathbf{v})$ repeatedly, which is computationally expensive because every evaluation of $J_h^{\Delta t}(\mathbf{v})$ for a given $\mathbf{v} \in [V_h]^N$ requires solving the state equation (3.31). To address this issue, we advocate the following step size seeking strategy, and a similar idea can also be found in [24].

We consider linearizing the fully discretized state equation (3.31) to find an appropriate step size. Recall that in the fully discretized optimal control problem (3.29)-(3.31), the states $(\mathbf{y}, \tilde{\mathbf{y}}, \boldsymbol{\theta}) = (\mathbf{y}^n, \tilde{\mathbf{y}}^n, \boldsymbol{\theta}^n)_{n=1}^N$ are given from $\mathbf{v} = (\mathbf{v}^n)_{n=1}^N$ by the solution of the fully discretized state equation (3.31). We introduce the control-to-state operator $S_h^{\Delta t}$ associated with the equation (3.31), which maps \mathbf{v} to $\mathbf{y} = S_h^{\Delta t}(\mathbf{v})$. Then the first-order approximation of mapping $\rho \rightarrow S_h^{\Delta t}(\mathbf{u}_k + \rho \mathbf{d}_k)$ at $\rho = 0$ is

$$S_h^{\Delta t}(\mathbf{u}_k) + \rho S_h^{\Delta t'}(\mathbf{u}_k) \mathbf{d}_k := \mathbf{y}_k + \rho \mathbf{w}_k,$$

where \mathbf{w}_k are given from \mathbf{u}_k , \mathbf{d}_k and $(\mathbf{y}_k, \tilde{\mathbf{y}}_k, \boldsymbol{\theta}_k)$ by the solutions of the following equations:

$$\left\{ \begin{array}{l} \mathbf{w}_k^0 = 0, \quad \phi_k^0 = 0 \text{ and } \mathbf{q}_k^0 = 0, \\ \int_{\Omega} \frac{\tilde{\mathbf{w}}_k^n - \mathbf{w}_k^{n-1}}{\Delta t} \varphi_h dx + \int_{\Omega} \nu_1 \nabla \tilde{\mathbf{w}}_k^n \cdot \nabla \varphi_h dx \\ \quad + \int_{\Omega} ((\mathbf{y}_k^{n-1} \cdot \nabla) \tilde{\mathbf{w}}_k^n + (\mathbf{w}_k^{n-1} \cdot \nabla) \tilde{\mathbf{y}}_k^n + \nabla \mathbf{q}_k^{n-1}) \varphi_h dx = \int_{\Omega} \phi_k^n \mathbf{e}_2 \cdot \varphi_h dx, \forall \varphi_h \in [V_{0h}]^2, \\ \int_{\Omega} \frac{\phi_k^n - \phi_k^{n-1}}{\Delta t} \psi_h dx + \nu_2 \int_{\Omega} \nabla \phi_k^n \cdot \nabla \psi_h dx + \int_{\Omega} (\mathbf{w}_k^{n-1} \cdot \nabla \boldsymbol{\theta}_k^n + \mathbf{y}_k^{n-1} \cdot \nabla \phi_k^n) \psi_h dx = \int_{\Gamma_c} \mathbf{d}_k^n \psi_h dx, \forall \psi_h \in V_h, \\ \int_{\Omega} \frac{\mathbf{w}_k^n - \tilde{\mathbf{w}}_k^n}{\Delta t} \varphi_h dx + \int_{\Omega} \nabla (\mathbf{q}_k^n - \mathbf{q}_k^{n-1}) \varphi_h dx = 0, \forall \varphi_h \in [V_h]^2, \\ \int_{\Omega} \nabla \cdot \mathbf{w}_k^n \mathbf{q}_H dx = 0, \forall \mathbf{q}_H \in P_H. \end{array} \right.$$

Consequently, the following quadratic function

$$Q_k(\rho) = \frac{1}{2} \Delta t \sum_{n=1}^N \int_{\Omega} |\rho \mathbf{w}_k^n + \mathbf{y}_k^n - \mathbf{y}_d^n|^2 dx + \frac{\alpha}{2} \Delta t \sum_{n=1}^N \int_{\Gamma_c} |\mathbf{u}_k^n + \rho \mathbf{d}_k^n|^2 dx$$

is an approximation of $P_k(\rho)$. Then it is easy to show that the minimizer of $Q_k(\rho)$ is

$$(4.1) \quad \hat{\rho}_k = - \frac{\Delta t \sum_{n=1}^N \int_{\Omega} \mathbf{g}_k^n \cdot \mathbf{d}_k^n dx}{\Delta t \sum_{n=1}^N \int_{\Omega} |\mathbf{w}_k^n|^2 dx + \alpha \Delta t \sum_{n=1}^N \int_{\Gamma_c} |\mathbf{d}_k^n|^2 dx},$$

where $(\mathbf{g}_k^n)_{n=1}^N = D J_h^{\Delta t}(\mathbf{u}_k)$. We take step size $\rho_k = \hat{\rho}_k$, which is indeed an approximate minimizer of $P_k(\rho)$.

4.3. Computation of the search direction \mathbf{d}_k . As mentioned in Section 4.1, due to the requirement of large memory, it is more appropriate to consider the L-BFGS method, which only requires storing a sequence of vectors $\delta \mathbf{u}_k$ and $\delta \mathbf{g}_k$ from the most recent iterations to compute $\mathcal{H}_k \mathbf{g}_k$ without constructing \mathcal{H}_k explicitly. It turns out that the L-BFGS method is fairly robust, inexpensive, and easy-to-implement, see e.g., [38, 39]. For the reader's convenience, we present a two-loop recursive procedure to efficiently compute the product $\mathcal{H}_k \mathbf{g}_k$ in Algorithm 4.2.

