Parallelized POD-based Suboptimal Economic Model Predictive Control of a State-Constrained Boussinesq approximation

Julian Andrej∗, Lars Grüne†, Luca Mechelli‡, Thomas Meurer§, Simon Pirkelmann† and Stefan Volkwein‡

Motivated by an energy efficient building application, we want to optimize a quadratic cost functional subject to the Boussinesq approximation of the Navier-Stokes equations and to bilateral state and control constraints. Since the computation of such an optimal solution is numerically costly, we design an efficient strategy to compute a sub-optimal (but applicationally acceptable) solution with significantly reduced computational effort. We employ an economic Model Predictive Control (MPC) strategy to obtain a feedback control. The MPC sub-problems are based on a linear-quadratic optimal control problem subjected to mixed control and state constraints and a convection-diffusion equation, reduced with proper orthogonal decomposition. To solve each sub-problem, we apply a primal-dual active set strategy. The method can be fully parallelized, which enables the solution of large problems with real-world parameters.

Keywords: Boussinesq approximation, Model predictive control, State constraints, Proper orthogonal decomposition, Parallel computing.

1 Introduction

Coupled heat-convection phenomena like those occurring in heating, cooling and air-conditioning (HVAC) of residual buildings can be accurately modeled by the incompress-
ible Navier-Stokes equations; see [2, 38, 40], for instance. These equations describe how the physical quantities temperature, air velocities and pressure are connected and how they influence each other. While the underlying physical principles are completely described by the equations, working with them directly for the purpose of control of HVAC processes has turned out to be challenging due to the high computational complexity. Instead, simplified models of varying granularity are used in control applications.

One common simplification that still retains much of the relevant physical relations in the HVAC regime is the Boussinesq approximation [38, 40]. The general idea is to reduce the coupling between heat and airflow by only considering buoyancy induced changes of the fluid velocity. More specifically, we only consider density variations along the gravity field which lead to, e.g., the rise of warmer air. Although we obtain a simplified model, this simplification is not enough to prevent high computational cost for computing its solution. Many degrees of freedom (DOFs) are in fact necessary to accurately capture the involved dynamics, especially when dealing with real-world parameters; see, e.g., [5].

One viable approach is to apply model order reduction (MOR). This field was extensively developed during the last two decades and several techniques were proposed. Among all, in case of computational fluid dynamics, proper orthogonal decomposition (POD) seems the one to guarantee best performances for nonlinear dynamical systems; see, e.g., [6, 24, 37]. In the particular case of the Boussinesq approximation, the POD-based MOR approach is also certified by an a-posteriori error analysis [32, 33], that guarantees the validity of approximation for the computed reduced order model. Due to the request of computing many PDE solutions, MOR techniques are naturally applied in the field of optimal control. In [22], for example, a POD-based model predictive control (MPC) algorithm is considered to approximate a feedback solution to the Boussinesq equation.

MPC is a well-established model-based control method, sometimes also called Receding Horizon Control; cf. [16, 34]. The key idea is to decompose an infinite time horizon optimal control problem into several finite horizon sub-problems and, therefore, to optimize predictions of the model behavior in relatively short time. After a sub-problem is solved, the initial part of the computed optimal control is stored and used as feedback control, the time horizon is shifted and it is possible to update the problem parameters before the procedure is iterated. This technique is particularly useful in HVAC applications. It permits, in fact, to merge predictions, such as weather forecast, with system measurements. It permits, in fact, to merge predictions, such as weather forecast, with system measurements [30, 43]. Let us mention that POD-based MPC is considered also in [10, 12, 27, 29] for different models. Furthermore, we are particularly interested in economic MPC; cf. [1, 9, 11, 17, 19, 31]. In this particular branch of MPC, more general cost functionals than those employing tracking type costs are allowed. In our specific application, the cost functional depends only on the control variables, without any dependence on the state variable. The focus is then to keep the system trajectory inside pre-defined bounds with less control effort as possible and it deals perfectly with the usual requests of HVAC applications.

This work aspires to unify the above mentioned contributions in a reliable method, in order to compute a suboptimal solution to an optimal control problem subject to the Boussinesq approximation and bilateral state and control constraints. In particular,
starting from [5][27][31], we propose a POD-based economic MPC scheme which considers linear-quadratic subproblems governed by convection-diffusion equations with a virtual control approach [26, 27] and possible updates of the reduced order model through parallel solution of the Boussinesq approximation [5]. Furthermore a primal-dual active set strategy (PDASS) [21, 25] is considered to solve each sub-problem. The goal is to have a robust and reliable method that speeds up the computational time.

In Section 2, we present the optimal control problem. Section 3 describes the virtual control approach and the PDASS for linear-quadratic optimal control problems, while Section 4 contains a brief explanation of the POD method and the a-posteriori error estimate. The POD-based economic MPC algorithm is explained in Section 5 and further details about the implementation are given in Section 6. Finally, we test our algorithm for a simplified problem in Section 7.

2 The setting of the optimal control problem

The Boussinesq approximation of the Navier-Stokes equations is described by the following PDE:

\[
\frac{\partial v}{\partial t} + v \cdot \nabla v = -\frac{1}{\rho} \nabla p + \nu \Delta v - g \alpha (y - \tilde{y}) \quad \text{in } Q := (0, T) \times \Omega, \tag{1a}
\]

\[\nabla \cdot v = 0 \quad \text{in } Q, \tag{1b}\]

\[v(0) = v_0, \quad \text{in } \Omega, \tag{1c}\]

\[
\frac{\partial y}{\partial t} + v \cdot \nabla y = \frac{\kappa}{\rho c_p} \Delta y \quad \text{in } Q, \tag{1d}
\]

\[y(0) = y_0 \quad \text{in } \Omega, \tag{1e}\]

where \(v : Q \to \mathbb{R}^2\) is the air velocity, \(p : Q \to \mathbb{R}\) stands for the air pressure, and \(y : Q \to \mathbb{R}\) is the temperature, \(\Omega \subset \mathbb{R}^2\) denotes an open and bounded domain with sufficiently smooth boundary, \(T > 0\) is the time horizon, \(g \in \mathbb{R}^2\) is the gravitational acceleration, \(v_0, y_0\) are given initial conditions and \(\tilde{y} \in \mathbb{R}\) is the reference temperature at which we measure:

- the (constant) density \(\rho > 0\),
- the (constant) kinematic viscosity \(\nu > 0\),
- the (constant) coefficient of thermal expansion \(\alpha > 0\),
- the (constant) thermal conductivity \(\kappa > 0\),
- the (constant) isobaric specific heat \(c_p > 0\)

of the studied fluid (air in our case). In addition, we consider time-varying boundary conditions of the following form

\[v = 0 \quad \text{on } (0, T) \times \Gamma, \tag{1f}\]
where \( \Gamma = \partial \Omega \) and \( \gamma, \gamma_c > 0 \) are fixed parameters. The function \( u \in U = L^2(0, T) \) is a control input placed at a predefined part of the domain’s boundary, namely \( \Gamma_c \), and it represents a device for controlling the temperature in the room, e.g. a radiator or an underfloor heating system; cf. Figure 1. The quantity \( y_{\text{out}} \in L^2(0, T) \) represents the outside temperature.

We are interested in the weak solution to (1). For more details of the straightforward variational formulation for (1) we refer to [2, 38] and [22] for more details.

The goal is to minimize the control effort while keeping the temperature \( y \) inside desired bounds, i.e. we are interested in minimizing

\[
J(u) = \frac{1}{2} \|u\|^2_U
\]

subject to (s.t.) the weak formulation of (1) and the inequality constraints

\[
\begin{align*}
&u_a(t) \leq u(t) \leq u_b(t) \quad \text{a.e. in } (0, T), \quad (2b) \\
&y_a(t, \mathbf{x}) \leq y(t, \mathbf{x}) \leq y_b(t, \mathbf{x}) \quad \text{a.e. in } Q. \quad (2c)
\end{align*}
\]

Here \( y \) is the (weak) solution of (1), \( u_a, u_b \in U \) given control constraints and \( y_a, y_b \in C(Q) \) given state constraints. It follows from [38] Chap. III, §3 that (1) admits a unique weak solution

\[
(v, y) \in \mathcal{Y}_B = W(0, T; \tilde{V}) \times W(0, T; V),
\]

where \( V = H^1(\Omega) \) and the Hilbert space \( \tilde{V} \) is the closure of the space \( \{v \in C_0^\infty(\Omega)^2 \mid \nabla \cdot v = 0\} \) with respect to the norm \( \|\cdot\|_{V \times V} \). Moreover, we have \( W(0, T; \tilde{V}) = L^2(0, T; \tilde{V}) \cap H^1(0, T; \tilde{V}') \), and the set \( W(0, T; V) \) is defined analogously; see [38] Chap. XVIII] for more details. Thus, considering the control-to-state map

\[
\mathcal{S}_B : U \rightarrow \mathcal{Y}_B, \quad u \mapsto \mathcal{S}_B(u) = (v(u), y(u)) \in \mathcal{Y}_B
\]
and the admissible set
\[ \mathcal{U}_{B,\text{ad}} = \{ u \in \mathcal{U} \mid u \text{ and } y = \mathcal{S}_B(u) \text{ satisfy (2b) and (2c), respectively} \} , \]

problem (2) can be rewritten as the following purely control-constrained optimization problem

\[ \min J(u) \quad \text{s.t.} \quad u \in \mathcal{U}_{B,\text{ad}} . \]

We call \( \hat{P}_B \) the reduced problem because – in contrast to (2) – it is an optimization problem in the control variable only. Note that \( \hat{P}_B \) admits a global solution \( \bar{u}_B \in \mathcal{U}_{B,\text{ad}} \) provided \( \mathcal{U}_{B,\text{ad}} \) has non-empty interior; cf., e.g., [23]. Throughout the work a bar indicates optimality.

