

Article

A Finite Element–Finite Volume Code Coupling for Optimal Control Problems in Fluid Heat Transfer for Incompressible Navier–Stokes Equations

Samuele Baldini [†], Giacomo Barbi [†], Giorgio Bornia [†], Antonio Cervone [†], Federico Giangolini [†], Sandro Manservigi ^{*,†} and Lucia Sirotti [†]

Laboratory of Montecuccolino, Department of Industrial Engineering, University of Bologna, Via dei Colli 16, 40136 Bologna, Italy; samuele.baldini2@unibo.it (S.B.); giacomo.barbi3@unibo.it (G.B.); giorgio.bornia@unibo.it (G.B.); a.cervone@unibo.it (A.C.); federico.giangolini2@unibo.it (F.G.); lucia.sirotti4@unibo.it (L.S.)

* Correspondence: sandro.manservigi@unibo.it

[†] These authors contributed equally to this work.

Abstract: In this work, we present a numerical approach for solving optimal control problems for fluid heat transfer applications with a mixed optimality system: an FEM code to solve the adjoint solution over a precise restricted admissible solution set and an open-source well-known code for solving the state problem defined over a different one. In this way, we are able to decouple the optimality system and use well-established and validated numerical tools for the physical modeling. Specifically, two different CFD codes, OpenFOAM (finite volume-based) and FEMuS (finite element-based), have been used to solve the optimality system, while the data transfer between them is managed by the external library MEDCOUPLING. The state equations are solved in the finite volume code, while the adjoint and the control are solved in the finite element code. Two examples taken from the literature are implemented in order to validate the numerical algorithm: the first one considers a natural convection cavity resulting from a Rayleigh–Bénard configuration, and the second one is a conjugate heat transfer problem between a fluid and a solid region.

Keywords: Boussinesq approximation; conjugate heat transfer; natural convection; optimal control; code coupling

MSC: 49M41



Academic Editors: Sotos C. Generalis and Philip Trevelyan

Received: 10 April 2025

Revised: 16 May 2025

Accepted: 20 May 2025

Published: 22 May 2025

Citation: Baldini, S.; Barbi, G.; Bornia, G.; Cervone, A.; Giangolini, F.; Manservigi, S.; Sirotti, L. A Finite Element–Finite Volume Code Coupling for Optimal Control Problems in Fluid Heat Transfer for Incompressible Navier–Stokes Equations. *Mathematics* **2025**, *13*, 1701. <https://doi.org/10.3390/math13111701>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

One of the most important aspects in designing an engineering system is foreseeing the physical behavior of the variables considered. Knowing the range in which a specific quantity can evolve is a crucial issue in checking the correct operation of a given apparatus. In this context, the numerical simulation of the partial differential equations that describe physical phenomena is essential. Specifically, the standard approach for a numerical simulation consists of building a parameter set, such as boundary conditions, source terms, and material physical properties that, given as input to the conservation equations, provides a numerical output that describes the evolution of the physical system. On the other hand, this approach is expensive from a computational point of view and leads to a try-and-fail procedure since we do not have direct control over the output results. However, numerical optimization is proposed as a viable alternative to the former approach because the desired outputs turn into input parameters that can be selected in the model. In fact, the idea

is to minimize a cost function that represents the distance between a variable and its desired target value under specific physical constraints that are the governing equations [1]. Therefore, we can introduce the objective term as a physical quantity to be obtained, such as a velocity or temperature matching, but also the enhancement or the reduction in a specific variable. The control term is a physical parameter used for minimization. It can be applied across the entire domain or exclusively on the boundaries. Additionally, the control term may alter the geometry of the problem, particularly in the context of shape optimization [2]. Ultimately, both objective and control terms must adhere to the governing equations, which serve as constraints describing the relevant physics.

Other optimization techniques come from various physical maximization/minimization principles. See, for example, the review in [3] and citations therein. The mathematical foundation of optimal control theory leads to adjoint and Lagrange multiplier techniques. The mathematical theory gives a solution defined by the optimality system over a set of admissible solutions. The uniqueness and existence of the solution are still open questions in many engineering fields.

The optimal control of the Navier–Stokes and energy equations is a difficult task. The difficulties arise from the regularity of the appropriate admissible solution set. The interested reader can find a historical perspective and mathematical background in [4]. Due to differentiability constraints, the Navier–Stokes optimal system solution can be found only in a set of restricted admissible solutions. This restricted admissible set is easy to manage in the FEM approach with Lagrange multiplier and variational formulation [5].

Furthermore, in many cases, well-known software must be used. The restriction of the admissible set can be achieved mainly with appropriate coupling. The coupling can interpolate a restricted admissible solution set from the FEM adjoint system to drive FV state solutions. Nevertheless, validated and state-of-the-art codes may already exist nowadays, at least for the state equations (i.e., the constraints of the optimal control system). Therefore, the aim of this paper is to leverage these codes to build the optimal control system. Here, we extend the work presented in [6,7] with the aim of coupling two CFD codes for the implementation of a heat transfer optimal control problem. The idea is to use the finite volume (FV) code OpenFOAM [8] for the solution of the state equation and the in-house finite element (FE) code FEMuS [9] for the adjoint and control equations. In fact, the latter equations arise naturally from a variational formulation of the considered problem, justifying the use of a finite element approach. In this framework, the latter equations should be solved because their derivation requires the concept of variation in the unknowns, with the possibility of also including the gradient of the variations. It appears clear from a mathematical point of view that the finite element approach lends itself to these representations, which are often missing or cumbersome to evaluate in standard finite volume frameworks. The code coupling, i.e., the data transfer between the different codes, is implemented with the external library MedCoupling [10].

Additionally, two main approaches can be followed when dealing with optimal control problems. The first one consists of discretizing the optimality systems after minimizing the functional in an infinite-dimensional framework (optimize-then-discretize). Differently, the second one discretizes the Lagrangian function, and then it derives the different equations to be solved (discretize-then-optimize). Under generic assumptions, these two approaches do not produce the same set of equations [1,4]. The numerical code coupling by using finite volume and finite element discretization can implement only the first approach, since with the second one, the discretized equations, such as the adjoint and the control, follow directly from the discretization used for the Lagrangian function.

In this work, we present a numerical optimization algorithm that considers generic heat transfer problems. In particular, we present two examples that are representative of

physical phenomena commonly present in several engineering applications. The first one involves the solution of the Navier–Stokes and energy equations (under the Boussinesq approximation) in the context of a natural convection problem considering the Rayleigh–Bénard setting as described in [11]. The second one is described by the conjugate heat transfer (CHT) model, which describes the heat transfer between a solid and a fluid region, as seen in [12]. Specifically, we adopt an adjoint-based optimization, which entails the solution of a coupled system given by the state, adjoint, and control equations [4].

Regarding the first application, several results have already been shown in the literature. One of the first works appears in [13], where a natural convection optimal control problem has been considered for a two-dimensional square cavity and for a temperature matching problem. In [5], an optimal control problem for the Boussinesq system has been presented in the context of liquid metals in a differentially heated cavity. Several papers have shown the existence, uniqueness, and regularity of the solution for this kind of problem. The interested reader can consult [14–25], where different settings are taken into account, including the use of different control approaches. For the specific problem of the boundary control problem considering the stationary Rayleigh–Bénard–Marangoni setting, one can consult [26]. Regarding the optimal control of the heat transfer between a fluid and solid region, or more in general heat transfer problems, we refer to [12,27,28].

This paper is organized as follows: in the first section, we derive the optimality systems for the heat transfer problems, one considering the Boussinesq equations, and a second one for the conjugate heat transfer between solid and fluid. After that, we provide a brief description of the numerical algorithm for the code coupling involving the two CFD codes via the external library. Lastly, we provide two numerical examples in order to validate the proposed numerical approach considering numerical benchmarks taken from the literature.

2. Optimal Control Problems

In this section, we present and describe the mathematical formulation of the system of equations considered for both numerical problems. We start by considering the state equations that indeed describe the different methods to transfer heat, i.e., the Boussinesq system and the conjugate heat transfer. After that, since the examples involve different objective and control terms, we derive each optimality system considering the specific application setup.

We first introduce some mathematical notation from functional analysis [29]. Considering a generic domain $\Omega \in \mathbb{R}^N$, $H^s(\Omega)$ denotes the standard Sobolev space $W^{s,2}(\Omega)$ of order $s \in \mathbb{R}$, and clearly the space of square-integrable functions $L^2(\Omega)$ refers to the zero-order Sobolev space $H^0(\Omega)$,

$$L^2(\Omega) = \left\{ f: \Omega \rightarrow \mathbb{R} \mid \int_{\Omega} |f|^2 d\Omega < +\infty \right\}.$$

Of particular interest is the space $H^1(\Omega)$, defined as

$$H^1(\Omega) = \left\{ f \in L^2(\Omega) \mid \frac{\partial f}{\partial x_k} \in L^2(\Omega), \quad k = 1, 2, \dots, N \right\},$$

and we also introduce the subspace of functions that vanish on the boundary $\Gamma \in \partial\Omega$,

$$H_{\Gamma}^1(\Omega) = \left\{ f \in H^1(\Omega) \mid f = 0 \quad \text{on } \Gamma \right\}.$$

For a generic element f of the Sobolev spaces, the norms can be written as

$$\|f\|_{L^2(\Omega)} = \int_{\Omega} |f|^2 d\Omega,$$

and

$$\|f\|_{H^1(\Omega)} = \int_{\Omega} (|f|^2 + |\nabla f|^2) d\Omega,$$

where $|\nabla f|^2 = \sum_{k=1}^N \left| \frac{\partial f}{\partial x_k} \right|^2$. For functions defined only on the boundary of the domain Γ , its norm is defined as

$$\|f\|_{H^1(\Gamma)} = \int_{\Gamma} (|f|^2 + |\nabla_s f|^2) d\Gamma,$$

where the surface gradient is $\nabla_s f = \nabla f - (\nabla f \cdot \mathbf{n})\mathbf{n}$. Lastly, for $(f, g) \in L^2$ such that $(fg) \in L^1(\Omega)$, where L^1 is the Sobolev space of integrable functions $W^{0,1}$, their inner product is defined as

$$(f, g) = \int_{\Omega} f g d\Omega.$$

In the case of vector-valued spaces, we indicate them with boldface letters $\mathbf{L}^2, \mathbf{H}^1$. Their norm is defined in a completely analogous manner to the scalar case. Taking $\mathbf{u}, \mathbf{v} \in \mathbf{L}^2(\Omega)$, the inner product is defined as

$$(\mathbf{u}, \mathbf{v}) = \int_{\Omega} \mathbf{u} \cdot \mathbf{v} d\Omega.$$

Next, we introduce the bilinear forms

$$\begin{aligned} a(f, g) &= \int_{\Omega} \nabla f \cdot \nabla g d\Omega && \forall f, g \in H^1(\Omega), \\ a(\mathbf{u}, \mathbf{v}) &= \frac{1}{2} \int_{\Omega} (\nabla \mathbf{u} + \nabla \mathbf{u}^T) : (\nabla \mathbf{v} + \nabla \mathbf{v}^T) d\Omega && \forall \mathbf{u}, \mathbf{v} \in \mathbf{H}^1(\Omega), \\ b(\mathbf{u}, f) &= - \int_{\Omega} f \nabla \cdot \mathbf{u} d\Omega && \forall \mathbf{u} \in \mathbf{H}^1(\Omega), \forall f \in L^2(\Omega), \end{aligned}$$

and the trilinear forms

$$\begin{aligned} c(\mathbf{u}, \mathbf{v}, \mathbf{w}) &= \int_{\Omega} ((\mathbf{u} \cdot \nabla) \mathbf{v}) \cdot \mathbf{w} d\Omega && \forall \mathbf{u}, \mathbf{v}, \mathbf{w} \in \mathbf{H}^1(\Omega), \\ c(\mathbf{u}, f, g) &= \int_{\Omega} (\mathbf{u} \cdot \nabla f) g d\Omega && \forall \mathbf{u} \in \mathbf{H}^1(\Omega), \forall f \in H^1(\Omega), \forall g \in L^2(\Omega), \end{aligned}$$

that will be useful in the description of the optimality systems.

We introduce the functional \mathcal{F} that we define as

$$\mathcal{F}(\mathbf{u}, T, q) = \mathcal{J}(\mathbf{u}, T) + \|q\|_{H^s},$$

where \mathcal{J} represents the generic objective functional, i.e., $\|\nabla \times \mathbf{u}\|_{L^2(\Omega)}$ or $\|T - T_d\|_{L^2(\Omega)}$, and $\|q\|_{H^s}$, the regularization term with $s = 0$ for distributed control and $s = 1$ or $s = \pm \frac{1}{2}$ for boundary control. In this paper, we will address Dirichlet boundary control problems with $H^1(\Gamma)$ regularization while Neumann boundary control problems with $L^2(\Gamma)$ regularization. In both cases, we are restricting the space where we are looking for the solution with respect to its native space, respectively, $H^{1/2}$ and $H^{-1/2}$. For more details regarding this aspect and the different numerical approaches to address this issue, the interested reader can consult [30–32].

2.1. Rayleigh–Bénard Problem

In this section, we derive the optimality system for the Rayleigh–Bénard problem considering a square domain, as reported in Figure 1. The Rayleigh–Bénard system exhibits buoyancy-driven flow, where the natural convection develops due to the temperature gradient imposed between the two horizontal walls Γ_{top} and Γ_1 . The objective of this

optimization case is to stabilize the flow inside the domain, thereby minimizing the vorticity. This will be achieved by introducing appropriate cavity heating, initially using a volumetric heat source within the domain, and then applying a heat flux condition on the boundary.

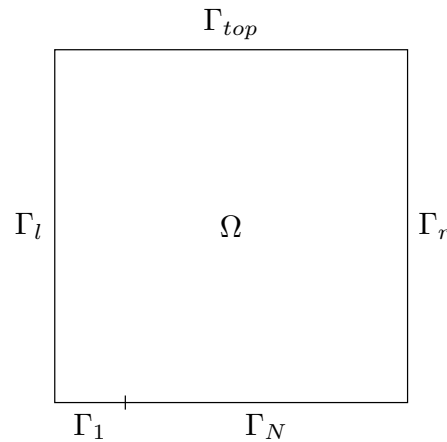


Figure 1. Domain of the Rayleigh–Bénard minimization problem.