Algorithm 4.2 Computation of search direction

```

Set  $\hat{\mathbf{d}} = -\mathbf{g}_k (= -D J_h^{\Delta t}(\mathbf{u}^k))$ ;
for  $i = k-1, \dots, k-m$  do
     $\eta_i = \frac{1}{\delta \mathbf{g}_i^T \delta \mathbf{u}_i}$ ,  $\tau_i = \eta_i \delta \mathbf{u}_i^T \hat{\mathbf{d}}$ 
     $\hat{\mathbf{d}} = \hat{\mathbf{d}} - \tau_i \delta \mathbf{g}_i$ 
end for
Compute  $\mathbf{d} = \mathcal{H}_k^0 \hat{\mathbf{d}}$ 
for  $i = k-m, k-m+1, \dots, k-1$  do
     $\kappa = \eta_i \delta \mathbf{g}_i^T \mathbf{d}$ 
     $\mathbf{d} = \mathbf{d} + \delta \mathbf{u}_i (\tau_i - \kappa)$ 
end for
return  $\mathbf{d} = -\mathcal{H}_k \mathbf{g}_k$ 

```

Note that, in contrast to the standard BFGS iteration, the initial inverse Hessian approximation \mathcal{H}_k^0 is allowed to vary from iteration to iteration. In our numerical implementation, we choose $\mathcal{H}_k^0 = I$ for simplicity.

4.4. An easily implementable L-BFGS method for the solution of problem (3.29)-(3.31).

With the discussions in Sections 4.2 and 4.3, we obtain an easily implementable L-BFGS method as listed in Algorithm 4.3 for the solution of problem (3.29)-(3.31).

Algorithm 4.3 An easily implementable L-BFGS method for the solution of problem (3.29)-(3.31)

Given an initial point \mathbf{u}_0 .
 Compute $\mathbf{g}_0 = DJ_h^{\Delta t}(\mathbf{u}_0)$; If $\mathbf{g}_0 = 0$, then $\mathbf{u} = \mathbf{u}_0$, else for $k \geq 0$
while not converged **do**
 Choose \mathcal{H}_k^0 ;
 Compute search direction \mathbf{d}_k by Algorithm 4.2:
 Compute a step size ρ_k by (4.1).
 Update $\mathbf{u}_{k+1} = \mathbf{u}_k + \rho_k \mathbf{d}_k$ and $\mathbf{g}_{k+1} = DJ_h^{\Delta t}(\mathbf{u}_{k+1})$
 if $k > m$ **then**
 Discard the vector pair $\{\delta \mathbf{u}_{k-m}, \delta \mathbf{g}_{k-m}\}$ from storage
 end if
 Compute and save $\delta \mathbf{u}_k = \mathbf{u}_{k+1} - \mathbf{u}_k$ and $\delta \mathbf{g}_k = \mathbf{g}_{k+1} - \mathbf{g}_k$
end while

5. Numerical results. In this section, we report some preliminary numerical results to validate the efficiency of our proposed numerical approach for solving the optimal control problem (1.3) with the objective function (1.4) or (1.5). In particular, we test two numerical examples for velocity-tracking and vorticity reduction by controlling the heat flux. All codes were written in MATLAB R2020b and numerical experiments were conducted on a MacBook Pro with mac OS Monterey, Intel(R) Core(TM) i7-9570h (2.60 GHz), and 16 GB RAM. For simplicity, all the linear equations originated from the numerical discretization are solved by the default backslash operator in Matlab.

Example 1. We consider the tracking-type optimal control of the Boussinesq equations (1.1)-(1.2) with the objective functional (1.4). The boundary and initial conditions are specified as

$$\begin{aligned} y &= 0 \text{ on } \Gamma \times (0, T), & y(0) &= 0. \\ \begin{cases} \theta = 0 & \text{on } \Gamma/\Gamma_c \times (0, T), \\ \nu_2 \frac{\partial \theta}{\partial \vec{n}} = v & \text{on } \Gamma_c \times (0, T), \end{cases} & \theta(0) &= 0. \end{aligned}$$

Furthermore, we set that

- the regularization parameter is $\alpha = 5 \times 10^{-5}$ and the coefficients are $\nu_1 = 1/100$ and $\nu_2 = 1/72$;
- the domain is $\Omega = (0, 1)^2$, $T = 5$, and the control acts on $\Gamma_c = \Gamma_l \cup \Gamma_r$ as shown in Figure 5.1 (left);
- the target velocity $y_d = (y_{1,d}, y_{2,d})^T$ is time-independent and given by

$$\begin{cases} y_{1,d}(x_1, x_2, t) = 100\phi(x_1)\phi'(x_2), \\ y_{2,d}(x_1, x_2, t) = -100\phi'(x_1)\phi(x_2), \end{cases} \quad \forall (x_1, x_2) \in \Omega, t \in (0, T)$$

where $\phi(z) = z^2(z-1)^2$ and $\nabla \cdot y_d = 0$. A schematic of the target velocity y_d is shown in Figure 5.1 (right).

If there is no control in Example 1, i.e., the control is set as $v = 0$, then the states y and θ are zeros in $(0, T)$. We aim at tracking a time-independent pinwheel-shape velocity field y_d from static initial states $y(0) = 0$ and $\theta(0) = 0$ by controlling the heat flux v on the side walls Γ_c of the square domain Ω . For this purpose, one can generate a buoyancy-driven swirling flow whose velocity field y is close to the target field y_d during $(0, T)$.