While the Boussinesq approximation already reduces numerical effort for simulation of the fluid processes compared to the full compressible Navier-Stokes equations, the nonlinearity induced by the two-way coupling of temperature and velocity still makes it computationally infeasible, when it is used, e.g., as the prediction model in an MPC scheme. This is because each step of the MPC algorithm typically requires many evaluations of the prediction model, which would be too costly for an online application of the method.

For that reason we elaborate a simplified strategy that aims at decreasing the computational cost significantly. We introduce several simplifications from a model and optimal control perspectives. At first, we notice that the buoyancy effect in the Boussinesq model has a big impact only for large time horizons, therefore in the MPC open-loop problem, one can consider to fix the current velocity field \( \vec{v} \) and solve only the linear convection-diffusion equation (1d).

However, we will still carry out occasional simulations of the Boussinesq approximation in order to generate the source for so-called snapshots of the system state. From those we can then construct a local (with respect to time) reduced-order model of the convection-diffusion equation (1d) system using POD. This reduced model summarizes information about the essential dynamics currently going on in the system and is much smaller than the original system which allows us to use it for fast computation of an approximated solution of optimal control problems within the MPC algorithm.

One may ask, why not performing POD directly for the coupled system, as in [32, 33]. Although this is a viable approach, the use of the velocity field \( \vec{v} \) as datum of the problem and not as unknown improves and simplifies further the POD approximation. This has the double advantage of using less basis functions and easier (mathematically and computationally) a-posteriori error estimator for the problem, which may result in tighter estimates compared to the one for the fully coupled system.

The last challenge is caused by the state constraints. These do not guarantee the existence of regular \( (L^2) \) Lagrange multipliers (cf. e.g. [12]) , introducing additional computational effort. To overcome this issue, we employ a virtual control approach together with a primal-dual active set strategy.
Virtual control approach and PDASS for the linear-quadratic problem

As mentioned in Section 2, as first simplification we assume that the velocity field $v$ is given and we focus on a model based only on the heat equation (1d)-(1e) with convection together with the boundary conditions (1g)-(1h). It is well-known that this equation admits a unique weak solution $y \in Y = W(0,T;V)$ for any given control $u \in U$; cf., e.g., [8]. Let $S : U \rightarrow \mathcal{Y}$ be the control-to-state map for the heat equation, then we define the admissible set of controls

$$U_{ad} = \{ u \in U \mid u \text{ and } y = S(u) \text{ satisfy (2b) and (2c), respectively} \}.$$

We are, thus, interested in solving the strictly convex optimal control problem

$$\min J(u) \quad \text{s.t.} \quad u \in U_{ad}, \quad (\hat{P})$$

which admits a unique solution $\bar{u}$ if the set $U_{ad}$ has non-empty interior; see, e.g., [23].

Note that the difference between $P_{B}$ and $P$ relies on the different control-to-state map, i.e. on the different underlying state model. As said, we expect that $\bar{u}_B$ and $\bar{u}$ will be close for short time horizon $T$, since the buoyancy effect influences the temperature trajectory $y$ only on larger time scales. This will allow computing a good enough approximation of the velocity field $v$ for (1d), to be used as datum of the problem. We will explain in Section 6 how we compute this approximation. What is important to clarify now is how to compute the solution to $(\hat{P})$. As mentioned, the presence of pointwise state constraints leads to non-regular Lagrange multipliers; cf. [7, 35, 36]. To overcome this issue, we apply a virtual control approach as in [26, 27, 28]. Let $w \in W = L^2(Q)$ be an additional (artificial) control. For given $\varepsilon > 0$ we relax the bilateral state constraints (2c) as follows:

$$y_a \leq S(u) + \varepsilon w \leq y_b \quad \text{a.e. in } Q \quad (3)$$

and introduce the artificial constraints

$$w_a \leq w(t,x) \leq w_b \quad \text{a.e. in } Q \quad (4)$$

with fixed (safeguard) scalars $w_a \ll 0$ and $w_b \gg 0$, i.e., sufficiently small and large, respectively. Furthermore, we define the admissible set of controls

$$Z_{ad}^\varepsilon = \{ (u,w) \in U \times W \mid u, w \text{ and } S(u) \text{ satisfy (2b), (3), (4}) \}.$$

Now we are interested in solving

$$\min J(u,w) \quad \text{s.t.} \quad (u,w) \in Z_{ad}^\varepsilon, \quad (\hat{P}^\varepsilon)$$

where

$$J(u,w) = J(u) + \frac{\sigma}{2} \| w \|^2_W$$

with $\sigma > 0$. Note that $\hat{P}$ admits a unique solution $(\bar{u}^\varepsilon, \bar{w}^\varepsilon) \in U \times W$ for each $\varepsilon > 0$ provided $Z_{ad}^\varepsilon$ has non-empty interior (cf. [23, 27]).
Theorem 1. Let $\hat{P}$ and $\bar{V}$ be given. Then the first-order necessary optimality conditions for $(\hat{P})$ as $\varepsilon \to 0$ with an order of convergence $O(\varepsilon)$; cf. [27]. They can be omitted if $\varepsilon$ is taken small but fixed. In this case, the existence of a unique solution is still guaranteed [28]. Moreover, we can choose $w_a$ and $w_b$ as large in absolute value as one desires, because they do not have any physical meaning [27].

Remark 1. We point out that the artificial bounds $w_a$, $w_b$ are necessary to guarantee $\bar{u}^\varepsilon \to \bar{u}$ in $\mathcal{U}$ as $\varepsilon \to 0$ with an order of convergence $O(\sqrt{\varepsilon})$; cf. [27, Section 1.3.3].

Remark 2. Let us define the non-symmetric continuous weakly-coercive bilinear form $a(t; \cdot, \cdot) : V \times V \to \mathbb{R}$ as

$$a(t; \varphi, \psi) = \int_{\Omega} \frac{\partial}{\partial t} \varphi \cdot \nabla \psi + (v(t, \cdot) \cdot \nabla \varphi) \psi \, dx + \gamma \int_{\Gamma_{T}} \varphi \psi \, ds + \gamma_{c} \int_{\Gamma_{c}} \varphi \psi \, ds.$$ 

Then the first-order necessary optimality conditions for $(\hat{P})$ are as follows (cf. [27]):

**Theorem 1.** Let $Z_{ad}^\varepsilon$ have a non-empty interior. Suppose that the pair $(\bar{u}^\varepsilon, \bar{w}^\varepsilon) \in Z_{ad}^\varepsilon$ is the solution to $(\hat{P})$ with associated optimal state $\bar{y}^\varepsilon = S(\bar{w}^\varepsilon)$, i.e. satisfying

$$\frac{d}{dt} \langle \bar{y}^\varepsilon(t) , \varphi \rangle_{H} + a(t; \bar{y}^\varepsilon(t) , \varphi) = \gamma y_{out}(t) \int_{\Gamma_{T}} \varphi \, ds + \gamma_{c} u(t) \int_{\Gamma_{c}} \varphi \, ds, \quad (5a)$$

for all $\varphi \in V$ and a.e. in $[0, T]$, where $H = L^{2}(\Omega)$. Then, there exist unique Lagrange multipliers $\bar{p}^\varepsilon \in \mathcal{Y}$, $\bar{\beta}^\varepsilon$, $\bar{\varphi}^\varepsilon \in \mathcal{W}$ and $\bar{\alpha}^\varepsilon \in \mathcal{U}$ satisfying the dual equation

$$-\frac{d}{dt} \langle \bar{\varphi}^\varepsilon(t) , \varphi \rangle_{H} + a(t; \varphi, \bar{\varphi}^\varepsilon(t)) = -\langle \bar{\beta}^\varepsilon(t) , \varphi \rangle_{H}, \quad (5b)$$

for all $\varphi \in V$ and a.e. in $[0, T]$ and the optimality system

$$\bar{u}^\varepsilon - \gamma_{c} \int_{\Gamma_{c}} \bar{p}^\varepsilon \, ds + \bar{\alpha}^\varepsilon = 0 \quad \text{ in } \mathcal{U}, \quad (5c)$$

$$\sigma \bar{w}^\varepsilon + \varepsilon \bar{\beta}^\varepsilon + \bar{\varphi}^\varepsilon = 0 \quad \text{ in } \mathcal{W}. \quad (5d)$$

Moreover,

$$\bar{\beta}^\varepsilon = \max \left\{ 0, \bar{\beta}^\varepsilon + \eta(\bar{y}^\varepsilon + \varepsilon \bar{w}^\varepsilon - y_{a}) \right\} + \min \left\{ 0, \bar{\beta}^\varepsilon + \eta(\bar{y}^\varepsilon + \varepsilon \bar{w}^\varepsilon - y_{a}) \right\}, \quad (5e)$$

$$\bar{\alpha}^\varepsilon = \max \left\{ 0, \bar{\alpha}^\varepsilon + \eta_{u}(\bar{u}^\varepsilon - u_{b}) \right\} + \min \left\{ 0, \bar{\alpha}^\varepsilon + \eta_{u}(\bar{u}^\varepsilon - u_{b}) \right\}, \quad (5f)$$

$$\bar{\varphi}^\varepsilon = \max \left\{ 0, \bar{\varphi}^\varepsilon + \eta_{w}(\bar{w}^\varepsilon - w_{b}) \right\} + \min \left\{ 0, \bar{\varphi}^\varepsilon + \eta_{w}(\bar{w}^\varepsilon - w_{b}) \right\} \quad (5g)$$

for arbitrarily chosen $\eta, \eta_{u}, \eta_{w} > 0$, where the max- and min-operations are interpreted componentwise in the pointwise everywhere sense.