The coupled Navier–Stokes system and energy equation form the Boussinesq system of equations that model the Rayleigh–Bénard problem. We write the stationary form of the Navier–Stokes equations as

$$\begin{cases} \nabla \cdot \mathbf{u} = 0 & \text{on } \Omega, \\ -2\nu \nabla \cdot \mathcal{D}(\mathbf{u}) + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\nabla p + \beta \mathbf{g}(T - T_{ref}) & \text{on } \Omega, \\ \mathbf{u} = 0 & \text{on } \partial\Omega, \end{cases} \quad (1)$$

where \mathbf{u} is the velocity field, p is the pressure field, ν is the kinematic viscosity, β is the thermal expansion coefficient, \mathbf{g} is a vector representing the gravitational acceleration in the negative y direction, and $\mathcal{D}(\mathbf{u}) = \frac{1}{2}(\nabla \mathbf{u} + \nabla \mathbf{u}^T)$ is the rate of deformation tensor of the flow. Next, we write the stationary energy equation in the temperature formulation as

$$\begin{cases} -\alpha \Delta T + \mathbf{u} \cdot \nabla T = Q & \text{on } \Omega, \\ T = g_D & \text{on } \Gamma_{top} \cup \Gamma_1, \\ \frac{\partial T}{\partial \mathbf{n}} = g_N & \text{on } \Gamma_N, \\ \frac{\partial T}{\partial \mathbf{n}} = 0 & \text{on } \Gamma_l \cup \Gamma_r, \end{cases} \quad (2)$$

where T is the temperature field, Q is a generic volumetric heat source, g_N is a generic wall heat flux, g_D is a given temperature profile at the boundary and α is the thermal diffusivity coefficient. The temperature gradient inside the cavity is generated by fixing the temperature along the top wall Γ_{top} and part of the bottom wall Γ_1 , according to g_D . As mentioned earlier, the vorticity minimization is achieved by controlling the volumetric heat source Q and the wall heat flux g_N separately; however, the generic derivation for both types of control is reported below.

The weak formulation of the state constraints (1) and (2) is written using the linear, bilinear, and trilinear forms introduced earlier

$$\begin{cases} b(\mathbf{u}, f) = 0 & \forall f \in L^2(\Omega), \\ va(\mathbf{u}, \boldsymbol{\psi}) + c(\mathbf{u}, \mathbf{u}, \boldsymbol{\psi}) = -b(\boldsymbol{\psi}, p) + (\beta \mathbf{g}(T - T_{ref}), \boldsymbol{\psi}) & \forall \boldsymbol{\psi} \in \mathbf{H}_{\partial\Omega}^1(\Omega), \\ \alpha a(T, \phi) + c(\mathbf{u}, T, \phi) = (Q, \phi) + (g_N, \phi)_{\Gamma_N} & \forall \phi \in H_{\Gamma_{top} \cup \Gamma_1}^1(\Omega). \end{cases} \quad (3)$$

Then, we specialize the generic functional to this case, i.e.,

$$\mathcal{F}(\mathbf{u}, Q, g_N) = \frac{1}{2} \int_{\Omega} |\nabla \times \mathbf{u}|^2 d\Omega + \frac{\lambda_Q}{2} \int_{\Omega} |Q|^2 d\Omega + \frac{\lambda_N}{2} \int_{\Gamma_N} |g_N|^2 d\Gamma, \tag{4}$$

where λ_Q, λ_N are the regularization coefficients for the distributed and the Neumann boundary control cases. Note that, in the case of boundary control, since we are dealing with the temperature derivative, the regularization is intended to be in L^2 . We also introduce the Lagrange multipliers (ξ, μ, θ) , referred to as the adjoint variables, corresponding to the adjoint velocity, adjoint pressure, and adjoint temperature, respectively. Finally, we write the Lagrangian as

$$\begin{aligned} \mathcal{L}(\mathbf{u}, p, T, \xi, \mu, \theta) = & \frac{1}{2} (\nabla \times \mathbf{u}, \nabla \times \mathbf{u})_{\Omega} + \frac{\lambda_Q}{2} (Q, Q)_{\Omega} + \frac{\lambda_N}{2} (g_N, g_N)_{\Gamma_N} + \\ & -b(\mathbf{u}, \mu) + \\ & + \nu a(\mathbf{u}, \xi) + c(\mathbf{u}, \mathbf{u}, \xi) + b(\xi, p) - (\beta \mathbf{g}(T - T_{ref}), \xi) + \\ & + \alpha a(T, \theta) + c(\mathbf{u}, T, \theta) - (Q, \theta) - (g_N, \theta)_{\Gamma_N}. \end{aligned} \tag{5}$$

Taking the variational derivative of the Lagrangian in the adjoint directions, we re-obtain the state equation system (3). Then, we take the variation in the Lagrangian in the state directions $(\tilde{\mathbf{u}}, \tilde{p}, \tilde{T})$, obtaining the adjoint equation system

$$\begin{cases} b(\xi, \tilde{p}) = 0 & \forall \tilde{p} \in L^2(\Omega), \\ \nu a(\tilde{\mathbf{u}}, \xi) + c(\tilde{\mathbf{u}}, \mathbf{u}, \xi) + c(\mathbf{u}, \tilde{\mathbf{u}}, \xi) - b(\tilde{\mathbf{u}}, \mu) = (\nabla \times \mathbf{u}, \nabla \times \tilde{\mathbf{u}}) - c(\tilde{\mathbf{u}}, T, \theta) & \forall \tilde{\mathbf{u}} \in \mathbf{H}_{\partial\Omega}^1(\Omega), \\ \alpha a(\tilde{T}, \theta) + c(\mathbf{u}, \tilde{T}, \theta) = (\beta \mathbf{g} \tilde{T}, \xi) & \forall \tilde{T} \in H_{\Gamma_{top} \cup \Gamma_1}^1(\Omega). \end{cases} \tag{6}$$

Finally, we differentiate the Lagrangian in the two control variables direction (Q, g_N)

$$\begin{aligned} \lambda_Q(Q, \tilde{Q}) = (\theta, \tilde{Q}) & \quad \forall \tilde{Q} \in L^2(\Omega), \\ \lambda_N(g_N, \tilde{g}_N)_{\Gamma_N} = (\theta, \tilde{g}_N)_{\Gamma_N} & \quad \forall \tilde{g}_N \in L^2(\Gamma_N), \end{aligned} \tag{7}$$

clearly taking $\lambda_N = 0$, we solve the distributed control case, and while taking $\lambda_Q = 0$, we solve the boundary control case. From these two equations, we can derive the functional gradient used in the steepest descent method as

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial Q} &= \theta + \lambda_Q Q, \\ \frac{\partial \mathcal{L}}{\partial g_N} &= \theta + \lambda_N g_N, \end{aligned} \tag{8}$$

respectively.

Finally, we write the strong form of the optimality system. We must isolate the velocity variation $\tilde{\mathbf{u}}$ to achieve this. To do so, we perform the following integrations by parts

$$\begin{aligned} (\nabla \times \mathbf{u}, \nabla \times \tilde{\mathbf{u}}) &= - \int_{\Omega} (\nabla \times \nabla \times \mathbf{u}) \cdot \tilde{\mathbf{u}} d\Omega + \int_{\Gamma} ((\nabla \times \mathbf{u}) \times \mathbf{n}) \cdot \tilde{\mathbf{u}} d\Gamma \\ &= - \int_{\Omega} \tilde{\mathbf{u}} \nabla (\nabla \cdot \mathbf{u}) d\Omega + \int_{\Omega} \tilde{\mathbf{u}} \Delta \mathbf{u} d\Omega + \int_{\Gamma} ((\nabla \times \mathbf{u}) \times \mathbf{n}) \cdot \tilde{\mathbf{u}} d\Gamma, \\ c(\mathbf{u}, \tilde{\mathbf{u}}, \xi) &= \int_{\Omega} \nabla \tilde{\mathbf{u}} : (\mathbf{u} \otimes \xi) d\Omega = - \int_{\Omega} \tilde{\mathbf{u}} \cdot (\nabla \cdot (\mathbf{u} \otimes \xi)) d\Omega + \int_{\Gamma} (\tilde{\mathbf{u}} \cdot \mathbf{n}) \cdot (\mathbf{u} \otimes \xi) d\Gamma \\ &= - \int_{\Omega} \tilde{\mathbf{u}} \cdot ((\nabla \cdot \mathbf{u}) \xi + \mathbf{u} \cdot \nabla \xi) d\Omega + \int_{\Gamma} \tilde{\mathbf{u}} \cdot (\mathbf{u} \cdot \mathbf{n}) \cdot \xi d\Gamma. \end{aligned} \tag{9}$$

Considering both the divergence of the velocity and the adjoint velocity on the boundary to be zero, we obtain the following strong form for the adjoint velocity system of equations, which corresponds with the steady formulation of the Oseen equation [33]

$$\begin{cases} \nabla \cdot \boldsymbol{\xi} = 0 & \text{on } \Omega, \\ -2\nu \nabla \cdot \mathcal{D}(\boldsymbol{\xi}) + \boldsymbol{\xi}(\nabla \mathbf{u})^T - (\mathbf{u} \cdot \nabla)\boldsymbol{\xi} - \nabla \mu + \theta \nabla T = \Delta \mathbf{u} & \text{on } \Omega, \\ \boldsymbol{\xi} = 0 & \text{on } \partial\Omega. \end{cases} \tag{10}$$

Isolating the temperature variation \tilde{T} in the adjoint system (6) after some manipulation

$$\begin{aligned} a(\tilde{T}, \theta) &= - \int_{\Omega} \tilde{T} \Delta \theta \, d\Omega + \int_{\Gamma} \tilde{T} (\nabla \theta \cdot \mathbf{n}) \, d\Gamma, \\ c(\mathbf{u}, \tilde{T}, \theta) &= - \int_{\Omega} \mathbf{u} \tilde{T} \cdot \nabla \theta + \int_{\Gamma} (\mathbf{u} \cdot \mathbf{n}) \tilde{T} \theta \, d\Gamma, \end{aligned} \tag{11}$$

where in the second one we used the divergence-free property of \mathbf{u} . Since the temperature variation \tilde{T} vanishes on the boundary Γ_i due to the Dirichlet condition on T , we obtain the following strong form system for the adjoint temperature θ

$$\begin{cases} -\alpha \Delta \theta - \mathbf{u} \nabla \theta = \beta \mathbf{g} \cdot \boldsymbol{\xi} & \text{on } \Omega, \\ \theta = 0 & \text{on } \Gamma_i, \\ \nabla \theta \cdot \mathbf{n} = 0 & \text{on } \partial\Omega \setminus \Gamma_i. \end{cases} \tag{12}$$

Lastly, the strong form of the control equations, both for the distributed and the boundary control case, is obtained directly from Equation (7), leading to

$$\begin{aligned} Q &= -\frac{\theta}{\lambda_Q}, \\ g_N &= -\frac{\theta}{\lambda_N}. \end{aligned} \tag{13}$$

2.2. Solid/Fluid Heat Transfer Problem

In this section, we derive the optimality system for the solid-fluid domain depicted in Figure 2. Moreover, during the following derivation, we will include some additional steps compared to the previous case for the sake of clarity. Unlike the previous case, the system here is considered to operate in a forced convection regime. As a result, the buoyancy term in the Navier–Stokes equations can be neglected. Furthermore, we impose a controlled inlet flow \mathbf{w} on the boundary Γ_i and an outflow condition on Γ_o . The corresponding system of equations reads

$$\begin{cases} \nabla \cdot \mathbf{u} = 0 & \text{on } \Omega_f, \\ -2\nu \nabla \cdot \mathcal{D}(\mathbf{u}) + (\mathbf{u} \cdot \nabla)\mathbf{u} + \nabla p = 0 & \text{on } \Omega_f, \\ \mathbf{u} = 0 & \text{on } \Gamma_c \cup \Gamma_4, \\ \mathbf{u} = \mathbf{w} & \text{on } \Gamma_i, \\ \sigma(\mathbf{u}, p) \cdot \mathbf{n} = 0 & \text{on } \Gamma_o, \end{cases} \tag{14}$$

where the stress tensor is defined as $\sigma(\mathbf{u}, p) = -p\mathbf{I} + 2\nu\mathcal{D}(\mathbf{u})$, and ν is the kinematic viscosity for the fluid. The temperature is solved both in the solid and the fluid domain, and on the interface, a conjugate heat transfer condition is imposed. In the following, we define a CHT interface condition as the condition of continuity of temperature as well as normal heat flux. We fix the temperature on Γ_i while we consider the other walls $\partial\Omega \setminus \Gamma_i$ to

be insulated. Furthermore, in the solid region, a volumetric heat source Q is enforced. We write separately the system of equations for the solid region

$$\begin{cases} -\alpha_s \Delta T_s = Q & \text{on } \Omega_s, \\ T_s = T_f & \text{on } \Gamma_c, \\ \alpha_s \frac{\partial T_s}{\partial \mathbf{n}} = \alpha_f \frac{\partial T_f}{\partial \mathbf{n}} & \text{on } \Gamma_c, \\ \frac{\partial T_s}{\partial \mathbf{n}} = 0 & \text{on } \partial\Omega_s \setminus \Gamma_c, \end{cases} \tag{15}$$

and for the fluid region

$$\begin{cases} -\alpha_f \Delta T_f + \mathbf{u} \cdot \nabla T_f = 0 & \text{on } \Omega_f, \\ T_f = 1 & \text{on } \Gamma_i, \\ T_f = T_s & \text{on } \Gamma_c, \\ \alpha_f \frac{\partial T_f}{\partial \mathbf{n}} = \alpha_s \frac{\partial T_s}{\partial \mathbf{n}} & \text{on } \Gamma_c, \\ \frac{\partial T_f}{\partial \mathbf{n}} = 0 & \text{on } \Gamma_o \cup \Gamma_4, \end{cases} \tag{16}$$

where α_s and α_f are the thermal diffusivity coefficients for the solid and fluid regions, respectively.

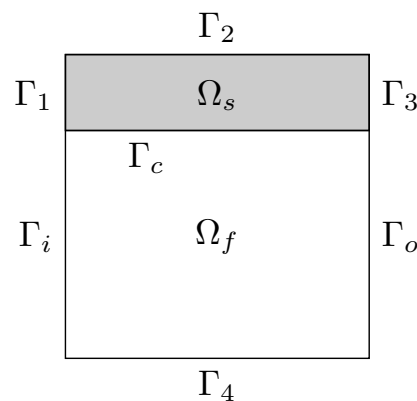


Figure 2. Domain of the solid/fluid heat transfer problem.