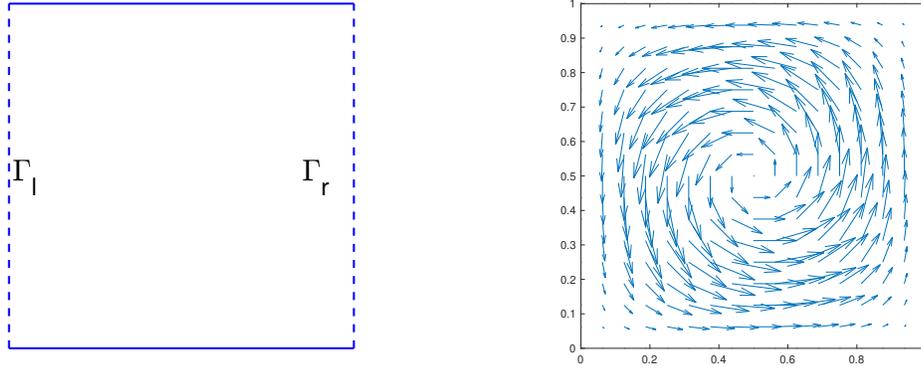


FIG. 5.1. The schematics of $\Gamma_c = \Gamma_l \cup \Gamma_r$ (left) and the target velocity y_d (right) for Example 1..

We implement Algorithm 4.3 to Example 1 with mesh size $h = 1/64$ for the space discretization and step size $\Delta t = 1/64$ for the time discretization. Moreover, we store the most recent vector pair $\{\delta \mathbf{u}_{k-m}, \delta \mathbf{g}_{k-m}\}$ during the L-BFGS iterations and terminate it when the following stopping criterion is satisfied:

$$\frac{\|DJ(\mathbf{u}_h^k)\|}{\|DJ(\mathbf{u}_h^0)\|} < 5 \times 10^{-3}.$$

The numerical results for Example 1 are displayed in Figures 5.2, 5.3, and 5.4.

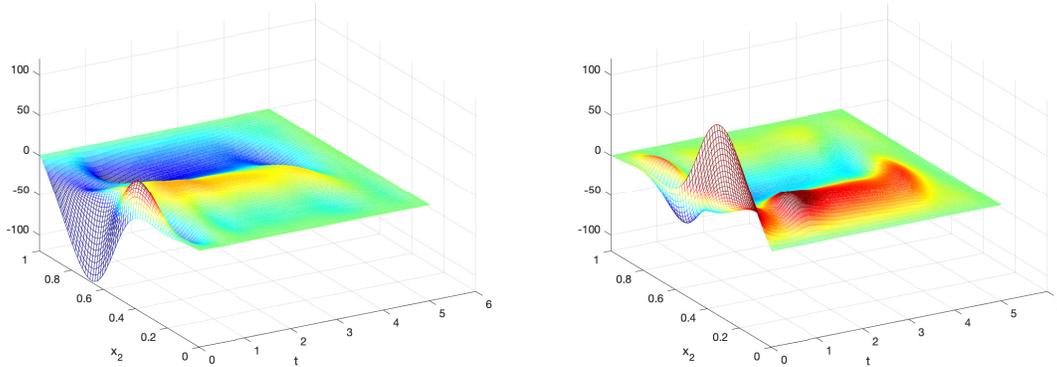


FIG. 5.2. The computed optimal control u on Γ_l (left) and Γ_r (right) for Example 1.

We observe from Figure 5.3 that the objective function values decrease rapidly, which implies fast convergence and hence high efficiency of our proposed Algorithm 4.3. The relative error $\|y(t; u) - y_d(t)\|/\|y_d(t)\|$ becomes small after about one second, which indicates that the velocity field gets close to the target field, and this observation coincides with that in Figure 5.4.

The computed velocity and temperature at different instants of time for Example 1 are displayed in Figure 5.4, from which we can observe how the buoyancy drives a swirling fluid and how the velocity influences the temperature. To be concrete, at the early stage of the control process ($t = 0.016, 0.25, 0.5, 0.75$), the temperature and the velocity field begin to change under the action of the optimal control. Two swirling flows are formed on the left and right sides and merge into a large swirling flow. Then, at $t = 1, 2, 3, 4$, both

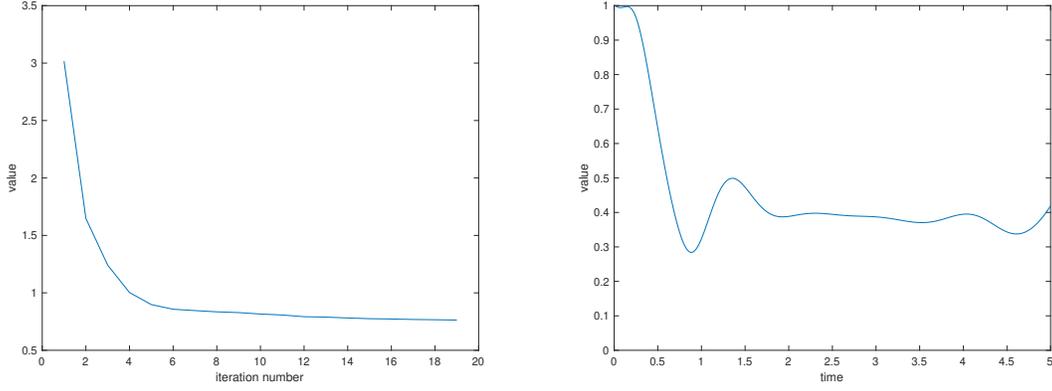


FIG. 5.3. Numerical results for Example 1. Left: the objective functional values with respect to L -BFGS iterations. Right: relative errors $\frac{\|y(t;u)-y_d(t)\|}{\|y_d(t)\|}$ at different time t .

the velocity and the temperature become steady and the velocity field is close to the target field y_d . In the final stage ($t = 4.25, 4.5, 4.75, 5$), the control is gradually removed and the temperature gradually goes to 0, but the computed velocity field remains close to the target field y_d .