Remark 2. The well-defined bounded solution operator $A : \mathcal{W} \to \mathcal{Y}$ is defined as follows: for given $\beta \in \mathcal{W}$ the function $p = A\beta$ is the unique solution to

$$-\frac{d}{dt} \langle p(t), \varphi \rangle_{H} + a(t; \varphi, p(t)) = -\langle \beta(t), \varphi \rangle_{H} \quad \forall \varphi \in V \text{ a.e. in } [0, T],$$

$$p(T) = -\beta(T)$$

for given $\beta \in \mathcal{W}$; cf. Remark 2.3 in [28]. Then, $\bar{\varphi}^\varepsilon = A(\bar{\beta}^\varepsilon)$ solves $(5b)$. \hfill \Box
Taking into account (5), conditions (7) lead to the following coupled equations:

\[ \mathcal{A}_a^\nu(\nu) = \{ t \in (0, T) \mid \alpha(\nu) + u - u_a < 0 \text{ a.e.} \}, \]
\[ \mathcal{A}_b^\nu(\nu) = \{ t \in (0, T) \mid \alpha(\nu) + u - u_b > 0 \text{ a.e.} \}, \]
\[ \mathcal{A}_a^W(\nu) = \{ (t, x) \in Q \mid \beta(\nu) + \frac{\sigma}{\varepsilon^2}(y(\nu) + \varepsilon w - y_a) < 0 \text{ a.e.} \}, \]
\[ \mathcal{A}_b^W(\nu) = \{ (t, x) \in Q \mid \beta(\nu) + \frac{\sigma}{\varepsilon^2}(y(\nu) + \varepsilon w - y_b) > 0 \text{ a.e.} \}, \]
\[ \mathcal{A}_{a,2}^W(\nu) = \{ (t, x) \in Q \mid w - w_a < 0 \text{ a.e.} \}, \]
\[ \mathcal{A}_{b,2}^W(\nu) = \{ (t, x) \in Q \mid w - w_b > 0 \text{ a.e.} \}. \]  

Similarly, the associated inactive sets are

\[ \mathcal{J}^u(\nu) = (0, T) \setminus (\mathcal{A}_a^\nu(\nu) \cup \mathcal{A}_b^\nu(\nu)), \]
\[ \mathcal{J}^W(\nu) = Q \setminus (\mathcal{A}_a^W(\nu) \cup \mathcal{A}_b^W(\nu)), \]
\[ \mathcal{J}_{2}^W(\nu) = Q \setminus (\mathcal{A}_{a,2}^W(\nu) \cup \mathcal{A}_{b,2}^W(\nu)). \]  

One can solve (5) applying a primal-dual active set strategy or, equivalently, a semi-smooth Newton method; see, e.g., [21, 25, 27]. Doing so, at each iteration \( k \) of the semi-smooth Newton method we have to solve the following system

\[ \gamma_c \int_{\Gamma_e} p^{k+1} \, ds - u^{k+1} = 0 \quad \text{in } \mathcal{J}^u(\nu^k), \]  
\[ \nu^{k+1} = u_a \quad \text{in } \mathcal{A}_a^\nu(\nu^k), \]  
\[ u^{k+1} = u_b \quad \text{in } \mathcal{A}_b^\nu(\nu^k), \]  
\[ w^{k+1} = 0 \quad \text{in } \mathcal{J}^W(\nu^k), \]  
\[ y^{k+1} + \varepsilon w^{k+1} = y_a \quad \text{in } \mathcal{A}_a^W(\nu^k), \]  
\[ y^{k+1} + \varepsilon w^{k+1} = y_b \quad \text{in } \mathcal{A}_b^W(\nu^k), \]  
\[ \theta^{k+1} = 0 \quad \text{in } \mathcal{J}_{2}^W(\nu^k), \]  
\[ \sigma w^{k+1} - \varepsilon \theta^{k+1} = \sigma w_a \quad \text{in } \mathcal{A}_{a,2}^W(\nu^k), \]  
\[ \sigma w^{k+1} - \varepsilon \theta^{k+1} = \sigma w_b \quad \text{in } \mathcal{A}_{b,2}^W(\nu^k). \]  

Taking into account (5), conditions (7) lead to the following coupled equations:

\[ \frac{d}{dt} \left( y^{k+1}(t), \varphi \right)_H + a(y^{k+1}(t), \varphi) - \gamma_c \mathcal{H}^k(t; p^{k+1}(t)) \int_{\Gamma_e} \varphi \, ds \]
\[ = \gamma y_{\text{out}}(t) \int_{\Gamma \setminus \Gamma_e} \varphi \, ds + \gamma_c r^{k}(t) \int_{\Gamma_e} \varphi \, ds, \]
\[ - \frac{d}{dt} \left( p^{k+1}(t), \varphi \right)_H + a(t; \varphi, p^{k+1}(t)) + \frac{\sigma}{\varepsilon^2} \left( \mathcal{G}^{k,\varepsilon}(t; y^{k+1}(t)), \varphi \right)_H \]
\[ = \frac{\sigma}{\varepsilon^2} \left( r^{k,\varepsilon}(t), \varphi \right)_H. \]
for all $\varphi \in V$ and a.e. in $(0, T)$, together with the conditions

$$y^{k+1}(0) = y_0,$$

$$p^{k+1}(T) + \frac{\sigma}{\varepsilon^2} \mathcal{G}^{k, \varepsilon}(T; y^{k+1}(T)) = \frac{\sigma}{\varepsilon^2} r^{k, \varepsilon}(T),$$

where

$$\mathcal{G}^{k, \varepsilon}(t; y^{k+1}(t)) = y^{k+1}(t) \sum_{i=1}^{6} \chi_{A_i(y^{k+1})}(t) + \frac{1}{\varepsilon} y^{k+1}(t) \sum_{i=3}^{6} \chi_{A_i(y^{k+1})}(t)$$

$$\mathcal{H}^{k}(t; p^{k+1}(t)) = \chi_{\mathcal{M}_t(z^{k})}(t) \gamma_c \int_{\Gamma_{\varepsilon}} p^{k+1}(t) \, ds$$

$$r^{k}(t) = \chi_{A_0^{(\varepsilon)}}(t) u_a(t) + \chi_{A_0^{(\varepsilon)}}(t) u_b(t)$$

$$r^{k, \varepsilon}(t) = y_a \chi_{A_1^{(\varepsilon)}}(t) + y_b \chi_{A_2^{(\varepsilon)}}(t) - w_a \left( \chi_{A_3^{(\varepsilon)}}(t) + \chi_{A_4^{(\varepsilon)}}(t) \right)$$

$$+ \frac{\varepsilon + 1}{\varepsilon} y_a \left( \chi_{A_5^{(\varepsilon)}}(t) + \chi_{A_6^{(\varepsilon)}}(t) \right)$$

$$+ \frac{\varepsilon + 1}{\varepsilon} y_b \left( \chi_{A_5^{(\varepsilon)}}(t) + \chi_{A_6^{(\varepsilon)}}(t) \right),$$

with $\chi_\bullet(t)$ being the indicator functions of the sets

$$\mathcal{A}_1^{\varepsilon}(\nu^k) = \mathcal{A}_a^{\varepsilon}(\nu^k) \cap \mathcal{J}_1^{\varepsilon}(\nu^k),$$

$$\mathcal{A}_2^{\varepsilon}(\nu^k) = \mathcal{A}_a^{\varepsilon}(\nu^k) \cap \mathcal{J}_2^{\varepsilon}(\nu^k),$$

$$\mathcal{A}_3^{\varepsilon}(\nu^k) = \mathcal{A}_a^{\varepsilon}(\nu^k) \cap \mathcal{J}_3^{\varepsilon}(\nu^k),$$

$$\mathcal{A}_4^{\varepsilon}(\nu^k) = \mathcal{A}_a^{\varepsilon}(\nu^k) \cap \mathcal{J}_4^{\varepsilon}(\nu^k),$$

$$\mathcal{A}_5^{\varepsilon}(\nu^k) = \mathcal{A}_b^{\varepsilon}(\nu^k) \cap \mathcal{J}_5^{\varepsilon}(\nu^k),$$

$$\mathcal{A}_6^{\varepsilon}(\nu^k) = \mathcal{A}_b^{\varepsilon}(\nu^k) \cap \mathcal{J}_6^{\varepsilon}(\nu^k),$$

respectively. Summarizing, the coupled system of equations [8] can be formulated only in the variables $y^{k+1}$ and $p^{k+1}$:

$$\begin{pmatrix} A_{11}^{k} & A_{12}^{k} \\ A_{21}^{k} & A_{22}^{k} \end{pmatrix} \begin{pmatrix} y^{k+1} \\ p^{k+1} \end{pmatrix} = \begin{pmatrix} Q_{1}^{k}(\nu^k; u_a, u_b, \gamma_c, \gamma, y_{\text{out}}) \\ Q_{2}^{k}(\nu^k; y_a, y_b, u_a, u_b, \varepsilon, \sigma) \end{pmatrix},$$

where the operators $A_{11}^{k}$ and $A_{22}^{k}$ are of the form

$$A_{11}^{k} = C + \tilde{A}_{11}^{k}, \quad A_{22}^{k} = C^* + \tilde{A}_{22}^{k},$$

and $C$ stands for the heat-convection operator, which does not depend on $k$. A discretization of [9] leads to a discretized system of the form

$$\begin{pmatrix} A_{11}^{k} & A_{12}^{k} \\ A_{21}^{k} & A_{22}^{k} \end{pmatrix} \begin{pmatrix} y^{k+1} \\ p^{k+1} \end{pmatrix} = \begin{pmatrix} Q_{1}^{k} \\ Q_{2}^{k} \end{pmatrix}$$

with

$$A_{11}^{k} = C + \tilde{A}_{11}^{k}, \quad A_{22}^{k} = C^* + \tilde{A}_{22}^{k},$$

where $C$ stands for the discretized heat-convection operator, which is again independent of $k$. For further details, we refer to [27]. We summarize the PDASS in Algorithm [1].
Algorithm 1 (Primal-dual active set strategy)