Now, we derive the weak form of the Navier–Stokes system after performing the following steps:

$$\begin{aligned} \int_{\Omega_f} \boldsymbol{\psi} \nabla \cdot \mathcal{D}(\mathbf{u}) \, d\Omega &= - \int_{\Omega_f} \mathcal{D}(\mathbf{u}) : \mathcal{D}(\boldsymbol{\psi}) \, d\Omega + \int_{\Gamma} \boldsymbol{\psi} \cdot \mathcal{D}(\mathbf{u}) \mathbf{n} \, d\Gamma, \\ \int_{\Omega_f} \nabla p \cdot \boldsymbol{\psi} \, d\Omega &= - \int_{\Omega_f} p (\nabla \cdot \boldsymbol{\psi}) \, d\Omega + \int_{\Gamma} p (\boldsymbol{\psi} \cdot \mathbf{n}) \, d\Gamma. \end{aligned}$$

While in the Boussinesq case, both boundary terms were zero due to the choice of the test function $\boldsymbol{\psi}$, and in this application, they give a contribution to Γ_i . Then, we can write the weak formulation for the Navier–Stokes equations system as follows:

$$\begin{cases} b(\mathbf{u}, f) = 0 & \forall f \in L^2(\Omega_f), \\ va(\mathbf{u}, \boldsymbol{\psi}) + c(\mathbf{u}, \mathbf{u}, \boldsymbol{\psi}) + b(\boldsymbol{\psi}, p) - (\boldsymbol{\sigma} \mathbf{n}, \boldsymbol{\psi})_{\Gamma_i} = 0 & \forall \boldsymbol{\psi} \in \mathbf{H}_{\Gamma_c \cup \Gamma_4}^1(\Omega_f), \\ (\mathbf{u} - \mathbf{w}, \mathbf{s})_{\Gamma_i} = 0 & \forall \mathbf{s} \in \mathbf{H}^{-1/2}(\Gamma_i), \end{cases} \tag{17}$$

where the Dirichlet boundary condition at the inlet is also formulated weakly in order to obtain the control in the desired function space. For the temperature system, we first integrate the Laplacian term in the solid and the fluid systems to obtain

$$\int_{\Omega_s} -\alpha_s \Delta T_s \phi \, d\Omega = - \int_{\Omega_s} \alpha_s \nabla T_s \cdot \nabla \phi \, d\Omega + \int_{\Gamma_c} \alpha_s (\nabla T_s \cdot \mathbf{n}) \phi \, d\Gamma,$$

$$\int_{\Omega_f} -\alpha_f \Delta T_f \phi \, d\Omega = - \int_{\Omega_f} \alpha_f \nabla T_f \cdot \nabla \phi \, d\Omega + \int_{\Gamma_c} \alpha_f (\nabla T_f \cdot \mathbf{n}) \phi \, d\Gamma.$$

After summation of the above equations, the two boundary terms on Γ_c cancel each other due to the heat flux continuity. Then, we can write the temperature equation for the whole domain in weak form as

$$\alpha a(T, \phi) + c(\mathbf{u}, T, \phi)_{\Omega_f} = (Q, \phi)_{\Omega_s} \quad \forall \phi \in L^2(\Omega_s), \tag{18}$$

where of course the diffusivity constant is $\alpha = \alpha_s H(\Omega_s) + \alpha_f H(\Omega_f)$, H being the Heaviside function.

Now, we introduce the cost functional

$$\mathcal{F}(T, \mathbf{w}) = \frac{1}{2} \int_{\Gamma_d} |T - T_d|^2 \, d\Gamma + \frac{\lambda_1}{2} \int_{\Gamma_i} |\mathbf{w}|^2 \, d\Gamma + \frac{\lambda_2}{2} \int_{\Gamma_i} |\nabla_s \mathbf{w}|^2 \, d\Gamma, \tag{19}$$

where $\Gamma_d \subset (\partial\Omega_f \cup \partial\Omega_s)$ is the observation boundary and an H^1 regularization is chosen, since we deal with a Dirichlet boundary control [34]. Even if we use two different weights, λ_1 and λ_2 , the existence of the solution for a Dirichlet boundary optimal control necessarily requires the regularization of the gradient term if we consider a CHT problem [12,35], i.e., $\lambda_2 > 0$. Using again the Lagrange multipliers (ξ, μ, τ, θ) , we write the Lagrangian \mathcal{L} as

$$\begin{aligned} \mathcal{L}(\mathbf{u}, p, T, \xi, \mu, \tau, \theta) = & \frac{1}{2} (T - T_d, T - T_d)_{\Gamma_d} + \frac{\lambda_1}{2} (\mathbf{w}, \mathbf{w})_{\Gamma_i} + \frac{\lambda_2}{2} (\nabla_s \mathbf{w}, \nabla_s \mathbf{w})_{\Gamma_i} + \\ & -b(\mathbf{u}, \mu) + \\ & +va(\mathbf{u}, \xi) + c(\mathbf{u}, \mathbf{u}, \xi) + b(\xi, p) - (\sigma \mathbf{n}, \xi)_{\Gamma_i} + (\mathbf{u} - \mathbf{w}, \tau)_{\Gamma_i} + \\ & +\alpha a(T, \theta) + c(\mathbf{u}, T, \theta)_{\Omega_f} - (Q, \theta)_{\Omega_s}. \end{aligned} \tag{20}$$

Taking the variational derivative of the Lagrangian in the adjoint direction, we re-obtain the state Equations (17) and (18). In order to derive the adjoint velocity system, we differentiate in the velocity \mathbf{u} and pressure p direction to obtain

$$\begin{cases} b(\xi, \tilde{p}) = 0 & \forall \tilde{p} \in L^2(\Omega_f), \\ va(\tilde{\mathbf{u}}, \xi) + c(\tilde{\mathbf{u}}, \mathbf{u}, \xi) + c(\mathbf{u}, \tilde{\mathbf{u}}, \xi) - b(\tilde{\mathbf{u}}, \mu) + (\tilde{\mathbf{u}}, \tau)_{\Gamma_i} + c(\tilde{\mathbf{u}}, T, \theta) = 0 & \forall \tilde{\mathbf{u}} \in \mathbf{H}_{\Gamma_c \cup \Gamma_4}^1(\Omega_f), \\ (\tilde{\sigma} \mathbf{n}, \xi)_{\Gamma_i} = 0 & \forall \tilde{\sigma} \mathbf{n} \in \mathbf{H}^{-1/2}(\Gamma_i). \end{cases} \tag{21}$$

Similarly, by taking the variation in the Lagrangian in the temperature T direction, we obtain the weak form of the adjoint temperature equation

$$\alpha a(\tilde{T}, \theta) + c(\mathbf{u}, \tilde{T}, \theta)_{\Omega_f} = -(T - T_d, \tilde{T})_{\Gamma_d} \quad \forall \tilde{T} \in H_{\Gamma_i}^1(\Omega). \tag{22}$$

Lastly, differentiating the Lagrangian with respect to the control variable \mathbf{w} we obtain the Euler equation in weak form

$$\frac{\lambda_1}{2} (\mathbf{w}, \tilde{\mathbf{w}})_{\Gamma_i} + \frac{\lambda_2}{2} (\nabla_s \mathbf{w}, \nabla_s \tilde{\mathbf{w}})_{\Gamma_i} = (\tau, \tilde{\mathbf{w}})_{\Gamma_i} \quad \forall \tilde{\mathbf{w}} \in \mathbf{H}^{-1/2}(\Gamma_i). \tag{23}$$

In order to iteratively solve this system through the gradient descent method, we need the explicit formulation for the Lagrangian derivative in the control direction, which clearly results in

$$\frac{\delta \mathcal{L}}{\delta \mathbf{w}} = \lambda_1 \mathbf{w} - \lambda_2 \Delta_s \mathbf{w} - \boldsymbol{\tau}. \tag{24}$$

Once again, we transition from the weak form of the adjoint velocity system to the strong form. Specifically, we start isolating the velocity variation $\tilde{\mathbf{u}}$. To do so, we perform the same steps reported in (9), i.e., we perform some manipulations to obtain

$$\begin{aligned} a(\tilde{\mathbf{u}}, \boldsymbol{\xi}) &= - \int_{\Omega} \tilde{\mathbf{u}} \nabla \cdot \mathcal{D}(\boldsymbol{\xi}) \, d\Omega + \int_{\Gamma} \mathbf{u} \cdot \mathcal{D}(\boldsymbol{\xi}) \mathbf{n} \, d\Gamma, \\ b(\tilde{\mathbf{u}}, \mu) &= \int_{\Omega} \tilde{\mathbf{u}} \cdot \nabla \mu \, d\Omega - \int_{\Gamma} (\tilde{\mathbf{u}} \cdot \mathbf{n}) \mu \, d\Gamma. \end{aligned}$$

Taking the velocity variation $\tilde{\mathbf{u}}$ to vanish on Γ_c and Γ_4 due to Dirichlet conditions, and the adjoint velocity $\boldsymbol{\xi}$ to vanish on $\Gamma_i, \Gamma_c, \Gamma_4$, as shown in the weak formulation of the system, we obtain the following strong form for the adjoint velocity system of equations

$$\begin{cases} \nabla \cdot \boldsymbol{\xi} = 0 & \text{on } \Omega_f, \\ -2\nu \nabla \cdot \mathcal{D}(\boldsymbol{\xi}) + \boldsymbol{\xi} (\nabla \mathbf{u})^T - (\mathbf{u} \cdot \nabla) \boldsymbol{\xi} - \nabla \mu + \theta \nabla T = 0 & \text{on } \Omega_f, \\ \boldsymbol{\xi} = 0 & \text{on } \Gamma_i \cup \Gamma_c \cup \Gamma_4, \\ -\mu \mathbf{n} + 2\nu \mathcal{D}(\boldsymbol{\xi}) \mathbf{n} + (\mathbf{u} \cdot \mathbf{n}) \boldsymbol{\xi} = 0 & \text{on } \Gamma_o, \\ \boldsymbol{\tau} = -\mu \mathbf{n} + 2\nu \mathcal{D}(\boldsymbol{\xi}) \mathbf{n} & \text{on } \Gamma_i. \end{cases} \tag{25}$$

Performing the same steps as in (11), we write the strong form equation system for the adjoint temperature θ as

$$\begin{cases} -\alpha \Delta \theta - \mathbf{u} \nabla \theta = (T - T_d) H_{\Gamma_d} & \text{on } \Omega, \\ \theta = 0 & \text{on } \Gamma_i, \\ \nabla \theta \cdot \mathbf{n} = 0 & \text{on } \partial\Omega \setminus \Gamma_i. \end{cases} \tag{26}$$

Lastly, the strong form of the control equation is easy to obtain directly from Equation (23)

$$\lambda_1 \mathbf{w} - \lambda_2 \Delta_s \mathbf{w} = \boldsymbol{\tau} \quad \text{on } \Gamma_i. \tag{27}$$

2.3. Minimization Method

The optimality system consists of the state equations for (T, \mathbf{u}, p) , the adjoint equations for $(\theta, \boldsymbol{\xi}, \mu)$, and the control equation, or optimality condition, for Q, g_N , and \mathbf{w} . These three systems of equations are clearly coupled but are solved in a segregated manner, as mentioned, to reduce computational effort. The state and adjoint systems are solved using a simple fixed-point iteration method, while for the control variables, to ensure convergence to the optimum, we employ a basic gradient method with a backtracking line search algorithm, as shown in Algorithm 1. Here, we refer to the control variable with the generic symbol q , and we set $\epsilon = 10^{-10}, \tau = 10^{-8}$. It can be demonstrated that, for a sufficiently small step size that the backtracking line search algorithm always guarantees, the gradient method yields descent direction and is globally convergent [36]. As mentioned, the control update in line 8 is performed using a gradient method. Firstly, the control variation δq in the three different cases is calculated as

$$\begin{aligned}\delta Q &= -\rho(\theta + \lambda_Q Q), \\ \delta g_N &= -\rho(\theta + \lambda_{Ng} g_N), \\ \lambda_1 \delta \mathbf{w} - \lambda_2 \Delta \delta \mathbf{w} &= -\rho(-\tau + \lambda_1 \mathbf{w} - \lambda_2 \Delta_s \mathbf{w}),\end{aligned}$$

and then the update is performed

$$q_{k+1} = q_k + \delta q.$$

Algorithm 1 Steepest gradient method

```

1: Set  $k = 0$ 
2: Compute state  $T_k, \mathbf{u}_k, p_k$ 
3: Compute functional  $\mathcal{F}_k$ 
4: Compute adjoint  $\theta_k, \boldsymbol{\zeta}_k, \mu_k$ 
5: while true do
6:   Set initial step size  $\rho = \rho_0$ 
7:   while  $\mathcal{F}_{k+1} > \mathcal{F}_k$  and  $\rho > \epsilon$  do
8:     Update control  $q_{k+1}$ 
9:     Compute state  $T_{k+1}, \mathbf{u}_{k+1}, p_{k+1}$ 
10:    Compute functional  $\mathcal{F}_{k+1}$ 
11:    Update step size  $\rho = \rho/2$ 
12:  end while
13:  if  $\left| \frac{\mathcal{F}_{k+1} + \mathcal{F}_k}{\mathcal{F}_{k+1}} \right| < \tau$  then
14:    break
15:  end if
16:  Compute adjoint  $\theta_{k+1}, \boldsymbol{\zeta}_{k+1}, \mu_{k+1}$ 
17: end while

```

3. Coupling Algorithm

In this section, we present the numerical algorithm for the code coupling that solves the state equations separately from the adjoint and control equations. The previous section has given a detailed explanation of the reasoning for adopting a finite element framework for solving the adjoint and control equations. Moreover, the coupling application leverages the well-validated OpenFOAM solvers, making it a reliable tool for solving the state equations and eliminating the need for re-implementing traditional conservation equations.

Data exchange between the two codes has been made possible by implementing three C++ classes. The first class, `med_coupling`, manages the data transfer routines with the external library MEDCOUPLING, specifically for data manipulation and interpolation. Furthermore, we have two specific wrapper classes for the CFD codes. The first one, referred to as `of_interface`, is responsible for the communication between the routines of OpenFOAM and the MED data structure. The corresponding class for the FEMuS code is labeled as `femus_interface`. More details regarding the specific implemented classes and the coupling strategy can be found in [6]. With the previously described code architecture, we describe the specific algorithms for the Rayleigh–Bénard problem and the solid/fluid heat transfer problem in the following sections.

