MAX-PLANCK-GESELLSCHAFT

Max Planck Institute Magdeburg
Preprints

Peter Benner Jan Heiland

Time-dependent Dirichlet Conditions in Finite Element Discretizations

MAX–PLANCK–INSTITUT
FÜR DYNAMIK KOMPLEXER
TECHNISCHER SYSTEME
MAGDEBURG

MPIMD/15-03 **March 10, 2015**

Abstract

For the modelling and the numerical approximation of problems with time-dependent Dirichlet boundary conditions one can call on several consistent and inconsistent approaches. We show that spatially discretized boundary control problems can be brought into a standard state space form accessible for standard optimization and model reduction techniques. We discuss several methods that base on standard finite-element discretizations, propose a newly developed problem formulation, and investigate their performance in numerical examples. We illustrate that penalty schemes require a wise choice of the penalization parameters in particular for iterative solves of the algebraic equations. Incidentally we confirm that standard finite element discretizations of higher order may not achieve the optimal order of convergence in the treatment of boundary forcing problems and that convergence estimates by the common method of *manufactured solutions* can be misleading.

Contents

Impressum:

Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg

Publisher:
Max Planck Institute for
Dynamics of Complex Technical Systems

Address:
Max Planck Institute for
Dynamics of Complex Technical Systems
Sandtorstr. 1
39106 Magdeburg

http://www.mpi-magdeburg.mpg.de/preprints/

1 Introduction

In practical applications, see [18, 19] for examples from flow control, a system is typically controlled via actuations at an interface. The mathematical model to use is, thus, a partial differential equation (PDE) with respect to space and possibly time posed on a domain and controls acting at the boundary. Depending on the application, the control may appear as a Dirichlet or a Neumann or Robin boundary condition.

Despite their importance in the modelling of control setups, cf. [20, Ch. 1], time-dependent inhomogeneous Dirichlet conditions have sparsely been investigated in terms of analysis and numerical approximation. Also for the elliptic or time independent case, in textbooks on optimal control of PDEs, inhomogeneous Dirichlet conditions are often not considered because they are not of *variational type*, i.e. the equations are not posed in a dual space of the solution space, see, e.g., [6, Ch. 2] and [34, Ch. 2.3]. Another rather obvious obstacle is that a standard choice of trial and test functions formulations implies a certain smoothness of the boundary data which may be impractical [34, Ch. 2.3].

For a general overview of the functional analysis for parabolic systems with Dirichlet boundary control, we refer to [6, 25]. One basic approach is to transpose the involved elliptic operator so that the boundary conditions appear in the dynamic equations. This approach considers test functions of higher regularity and allows for rough solutions and boundary values. In the books mentioned, this method is referred to as *Method of Transposition*. More recently, in the literature on numerical approximation of this type of solutions, the term *very* or *ultra weak solutions* has been used. The elliptic case is treated in [7, 17, 28], time-dependent formulations are considered in [21]. An alternative approach of relaxing the boundary constraint via a penalization term in Robin boundary conditions has been investigated in [4, 9].

The scope of the work presented is the assessment of the numerical treatment of boundary control problems in view of employing standard finite dimensional state space system theory for optimal control and model reduction, see [5] for an application example. As the main criterion we set that we can use standard continuous Galerkin schemes and that the spatially discretized problem can be written in the form

$$\dot{v}(t) = g(t, v, u) \tag{1}$$

or, in the linear case, in the form

$$\dot{v}(t) = A(t)v(t) + B(t)u(t). \tag{2}$$

We will consider algebraic manipulations of spatial discretizations of the standard formulation, as well as reformulations of the abstract equations and discuss their performance in numerical approximation of convection-diffusion problems. Apart from the value of an overview and a comparison of more or less well-known approaches, this paper provides evidence and insight to two phenomena that are important for the numerical analysis but that have not gained particular attention yet:

1. The convenient and analytically well understood approach of approximative Robin boundary conditions will likely fail if the state equations are solved only up to a given relative residual.