1: Choose starting value \( \nu^0 = (u^0, w^0, \theta^0) \in \mathbb{N} \), set \( k = 0 \) and \( \text{flag} = \text{false} \);
2: Determine \( y^0 = S u^0 \) and \( p^0 = -A(\sigma w^0 + \theta^0)/\varepsilon \).
3: Determine \( A^U_a(\nu^0), A^U_b(\nu^0), g^U(\nu^0) \) for \( i = 1, \ldots, m \), \( A^W_a(\nu^0), A^W_b(\nu^0), \}^W(\nu^0) \) and \( A^W_{a,2}(\nu^0), A^W_{b,2}(\nu^0) \) and \( \}^W_{2}(\nu^0) \) from (8);
4: repeat
5: Compute the solution \( (y^{k+1}, p^{k+1}) \) by solving (10);
6: Compute \( \nu^{k+1} = (u^{k+1}, w^{k+1}, \theta^{k+1}) \in \mathbb{N} \) from (7) and set \( k = k + 1 \);
7: Determine \( A^U_a(\nu^k), A^U_b(\nu^k), g^U(\nu^k) \) for \( i = 1, \ldots, m \), \( A^W_a(\nu^k), A^W_b(\nu^k), \}^W(\nu^k) \) and \( A^W_{a,2}(\nu^k), A^W_{b,2}(\nu^k) \) and \( \}^W_{2}(\nu^k) \) from (9);
8: if \( A^U_a(\nu^k) = A^U_a(\nu^{k-1}) \) and \( A^U_b(\nu^k) = A^U_b(\nu^{k-1}) \) for \( i = 1, \ldots, m \) then
9: if \( A^W_a(\nu^k) = A^W_a(\nu^{k-1}) \) and \( A^W_b(\nu^k) = A^W_b(\nu^{k-1}) \) then
10: if \( A^W_{a,2}(\nu^k) = A^W_{a,2}(\nu^{k-1}) \) and \( A^W_{b,2}(\nu^k) = A^W_{b,2}(\nu^{k-1}) \) then
11: Set \( \text{flag} = \text{true} \);
12: end if
13: end if
14: end if
15: until \( \text{flag} = \text{true} \);

Remark 3. The discrete linear system (10) can be obtained discretizing the primal and dual equations (8), for example, with piecewise linear finite elements (FE) in space and the implicit Euler scheme in time. Let \( V_h \subset V \) be the FE space with \( \dim V_h = N_x \) and \( N_t \) be the number of time steps. Notice that the matrix in (10) has the size \((2N_tN_x) \times (2N_tN_x)\). Clearly, its dimension may cause problems regarding memory consumption and computational time. In our application, in fact, \( N_x \) is not small, since a coarse FE grid can not capture the complexity of the dynamics involved. Therefore, we apply POD-based reduced-order modeling to approximate the solution of (10) by a reduced-order system of dimension \( 2\ell N_t \) with \( \ell \ll N_x \), gaining computational time but paying in approximation; cf. Section 4. The number of time steps may not be small either, since the time horizon \( T \) could be extremely large (e.g. one month) and the time step small (e.g. a minute). This is not a disadvantage for us, since because we apply an MPC method, we iteratively solve optimal control subproblems on a smaller time horizon, computing a feedback control that approximates the optimal control \( \bar{u}^* \); cf. Section 5. Finally, note that the matrix in (10) has a sparse block structure that allows to store only the FE matrices, e.g., the mass and stiffness matrices, which have dimension \( N_x \times N_x \). These blocks will then be projected into the POD space. Doing that we can save memory significantly, in particular when the number of FE nodes is large. Note that, in comparison to the Boussinesq approximation, we gain even more, since in general computing the solution of this model requires an additional piecewise quadratic FE approximation of the velocity space, which would definitely become infeasible in combination with the PDASS.

10
4 POD and a-posteriori error estimate

Model order reduction and, in particular, POD is a well established field of numerical analysis, which was intensively developed in the past 20 years. The key idea is to construct a low-dimensional subspace \( V_\ell \subset V_h = \text{span} \{ \varphi_1, \ldots, \varphi_{N_x} \} \), spanned by the so-called POD basis functions. For the sake of completeness, in the first part of this section, we describe briefly the POD method in the discrete case, but we focus on one snapshot ensemble for brevity. For more details we refer the reader to [20], for instance.

Let \( t_j = j \Delta t, \ j = 0, \ldots, n, \) be the time discretization for a suitable time interval with constant time step \( \Delta t > 0 \). By \( \{ y_j \}_{j=0}^n \subset V_h \) we denote a given discrete trajectory, also called snapshots ensemble. We define the snapshot subspace \( V_n = \text{span} \{ y_0, \ldots, y_n \} \subset V_h \). Let \( d_n = \text{dim} V_n \leq n + 1 \). For any \( \ell \in \{ 1, \ldots, d_n \} \) we are interested in finding an orthonormal set \( \{ \psi_i \}_{i=1}^\ell \subset V_n \), which is the solution of the following minimization problem

\[
\min \sum_{j=0}^n \alpha_j \left\| y_j - \sum_{i=1}^\ell \langle y_j, \psi_i \rangle_V \psi_i \right\|_V^2 \quad \text{s.t.} \quad \langle \psi_i, \psi_j \rangle_V = \delta_{ij},
\]

where \( \delta_{ij} \) is the Kronecker delta and \( \alpha_j > 0 \) are given (trapezoidal) weights. We set \( V_\ell = \text{span} \{ \psi_1, \ldots, \psi_\ell \} \subset V_h \). Obviously, if \( \text{dim} V_\ell = \ell \ll m \) holds we gain a computational speed-up. From \( y_j \in V_h \ (0 \leq j \leq n) \) we infer that there exists a snapshot matrix \( Y \in \mathbb{R}^{N_x \times (n+1)} \), such that

\[
y_j(x) = \sum_{i=1}^{N_x} Y_{ij} \varphi_i(x) \quad \text{for} \ x \in \Omega.
\]

Introducing the weighting matrices \( D = \text{diag} (\alpha_0, \ldots, \alpha_n) \in \mathbb{R}^{(n+1) \times (n+1)} \) and \( W = ((W_{ij})) \in \mathbb{R}^{N_x \times N_x} \) with \( W_{ij} = \langle \varphi_j, \varphi_i \rangle_V \), a solution \( \{ \psi_j \}_{j=1}^\ell \) to (11) can be computed as follows:

- Solve the symmetric eigenvalue problem
  \[
  (D^{1/2}Y^TWYD^{1/2})\Phi_j = \lambda_j \Phi_j \quad \text{for} \ j = 1, \ldots, \ell
  \]
  with \( \lambda_1 \geq \ldots \geq \lambda_\ell > 0 \).

- Define the vectors \( \psi_j = YD^{1/2}\Phi_j/\sqrt{\lambda_j} \in \mathbb{R}^{N_x} \) for \( 1 \leq j \leq \ell \) and the matrix \( \Psi = [\psi_1|\ldots|\psi_\ell] \in \mathbb{R}^{N_x \times \ell} \).

- Set
  \[
  \psi_j(x) = \sum_{i=1}^{N_x} \Psi_{ij} \varphi_i(x) \quad \text{for} \ j = 1, \ldots, \ell \text{ and } x \in \Omega.
  \]

Using a Galerkin ansatz, the reduced-order version of (8) can be derived replacing the test functions \( \varphi \) by the POD basis \( \{ \psi_i \}_{i=1}^\ell \). We indicate with \( S^\ell \) the POD solution operator corresponding to the reduced-order model. We can then construct the discretized
system

\[
\begin{pmatrix}
A_{11}^{k,\ell} & A_{12}^{k,\ell} \\
A_{21}^{k,\ell} & A_{22}^{k,\ell}
\end{pmatrix}
\begin{pmatrix}
y^{k+1,\ell} \\
p^{k+1,\ell}
\end{pmatrix}
= \begin{pmatrix}
Q_1^{k,\ell} \\
Q_2^{k,\ell}
\end{pmatrix},
\] (12)

of which the matrix belongs to \(\mathbb{R}^{(2\ell N_t) \times (2\ell N_t)}\). From [12], one can easily implement a POD-based PDASS following the structure of Algorithm 1. Now, a first question is how to check that the chosen number of POD basis \(\ell\) is good enough to guarantee a small approximation error in reconstructing the snapshots. For this, the following error formula holds

\[
\sum_{j=0}^{n} \alpha_j \| y_j - \sum_{i=1}^{\ell} \langle y_j, \psi_i \rangle V \psi_i \|_V^2 \leq \sum_{i=\ell+1}^{d_n} \lambda_i.
\] (13)