Example 2. This example is extended from [31], which is motivated by the transport process in high-pressure chemical vapor deposition (CVD) reactors. The original example focuses on the steady case and we extend it to an unsteady example in $(0, T)$ with $T = 15$.

A typical vertical reactor Ω , shown in Figure 5.5, is a classical configuration for the growth of compound semiconductors by metalorganic vapor phase epitaxy. The geometry of the prototype reactors has two outlet portions, Γ_o , and an inlet, Γ_i , whose widths are $1/3$. The size of the susceptor region Γ_0 and that of the side walls Γ_l and Γ_r are 1; the height of the inlet port Γ_1 is $1/3$. The reactant gases enter the reactor from Γ_i and flow down to the substrate Γ_0 which is kept at a high temperature. This means that the least dense gas is closest to the substrate and the flow is likely to be affected by buoyancy-driven convection. To have uniform growth rates and better compositional variations, it is crucial to have a flow field without recirculation.

To achieve the aforementioned goal, one can minimize the vorticity by controlling the temperature at the side walls $\Gamma_c = \Gamma_l \cup \Gamma_r$ in order to obtain a flow field without recirculation and thereby obtain better vertical transport. For this purpose, we consider to minimize the objective functional

$$J(v) = \frac{1}{2} \iint_Q |\nabla \times y|^2 dxdt + \frac{\alpha}{2} \iint_{\Sigma_c} v^2 dxdt,$$

where $y := y(v)$ is the solution of the state equation (1.1) complemented with the following initial and boundary conditions:

$$(5.1) \quad \begin{cases} y(0) = 0, \theta(0) = 0 \\ \Gamma_0 : y = (0, 0), \quad \theta = 1, \\ \Gamma_1 : y = (0, 0), \quad \theta = 0, \\ \Gamma_i : y = (0, -4(x_1 - 1/3)(2/3 - x_1)\phi(t)), \quad \theta = 0, \\ \Gamma_0 : \frac{\partial y}{\partial \vec{n}} = (0, 0), \quad \frac{\partial \theta}{\partial \vec{n}} = 0, \\ \Gamma_c, y = (0, 0), \quad \frac{\partial \theta}{\partial \vec{n}} + \theta = v. \end{cases}$$