3.1. Rayleigh–Bénard Problem

For the first application, we want to couple FEMuS and OpenFOAM in order to solve a Rayleigh–Bénard problem describing the natural convection inside a cavity. The state equations to be solved are the Navier–Stokes equations in the incompressible formulation together with the temperature equation, as described in the previous section. This system of

equations is then solved by adding the adjoint and the control equations, which can be both a volumetric source or a boundary condition for the temperature equation. In this work, we present these two different data type transfers, i.e., a volumetric and a boundary approach. For more details regarding the data transfer with the external library MEDCOUPLING, the interested reader can consult [6], where each routine used in this work is explained.

The data exchange from OpenFOAM to FEMuS involves transferring the state variables: the velocity \mathbf{u} and the temperature T fields. Thus, the `obtain` and `set` functions extract these volumetric fields from OpenFOAM and write them into FEMuS for the solution of the adjoint equations.

The problem is always solved in a restricted admissible set of controls. For the variational formulation, the admissible Neumann control sets are subspaces of $L^2(\Gamma_N)$, while the distributed controls are subspaces of $L^2(\Omega)$. The velocity, pressure, and temperature fields are then solved in the spaces dictated by the constraint. The discretized unknowns belong to the smallest finite-dimensional spaces involved in the hybrid FE-FV formulation. Such spaces are the spaces admitted by the variational formulation that underlies the FE discretization. The discretized solutions of the optimality system will eventually converge to these spaces [37].

Note that this data exchange implies a transfer from a finite volume representation to a finite element data structure. The same approach applies to the control variable q , where the `get` and `set` functions transfer this variable from FEMuS to OpenFOAM. This data exchange can be used in both volumetric and boundary-based methods, with data interpolation now originating from a finite element grid and converting data to a finite volume grid. For the volumetric data transfer, q is the volumetric source of the temperature equation, while for boundary control, q is transferred to the Neumann boundary condition of the same equation. A schematic representation of this algorithm is reported in Figure 3.

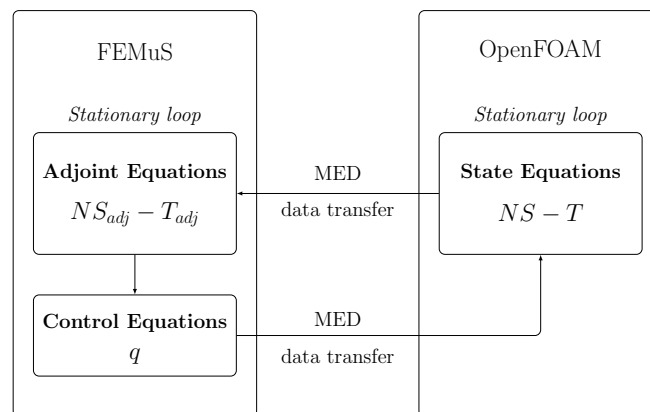


Figure 3. Code coupling scheme for the optimal control of the Boussinesq system of equations.

3.2. Solid/Fluid Heat Transfer Problem

For this test, we exploit already validated numerical tools available in OpenFOAM for the solution of the state equations. In particular, we refer to the multi-region technique, which provides tools to exchange data between two regions, such as the conjugate heat transfer approach.

The schematic representation for the application is reported in Figure 4. The data transfer between the CFD codes follows the same volumetric and boundary data exchange described in the previous application. However, the `of_interface` class has been improved to support the multi-region solver of OpenFOAM, allowing for the retrieval of field values within different domains. In this application, the temperature field is transferred from OpenFOAM to FEMuS across the entire fluid domain. OpenFOAM supplies the volumetric velocity field to FEMuS, while FEMuS provides the boundary control q to the fluid domain

in OpenFOAM. The FEM approach follows strictly the optimality system over a restrictive admissible set of control solutions. The interpolation errors are significant when the FEM boundary control is interpolated on the FV computational domain. The FEM adjoint approximation feels the interpolation over the volume of the state errors concerning the target. The convergent solution is the solution imposed by the restricted admissible control set on the FEM discretization in agreement with the standard FEM-only formulation.

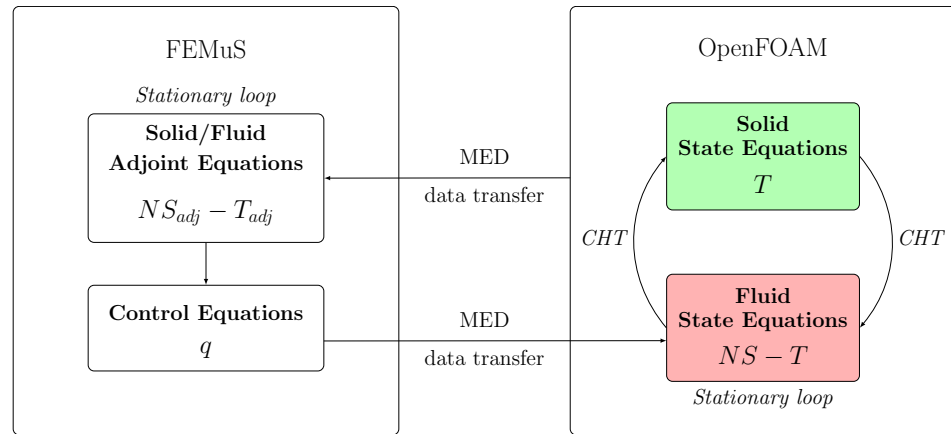


Figure 4. Code-coupling scheme for the optimal control of the CHT system.

It is worth noting that OpenFOAM exploits a segregated approach in the implementation of the CHT boundary condition between the solid and the fluid, i.e., the two regions are solved separately, and then the boundary conditions are enforced on Γ_c , taking into account the continuity of the temperature field and the conservation of the heat fluxes. On the other hand, FEMuS solves the adjoint temperature in a monolithic way on a single domain which includes both the solid and the fluid region. Therefore, we do not enforce additional boundary conditions on the common interface Γ_c since the assembling of a unique matrix allows us to satisfy the continuity conditions on temperature and heat flux in a natural way. Of course, different material properties and source terms are taken into account depending on the specific domain where the local element of the mesh is assembled.

Lastly, we remark that OpenFOAM solves the Navier–Stokes system only by a projection method, and we selected the PIMPLE algorithm. On the other hand, a monolithic approach is implemented in FEMuS for the adjoint Navier–Stokes, which indeed couples the adjoint velocity and pressure by assembling and solving them simultaneously.

3.3. Numerical Discretization

We briefly discuss the discretization framework of the equations. We already know that the state equations are solved with a finite volume approach while the adjoint and control equations use a finite element discretization. In the introduction, we also mentioned the use of the optimize-then-discretize concept to build our optimal control problem that supports different discretization spaces for the approximation of the state and adjoint equations. We introduce the finite-dimensional spaces $X_h \subset X = H^1_{\Gamma_D}(\Omega)$, $Q_h \subset L^2(\Omega)$ and $S_h \subset L^2(\Omega)$. Specifically, given a domain partition \mathcal{T}_h , the element K , and the space \mathcal{P}_r of polynomials of degree at most r , we set

$$\begin{aligned} X_h &= \{v_h \in C^0(\bar{\Omega}) : v_h|_K \in \mathcal{P}_2 \quad \forall K \in \mathcal{T}_h\}, \\ Q_h &= \{v_h \in C^0(\bar{\Omega}) : v_h|_K \in \mathcal{P}_1 \quad \forall K \in \mathcal{T}_h\}, \\ S_h &= \{v_h \in L^2(\Omega) : v_h|_K \in \mathcal{P}_0 \quad \forall K \in \mathcal{T}_h\}. \end{aligned}$$

We refer to the vector-valued version of the same spaces with a bold symbol. Regarding the discretized adjoint variables, we consider a standard Taylor–Hood pair for the adjoint velocity and pressure [38,39]; thus, we have $(\boldsymbol{\xi}_h, \mu_h) \in \mathbf{X}_h \times Q_h$. For the adjoint temperature and the control, we consider biquadratic finite elements, i.e., $\theta_h, q_h \in X_h$.

The finite volume approximation of the state equations requires only one type of discretized space. Therefore, we consider the piecewise constant approximation for velocity, pressure, and temperature, i.e., $\mathbf{u}_h \in \mathbf{S}_h$ and $p_h, T_h \in S_h$.

These discretization spaces interact through the numerical interpolation of the physical variables from a finite volume to a finite element grid and vice versa. A comprehensive analysis of the interpolation techniques used can be found in [40,41].

4. Numerical Results

In this section, we present two numerical examples in order to validate the effectiveness of the above-presented algorithm. As a first numerical test, a Rayleigh–Bénard natural convection problem is described considering a square cavity. Later, we present a multi-region problem with a solid and a fluid domain that interact via a standard conjugate heat transfer model. For each application, we recall the objective functional, the control term in the state equations, and the corresponding boundary conditions. Numerical results are then compared with already published works with the same problem setting.

4.1. Rayleigh–Bénard Problem

For the first numerical test, we refer to the example reported by Lee [11], where a vorticity minimization has been studied on a square domain. Specifically, the natural convection obtained from the Rayleigh–Bénard configuration has been simulated with both a distributed and a Neumann boundary control for the temperature equation. Among the several cases reported in [11], we selected two of them that present different behaviors of the control and the optimal solution. Therefore, after a comparison of the uncontrolled solution, the cases with $\lambda_Q = 10^{-3}$ and $\lambda_Q = 10^{-5}$ will be discussed (same values have also been considered for the Neumann case, i.e., for λ_N).

In Figure 1, the domain $\Omega = [0, 1]^2$ is shown, with the partition of the bottom boundary into two sections, Γ_1 and Γ_N . The boundary conditions for the temperature field are as follows: a homogeneous Neumann is enforced on Γ_l and Γ_r ; on Γ_{top} , we have a homogeneous Dirichlet boundary condition, while on Γ_1 we have a fixed temperature value $T = 10$. Lastly, on Γ_N we have the nonhomogeneous Neumann boundary condition that acts as control.

The grid used for the simulation consists of 20×20 elements for the FE code and 40×40 elements for the FV code, both equally spaced. This discretization setup leads to a very similar amount of degrees of freedom for each code (the finite element grid is indeed equipped with biquadratic elements, therefore implying 1681 dots).

In the case of the distributed control, we fix $\lambda_N = 0$, while for the Neumann boundary control, we set $\lambda_Q = 0$. Regarding the material properties of the fluid, we set $\nu = \beta = \alpha = 1$.

We recall the optimality system in strong form for this problem, given by (28), (29), and (34), along with the corresponding boundary conditions reported in Table 1.

$$State : \begin{cases} \nabla \cdot \mathbf{u} = 0 & \text{on } \Omega, \\ -2\nu \nabla \cdot \mathcal{D}(\mathbf{u}) + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\nabla p + \beta \mathbf{g}(T - T_{ref}) & \text{on } \Omega, \\ -\alpha \Delta T + (\mathbf{u} \cdot \nabla)T = Q & \text{on } \Omega, \end{cases} \quad (28)$$

$$\text{Adjoint : } \begin{cases} \nabla \cdot \boldsymbol{\zeta} = 0 & \text{on } \Omega, \\ -2\nu \nabla \cdot \mathcal{D}(\boldsymbol{\zeta}) + \boldsymbol{\zeta}(\nabla \mathbf{u})^T - (\mathbf{u} \cdot \nabla)\boldsymbol{\zeta} = \nabla \mu - \theta \nabla T + \Delta \mathbf{u} & \text{on } \Omega, \\ -\alpha \Delta \theta - (\mathbf{u} \cdot \nabla)\theta = \beta \mathbf{g} \cdot \boldsymbol{\zeta} & \text{on } \Omega, \end{cases} \quad (29)$$

$$\text{Control : } \quad Q = -\frac{\theta}{\lambda_Q} \quad \text{on } \Omega, \quad \text{or} \quad g_N = -\frac{\theta}{\lambda_N} \quad \text{on } \Gamma_N. \quad (30)$$

Table 1. Boundary conditions for the Rayleigh–Bénard problem for each unknown $(\mathbf{u}, T, \boldsymbol{\zeta}, \theta)$ on each portion of the boundary

	Γ_l	Γ_r	Γ_{top}	Γ_1	Γ_N
\mathbf{u}	$\mathbf{u} = 0$	$\mathbf{u} = 0$	$\mathbf{u} = 0$	$\mathbf{u} = 0$	$\mathbf{u} = 0$
T	$\nabla T \cdot \mathbf{n} = 0$	$\nabla T \cdot \mathbf{n} = 0$	$T = 0$	$T = 10$	$\nabla T \cdot \mathbf{n} = g_N$
$\boldsymbol{\zeta}$	$\boldsymbol{\zeta} = 0$	$\boldsymbol{\zeta} = 0$	$\boldsymbol{\zeta} = 0$	$\boldsymbol{\zeta} = 0$	$\boldsymbol{\zeta} = 0$
θ	$\nabla \theta \cdot \mathbf{n} = 0$	$\nabla \theta \cdot \mathbf{n} = 0$	$\theta = 0$	$\theta = 0$	$\nabla \theta \cdot \mathbf{n} = 0$

4.1.1. Uncontrolled Case

The results for the uncontrolled case (namely, the control variable is set to zero) are shown in Figure 5. The contour plot of the temperature field is shown on the left side, while the velocity field is shown on the right.

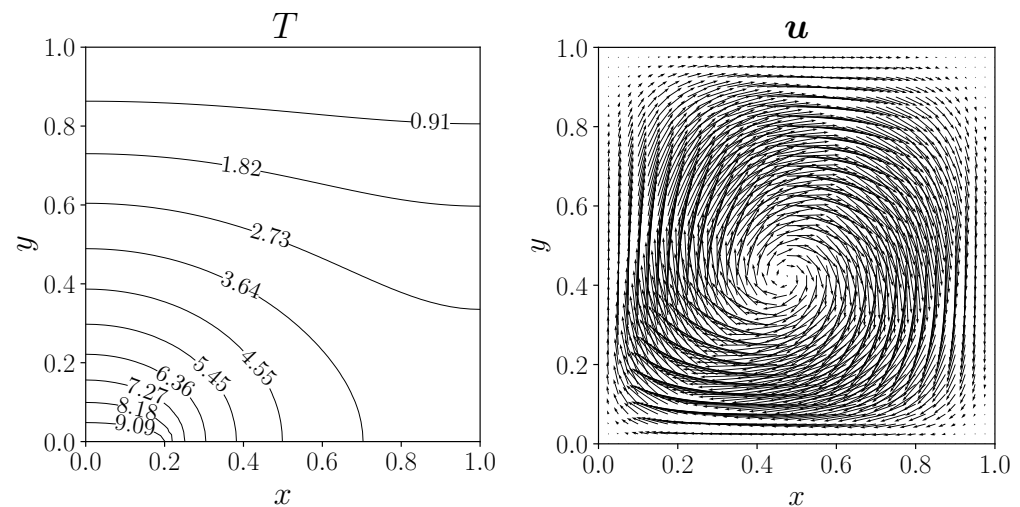


Figure 5. The uncontrolled case: on the left, a contour plot of the temperature field; on the right, velocity glyph representation.