More importantly, to check that the constructed POD basis is able to reconstruct the full-order optimal solution \(\bar{u}^\varepsilon\), we need an a-posteriori error estimator. Let \(\bar{u}^\varepsilon, \bar{w}^\varepsilon\) be the solution of the POD-based PDASS, then the following result from [27] holds:

**Proposition 1.** Let \((\bar{u}^\varepsilon, \bar{w}^\varepsilon)\) be the optimal solution to (P\(^e\)). Suppose that \(z^{ap} = (u^{ap}, w^{ap}) \in Z^{e}_{ad}\) is given arbitrarily. Then, there exists a perturbation \(\zeta = (\zeta^u, \zeta^w) \in \mathcal{U} \times \mathcal{W}\), which is independent of \((\bar{u}^\varepsilon, \bar{w}^\varepsilon)\), so that

\[
\| \bar{u}^\varepsilon - u^{ap} \|_\mathcal{U} + \| \bar{w}^\varepsilon - w^{ap} \|_\mathcal{W} \leq \frac{1}{\sigma^{ap}} \| T^* \zeta \|_\mathcal{U},
\] (14)

where \(\sigma^{ap} := \min\{\sigma, 1\} > 0\) and \(T^*\) is defined as

\[
T^* = \begin{pmatrix}
I_\mathcal{U} & S^* \\
0 & I_\mathcal{W}
\end{pmatrix} : \mathcal{U} \times \mathcal{W} \to \mathcal{U} \times \mathcal{W},
\]

where \(I_\mathcal{U}\) and \(I_\mathcal{W}\) are the identities on \(\mathcal{U}\) and \(\mathcal{W}\), respectively. The perturbation \(\zeta\) is computed as follows: Let \(\xi = (\xi^u, \xi^w) \in \mathcal{U} \times \mathcal{W}\) be given as the solution of the linear system \(T^* \xi = \nabla \hat{J}(z^{ap})\), i.e.,

\[
\begin{pmatrix}
I_\mathcal{U} & S^* \\
0 & I_\mathcal{W}
\end{pmatrix}
\begin{pmatrix}
\xi^u \\
\xi^w
\end{pmatrix}
= \begin{pmatrix}
u^{ap} \\
\sigma w^{ap}
\end{pmatrix},
\] (15)

then

\[
\zeta^u(t) = \begin{cases}
-\min\{0, \xi^u(t)\} & \text{for } t \in A^U_2(z^{ap}), \\
-\max\{0, \xi^u(t)\} & \text{for } t \in A^U_1(z^{ap}), \\
-\xi^u(t) & \text{for } t \in I^U(z^{ap})
\end{cases}
\] (16a)

and

\[
\zeta^w(t, x) = \begin{cases}
-\min\{0, \xi^w(t, x)\} & \text{for } (t, x) \in A^W_2(z^{ap}), \\
-\max\{0, \xi^w(t, x)\} & \text{for } (t, x) \in A^W_1(z^{ap}), \\
-\xi^w(t, x) & \text{for } (t, x) \in I^W(z^{ap})
\end{cases}
\] (16b)
Remark 4. Note that we can easily decouple the equations in system (15) by computing \( \xi^w = \sigma w^{ap} / \epsilon \) from the second equation. Then \( \xi^u = -S^* \xi^w + u^{ap} = -\sigma S^* w^{ap} / \epsilon + u^{ap} \). Moreover, we recall that \( S^* : W_{\epsilon}^0(0,T)^' \rightarrow U \) denotes the dual operator of the linear solution operator \( S \); see [41, Lemma 2.4]. Finally, we remark that in our numerical realization \( z^{ap} \) is given by the POD suboptimal solution pair \( (\bar{u}^{\epsilon,\ell}, \bar{w}^{\epsilon,\ell}) \). Thus, (14) can be utilized as an a-posteriori error estimate. For further details regarding the a-posteriori error estimator, we refer to [27].

5 Economic MPC

MPC is a well-established method for computing a closed-loop control for a dynamical system on an infinite (or large) time horizon. MPC splits the solution of such problems into the consecutive solution of problems on a small finite time horizon, the so called prediction horizon. The goal is to reduce the complexity of the problem in time, while at the same time being able to react to model uncertainties, disturbances, and possible parameter changes. We are interested in economic MPC [4, 11]. This term comprises all MPC schemes in which the cost function does not merely penalize the distance to a pre-defined target. Here we do not have such a pre-defined target. Rather, the goal is to minimize the control effort, while satisfying certain control and state constraints; see, e.g., [17, 19, 31]. Applying MPC to the virtual control problem (\( \hat{P}^{\epsilon} \)) has several advantages, which will be more clear after having introduced the method. Let \( N \Delta t, N \in \mathbb{N}, \) be the prediction horizon. We define the Hilbert spaces

\[
U^j_N = L^2(t_j, t_{j+N}), \quad W^j_N = L^2((t_j, t_{j+N}) \times \Omega),
\]

for \( j = 0, \ldots, N_t \) and with the notation \( t_j = j \Delta t \) for \( j > N_t \). Then the MPC cost functional is

\[
J^j_N(u, w) = \frac{1}{2} \| u \|^2_{U^j_N} + \frac{\sigma}{2} \| w \|^2_{W^j_N},
\]

and the MPC admissible sets are

\[
Z^{\epsilon,j}_{N,ad} = \{(u, w) \in U^j_N \times W^j_N : u, w, Su + \epsilon w \text{ satisfy}
\]

\[
\begin{align*}
& u_a(t) \leq u(t) \leq u_b(t) \text{ a.e. in } (t_j, t_{j+N}], \\
& w_a \leq w(t, x) \leq w_b, y_a \leq Su + \epsilon w \leq y_b \text{ a.e. in } (t_j, t_{j+N}] \times \Omega, \\
& \text{respectively}
\end{align*}
\]

for \( j = 0, \ldots, N_t \). Furthermore, note that the fully discretized version of (5a) can be rewritten as the following discrete dynamical system

\[
y_{j+1} = f(j, y_j, u_j) \text{ for } j = 0, \ldots, N_t, \quad y_0 = y_0,
\]

where the function \( f \) comprises all the formulas that come from the discretization and the model, which we avoid to write for brevity. The MPC method is then summarized in Algorithm 2. To guarantee well-posedness and existence of the optimal control to each
Algorithm 2 Model predictive control

1: Choose an initial state $y_0$, an MPC prediction horizon $N$ and the regularization parameter $\varepsilon > 0$;
2: for each time instant $j = 0, 1, 2, \ldots, N_t$ do
3: Measure the current state of the system $y^j$ and the model parameters;
4: Solve the optimal control problem
\[
\min J_N^j(u, w) \quad \text{s.t.} \quad (u, w) \in Z_{N, \text{ad}}^{\varepsilon, j}
\]
\hspace{1cm} to obtain the open-loop optimal control sequence $\tilde{u}_N^j$;
5: Apply the first element of $\tilde{u}_N^j$ as a control to the system (17) during the next sampling period, i.e. use the feedback law $\mu_N(y_j) = \tilde{u}_N^j(0)$.
6: end for

Algorithm 2 produces a sequence of optimal controls in discrete time and uses the first element of each optimal control sequence as a feedback control value. This results in a closed-loop sub-optimal trajectory, which solves (17) for $u_j = \mu_N(y_j)$, $j = 0, 1, \ldots, N_t$. Note that in step 3 of the MPC algorithm it is possible to measure (and update) the parameters of the model. This is particularly advantageous for our application, since usually we do not have, e.g., a precise weather forecast for the entire long-time horizon $T$. In this way, we can update the outside temperature while the algorithm runs. To determine $y^j$ typically an estimator is needed to process available sensor data properly to reconstruct the spatial-temporal distribution. In addition, we can take advantage for the velocity field $v$. We mentioned already that this will be provided from an approximation of the Boussinesq velocity fields. We thus gain the possibility of updating this approximation while the simulation runs, improving the accuracy of our simplified model.

The optimal control problems (18) are solved on a much smaller time horizon $(t_j, t_{j+N})$ since $N \ll N_t$. This guarantees a good computational speed-up. For general optimal control problems it is not necessarily the case that the MPC closed-loop trajectory is approximately optimal on the long time horizon $N_t$. However, in recent years structural conditions were discovered that allow to conclude this property for $N$ large enough; cf. [16, Chapter 7] and [11, 19]. The most important among these conditions are the turnpike property and the closely related strict dissipativity property [39, 14]. The turnpike property can be seen as a similarity condition for optimal trajectories on different time horizons and we will check numerically that this property holds for the optimal control problem considered in this paper; cf. Section 7.1. It is also possible to check the quality of the MPC horizon and trajectory with a-posteriori error estimators, such as in [13, 15], but we do not utilize them in this work.