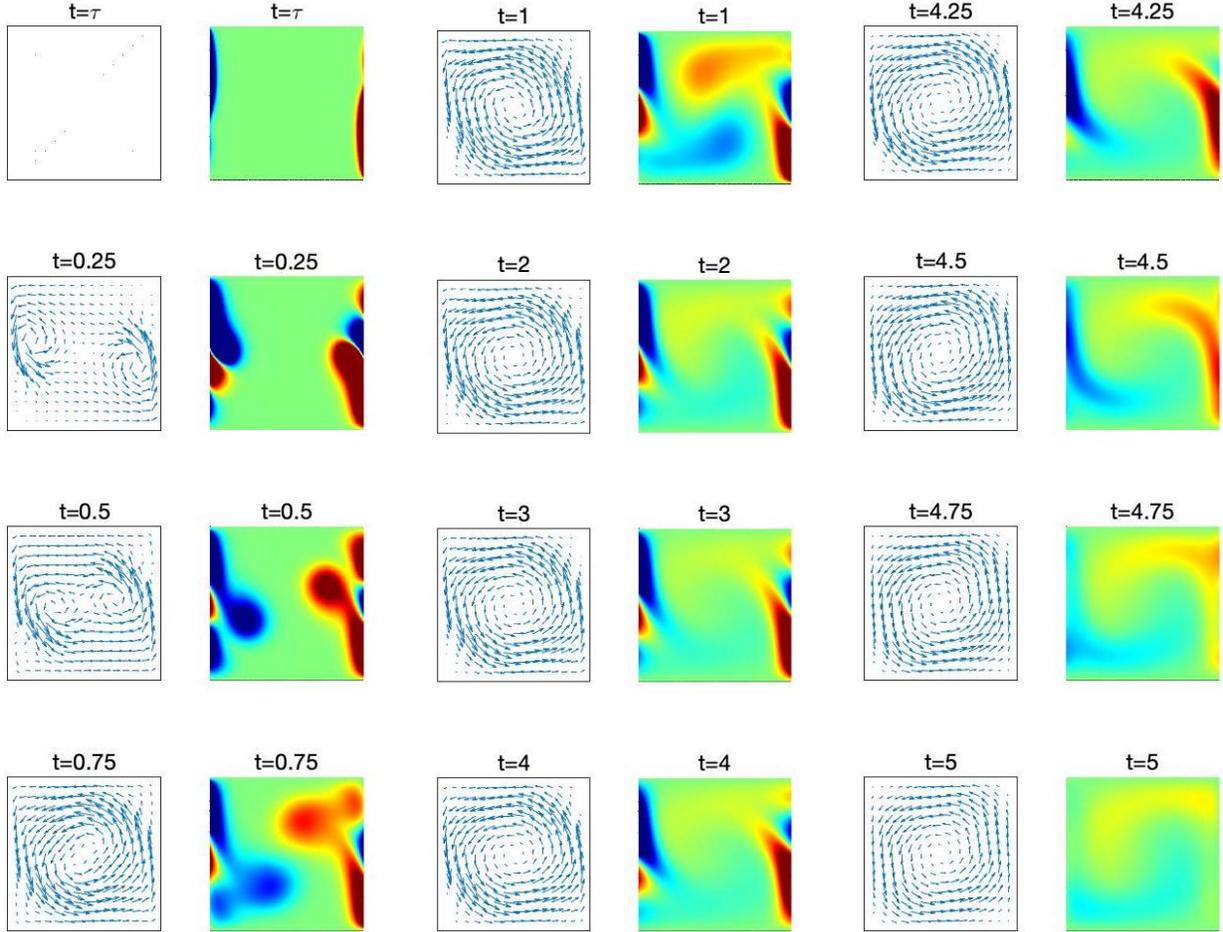


FIG. 5.4. Computed velocity field y (the 1st, 3rd, and 5th columns) and computed temperature θ (the 2nd, 4th, and 6th columns) at different time t for Example 1. Here, blue/red color means low/high temperature.

Here the function

$$\phi(t) := \begin{cases} t/5, & 0 < t \leq 5, \\ 1, & 5 < t < 10, \\ (15 - t)/5, & 10 \leq t < 15, \end{cases}$$

determines the maximum velocity of the fluid on Γ_i .

Throughout, the regularization parameter is $\alpha = 10^{-4}$ and the coefficients are $\nu_1 = 1/100$ and $\nu_2 = 1/72$ respectively. We use a uniform triangulation with mesh size $h = 1/64$ for the space discretization and step size $\Delta t = 1/64$ for the time discretization. We keep the $m = 5$ most recent vector pair $\{\delta \mathbf{u}_{k-i}, \delta \mathbf{g}_{k-i}\}_{i=1}^m$ during the L-BFGS iterations and terminate it if $\frac{\|DJ(\mathbf{u}_h^k)\|}{\|DJ(\mathbf{u}_h^0)\|} < 5 \times 10^{-3}$.

The numerical results obtained by Algorithm 4.3 for Example 2 are displayed in Figures 5.6, 5.7, and 5.8, where the notation "uncontrolled" means that there is no control (i.e. $v = 0$) acting in the state system specified by (1.1) and (5.1). We observe from Figure 5.6 that the computed controls on the left and right sides are the same at any time, which is due to the symmetry structure of the example under investigation. From Figure 5.7, we see that the objective function values decrease rapidly which implies fast convergence of Algorithm 4.3. Moreover, we observe that the vorticity $\|\nabla \times y(t, u)\|$ of the controlled system goes to zeros

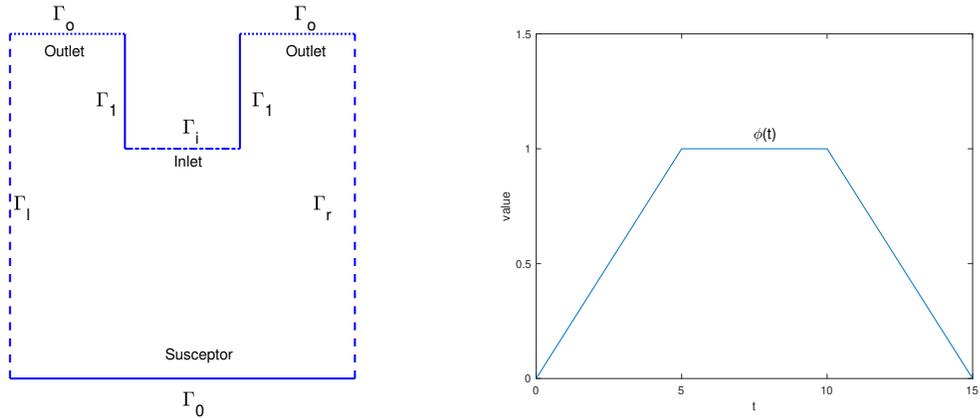


FIG. 5.5. The schematics of the domain Ω (left) and the graph of $\phi(t)$ (right) for Example 2. a) The reactant gases entered the reactor from Γ_i , then flow down to the substrate Γ_0 which is kept at high temperature, and finally go out from Γ_o . b) The function $\phi(t)$ determines the maximum velocity of the fluid on Γ_i . It increases linearly from zero to one during $(0, 5)$, then remains unchanged during $(5, 10)$, and finally decreases linearly to zero during $(10, 15)$.

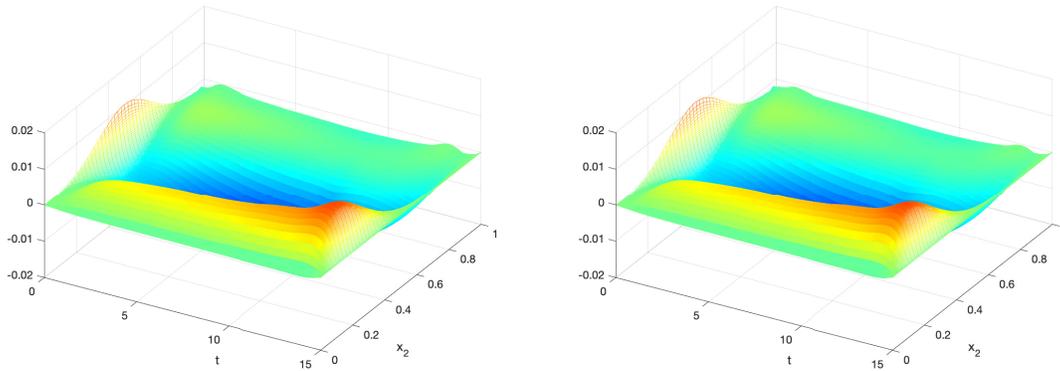


FIG. 5.6. The computed optimal control u on Γ_l (left) and Γ_r (right) for Example 2.