2. In the considered example, the convergence order of standard finite element schemes of polynomial degree 2 for time-dependent boundary driven problems is lower then one would expect from the convergence order for stationary problems. This lower convergence rate is not detected by the *method of manufactured solution* that is often used to numerically determine the convergence.

In this manuscript, we define consistency, i.e. the reformulation does not change the solution, on the semi-discrete level. Hence, we take the point of view that the solution of the equivalent representation will converge, if the chosen discretization scheme converges. However, this might not be the case, see [15, Ch. 1] for an example considering the Navier-Stokes equations. In short, the consistency of the algebraic manipulations with reformulations of the abstract equations is of highest importance for stable and convergent approximations. We will consider this issue for the treatment of Dirichlet conditions separately in a forth-coming paper.

The paper is organized as follows. In Section 2 we state the type of problems that we will consider both in an abstract setting and after a spatial discretization. In Section 3, we consider approaches that reformulate the spatially discretized equations into the desired form. In Section 4, we discuss methods that reformulate the abstract equations such that a spatial discretization is a system of *distributed type*. In Section 5 we report on numerical tests on the approximation properties of the introduced methods. We conclude the paper by summarizing remarks and an outlook.

2 Generic Problem Formulations

We will define a general continuous formulation that covers weak formulations of many PDEs from the modelling of physical phenomena. Also, we state the generic form of a spatial semi-discretization. We will restrict the considerations to the scalar case.

2.1 Continuous Equations

Let $\Omega \in \mathbb{R}^d$, $d \in \{2, 3\}$, be a bounded and regular domain such that the *trace theorem* as formulated in [8, Thm. 3.1] applies. Let Γ be its boundary. We define the Sobolev spaces $\mathcal{V} := W^{1,2}(\Omega)$ and $\mathcal{H} := L^2(\Omega)$ and the dual space $\mathcal{V}'$ of $\mathcal{V}$ with respect to the continuous embedding of $\mathcal{V}$ in $\mathcal{H}$ to get

$$\mathcal{V} \hookrightarrow \mathcal{H} \hookrightarrow \mathcal{V}'.$$

We also introduce abbreviations for the trace spaces, cf. [12, Ch. 1.1], via

$$\mathcal{Q}' = [W^{\frac{1}{2},2}(\Gamma)]^d \quad \text{and} \quad \mathcal{Q} = \mathcal{Q}'' := \mathcal{L}(\mathcal{Q}', \mathbb{R}).$$

Let

$$\gamma \colon \mathcal{V} \to \mathcal{Q}' \tag{3}$$

be the trace operator as defined, e.g., in [12, Thm. 1.5].

We state the prototype of the continuous problem

Problem 2.1. *Let $T > 0$ and consider $\mathcal{A}\colon (0,T) \times \mathcal{V} \to \mathcal{V}'$. For $\mathcal{F} \in L^2(0,T;\mathcal{V}')$, for $\upsilon_0 \in \mathcal{H}$, and $\mathcal{U} \in L^2(0,T;\mathcal{Q}')$, find υ with $\upsilon(t) \in \mathcal{V}$ and $\dot{\upsilon}(t) \in \mathcal{V}'$, a.e. on $(0,T)$, so that*

$$\dot{\upsilon}(t) - \mathcal{A}(t, \upsilon(t)) = \mathcal{F}(t), \tag{4a}$$

$$\gamma\upsilon(t) = \mathcal{U}(t), \tag{4b}$$

holds for almost all $t \in (0,T)$, and so that $\upsilon(0) = \upsilon_0$ in $\mathcal{H}$.

The system of abstract equations (4) contains common weak formulations of PDEs that model physical phenomena, cf. [32]. We will not address time regularity here and, thus, leave the properties of the mappings $t \mapsto \mathcal{A}(t, \upsilon(t))$ and, e.g., $t \mapsto \dot{\upsilon}(t)$ undefined in the statement of Problem 2.1.

As an example we consider the convection diffusion equation that models the propagation of a scalar quantity due ρ to convection and diffusion in a domain.