What remains to be explained is how to successfully combine MPC and POD. The challenge is to guarantee a small approximation error for the reduced-order model while
Algorithm 3 MPC-POD

1: Choose an initial state $y_0$, an MPC prediction horizon $N$, the regularization parameter $\varepsilon > 0$, and tolerances $\tau_1$ for the a-priori estimate \[13\] and $\tau_2$ for the a-posteriori estimate;
2: Set $t_0 = 0$, $y_0(0) = y_0$, and $\text{flag} = \text{true}$;
3: for $j = 0, 1, 2, \ldots, N$ do
4: Measure the current state $y_j$ of the system at time $t_j$;
5: if $\text{flag} = \text{true}$ then
6: Update the model parameters;
7: Compute primal and dual snapshots;
8: Compute a POD basis $\{\psi_i\}_{i=1}^\ell$ of rank $\ell$ according to $\tau_1$;
9: Set $\text{flag} = \text{false}$;
10: end if
11: Solve the MPC-POD optimal control problem

$$\min J_N^j(u, w) \text{ s.t. } (u, w) \in Z_{N, \text{ad}}^{\varepsilon, \ell, j}$$

to obtain the open-loop suboptimal control sequence $\tilde{u}_N^{\varepsilon, \ell}$;
12: Apply the first element of $\tilde{u}_N^{\varepsilon, \ell}$ as a control to the system \[17\] during the next sampling period, i.e. use the feedback law $\mu_N(y_j) = \tilde{u}_N^{\varepsilon, \ell}(0)$;
13: Compute the a-posteriori error estimate $e = \|T^*\zeta\|_{U \times W}$ from \[14\];
14: if $e > \tau_2$ then
15: Set $\text{flag} = \text{true}$;
16: end if
17: end for

The MPC algorithm proceeds. This is clearly a big challenge if one thinks about the fact that the problem parameters are not only time-variant, but that the parameter values at a certain time instant in the future may also change when new measurement information becomes available. When this happens, it means that we need to update the POD model. Luckily, the a-posteriori error estimate from Proposition \[1\] is a useful tool to check whether this should be done or not. We report the MPC-POD algorithm in Algorithm 3. For further details, we refer to \[27, 29\]. The admissible sets in \[19\] are defined as

$$Z_{N, \text{ad}}^{\varepsilon, \ell, j} = \{(u, w) \in U_N^j \times W_N^j : u, w, S^\ell u + \varepsilon w \text{ satisfy}\}
\begin{align*}
&u_a(t) \leq u(t) \leq u_b(t) \text{ a.e. in } (t_j, t_j + N], \\
&w_a \leq w(t, x) \leq w_b, \ y_a \leq S^\ell u + \varepsilon w \leq y_b \text{ a.e. in } (t_j, t_j + N] \times \Omega,
\end{align*}$$

respectively,

where we recall that $S^\ell$ is the POD solution operator for the reduced-order state equation. Obviously, to guarantee existence and uniqueness of the suboptimal controls solution to \[19\], we assume that the sets $Z_{N, \text{ad}}^{\varepsilon, \ell, j}$ have non-empty interior for all $j = 0, \ldots, N$. 
The only aspect to be explained is how to compute the snapshots in Algorithm 3. We will explain this in the next section. Let us mention that other possibilities are explored in [10, 12, 27, 29].

6 Computational details and additional parallelization issues

In this section, we clarify some aspects of our simplification strategy to compute an approximate solution to \( \hat{P}_B \). We give details about the implementation of the algorithm, mentioning which parts can be performed in parallel.

Looking at Algorithm 3, there are still some unclear aspects. At first, it is not clear how to compute the POD snapshots in Step 7. At iteration \( j = 0 \), we initialize our model by solving the Boussinesq approximation (1) with a given initial control \( u_0 \) for the time horizon \([0, T + N\Delta t]\) using the FE method for the spatial and the implicit Euler method for the temporal discretization. The interval \([0, T + N\Delta t]\) is discretized by \( t_j = j\Delta t \), \( j = 0, \ldots, m \) with \( m = N_t + N \). Exploiting the power of parallel computing we obtain, therefore, the solution sequence \( \{(v_{j,h}^0, y_{j,h}^0)\}_{m,j=0}^m \). Then, we compute the sequence \( \{p_{j,h}^0\}_{m,j=0}^m \) by solving in parallel

\[
\frac{1}{\Delta t} \langle p_j^0 - p_{j+1}^0, \varphi_h \rangle_H + a(t_j; \varphi_h, p_j^0) + \frac{\sigma}{\varepsilon^2} \langle G^{0,\varepsilon}(t_j; y_j^0), \varphi_h \rangle_H
\]

for all \( \varphi_h \in V_h \) and \( j = m - 1, \ldots, 1 \),

\[
\langle p_{m,h}^0 + \frac{\sigma}{\varepsilon^2} G^{0,\varepsilon}(t_m; y_m^0), \varphi_h \rangle_H = \frac{\sigma}{\varepsilon^2} \langle r^{0,\varepsilon}(t_m), \varphi_h \rangle_H
\]

for all \( \varphi_h \in V_h \) (compare (8b) and (8d)), where the terms \( G^{0,\varepsilon} \) and \( r^{0,\varepsilon} \) involve the computation of initial active and inactive sets (6a)-(6b), which can be also performed in parallel. The snapshots \( \{y_{j,h}^0\}_{j=0}^m \) and \( \{p_{j,h}^0\}_{j=0}^m \) are used to compute the POD basis in step 8. Note that the discrete POD eigenvalue problem has dimension \( 2m \), which may be large. We discuss later about this issue. For subsequent basis updates at an MPC step \( j \), triggered by the a-posteriori error estimator, we will compute our snapshots with the same procedure, but using the control \( \tilde{u}_{\varepsilon,\ell}^N \) computed at the previous iteration \( j - 1 \). Note that we get also an update for the velocity field \( v \).

We use the velocity field \( v \) (computed at step 7) as fixed one for the open-loop optimal control problem (19), solved with POD-PDASS. In such a way, every time the basis and the velocity field are updated, the MPC can correct its approximation. We remark that the POD-PDASS matrix for problem (19) has the size \( (2N\ell) \times (2N\ell) \) (and is thus small), therefore there is no need of parallelization for solving the linear system (12). We anyway compute the active and inactive sets (6a)-(6b) in parallel, since they depend on \( N_x \) and not on \( \ell \).

With the current information, one can already run Algorithm 3. In addition, we use a couple of improvements, which we describe further. A first disadvantage of the POD
basis updates is that any time we need to compute the snapshots, we need to solve the Boussinesq equation for \( t \in [t_j, T + N \Delta t] \). Although the measure of this time domain decreases for increasing \( j \), computing the solution of the Boussinesq approximation (1) for this large time horizon can be computationally demanding even using parallel computing. This computational effort can be reduced, considering to generate snapshots in a smaller time interval \([t_j, t_j + (N + M) \Delta t]\), where \( M > 0 \) is chosen to be small. In such a way, we have an updated velocity field \( \mathbf{v} \) (and snapshots) for the subsequent \( M \) iterations of the algorithm. If a POD basis update is performed in one of these steps, we have saved computational time. If not, we need to solve (1) for the remaining time horizon up to the end (or for other \( M \) steps, for example). Having only \( 2(N + M) \) snapshots is also beneficial for computing the POD basis, since we have to solve a smaller eigenvalue problem.

A second disadvantage of the POD basis update is that overwriting the past information while computing a new basis may not be always a good choice. The old snapshots, in fact, may still carry useful information for the future computations. Yet, adding only snapshots over snapshots will increase the number of POD basis needed to achieve the a-priori tolerance \( \tau_1 \). We employ, therefore, the snapshots selection strategy proposed in [27]. We report it in Algorithm 4. This strategy consists in keeping only those old snapshots that represent dynamics close to the new ones. This procedure is motivated by the fact that the MPC algorithm gives a forecast of the future dynamics (new computed snapshots). Therefore, it makes sense to use them as starting point and add eventually previously computed information, if this describes similar trajectories. Furthermore, in

\begin{algorithm}
\caption{Snapshots selection from [27].}
\begin{algorithmic}[1]
\Require Snapshots previously computed and stored in a list \( L \) and tolerances \( 0 < \rho < \varrho \ll 1 \).
\State Compute the new \( 2(N + M) \) snapshots and store them in a list \( S \);
\For {\( i \leq \text{length}(L) \)}
  \For {\( j \leq 2(N + M) \)}
    \If {\( (1 - \varrho) \| S[j] \| \| L[i] \| \leq |\langle S[j], L[i] \rangle| \leq (1 - \rho) \| S[j] \| \| L[i] \| \)}
      \State Add \( L[i] \) to \( S \);
      \State \textbf{break}
    \EndIf
  \EndFor
\EndFor
\end{algorithmic}
\end{algorithm}

step [12] of Algorithm 3 one can solve in parallel one time step of the Boussinesq model (1) applying the feedback law, instead of solving (5a). In such a way, we further improve the MPC approximation. This operation is not costly if one considers that it is done only for one time-step. At last, note that the computation of the a-posteriori error estimate (14) requires full-order solves of state and adjoint equations. This can also be carried out in parallel. To conclude, we resume every step of Algorithm 3 which is performed in parallel.
• Computation of POD snapshots: this involves solving the Boussinesq approximation (1) and the adjoint equation (8b);
• Computation of the active and inactive sets (6a)-(6b) of the POD-PDASS;
• Computation of the initial guess for the next MPC iteration $j$: this involves solving one time step of (1);
• The POD a-posteriori error estimator (14).

7 Numerical tests

The numerical tests were performed on the compute server of the Chair of Applied Mathematics at the University of Bayreuth, a computer with 32 physical cores (64 logical) Intel(R) Xeon(R) CPU E7-2830 @ 2.13GHz and a total RAM of 512GB. For the numerical realization of the Boussinesq approximation, we refer to [5].