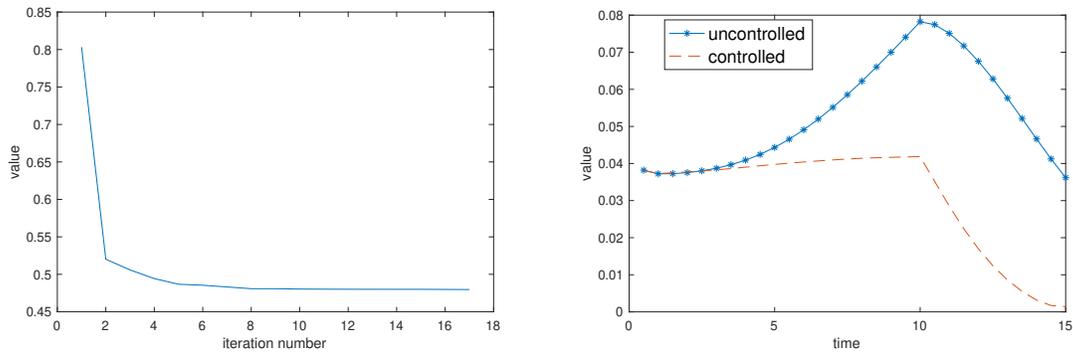


FIG. 5.7. Numerical results for Example 2. Left: the objective functional values with respect to L-BFGS iterations. Right: the vorticity values $\|\nabla \times y(t; u)\|$ at different times t .

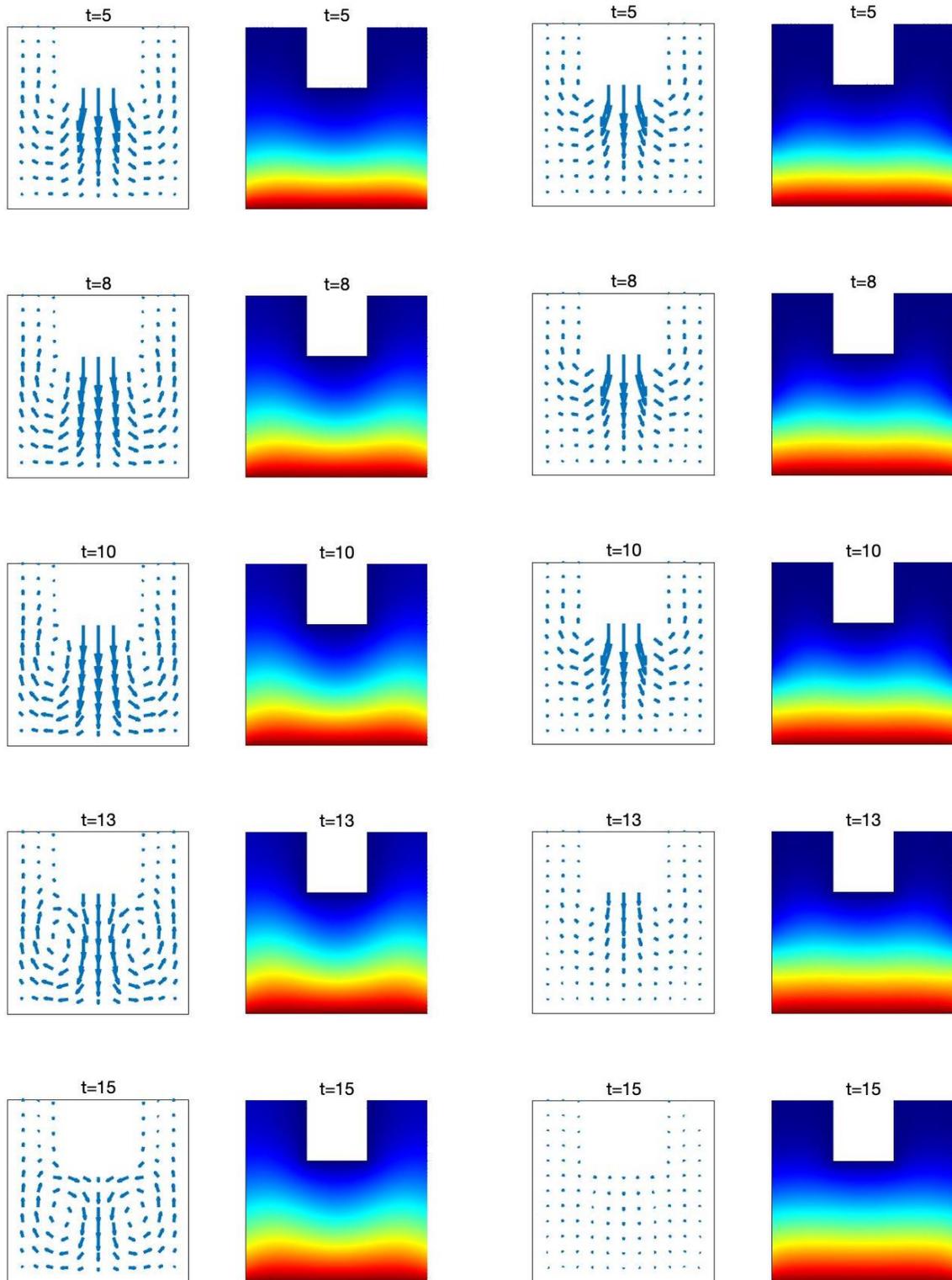


FIG. 5.8. Computed velocity field y (the 1st and 3rd columns) and temperature θ (the 2nd and 4th columns) of the uncontrolled (left) and controlled (right) system at different t for Example 2.