Problem 2.2. *Given a domain $\Omega \in \mathbb{R}^d$, a diffusion parameter ν, a convection wind β, with $\beta(x,t) \in \mathbb{R}^d$ for time $t > 0$ and $x \in \Omega$, an initial value ρ_0, and a function g, with $g(t)\colon \Gamma \to \mathbb{R}$ prescribing the boundary conditions, find a function ρ of space and time that satisfies*

$$\dot{\rho}(t) + \beta \cdot \nabla\rho(t) - \nu\Delta\rho(t) = 0, \tag{5a}$$

$$\upsilon|_\Gamma(t) = g(t), \tag{5b}$$

and $\rho(0) = \rho_0$.

In standard weak formulations, assuming $\upsilon \in \mathcal{V} := W^{1,2}(\Omega)$, Problem 2.2 is of the type of Problem 2.1, with, e.g., $\mathcal{A}$ defined via

$$\langle \mathcal{A}(t, \upsilon(t)), \phi \rangle_{\mathcal{V}',\mathcal{V}} = \int_\Omega \bigl(w \cdot \nabla\upsilon(t), \phi\bigr) + \nu\bigl(\nabla\upsilon(t), \nabla\phi\bigr)\, \mathrm{d}\omega - \nu \int_\Gamma \bigl(\frac{\partial \upsilon}{\partial \nu}(t), \phi\bigr)\, \mathrm{d}\gamma, \tag{6}$$

for all $\phi \in \mathcal{V}$ and with $\frac{\partial}{\partial \nu}$ denoting the normal derivative. Here and in what follows, the pairing $(\cdot,\cdot)$ denotes the inner product in the spaces under consideration.

Note that there are other possible choices for a weak formulation [21].

The boundary condition in Equation (4), viewed as a constraint, can also be incorporated using the dual operator of $\gamma\colon \mathcal{V} \to \mathcal{Q}'$ and a so called *Lagrange multiplier*. Then, under certain smoothness and consistency conditions [13], Problem 2.1 is equivalent to

Problem 2.3. *Let $T > 0$ and consider $\mathcal{A}\colon (0,T) \times \mathcal{V} \to \mathcal{V}'$. For $\mathcal{F} \in L^2(0,T;\mathcal{V}')$, $\upsilon_0 \in \mathcal{H}$, and $\mathcal{U} \in L^2(0,T;\mathcal{Q}')$, find υ with $\upsilon(t) \in \mathcal{V}$ and $\dot{\upsilon}(t) \in \mathcal{V}'$ and Λ with $\Lambda(t) \in \mathcal{Q}$, a.e. on $(0,T)$, so that*

$$\dot{\upsilon}(t) - \mathcal{A}(t, \upsilon(t)) - \gamma'\Lambda(t) = \mathcal{F}(t), \tag{7a}$$

$$\gamma\upsilon(t) = \mathcal{U}(t), \tag{7b}$$

holds for almost all $t \in (0,T)$, and so that $\upsilon(0) = \upsilon_0$ in $\mathcal{H}$.

2.2 Spatially Discretized Equations

We consider a generic spatial discretization of the introduced equations. Let $V \subset \mathcal{V}$ be a finite dimensional subspace spanned by the basis functions $\{\psi_i\}_{i=1}^{n_v}$. As it is standard for spatial discretizations of PDEs, we consider nodal bases, i.e. the basis functions are associated with nodes of a mesh and they have local support. We consider the decomposition

$$V = V_I \oplus V_\Gamma,$$

where the subspace $V_\Gamma \subset V$ is spanned by the basis functions $\{\psi_i\}_{i=n_I+1}^{n_v}$ that have nonzero values at the boundary. Accordingly, n_I is the number of nodes in the inner and V_I, which is the span of the remaining basis functions, contains only functions that are zero at the boundary. We will use the abbreviation *dof* to address a degree of freedom that is represented by a basis function of V. Note that the considered splitting of V is not necessarily orthogonal.

Thus, at time t, the function $v(t) \in \mathcal{V}$ is to be approximated by a finite dimensional function $v(t) \in V$ or the vector $v(t) \in \mathbb{R}^{n_v}$ containing the coefficients of the expansion in the considered basis. We will assume that $v = (v_I, v_\Gamma)$ is partitioned, with v_I being associated with V_I and v_Γ being associated with V_Γ, i.e. the parts of V that live in the inner and at the boundary of the considered domain.