7.1 Numerical verification of the turnpike property

In this subsection we check numerically that the turnpike property holds for (18), because this is the most important ingredient for proving that MPC provides nearly optimal solutions, see [19]. Before stating it in a mathematical formalism, we need to introduce the concept of optimal operation for an infinite horizon optimal control problem. Consider the discrete-time time-varying control system (17). In this subsection, we denote with $y_u(\cdot; t_j, \tilde{y}_0)$ the solution of (17) for a given control $u \in U_N$ starting in $\tilde{y}_0 \in H$ at time $t_j$. Considering the stage cost function $l : \mathbb{R} \times H \to \mathbb{R}$, $l(u, w) = (|u|^2 + \sigma \|w\|_H^2)/2$, the cost functional $J_N^j$ can be written as

$$J_N^j(u, w) = \sum_{i=0}^N l(u(t_{j+i}), w(t_{j+i})).$$

We recall that the finite horizon optimal control problem (18) is

$$\min_{(u, w) \in Z_{N,ad}^j} J_N^j(u, w)$$

with (locally) optimal solution $(\tilde{u}_N^j, \tilde{w}_N^j) \in Z_{N,ad}^j$. By $Z^j_{\infty,ad}$, $J^j_\infty$, $\tilde{u}_\infty$ and $\tilde{w}_\infty$ we denote the extension of the above quantities to infinite time horizon problem. Note that $Z^j_{N,ad}$ changes if the starting value $\tilde{y}_0 \in H$ of the PDE is different. Therefore, in what follows we indicate this dependency with $Z^j_{N,ad}(\tilde{y}_0)$.

**Definition 1.** Let $\tilde{y}_0 \in H$ be an arbitrary feasible starting value at time $t_j$. Consider a control sequence $(\bar{u}, \bar{w}) \in Z^j_{\infty,ad}(\tilde{y}_0)$ with corresponding trajectory $\bar{y}$. We say that the system (17) is optimally operated at $(\bar{y}, \bar{u}, \bar{w})$ if

$$\liminf_{I \to \infty} \sum_{i=0}^{I-1} l(u(t_{j+i}), w(t_{j+i})) - l(\bar{u}(t_{j+i}), \bar{w}(t_{j+i})) \geq 0$$

18
for all feasible $\hat{y} \in H$ and all $(u, w) \in Z_{\epsilon,j}^\infty(\hat{y})$. The trajectory $\bar{y}$ is called optimal operation trajectory.

We remark that there may exist more than one solution to (17) that satisfies Definition 1. Moreover, it is well-known that, in general, it is not easy to compute an optimal operation trajectory explicitly. Still, one can expect that the MPC trajectory converges to this trajectory for sufficiently large MPC horizon $N$. To ensure this behavior one has to assume structural conditions such as the turnpike property. This property demands that the solution to the finite (and infinite) time horizon optimal control problem is close for most of the time to an optimal operation trajectory $\bar{y}$, according to the following definition.

**Definition 2.** Consider $(\bar{y}, \bar{u}, \bar{w})$ at which system (17) is optimally operated. The finite horizon optimal control problem (18) has the turnpike property at $(\bar{y}, \bar{u}, \bar{w})$ if there exist $\eta \in L^\eta$ such that for each $j \in \mathbb{N}_0$, each feasible initial guess $y_j$, each optimal trajectory $y_{\eta_N}(\cdot; t_j, y_j)$ and all $N, P \in \mathbb{N}$ there is a set $Q(t_j, y_j, P, N) \subseteq \{0, \ldots, N\}$ with at most $P$ elements and

$$\|y_{\eta_N}(t_{j+M}; t_j, y_j) - \bar{y}(t_{j+M})\|_H + |\bar{u}_N(t_{j+M}) - \bar{u}(t_{j+M})|$$

$$+ \|\bar{w}_N(t_{j+M}) - \bar{w}(t_{j+M})\|_H \leq \eta(P)$$

for all $M \in \{0, \ldots, N\} \setminus Q(t_j, y_j, P, N)$.

The definition of the turnpike property seems technically elaborated but it is easy to illustrate graphically; cf. Figure 2. This property demands that for each optimal open-loop trajectory there can only be a finite number $P$ of time steps, $P$ independent

\[ L = \{\eta : \mathbb{R}_0^+ \to \mathbb{R}_0^+ | \eta \text{ is continuous and strictly decreasing with } \lim_{s \to \infty} \eta(s) = 0\} \]
from $N$, at which the trajectory is far from the optimal operation trajectory. Moreover, it permits to define a bound on the distance for the remaining close points. Contrary to the computation of the optimal operation trajectory, it is easy to find numerical evidence that a given optimal control problem has the turnpike property [18]. For this test, we choose $\Omega = (0, 1)^2$ discretized with a structured quadrilateral mesh of $N_x = 1769$ nodes, $\Gamma_c = (0, 1) \times \{0\}$, $\rho = C_p = \alpha = 1.0$, $\nu = 0.1$, $\kappa = 0.025$ and $g = (0, -9.8)$. Moreover, we consider $\gamma = 0.1$ and $\gamma_c = 10^5$ for the boundary conditions, so that the action of the control on the boundary can be seen (numerically) as the one of a Dirichlet boundary condition. We solve at first (1) with fixed control $u(t) = 22.5$, outside temperature $y_{out} = \max(13, 16-t)$ and initial guess $(v_0, y_0)(t, x) = (0, 20)$ for all $x \in \Omega$ and $t \in [0, T]$, in order to generate the advection field $v$ to use in (18); cf. Figure 3. Then, we solve (18) with $u_a(t) = 0$, $u_b(t) = 10^6$, $w_b = -w_a = 10^3$, $\varepsilon = 0.025$, $\sigma = 1.0$, $y_a(t) = 17.5 + \min(t, 2)$ and $y_b(t) = 23$ for all $t \in [0, T]$. We use a time step $\Delta t = 0.01$ and compute the open-loop problem solution for $j = 0$ (i.e. $t_0 = 0$) for different horizons $N$, thus $T = N\Delta t$ in each test. As shown in Figure 4(a) the $L^2$ norm of the solution to (18) is the same for a larger number of time steps as $N$ increases. Moreover, the solutions show a similar behavior. This provides clear evidence for the turnpike behavior. This evidence is further
strengthened by the results from Figure 4(b), where the open-loop trajectories starting from different initial guess are converging to the same region as the time passes. For more information about how to check the conditions for near optimal performance of MPC numerically, we refer to [18].

7.2 MPC-POD algorithm for the optimal control problem

In this section we test Algorithm 3 with the same data of Section 7.1, choosing an MPC horizon $N = 300$. This horizon is a good compromise between computational time and turnpike property; cf. Figure 4(a). We fix also $M = 60$, the a-priori tolerance $\tau_1 = 10^{-8} \sum_{i=1}^\ell \lambda_i$ and the a posteriori error estimator tolerance $\tau_2 = 3.5$. Although $\tau_2$ seems large, one has to consider that the control will have a value close to the state constraints range, roughly between 20 and 30, due to the fact that the control boundary condition is (numerically) close to a Dirichlet one. Moreover, the MPC horizon $N = 300$ corresponds to a time horizon for the open-loop problem equal to three. Therefore, one can expect an average $L^2$ norm for the control of $43 (= 25\sqrt{3})$. Hence, fixing $\tau_2 = 3.5$ equals asking a relative error approximately of 8%. We expect, moreover, an overestimation factor for the a-posteriori error estimator, since $(\sigma_{ap})^{-1} \|T^* \zeta\|_{U}$ in (14) gives an upper-bound also for the error in reconstructing the artificial control, which might attain significantly greater values according to the chosen $\varepsilon$. In addition, a too small $\tau_2$ will trigger the basis updates too often and thus worsen the performance of the algorithm.

Figure 5(a) shows the minimum, maximum and average value of the optimal MPC solution together with the desired bounds. One can notice that the average temperature is kept inside the bounds, where instead the minimum and the maximum one exceed them. This is typical of the $L^2$ regularization effect of the virtual control approach and
can be mitigated taking a small $\varepsilon$. This effect is accentuated also by the POD method, which provides a sub-optimal solution as well-known from the literature \[32, 41\]. As a result, the average number of active points at each time step amounts to 3.5\% of the total number of grid points. This can be seen in Figure 5(b). Note that the number of points below the lower bound $y_a$ increases as the time passes. The reason is that the upper bound $y_b$ prevents the control at the boundary from growing significantly to counteract the decreasing outside temperature. This together with the sub-optimality of the POD solution will prevent the control from reacting faster and stronger to this external inputs. A possible solution is to consider local constraints in the domain $\Omega$ \[31\] and/or local POD basis functions \[1, 3\]. Nevertheless, the fact that the average temperature is inside the bounds and only a small portion of discretization points are active make Algorithm 3 a viable approach in HVAC of residual buildings.

For smaller $\gamma = 0.01$ and $\varepsilon = 0.0075$, we can significantly improve the results in terms of violation of the state constraints; cf. Figure 6. In this test, in fact, the influence of the outside temperature is lower and the method is capable to reduce significantly the number of active points; see Figure 6(b). In particular, the minimum temperature in the room never lies below the given lower bound $y_a$ (Figure 6(a)). The maximum temperature, instead, is slightly above the upper bound $y_b$. This can be justified from the chosen value of the relaxation parameter $\varepsilon$ and also from the fact that we are using the POD method, thus the reduced-order model approximation introduce a further discretization error, which may not be canceled even when passing to the limit $\varepsilon \to 0$. This violation is anyway negligible for HVAC of residual buildings, since the maximum value of the temperature attained all over the time-space domain is 23.07. Furthermore, we already discuss the conflict (and a possible solution) between the state constraints and the control. This can be clearly seen here, since the 33 active points are exactly the grid points belonging to $\Gamma_c$.

Figure 6: Results for Algorithm 3 with $\gamma = 0.01$ and $\varepsilon = 0.0075$. 

(a) Minimum (blue), maximum (red) and average (purple) temperature in the room and state constraints $y_a$ and $y_b$ (black).