4.1.2. Distributed Control

Figure 6 shows the results for the temperature fields in the distributed control case, compared to the same contour plot from [11], which is shown with a dashed line. We observe that both temperature distributions get closer and closer to a purely vertical temperature gradient. It appears that the reduction in the horizontal temperature gradient leads to a temperature profile and hence a buoyancy force that is nearly constant along a single horizontal line. This evens out the horizontal differences in buoyancy and contributes to the minimization of the vorticity. A good agreement with respect to the reference paper can be observed for the isolines of the temperature field, in particular for the lowest λ_Q value.

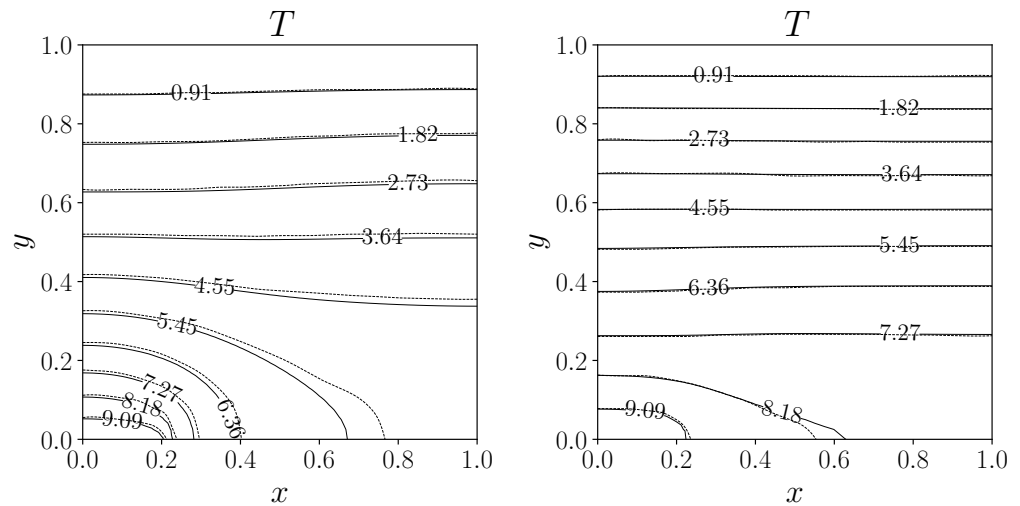


Figure 6. Contour plot of the temperature field for the controlled cases, $\lambda_Q = 10^{-3}$ on the left and $\lambda_Q = 10^{-5}$ on the right. Solid line for the present results and dashed line for the reference case [11].

The contour plot of the control Q is shown in Figure 7, on the left for the case with $\lambda_Q = 10^{-3}$ and on the right for $\lambda_Q = 10^{-5}$. We recall that the control Q acts as the volumetric source of the temperature equation, and we can see the different distributions for the two values of λ_Q . The lower the regularization factor λ_Q , the more the Q field concentrates in magnitude at the bottom of the domain.

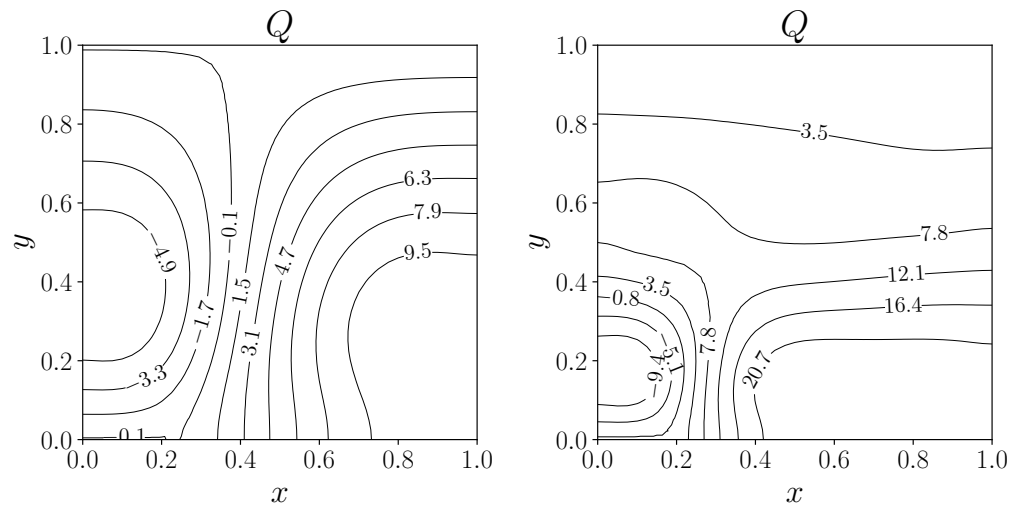


Figure 7. Contour plot of control Q for the distributed case. On the left with $\lambda_Q = 10^{-3}$, on the right with $\lambda_Q = 10^{-5}$.

Figure 8 shows the adjoint velocity vector glyphs for each controlled case.

Similarly, Figure 9 shows the velocity vector glyphs for each controlled case. Note that in this case, we have rescaled the velocity profiles by a common scalar in order to visualize the overall reduction in magnitude.

Specifically, the computed values of $|\mathbf{u}|_{max}$ are reported in Table 2, where it can be noticed that the velocity magnitude decreases by increasing the control contribution, i.e., with a lower λ_Q value.

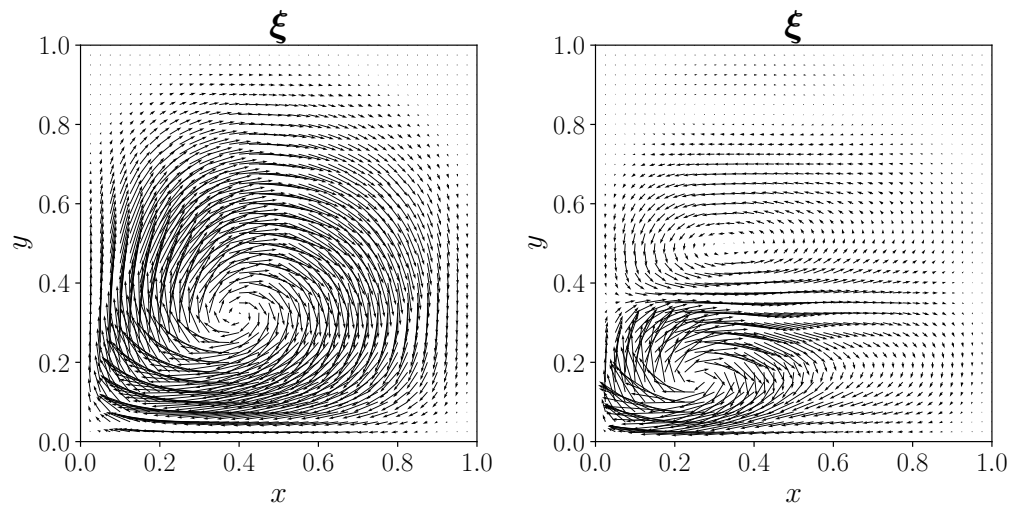


Figure 8. Adjoint velocity field for the distributed case, on the left for $\lambda_Q = 10^{-3}$, on the right for $\lambda_Q = 10^{-5}$.

Table 2. Maximum velocity magnitude for the uncontrolled and distributed controlled cases.

	Uncontrolled	$\lambda_Q = 10^{-3}$	$\lambda_Q = 10^{-5}$
$ \mathbf{u} _{max}$	1.034×10^{-1}	3.624×10^{-2}	4.039×10^{-3}

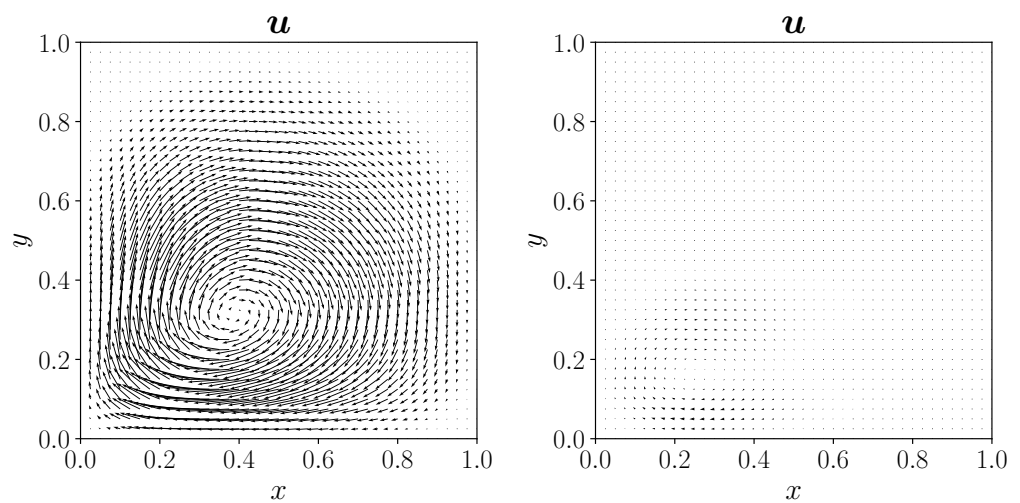


Figure 9. Velocity field for the distributed case, on the left for $\lambda_Q = 10^{-3}$, on the right for $\lambda_Q = 10^{-5}$.

In Table 3, we report the value of the objective term, the regularization term, and the total functional for the distributed case, with a comparison to the same results from [11]. In the last column, we have reported the relative error of the total functional, computed as

$$\varepsilon_r = \frac{|\mathcal{F}_C - \mathcal{F}_R|}{\mathcal{F}_R}, \tag{31}$$

where the subscripts C and R stand for Coupled and Reference, respectively. For the same reason, the results presented in this work are labeled with C in the table. As we can see, the proposed algorithm is able to reproduce the same values of the cost functional terms with an error below 2%, validating the accuracy of the numerical code coupling. A good agreement is evident in each of the above cases. Slight differences appear for the

uncontrolled case, while the results of the code-coupling algorithm become closer and closer to the literature ones by decreasing the control parameter λ_Q .

Table 3. Cost functionals for the vorticity minimization with a distributed control. In the last column, the relative error of the total functional is reported.

λ_Q	$\ \nabla \times \mathbf{u}\ $		$\ Q\ $		$\mathcal{F}(\mathbf{u}, p, T, Q)$		ϵ_r
	C	[11]	C	[11]	C	[11]	
Uncontrolled	4.326×10^{-1}	4.291×10^{-1}	-	-	9.356×10^{-2}	9.207×10^{-2}	1.6%
10^{-3}	1.383×10^{-1}	1.392×10^{-1}	5.672×10^0	5.719×10^0	2.565×10^{-2}	2.605×10^{-2}	1.5%
10^{-5}	1.553×10^{-2}	1.651×10^{-2}	1.201×10^1	1.177×10^1	8.414×10^{-4}	8.288×10^{-4}	1.5%

The number of iterations of the steepest descent method for the cases $\lambda_Q = 1 \times 10^{-3}$ and 1×10^{-5} is 121 and 786, respectively. The value increases as we reduce the control parameter. These iteration counts are comparable with the data reported in the reference paper [11].

4.1.3. Neumann Boundary Control

Regarding the Neumann boundary control, in Figure 10, we plot the control g_N along the boundary Γ_N for each controlled case ($\lambda_N = 10^{-3}$ and $\lambda_N = 10^{-5}$). In this case, this control represents the Neumann boundary condition for the temperature equation, which can be understood as the wall heat flux provided in order to minimize the vorticity. For the highest λ_N case, the control g_N seems to grow monotonically, while for $\lambda_N = 10^{-5}$, we can notice a peak in the left part of Γ_N , which is attached to Γ_1 where we have a fixed temperature boundary condition. We remark that on the point between Γ_1 and Γ_N , the adjoint temperature θ is zero. Hence, θ cannot help pushing the control at this point and so the control g_N tends to have a local maximum nearby in order to compensate for the lack of push.

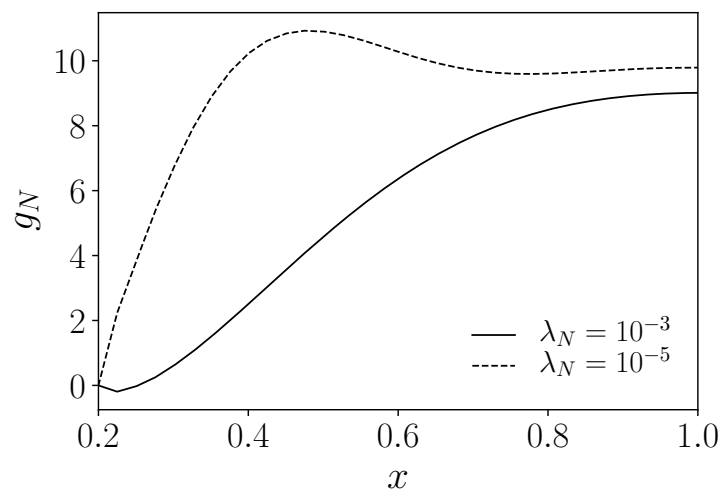


Figure 10. Neumann boundary control g_N for the two controlled cases on Γ_N .

In Figure 11, the contour plot for the temperature is reported for the Neumann boundary control case for each λ_N , together with the reference case [11], represented by dashed lines. Some discrepancies appear between the solid isolines and the corresponding dotted lines as reported in [11], especially for the case $\lambda_N = 10^{-3}$ in the bottom part

of the domain, where indeed we have the control g_N . This situation improves with the lower λ_N .