Without further mentioning, for a function $v \in V$, we will identify v_I and v_Γ with their coefficient vectors of the expansion in (8) and simply write

$$v = v_I + v_\Gamma = \sum_{i=1}^{n_I} v_i\psi_i + \sum_{i=n_I+1}^{n_v} v_i\psi_i. \tag{8}$$

We will consider test spaces that are subspaces of V. If only Dirichlet conditions are posed, the standard test space will be V_I, if there are only Neumann conditions given, then the standard choice for the test space will be the *full* space V.

Generally, in the assembled coefficient matrices, rows will correspond to dofs in the test space and columns will correspond to dofs in the ansatz space. In particular, we will consider complying partitions of the coefficient matrices like the mass matrix

$$M := \left[(\psi_i, \psi_j)_{\mathcal{H}}\right]_{i,j=1\cdots,n_v}$$

with respect to the test space,

$$M = \begin{bmatrix} M_I \\ M_\Gamma \end{bmatrix},$$

and, once more, with respect to the trial space,

$$M_I = \begin{bmatrix} M_{II} & M_{I\Gamma} \end{bmatrix}, \tag{9}$$

where

$$M_{II} := \left[(\psi_i, \psi_j)_{\mathcal{H}}\right]_{i,j=1\cdots,n_I} \quad \text{and} \quad M_{I\Gamma} := \left[(\psi_i, \psi_j)_{\mathcal{H}}\right]_{i=1,\cdots,n_I, j=n_I+1,\dots n_v}$$

are the parts associated with the inner dofs and the part of the mass matrix that relates to the boundary dofs tested against the inner nodes, respectively.

Thus, if we assume $v(t) \in V$ and if we test against the basis functions of V_I, the generic spatial discretization of Problem 2.1, that treats the boundary separately from the differential equation is of the form

Problem 2.4. *Let $T > 0$, n_v, $n_I \in \mathbb{N}$, and $n_d := n_v - n_I$ and consider $A_I \colon (0,T) \times \mathbb{R}^{n_v} \to \mathbb{R}^{n_I}$, $G \in \mathbb{R}^{n_d,n_v}$, and $M_I \in \mathbb{R}^{n_I,n_v}$. For $f \in L^2(0,T;\mathbb{R}^{n_I})$, $\alpha \in \mathbb{R}^{n_v}$, and $u \in L^2(0,T;\mathbb{R}^{n_d})$ find v with $v(t)$ and $\dot{v}(t) \in \mathbb{R}^{n_v}$, a.e. on $(0,T)$, so that*

$$M_I\dot{v}(t) - A_I(t, v(t)) = f(t) \tag{10a}$$

$$Gv(t) = u(t) \tag{10b}$$

holds for almost all $t \in (0,T)$ and $v(0) = \alpha$.

For the system of Problem 2.3 with the multiplier, a possible spatial discretization defines a differential equation considering also the boundary parts, cf. [2, 13]. It generically takes the form

Problem 2.5. *Let $T > 0$, n_v, $n_I \in \mathbb{N}$, and $n_d := n_v - n_I$ and consider $A \colon (0,T) \times \mathbb{R}^{n_v} \to \mathbb{R}^{n_v}$, $G \in \mathbb{R}^{n_d,n_v}$, and $M \in \mathbb{R}^{n_v,n_v}$. For $f \in L^2(0,T;\mathbb{R}^{n_v})$, $\alpha \in \mathbb{R}^{n_v}$, and $u \in L^2(0,T;\mathbb{R}^{n_d})$ find v with $v(t)$ and $\dot{v}(t) \in \mathbb{R}^{n_v}$ and λ with $\lambda(t) \in \mathbb{R}^{n_d}$, a.e. on $(0,T)$, so that*

$$M\dot{v}(t) - A(t, v(t)) - G^T\lambda(t) = f(t), \tag{11a}$$

$$Gv(t) = u(t), \tag{11b}$$

hold for almost all $t \in (0,T)$ and $v(0) = \alpha$.