eventually. We observe from Figure 5.8 that swirling flow appears near the susceptor for the uncontrolled system while there is no swirling flow for the controlled system; and the difference is more discernible after $t = 10$. This implies that the computed control works very well and it indeed leads to a flow field without recirculation. This, together with the results shown in Figure 5.7, validates a significant vorticity reduction by the computed control.

6. Conclusions. In this paper, we proposed an efficient numerical approach to the optimal control of thermally convective flows, which can be mathematically modeled as optimally controlling the Boussinesq equations consisting of the Navier-Stokes equations for incompressible viscous flow coupled with an advection-diffusion equation for temperature. Our main techniques included the Marchuk-Yanenko operator splitting method for the time discretization to untie the advection-diffusion equation from the Navier-Stokes component and to decouple the nonlinearity from the incompressibility condition. Computing the gradient of the objective functional became possible, and it was reduced to solving four easy linear advection-diffusion equations and two degenerated Stokes equations at each time step. With the Bercovier-Pironneau finite element method for space discretization, we also proposed an efficient and easily implementable BFGS method with limited memory to solve the fully discretized optimal control problem. The proposed algorithm appears to be the first efficient numerical approach to the optimal control of unsteady thermally convective flows.

We focused on the numerical study for optimal control problems modeled by the Boussinesq equations, and its validated efficiency clearly justifies the necessity to investigate the underlying theoretical issues such as the convergence analysis for the proposed numerical approach and the error estimate for the operator splitting time discretization scheme. Moreover, we conjecture that our philosophies in algorithmic design and techniques for numerical implementation can be extended to other important optimal control problems modeled by coupled systems, such as the optimal mixing [29], the optimal control of coupled Cahn-Hilliard Navier-Stokes system [28], and the optimal control of a diffuse interface model for tumor growth [17].

Acknowledgments. The authors would like to thank Prof. Ming-Chih Lai for his valuable suggestions about the numerical experiments.

REFERENCES

- [1] F. Abergel and R. Temam, *On some control problems in fluid mechanics*, Theoretical and Computational Fluid Dynamics, 1 (1990), pp. 303–325.
- [2] G. Bärwolff, *Boundary control of a Boussinesq equation for a crystal growth problem – aspects of numerical solution*, International Journal of Computer Mathematics, 85 (2008), pp. 329–343.
- [3] G. Bärwolff and M. Hinze, *Optimization of semiconductor melts*, ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik: Applied Mathematics and Mechanics, 86 (2006), pp. 423–437.
- [4] G. Bärwolff and M. Hinze, *Analysis of optimal boundary control of the Boussinesq approximation*, Fachbereich Mathematik, Univ., [Department Mathematik], 2007.
- [5] A. Belmiloudi, *Regularity results and optimal control problems for the perturbation of the Boussinesq equations of the ocean*, Numerical Functional Analysis and Optimization, 21 (2000), pp. 623–651.
- [6] A. Belmiloudi, *Robin-type boundary control problems for the nonlinear Boussinesq type equations*, Journal of Mathematical Analysis and Applications, 273 (2002), pp. 428–456.
- [7] M. Bercovier and O. Pironneau, *Error estimates for finite element method solution of the Stokes problem in the primitive variables*. Numerische Mathematik, 33 (1979), pp. 211–224.
- [8] M. Berggren, *Numerical solution of a flow-control problem: vorticity reduction by dynamic boundary action*, SIAM Journal on Scientific Computing, 19 (1998), pp. 829–860.
- [9] B. Bermúdez and A. Nicolás, *An operator splitting numerical scheme for thermal/isothermal incompressible viscous flows*, International Journal for Numerical Methods in Fluids, 29 (1999), pp. 397–410.
- [10] J. Borggaard, J. A. Burns, A. Surana and L. Zietsman, *Control, estimation, and optimization of energy efficient buildings*, Proceedings of the 2009 American Control Conference, St. Louis, MO, (2009), pp. 837–841.
- [11] J. A. Burns and W. Hu, *Approximation methods for boundary control of the Boussinesq equations*, 52nd IEEE Conference on Decision and Control, IEEE, 2013, pp. 454–459.
- [12] J. A. Burns, X. He and W. Hu, *Feedback stabilization of a thermal fluid system with mixed boundary control*, Computers & Mathematics with Applications, 71 (2016), pp. 2170–2191.