(b) Number of active points with respect to the state constraints.
8 Conclusion

We presented a POD-based economic MPC algorithm to handle HVAC of residual buildings. The strategy consists in computing a sub-optimal solution to an optimal control problem subjected to the Boussinesq approximation of Navier-Stokes and bilateral state and control constraints. The goal was to have a good compromise between robustness and computational cost. The first is achieved via the MPC and a goal-oriented a-posteriori error estimate, while the second is obtained through POD and parallel computing. The numerical results confirm the expectations. While some limitations of the strategy show up, particularly the emergence of several discretization points over time in which the state constraints are violated, the resulting control strategy can be regarded accurate enough for typical HVAC applications. The performance can be further improved, for example, by considering localized MOR and localized constraints. This will be the focus of future work.

Acknowledgments

The authors gratefully acknowledge support of DFG grants 274852524, 274852737 and 274853298. This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344 (LLNL-JRNL-818244). This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

References

[1] A. Abdulle and P. Henning. A reduced basis localized orthogonal decomposition. *Journal of Computational Physics*, 295:379–401, 2015.

[2] F. Abergel and R. Temam. *On Some Control Problem in Fluid Mechanics. Theoretical and Computational Fluid Dynamics*. Springer, 1990.

[3] F. Albrecht, B. Haasdonk, S. Kaulmann, and M. Ohlberger. The localized reduced basis multiscale method. In *Proceedings of Algoritmy 2012, Conference on Scientific Computing*,

23
[4] R. Amrit, J. B. Rawlings, and D. Angeli. Economic optimization using model predictive control with a terminal cost. *Annual Rev. Control*, 35:178–186, 2011.

[5] J. Andrej. *Modeling and optimal control of multiphysics problems using the finite element method*. PhD thesis, Kiel University, 2019.

[6] F. Ballarin, A. Manzoni, A. Quarteroni, and G. Rozza. Supremizer stabilization of POD–Galerkin approximation of parametrized steady incompressible Navier–Stokes equations. *International Journal for Numerical Methods in Engineering*, 102(5):1136–1161, 2015.

[7] E. Casas. Pontryagin’s principle for state-constrained boundary control problems of semilinear parabolic equations. *SIAM Journal on Control and Optimization*, 35(4):1297–1327, 1997.

[8] R. Dautray and P.-L. Lions. *Mathematical Analysis and Numerical Methods for Science and Technology. Volume 5: Evolution Problems I*. Springer, 2000.

[9] M. Ellis, J. Liu, and P. D. Christofides. *Economic Model Predictive Control: Theory, Formulations and Chemical Process Applications*. Springer Berlin, 2017.

[10] G. Fabrini, M. Falcone, and S. Volkwein. Coupling MPC and HJB for the computation of POD-based feedback laws. In F. A. Radu, K. Kumar, I. Berre, J. M. Nordbotten, and I. S. Pop, editors, *Numerical Mathematics and Advanced Applications ENUMATH 2017*, pages 941–949, Cham, 2019. Springer International Publishing.

[11] T. Faulwasser, L. Grüne, and M. A. Müller. Economic nonlinear model predictive control. *Foundations and Trends® in Systems and Control*, 5(1):1–98, 2018.

[12] J. Ghiglieri and S. Ulbrich. Optimal flow control based on POD and MPC and an application to the cancellation of Tollmien–Schlichting waves. *Optimization Methods and Software*, 29(5):1042–1074, 2014.

[13] L. Grüne, L. Mechelli, S. Pirkelmann, and S. Volkwein. Performance estimates for economic model predictive control and their application in proper orthogonal decomposition-based implementations. *To appear in Mathematical Control and Related Fields*, Preprint available at [http://nbn-resolving.de/urn:nbn:de:bsz:352-2-ywxqb2eru5d19](http://nbn-resolving.de/urn:nbn:de:bsz:352-2-ywxqb2eru5d19), 2020.

[14] L. Grüne and M. A. Müller. On the relation between strict dissipativity and the turnpike property. *Syst. Contr. Lett.*, 90:45–53, 2016.

[15] L. Grüne and J. Pannek. Practical NMPC suboptimality estimates along trajectories. *Systems & Control Letters*, 58:161–168, 2009.

[16] L. Grüne and J. Pannek. *Nonlinear Model Predictive Control. Theory and Algorithms. 2nd Edition*. Communications and Control Engineering. Springer International Publishing, 2017.
[17] L. Grüne and S. Pirkelmann. Closed-loop performance analysis for economic model predictive control of time-varying systems. In 2017 IEEE 56th Annual Conference on Decision and Control (CDC), page 5563–5569, 2017.

[18] L. Grüne and S. Pirkelmann. Numerical verification of turnpike and continuity properties for time-varying pdes. IFAC-PapersOnLine, 52(2):7 – 12, 2019. 3rd IFAC Workshop on Control of Systems Governed by Partial Differential Equations CPDE 2019.

[19] L. Grüne and S. Pirkelmann. Economic model predictive control for time-varying system: Performance and stability results. Optimal Control Applications and Methods, 41(1):42–64, 2020.

[20] M. Gubisch and S. Volkwein. Proper orthogonal decomposition for linear-quadratic optimal control. In P. Benner, A. Cohen, M. Ohlberger, and K. Willcox, editors, Model Reduction and Approximation: Theory and Algorithms, pages 5–66. SIAM, Philadelphia, PA, 2017.

[21] M. Hintermüller, K. Ito, and K. Kunisch. The primal-dual active set strategy as a semismooth Newton method. SIAM Journal on Optimization, 13(3):865–888, 2002.

[22] M. Hinze and U. Matthes. Optimal and model predictive control of the Boussinesq approximation. In K. Kunisch, J. Sprekels, G. Leugering, and F. Tröltzsch, editors, Control of Coupled Partial Differential Equations, pages 149–174, Basel, 2007. Birkhäuser Basel.

[23] M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich. Optimization with PDE Constraints. Springer Netherlands, 2009.

[24] P. Holmes, J. Lumley, G. Berkooz, and C. Rowley. Turbulence, Coherent Structures, Dynamical Systems and Symmetry. 2nd Edition. Cambridge Monographs on Mechanics. Cambridge University Press, 2012.

[25] K. Ito and K. Kunisch. Semi-smooth Newton methods for variational inequalities of the first kind. ESAIM: Mathematical Modelling and Numerical Analysis, 37(1):41–62, 2003.

[26] K. Krumbiegel and A. Rösch. A virtual control concept for state constrained optimal control problems. Computational Optimization and Applications, 43:213–233, 2009.

[27] L. Mechelli. POD-based State-Constrained Economic Model Predictive Control of Convection-Diffusion Phenomena. PhD thesis, University of Konstanz, 2019.

[28] L. Mechelli and S. Volkwein. POD-based economic optimal control of heat-convection phenomena. In M. Falcone, R. Ferretti, L. Grüne, and W. M. McEneaney, editors, Numerical Methods for Optimal Control Problems, pages 63–87, Cham, 2018. Springer International Publishing.

[29] L. Mechelli and S. Volkwein. POD-based economic model predictive control for heat-convection phenomena. In F. A. Radu, K. Kumar, I. Berre, J. M. Nordbotten, and I. S. Pop, editors, Numerical Mathematics and Advanced Applications ENUMATH 2017, pages 663–671, Cham, 2019. Springer International Publishing.
F. Oldewurtel, A. Parisio, C. N. Jones, D. Gyalistras, M. Gwerder, V. Stauch, B. Lehmann, and M. Morari. Use of model predictive control and weather forecasts for energy efficient building climate control. *Energy and Buildings*, 45:15–27, 2012.

S. Pirkelmann. *Economic Model Predictive Control and Time-Varying Systems*. PhD thesis, University of Bayreuth, 2020.

S. S. Ravindran. A reduced-order approach for optimal control of fluids using proper orthogonal decomposition. *International Journal for Numerical Methods in Fluids*, 34(5):425–448, 2000.

S. S. Ravindran. Error analysis for Galerkin POD approximation of the nonstationary boussinesq equations. *Numerical Methods for Partial Differential Equations*, 27(6):1639–1665, 2011.

J. Rawlings, D. Mayne, and M. Diehl. *Model Predictive Control: Theory and Design. Second edition*. Nob Hill Publishing, 2017.

J. P. Raymond. Nonlinear boundary control of semilinear parabolic problems with point-wise state constraints. *Discrete & Continuous Dynamical Systems - A*, 3:341–370, 1997.

J. C. G. Reyes, P. Merino, J. Rehberg, and F. Tröltzsch. Optimality conditions for state-constrained PDE control problems with time-dependent controls. *Control and Cybernetics*, 37:5–38, 2008.

G. Stabile and G. Rozza. Finite volume POD-Galerkin stabilised reduced order methods for the parametrised incompressible Navier–Stokes equations. *Computers & Fluids*, 173:273 – 284, 2018.

R. Temam. *Navier-Stokes Equations, Theory and Numerical Analysis*. North-Holland, 1984.

E. Trélat and E. Zuazua. The turnpike property in finite-dimensional nonlinear optimal control. *J. Differential Equations*, 258(1):81–114, 2015.

D. J. Tritton. *Physical Fluid Dynamics*. Springer Netherlands, 1977.

F. Tröltzsch and S. Volkwein. POD a-posteriori error estimates for linear-quadratic optimal control problems. *Computational Optimization and Applications*, 44:83–115, 2009.

F. Tröltzsch. Regular Lagrange multipliers for control problems with mixed pointwise control- state constraints. *SIAM Journal on Optimization*, 22:616–634, 2013.

J. Šiřeky, F. Oldewurtel, J. Cigler, and S. Prívara. Experimental analysis of model predictive control for an energy efficient building heating system. *Applied Energy*, 88:3079–3087, 2011.