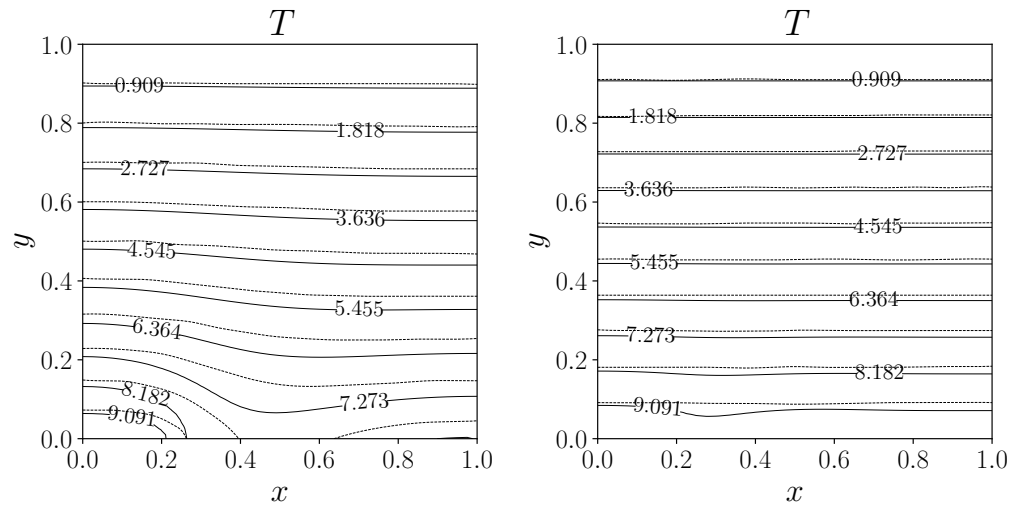


Figure 11. Contour plot of the temperature field for the controlled cases, $\lambda_N = 10^{-3}$ on the left and $\lambda_N = 10^{-5}$ on the right. Solid line for the present results and dashed line for the reference case [11].

The velocity fields are reported in Figure 12 for both controlled cases. Moreover, in Table 4, we report a comparison of the maximum value of the velocity magnitude for the computed cases, and we notice a difference of one order of magnitude between the results of the two λ_N values.

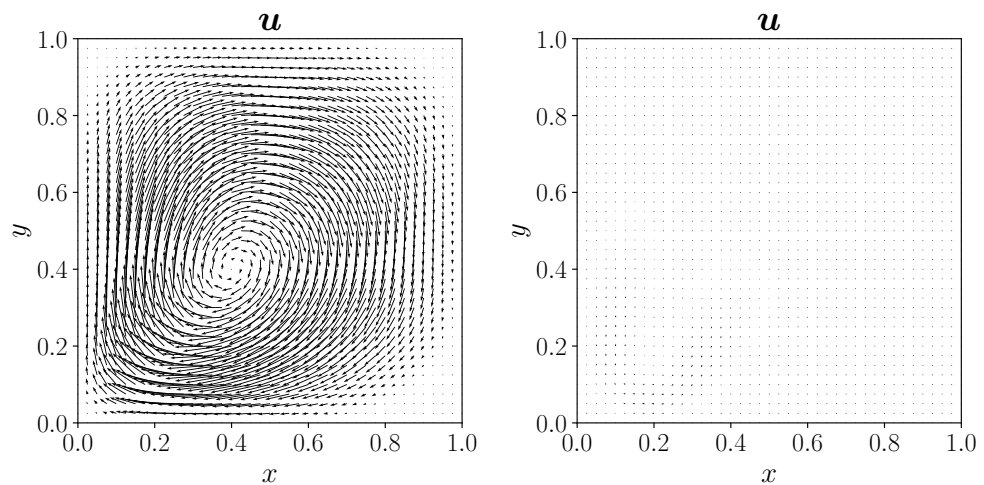


Figure 12. Velocity field for the Neumann boundary control case, on the left for $\lambda_N = 10^{-3}$, on the right for $\lambda_N = 10^{-5}$.

Table 4. Maximum velocity magnitude for the uncontrolled and Neumann boundary-controlled cases.

	Uncontrolled	$\lambda_N = 10^{-3}$	$\lambda_N = 10^{-5}$
$ \mathbf{u} _{max}$	1.034×10^{-1}	2.554×10^{-2}	1.412×10^{-3}

In Table 5, we report the value of the objective term, the regularization term, and the total functional for the Neumann boundary control case, with a comparison of the same results from [11].

Table 5. Cost functionals for the vorticity minimization with a Neumann boundary control. In the last column, the relative error of the total functional is reported.

λ_N	$\ \nabla \times \mathbf{u}\ $		$\ g_N\ $		$\mathcal{F}(\mathbf{u}, p, T, g_N)$		ε_r
	C	[11]	C	[11]	C	[11]	
Uncontrolled	4.326×10^{-1}	4.291×10^{-1}	-	-	9.356×10^{-2}	9.207×10^{-2}	1.6%
10^{-3}	1.071×10^{-1}	1.033×10^{-1}	5.718×10^0	6.677×10^0	2.209×10^{-2}	2.763×10^{-2}	20%
10^{-5}	6.250×10^{-3}	2.187×10^{-3}	8.429×10^0	8.838×10^0	3.748×10^{-4}	3.929×10^{-4}	4.6%

Differently from the distributed control case, the values computed in Table 5 for the Neumann control are sometimes significantly distant from the literature. Part of these differences may derive both from interpolation error on the boundary and from the different way of handling mixed Dirichlet–Neumann boundary conditions in the FE-FV frameworks. This effect will be better analyzed in the next numerical example, where it will be a key factor. Anyhow, the total value of the functional \mathcal{F} with the proposed algorithm is smaller than the one obtained in [11].

For this case, the number of iterations of the Steepest descent method is 46 and 211, when $\lambda_Q = 1 \times 10^{-3}$ and 1×10^{-5} , respectively. Like the previous case, the value increases as we decrease the control parameter and the iteration count is similar to the values in [11].

4.2. Solid/Fluid Heat Transfer Problem

In the second test, we reproduce the numerical examples proposed in [12], which we use as a benchmark. Here, a fluid domain exchanges heat with a neighboring solid region through a common boundary. This configuration is typically referred to as a conjugate heat transfer problem. The geometry is described in Figure 2, which shows two neighboring regions, Ω_s for the solid and Ω_f for the fluid. The boundary walls of the fluid domain are given by Γ_4 and Γ_c , which is the common interface with the solid. On the walls, a no-slip condition is enforced, i.e., a homogeneous Dirichlet boundary condition for the velocity field. On the other hand, Γ_i and Γ_o are the inlet and the outlet boundaries of the channel through which the fluid flows. We impose a Dirichlet boundary condition at the inlet and a stress-free condition at the outlet.

The boundaries of the solid domain are $\Gamma_1, \Gamma_2, \Gamma_3$, and Γ_c . The boundary conditions for the solid region are given by a Neumann boundary condition for each external edge, while on the common interface Γ_c , we enforce the continuity of the temperature and of the heat flux between the two domains. The aforementioned benchmark case has been obtained within a finite element environment. This presents some challenges when transitioning to the finite volume method.

In the FE formulation, the point at the junction $\Gamma_i \cap \Gamma_1$ belongs to boundary sections that require a temperature Dirichlet condition on one side and a Neumann condition on the other. In the FE context with Lagrange shape functions, the point itself is a degree of freedom carrier, and the Dirichlet condition can be enforced in essential way.

On the other hand, in the FV framework, the degrees of freedom are given by the cell centers, and a Dirichlet boundary condition involves the face of an element. In this situation, the intersection point will not behave in the FV code in the same way as in the FE code. Therefore, the FE versus the FV solutions show a rather different behavior at this intersection point. We have observed with auxiliary numerical tests that the results in [12] have been obtained with a mesh size that yields an FE solution that is quite far from grid convergence. These numerical tests are presented in Figure 13, which shows a grid convergence analysis for the state variable T on Γ_c . This solution will serve as the target for

the following numerical examples. Indeed, the temperature behavior obtained with the finest grid (2560×2560 elements) is quite far from the reference solution (40×40 elements).

In spite of that, in order to compare with the reference paper values, we decided to choose an FV mesh with approximately the same dof count. At such a resolution, the Dirichlet value computed by the FV code on the lower-left corner of the solid domain turns out to be quite far from the FE value. We were able to fix this difference by simply enforcing a Dirichlet condition also on the first solid element above Γ_c . Since in the FV framework a Dirichlet boundary condition fixes only the average value on the cell face, the only way to approximately fix the corner is to impose a Dirichlet condition on both faces that insist on that node. We point out that this action produces the same effect as enforcing the Dirichlet condition instead of the Neumann one on a boundary intersection point in the FE framework. The h -independent solution is not affected by this choice, both in the FV and the FE cases. However, coarser solutions show differences depending on which choice has been made for imposing such boundary conditions. For this reason, to compare the results with the 40×40 reference case shown in Figure 13, we need to make the same assumptions on the corner node.

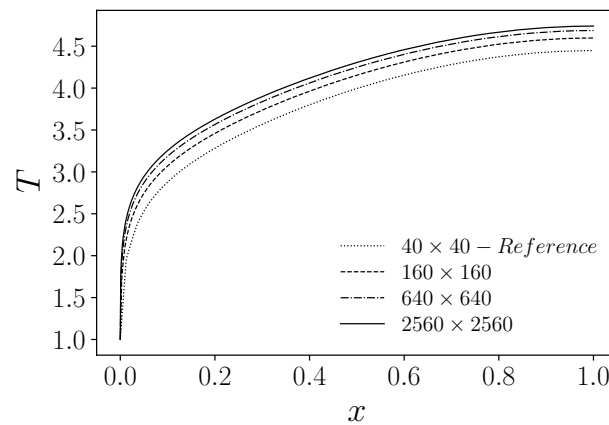


Figure 13. Target temperature profile on Γ_c for different grid refinements.

We use a 40×40 element grid for the FE code and a similar-sized (from the dof point of view) 80×80 cell grid for the FV code to replicate the results proposed in [12]. The latter grid is divided as a 60×80 grid in the fluid region and a 20×80 grid in the solid region, refined towards the solid/fluid interface Γ_c . Concerning the thermal diffusivity, we have $\alpha_s = 1$ and $\alpha_f = 0.1$, while the kinematic viscosity for the momentum equations is $\nu = 1/1000$. Lastly, the volumetric heat source in the solid region is set as $Q = 6$.

We recall the optimality system in strong form for this problem, given by Equations (32)–(34), along with the boundary conditions in Table 6:

$$\text{State : } \begin{cases} \nabla \cdot \mathbf{u} = 0 & \text{on } \Omega_f, \\ -2\nu \nabla \cdot \mathcal{D}(\mathbf{u}) + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\nabla p & \text{on } \Omega_f, \\ -\alpha_f \Delta T_f + (\mathbf{u} \cdot \nabla)T_f = 0 & \text{on } \Omega_f, \\ -\alpha_s \Delta T_s = Q & \text{on } \Omega_s, \\ T_s = T_f & \text{on } \Gamma_c, \\ \mathbf{n} \cdot (k_s \nabla T_s - k_f \nabla T_f) = 0 & \text{on } \Gamma_c, \end{cases} \tag{32}$$

$$Adjoint : \begin{cases} \nabla \cdot \boldsymbol{\zeta} = 0 & \text{on } \Omega_f, \\ -2\nu \nabla \cdot \mathcal{D}(\boldsymbol{\zeta}) + \boldsymbol{\zeta}(\nabla \mathbf{u})^T - (\mathbf{u} \cdot \nabla)\boldsymbol{\zeta} = \nabla \mu - \theta \nabla T + \Delta \mathbf{u} & \text{on } \Omega_f, \\ \boldsymbol{\tau} = -\mu \mathbf{n} + 2\nu \mathcal{D}(\boldsymbol{\zeta})\mathbf{n} & \text{on } \Gamma_i, \\ -\alpha \Delta \theta - (\mathbf{u} \cdot \nabla)\theta = \beta \mathbf{g} \cdot \boldsymbol{\zeta} & \text{on } \Omega, \end{cases} \quad (33)$$

$$Control : \quad \lambda_1 \mathbf{w} - \lambda_2 \Delta_s \mathbf{w} = \boldsymbol{\tau} \quad \text{on } \Gamma_i. \quad (34)$$

Notice that the state equations are reported explicitly over the two regions since OpenFOAM solves in a segregated way, while the adjoint uses the monolithic formulation. We also remark that, as in [12], the Dirichlet condition on the inlet velocity $\mathbf{u}|_{\Gamma_i}$ is enforced only in the normal direction.

Table 6. Boundary conditions for the solid/fluid CHT problem.

	Γ_i	Γ_1	Γ_2	Γ_3	Γ_o	Γ_4	Γ_c
\mathbf{u}	$\mathbf{u} = \mathbf{w} \cdot \mathbf{n}$	-	-	-	$\sigma(\mathbf{u}, p)\mathbf{n} = 0$	$\mathbf{u} = 0$	$\mathbf{u} = 0$
T	$T = 0$	$\nabla T \cdot \mathbf{n} = 0$	$\nabla T \cdot \mathbf{n} = 0$	$\nabla T \cdot \mathbf{n} = 0$	$\nabla T \cdot \mathbf{n} = 0$	$\nabla T \cdot \mathbf{n} = 0$	CHT
$\boldsymbol{\zeta}$	$\boldsymbol{\zeta} = 0$	-	-	-	$\sigma(\boldsymbol{\zeta}, \mu)\mathbf{n} = 0$	$\boldsymbol{\zeta} = 0$	$\boldsymbol{\zeta} = 0$
θ	$\theta = 0$	$\nabla \theta \cdot \mathbf{n} = 0$	$\nabla \theta \cdot \mathbf{n} = 0$	$\nabla \theta \cdot \mathbf{n} = 0$	$\nabla \theta \cdot \mathbf{n} = 0$	$\nabla \theta \cdot \mathbf{n} = 0$	CHT

4.2.1. Definition of the Target Temperature

Following [12], the target of the control problem is the temperature profile T_d on $\Gamma_d \subset \Gamma_c$, which is obtained with an analytical input velocity $\mathbf{u}_0|_{\Gamma_i} = (2(1.5y - 2y^2), 0)^T$, i.e., a Poiseuille flow. In the following, we refer to this as the uncontrolled solution. The target T_d is then fed to the optimal control system. For the solution of the optimality system, the first control in the steepest descent method is initialized with a different inlet boundary condition for the velocity field, i.e., $\mathbf{u}|_{\Gamma_i} = ((1.5y - 2y^2), 0)^T$, and therefore the control acts on the interface boundary to reach a velocity field that matches the temperature distribution T_d as close as possible. It appears clear that the optimal solution should provide a control on the boundary that reproduces the original boundary inlet condition for the velocity field \mathbf{u}_0 . In this way, it is easy to check the quality of the optimal control solution. Of course, due to the presence of a regularization term, i.e., a parameter $\lambda_i > 0$, the control is not able to reproduce exactly that original velocity profile, but it provides the closest solution according to the value of λ_i .