- [13] J. A. Burns, B. B. King and D. Rubio, *Feedback control of thermal fluid using state estimation*, International Journal of Computational Fluid Dynamics, 11 (1998), pp. 93–112.
- [14] E. Casas, *The Navier-Stokes equations coupled with the heat equation: analysis and control*, Control Cybernet, 23 (1994), pp. 605–620.
- [15] A. J. Chorin, *Numerical solution of the Navier–Stokes equations*, Mathematics of Computation, 22 (1968), pp. 745–762.
- [16] J. Douglas and H. H. Rachford, *On the numerical solution of heat conduction problems in two and three space variables*, Transactions of the American Mathematical Society, 82 (1956), pp. 421–439.
- [17] M. Ebenbeck and P. Knopf, *Optimal control theory and advanced optimality conditions for a diffuse interface model of tumor growth*, ESAIM: Control, Optimisation and Calculus of Variations, 26 (2020), 71.
- [18] A. V. Fursikov and O. Y. Imanuvilov, *Exact controllability of the Navier-Stokes and Boussinesq equations*, Russian Mathematical Surveys, 54 (1999), p. 565–618.
- [19] H. Gajewski, *Zur iterativen lösung der zweidimensionalen boussinesq-gleichungen*, ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik, 55 (1975), pp. 571–581.
- [20] R. Glowinski, *Viscous flow simulations by finite element methods and related numerical techniques*, in Progress in Supercomputing in Computational Fluid Dynamics, E. M. Murman and S. S. Abarbanel, eds., Birkhäuser, Boston, 1985, pp. 173–210.
- [21] R. Glowinski, *Splitting methods for the numerical solution of the incompressible Navier-Stokes equations*, in Vistas in Applied Mathematics, A. V. Balakrishnan, A. A. Dorodnitsyn, and J.-L. Lions, eds., Optimization Software, New York, 1986, pp. 57–95.
- [22] R. Glowinski, *Finite Element Methods for Incompressible Viscous Flow*, Handbook of Numerical Analysis, 9 (2003), pp. 3–1176.
- [23] R. Glowinski and J. He, *On shape optimization and related issues*, In Computational Methods for Optimal Design and Control, J. Borggaard, J. Burns, E. Cliff and S. Schreck (eds.), Birkhäuser, Boston, MA, 1998, pp. 151–179.
- [24] R. Glowinski, Y. Song, X. Yuan and H. Yue, *Bilinear optimal control of an advection-reaction-diffusion system*. SIAM Review, 64 (2022), pp. 392–421.
- [25] J. L. Guermond, P. Mineev and J. Shen, *An overview of projection methods for incompressible flows*, Computer Methods in Applied Mechanics and Engineering, 195 (2006), pp. 6011–6045.
- [26] M. D. Gunzburger, *Perspectives in Flow Control and Optimization*. SIAM, Philadelphia, 2002.
- [27] M. D. Gunzburger and H. C. Lee, *Analysis, approximation, and computation of a coupled solid/fluid temperature control problem*, Computer Methods in Applied Mechanics and Engineering, 118 (1994), pp. 133–152.
- [28] M. Hintermüller and D. Wegner, *Optimal control of a semidiscrete Cahn–Hilliard–Navier–Stokes system*, SIAM Journal on Control and Optimization, 52 (2014), pp. 747–772.
- [29] W. Hu and X. Zheng, *Boundary control for optimal mixing via Stokes flows and numerical implementation*, arXiv preprint, arXiv:2108.09533, 2021.
- [30] K. Huhtala, L. Paunonen and W. Hu, *Robust output regulation of the linearized Boussinesq equations with boundary control and observation*, Mathematics of Control, Signals, and Systems, 34 (2022), pp. 361–391.
- [31] K. Ito and S. S. Ravindran, *Optimal control of thermally convected fluid flows*, SIAM Journal on Scientific Computing, 19 (1998), pp. 1847–1869.
- [32] H. Lee, *Analysis and computational methods of Dirichlet boundary optimal control problems for 2d Boussinesq equations*, Advances in Computational Mathematics, 19 (2003), pp. 255–275.
- [33] H. C. Lee and O. Y. Imanuvilov, *Analysis of optimal control problems for the 2-D stationary Boussinesq equations*, Journal of Mathematical Analysis and Applications, 242 (2000), pp. 191–211.
- [34] H. Lee and O. Y. Imanuvilov, *Analysis of Neumann boundary optimal control problems for the stationary Boussinesq equations including solid media*, SIAM Journal on Control and Optimization, 39 (2000), pp. 457–477.
- [35] J. L. Lions, *Optimal Control of Systems Governed by Partial Differential Equations (Grundlehren der Mathematischen Wissenschaften)*, Vol. 170, Springer Berlin, 1971.
- [36] G. I. Marchuk, *Splitting and Alternating Direction Methods*, in Handbook of Numerical Analysis, 1, pp. 197–462, 1990.
- [37] P. A. Nguyen and J. P. Raymond, *Pointwise control of the Boussinesq system*, Systems & Control Letters, 60 (2011), pp. 249–255.
- [38] J. Nocedal, *Updating quasi-Newton matrices with limited storage*, Mathematics of Computation, 35 (1980), pp. 773–782.
- [39] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, 2006.
- [40] D. H. Peaceman and H. H. Rachford, *The numerical solution of parabolic elliptic differential equations*, SIAM Journal on Applied Mathematics, 3 (1955), pp. 28–41.
- [41] S. S. Ravindran, *Numerical solutions of optimal control for thermally convective flows*, International Journal for Numerical Methods in Fluids, 25 (1997), pp. 205–227.
- [42] S. S. Ravindran, *Adaptive reduced-order controllers for a thermal flow system using proper orthogonal decomposition*, SIAM Journal on Scientific Computing 23 (2002), pp. 1924–1942.
- [43] E. Zuazua, *Numerics for the control of partial differential equations*, In Encyclopedia of Applied and Computational Mathematics, B. Engquist (eds.), Springer, Berlin, Heidelberg, 2015.