In Figure 14, we report the target temperature solution. On the left, we show the overall behavior as a contour map and on the right, we plot the trend on a portion of the interface Γ_c , which corresponds to the target temperature T_d . We recall that the objective functional was defined only on a portion of the solid/fluid interface, i.e., the region $\Gamma_d = \{(x, y) \in \mathbb{R}^2 : x \in [0.075, 1], y = 0.75\}$. The isolines in the left figure confirm the interface conditions of the CHT model, i.e., the continuity of the temperature and the change in slope due to the continuity of the heat flux across different conductivities.

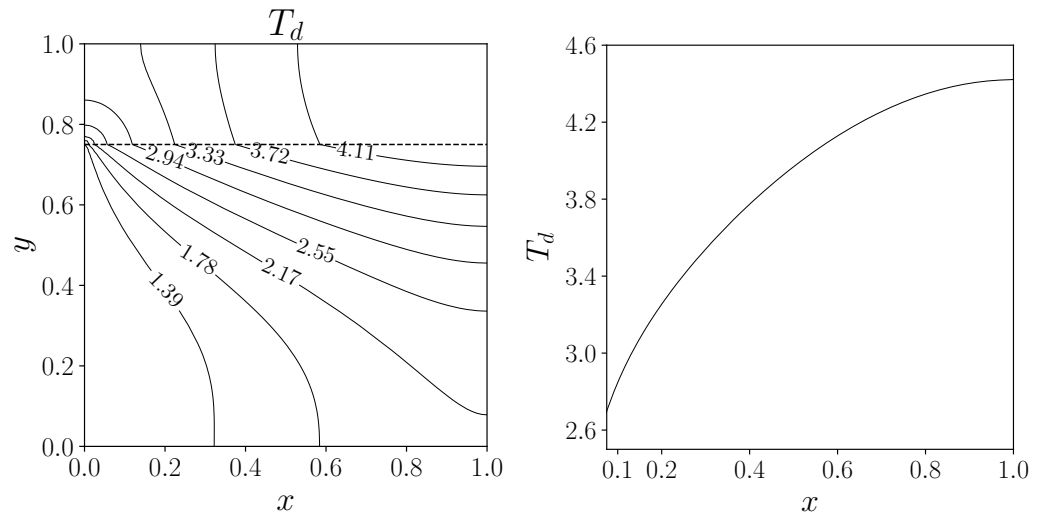


Figure 14. On the left, target temperature isolines over Ω (the dashed line marks the interface between fluid and solid); on the right, target temperature profile on Γ_d .

4.2.2. Dirichlet Boundary Control

In Figure 15, the Dirichlet velocity control on Γ_i is reported as a solid line for two controlled cases. In particular in Case 1 we set the pair (λ_1, λ_2) to $(5 \times 10^{-2}, 5 \times 10^{-3})$, while in Case 2 we set $(\lambda_1, \lambda_2) = (8 \times 10^{-3}, 8 \times 10^{-4})$. The comparison with the reference data of [12] is shown with circular markers.

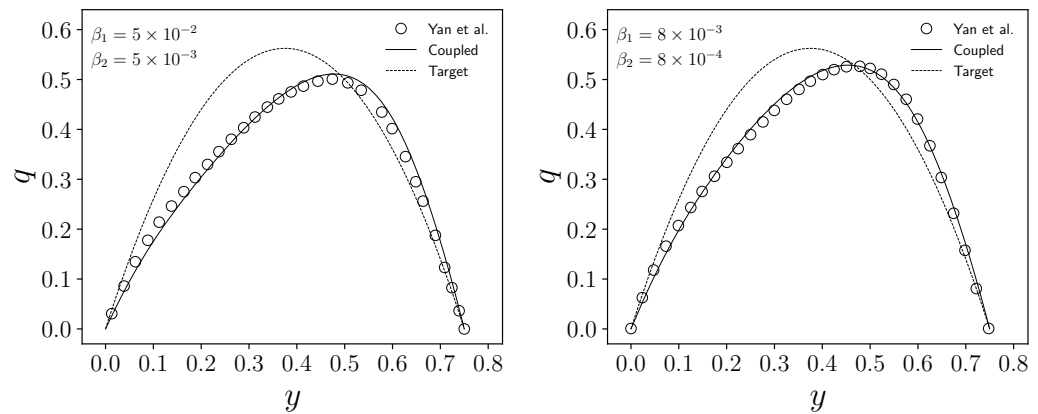


Figure 15. Normal component of the control \mathbf{w} with respect to Γ_i for the two controlled cases with different λ . Reference data [12] are reported with the circular markers.

We notice a good agreement between our results and the reference data for each controlled case. The inlet velocity control that we obtain is not symmetric with respect to the midpoint $(0, 0.375)$ of Γ_i . In fact, since we want to match the temperature on a portion of Γ_c , and since the solid has a volumetric heat source, we expect to have an advection contribution of the fluid that cools down the solid, i.e., a velocity field which is able to remove the heat generated. Therefore, the peak of the velocity profile is shifted towards the interface boundary. We also notice, as expected, that a lower combination of the values of λ_i yields a velocity profile closer to the target one represented by the dashed line.

In Figure 16, we show the temperature profiles on Γ_d and the corresponding relative percentage errors with respect to the target value for both controlled cases. The temperature profile obtained with the initial guess of the inlet velocity $\mathbf{u}|_{\Gamma_i}$ is also reported. Regarding the relative error with respect to T_d , we define

$$\varepsilon_r = \frac{|T_i - T_d|}{T_d} \cdot 100, \tag{35}$$

where T_i corresponds to the two controlled cases.

As expected, the numerical temperature field does not coincide with the target temperature T_d since we weight the control contribution with the λ_i parameters. As a result, the case with the lowest λ_i values provides a temperature distribution on Γ_c that is closer to T_d . In fact, Case 2 presents a maximum error below 0.2%, almost three times less than Case 1, which reaches a value close to 0.6%. This aspect is confirmed later when we provide the values of the cost functional terms, specifically the objective term.

The adjoint velocity and the adjoint temperature have been reported in Figure 17 for both controlled cases. More precisely, we report the vector glyph for the adjoint velocity on the left and the contour map for the adjoint temperature on the right. With regards to this latter plot, the adjoint temperature is multiplied by 10^3 , since its absolute values are relatively low.

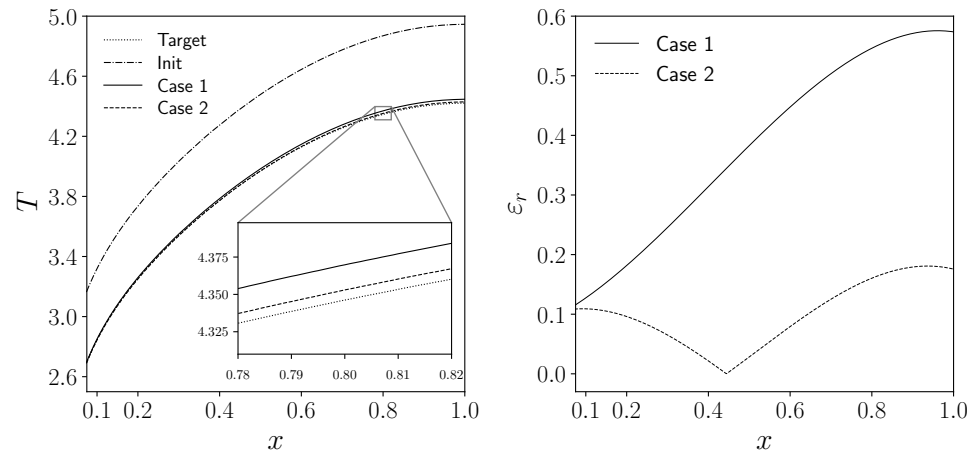


Figure 16. Temperature profile on Γ_d on the left, and corresponding relative temperature error ε_r on the right. Solid line for Case 1, dashed line for Case 2, dotted line for T_d , and dash-dotted line for the initial temperature distribution associated to $\mathbf{u}|_{\Gamma_i}$.

As we know from the literature [12,42], the adjoint velocity of a fluid channel configuration with an inlet and an outlet behaves like a flow moving in the opposite direction. Indeed, the adjoint inlet is now Γ_o , while the other boundaries behave as a wall for the adjoint velocity, since velocity Dirichlet boundary conditions are enforced on them.

In analogy with the previous test, in Table 7 we report the values of the cost functional terms, both for the uncontrolled case and for the two controlled cases. Unlike the boundary control of the Rayleigh–Bénard application, in this case, the boundary control is in $\mathbf{H}^1(\Gamma_i)$, hence we take into account two contributions for the regularization part, which are given in separate columns.

Table 7. Cost functionals for the temperature minimization with a Dirichlet boundary control on the velocity inlet value.

λ_1	λ_2	$\ T - T_d\ $	$\ q\ $	$\ \nabla_s q\ $	$\mathcal{F}(\mathbf{u}, p, T, q)$
Uncontrolled		2.391×10^{-1}	1.778×10^{-1}	7.506×10^{-1}	-
5×10^{-2}	5×10^{-3}	1.677×10^{-2}	3.160×10^{-1}	1.416×10^0	7.647×10^{-3}
8×10^{-3}	8×10^{-4}	4.538×10^{-3}	3.313×10^{-1}	1.448×10^0	1.287×10^{-3}

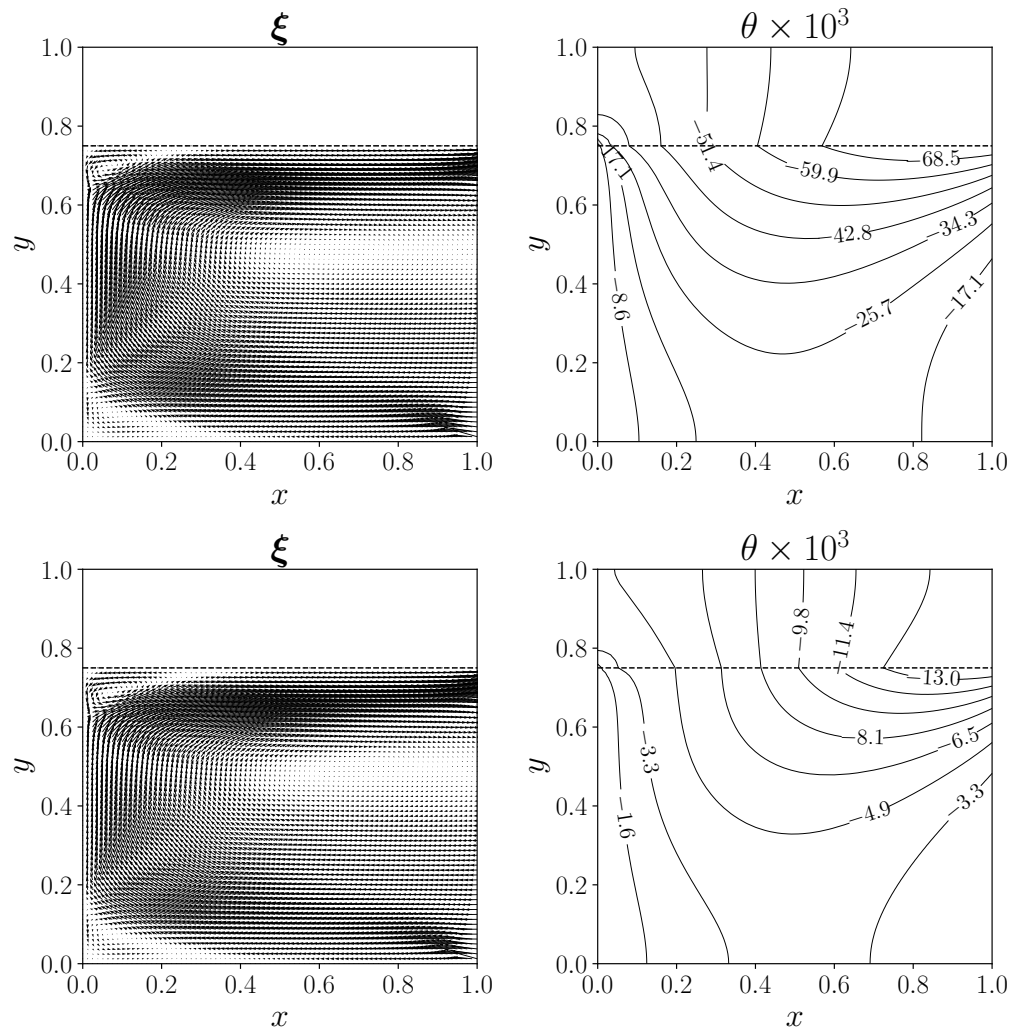


Figure 17. Adjoint velocity ξ (vector glyph) on the left and adjoint temperature θ (isolines) on the right. The dashed line marks the interface between fluid and solid. Case 1 on top, Case 2 on bottom.

The table confirms the expected behavior of the functional value in the last column, showing a lower number for the better-controlled case, i.e., the one with the lowest λ_i . In fact, with the latter values of λ_i , we obtain a total functional almost six times lower than the other controlled case. This reduction is mainly due to the objective term, which has a decrease of one order of magnitude, despite the regularization terms being similar. When $\lambda_1 = 5 \times 10^{-2}$ and $\lambda_2 = 5 \times 10^{-3}$, the steepest descent requires 333 iterations to reach convergence. With the smaller values $\lambda_1 = 8 \times 10^{-3}$ and $\lambda_2 = 8 \times 10^{-4}$, the number of iterations required decreases to 228. Contrary to the previous cases, we see a reduction in the value with smaller control parameters.

4.3. Steepest Descent Convergence Order

In order to verify the correctness of our numerical implementation, we computed the order of convergence of the steepest descent algorithm for all the proposed cases. We know from the theory that the steepest descent method has a linear order of convergence [43]. This test allows us to check the reliability of the coupling routines and the magnitude of the error made in the coupling process. We recall the formula for the convergence order p as

$$p = \lim_{k \rightarrow \infty} \frac{\log(\|\mathcal{F}_{k+1} - \mathcal{F}_{min}\|)}{\log(\|\mathcal{F}_k - \mathcal{F}_{min}\|)}, \tag{36}$$

where k represents the numerical iterations and \mathcal{F}_{min} is the minimum.

Tables 8 and 9 present the computed convergence order p for all the analyzed cases. This value is obtained by averaging the convergence order values of the steepest descent iterations computed using Formula (36). The results show good agreement with the theoretical value of the gradient descent algorithm, which is equal to one. Therefore, we can state that the use of segregated codes with different discretization techniques for the solution of optimal control systems does not affect the convergence order of the steepest descent method.

Table 8. Convergence order p for the Rayleigh–Bénard control case.

	λ_Q		λ_N	
	10^{-3}	10^{-5}	10^{-3}	10^{-5}
p	9.940×10^{-1}	$1.000 \times 10^{+0}$	9.936×10^{-1}	9.996×10^{-1}

Table 9. Convergence order p for the solid–fluid control case.

	Case 1	Case 2
p	9.818×10^{-1}	9.948×10^{-1}

4.4. Data Transfer Overhead

Lastly, we assess the impact of coupling overhead time on the simulation. To this end, we compare the total data transfer time—accounting for both forward and backward communication—with the time required by a single solver iteration. The resulting relative time t_{rel} , defined as the ratio between the data transfer time and the total iteration time given by the sum of coupling and solving time, is shown in Figure 18 for different mesh refinements N_{lev} . Specifically, we consider the mesh at the first level with 1600 degrees of freedom, the mesh at the second with 6400, and so on. As observed, the relative importance of data transfer decreases as the mesh size increases. This is due to the solver time growing roughly quadratically with the mesh size, while the coupling routines scale linearly. Furthermore, it is worth noting that in stationary optimization cases, such as those presented in this paper, data transfer occurs only once per stationary iteration, and therefore its relative impact is significantly lower.

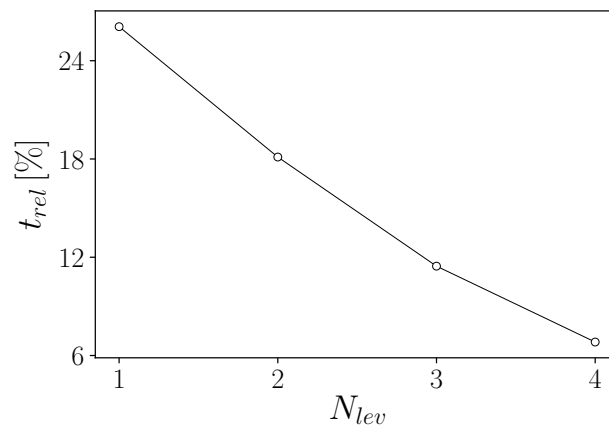


Figure 18. Data transfer relative time as a function of the mesh size with 4 levels of refinement.

We also notice that for the finest computational grid, the relative cost of the data transfer is much smaller than the total computational cost of the simulation, a value close to 6%.

5. Conclusions

This paper presents an extension of a code-coupling technique for optimal control problems already published by the authors. Specifically, we have considered optimal control problems for two fluid heat transfer applications, the first one is described by the Rayleigh–Bénard system for a natural convection cavity configuration while the second one considers the conjugate heat transfer between a solid and a fluid region. The presented approach is based on the solution of the optimality system employing two different CFD codes and the data exchange between them using the external MEDCOUPLING library. In fact, the state equations are solved in the finite volume code OpenFOAM while the adjoint and the control system are discretized with the finite element code FEMuS.

Two numerical examples have been reported with a comparison of the same application setup taken from the literature. We achieve good agreement with reference literature data obtained with a monolithic approach, validating the goodness of the numerical approach.

Future lines of research will focus on leveraging the versatility of this approach to apply optimal control to real multiphysics and multiscale environments. Additionally, we plan to implement more advanced minimization algorithms in the coupling framework, such as Newton and quasi-Newton methods, to enhance the computational efficiency.

The computational approach described in this paper can be a valuable tool for spreading the optimal control features into widely used CFD codes, enabling an engineering design approach based on physical target values to be reached.

Author Contributions: Conceptualization, S.B., G.B. (Giacomo Barbi), G.B. (Giorgio Borna), A.C., F.G., S.M. and L.S.; methodology, S.B., G.B. (Giacomo Barbi), G.B. (Giorgio Borna), A.C., F.G., S.M. and L.S.; writing—original draft, S.B., G.B. (Giacomo Barbi), G.B. (Giorgio Borna), A.C., F.G., S.M. and L.S.; writing—review and editing, S.B., G.B. (Giacomo Barbi), G.B. (Giorgio Borna), A.C., F.G., S.M. and L.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Manzoni, A.; Quarteroni, A.; Salsa, S. *Optimal Control of Partial Differential Equations*; Springer: Berlin/Heidelberg, Germany, 2021.
2. Jan, S.; Jean-Paul, Z. *Introduction to Shape Optimization*; Springer: Berlin/Heidelberg, Germany, 1992.
3. Chertovskih, R.; Ribeiro, V.M.; Gonçalves, R.; Aguiar, A.P. Sixty Years of the Maximum Principle in Optimal Control: Historical Roots and Content Classification. *Symmetry* **2024**, *16*, 1398. [[CrossRef](#)]
4. Gunzburger, M.D. *Perspectives in Flow Control and Optimization*; SIAM: Bangkok, Thailand, 2002.
5. Chierici, A.; Giovacchini, V.; Manservigi, S. Analysis and Computations of Optimal Control Problems for Boussinesq Equations. *Fluids* **2022**, *7*, 203. [[CrossRef](#)]
6. Barbi, G.; Cervone, A.; Giangolini, F.; Manservigi, S.; Sirotti, L. Numerical Coupling between a FEM Code and the FVM Code OpenFOAM Using the MED Library. *Appl. Sci.* **2024**, *14*, 3744. [[CrossRef](#)]
7. Baldini, S.; Barbi, G.; Cervone, A.; Giangolini, F.; Manservigi, S.; Sirotti, L. Optimal Control of Heat Equation by Coupling FVM and FEM Codes. *Mathematics* **2025**, *13*, 238. [[CrossRef](#)]
8. Jasak, H.; Jemcov, A.; Tukovic, Z. OpenFOAM: A C++ library for complex physics simulations. In Proceedings of the International Workshop on Coupled Methods in Numerical Dynamics, Dubrovnik, Croatia, 19–21 September 2007; Volume 1000, pp. 1–20.
9. Barbi, G.; Borna, G.; Cerroni, D.; Cervone, A.; Chierici, A.; Chirco, L.; Viá, R.; Giovacchini, V.; Manservigi, S.; Scardovelli, R.; et al. FEMuS-Platform: A numerical platform for multiscale and multiphysics code coupling. In Proceedings of the 9th International Conference on Computational Methods for Coupled Problems in Science and Engineering, COUPLED PROBLEMS, Chia Bay, Italy, 13–16 June 2021; International Center for Numerical Methods in Engineering: Barcelona, Spain, 2021; pp. 1–12.

10. Ribes, A.; Caremoli, C. Salome platform component model for numerical simulation. In Proceedings of the 31st Annual International Computer Software and Applications Conference (COMPSAC 2007), Beijing, China, 23–37 July 2007; Volume 2, pp. 553–564.
11. Lee, H.C. Optimal control problems for the two dimensional Rayleigh–Bénard type convection by a gradient method. *Jpn. J. Ind. Appl. Math.* **2009**, *26*, 93. [[CrossRef](#)]
12. Yan, Y.; Keyes, D.E. Smooth and robust solutions for Dirichlet boundary control of fluid–solid conjugate heat transfer problems. *J. Comput. Phys.* **2015**, *281*, 759–786. [[CrossRef](#)]
13. Cutelier, C. Optimal Control of a System Governed by the Navier-Stokes Equations Coupled with the Heat Equation. In *North-Holland Mathematics Studies*; Elsevier: Amsterdam, The Netherlands, 1976; Volume 21, pp. 81–98.
14. Alekseev, G.; Tereshko, D. Stability of optimal controls for the stationary Boussinesq equations. *Int. J. Differ. Equ.* **2011**, *2011*, 535736. [[CrossRef](#)]
15. Baranovskii, E.S.; Domnich, A.A.; Artemov, M.A. Optimal boundary control of non-isothermal viscous fluid flow. *Fluids* **2019**, *4*, 133. [[CrossRef](#)]
16. Belmiloudi, A. Robin-type boundary control problems for the nonlinear Boussinesq type equations. *J. Math. Anal. Appl.* **2002**, *273*, 428–456. [[CrossRef](#)]
17. Lee, H.C.; Imanuvilov, O.Y. Analysis of Neumann boundary optimal control problems for the stationary Boussinesq equations including solid media. *SIAM J. Control Optim.* **2000**, *39*, 457–477. [[CrossRef](#)]
18. Lee, H.C.; Imanuvilov, O.Y. Analysis of optimal control problems for the 2-D stationary Boussinesq equations. *J. Math. Anal. Appl.* **2000**, *242*, 191–211. [[CrossRef](#)]
19. Lee, H.C.; Shin, B.C. Piecewise optimal distributed controls for 2D Boussinesq equations. *Math. Methods Appl. Sci.* **2000**, *23*, 227–254. [[CrossRef](#)]
20. Lee, H.C.; Shin, B.C. Dynamics for controlled 2-D Boussinesq systems with distributed controls. *J. Math. Anal. Appl.* **2002**, *273*, 457–479. [[CrossRef](#)]
21. Lee, H.C. Analysis and computational methods of Dirichlet boundary optimal control problems for 2D Boussinesq equations. *Adv. Comput. Math.* **2003**, *19*, 255–275. [[CrossRef](#)]
22. Li, S. Optimal controls of Boussinesq equations with state constraints. *Nonlinear Anal. Theory Methods Appl.* **2005**, *60*, 1485–1508. [[CrossRef](#)]
23. Li, S.; Wang, G. The time optimal control of the Boussinesq equations. *Numer. Funct. Anal. Optim.* **2003**, *24*, 163–180. [[CrossRef](#)]
24. Fursikov, A.V.; Imanuvilov, O.Y. Local exact boundary controllability of the Boussinesq equation. *SIAM J. Control Optim.* **1998**, *36*, 391–421. [[CrossRef](#)]
25. Mallea-Zepeda, E.; Lenes, E.; Valero, E. Boundary control problem for heat convection equations with slip boundary condition. *Math. Probl. Eng.* **2018**, *2018*, 7959761. [[CrossRef](#)]
26. Rueda-Gómez, D.A.; Villamizar-Roa, E.J. On the Rayleigh–Bénard–Marangoni system and a related optimal control problem. *Comput. Math. Appl.* **2017**, *74*, 2969–2991. [[CrossRef](#)]
27. Gunzburger, M.D.; Lee, H.C. Analysis, approximation, and computation of a coupled solid/fluid temperature control problem. *Comput. Methods Appl. Mech. Eng.* **1994**, *118*, 133–152. [[CrossRef](#)]
28. Gunzburger, M.D.; Hou, L.S.; Svobodny, T.P. The approximation of boundary control problems for fluid flows with an application to control by heating and cooling. *Comput. Fluids* **1993**, *22*, 239–251. [[CrossRef](#)]
29. Adams, R.A.; Fournier, J.J. *Sobolev Spaces*; Elsevier: Amsterdam, The Netherlands, 2003.
30. Bornia, G.; Ratnavale, S. Different approaches for Dirichlet and Neumann boundary optimal control. *AIP Conf. Proc.* **2018**, *1978*, 270006.
31. Bornia, G.; Chierici, A.; Ratnavale, S. A comparison of regularization methods for boundary optimal control problems. *Int. J. Numer. Anal. Model.* **2022**, *19*, 329–346.
32. D’Elia, M.; Gunzburger, M. The fractional Laplacian operator on bounded domains as a special case of the nonlocal diffusion operator. *Comput. Math. Appl.* **2013**, *66*, 1245–1260. [[CrossRef](#)]
33. Benzi, M.; Olshanskii, M.A. An Augmented Lagrangian-Based Approach to the Oseen Problem. *SIAM J. Sci. Comput.* **2006**, *28*, 2095–2113. [[CrossRef](#)]
34. Furkisso, A.V.; Gunzburger, M.D.; Hou, L.S. Boundary Value Problems and Optimal Boundary Control for the Navier-Stokes systems: The two-dimensional case. *SIAM J. Control Optim.* **1998**, *36*, 852–894.
35. Gunzburger, M.; Hou, L.; Svobodny, T.P. Analysis and finite element approximation of optimal control problems for the stationary Navier-Stokes equations with Dirichlet controls. *ESAIM Math. Model. Numer. Anal.* **1991**, *25*, 711–748. [[CrossRef](#)]
36. David, G.L.; Yinyu, Y. *Linear and Nonlinear Programming*; Springer: Berlin/Heidelberg, Germany, 2021.
37. Ern, A.; Guermond, J.L. *Theory and Practice of Finite Elements*; Springer: Berlin/Heidelberg, Germany, 2004; Volume 159.
38. Vivette, G.; Pierre-Arnaud, R. *Finite Element Methods for Navier-Stokes Equations*; Springer: Berlin/Heidelberg, Germany, 2012.

39. Ciarlet, P.G. *The Finite Element Method for Elliptic Problems*; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2002.
40. Silva, G.H.; Le Riche, R.; Molimard, J.; Vautrin, A. Exact and efficient interpolation using finite elements shape functions. *Eur. J. Comput. Mech.* **2009**, *18*, 307–331. [[CrossRef](#)]
41. Da Vià, R. Development of a Computational Platform for the Simulation of Low Prandtl Number Turbulent Flows. Ph.D. Thesis, University of Bologna, Bologna, Italy, 2019.
42. Pichi, F.; Strazzullo, M.; Ballarin, F.; Rozza, G. Driving bifurcating parametrized nonlinear PDEs by optimal control strategies: Application to Navier–Stokes equations with model order reduction. *ESAIM Math. Model. Numer. Anal.* **2022**, *56*, 1361–1400. [[CrossRef](#)]
43. Meza, J.C. Steepest descent. *Wiley Interdiscip. Rev. Comput. Stat.* **2010**, *2*, 719–722. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.