For illustration purposes, we will use the linear time-invariant case of Problem 2.4, for which A_I is a linear map given as a matrix $A_I \in \mathbb{R}^{n_I,n_v}$ and write (10) as

$$M_I\dot{v}(t) - A_I v(t) = f(t) \tag{12a}$$

$$Gv(t) = u(t). \tag{12b}$$

More often than not, we will omit the time dependency of the variables and functions.

Remark 2.6. Until now we have not addressed time regularity, but, for sufficiently smooth input functions, we expect to obtain solutions in the classical sense. This, however, means that an initial value has to be consistent with the constraints given by the boundary conditions [22], which is not possible for any input. For the forward simulation, we circumvent this problem by adjusting the initial value to the input. For optimal control setups, this is a severe issue. Note that in the infinite dimensional setting this problem does not appear since the solution is typically only continuous in $\big(t \to \mathcal{H}\big)$, with $\mathcal{H} = L^2(\Omega)$ where boundary conditions do not play a role.

3 Rewriting the Spatially Discretized Equations

In this section, we consider the spatially discretized equations introduced in Section 2.2. For the sake of illustration, we assume that we only have Dirichlet boundary conditions. This is not a restriction, since one can always split the boundaries and consider the parts separately.

3.1 Direct Assignment of the Boundary Dofs

We now illustrate that the immediate way of assigning the dofs at the boundary, as it is commonly done for inhomogeneous Dirichlet conditions for stationary problems [26], does not simply lead to a system of the form (1).

Consider Problem 2.4 with the assignment of the boundary conditions as in (12b):

$$
\begin{aligned}
M_I\dot{v} - A_I(v) &= f, && \text{(13a)}\\
Gv = v_\Gamma &= u, && \text{(13b)}\\
v(0) &= \alpha. && \text{(13c)}
\end{aligned}
$$

Then, with the partitioning of M_I as in (9), the state equation reads

$$
\begin{bmatrix} M_{II} & M_{I\Gamma} \end{bmatrix} \begin{bmatrix} \dot{v}_I \\ \dot{v}_\Gamma \end{bmatrix} = A_I(v_I + v_\Gamma) + f
$$

which, having inserted (13b), gives

$$
M_{II}\dot{v}_I = A_I(v_I + u) + f - M_{I\Gamma}\dot{u}. \tag{14}
$$

System (14) is not of the form (2) because of the appearance of $\dot{u}$.

However, one can define a new input as $\tilde{u} := \dot{u}$ and consider the system

$$
\begin{bmatrix} 1 & 0 \\ 0 & M_{II} \end{bmatrix} \frac{d}{dt} \begin{bmatrix} u \\ v_I \end{bmatrix} - \begin{bmatrix} 0 \\ A_{II}(v_I + u) + f \end{bmatrix} = \begin{bmatrix} 1 \\ -M_{I\Gamma} \end{bmatrix} \tilde{u}.
$$

This approach uses a so called *dynamical controller*, that is defined via a differential relation. As pointed out in [3], for a dynamical controller one can set the initial value to zero to circumvent the expected inconsistencies mentioned in Remark 2.6.

3.2 Lifting of the Boundary Conditions

In these approaches, one defines a lifting $\tilde{v}$ that fulfills the boundary conditions for all time and considers the decoupling of the solution $v = y + \tilde{v}$, see [31] for an example with linearized Navier-Stokes equations.

We consider the linear time-invariant case (13) and assume that $f = 0$. At time $t \in [0, T]$, we define a lifting as

$$
\tilde{v}(t) = \begin{bmatrix} \tilde{v}_I(t) \\ u(t) \end{bmatrix}. \tag{15}
$$

Then, considering (8) with $v = y + \tilde{v}$ and splitting A_I and M_I as in (9), we find that $y_\Gamma = 0$ and we obtain the relation

$$M_{II}\dot{y}_I = A_{II}y_I + A_I\tilde{v} - M_I\dot{\tilde{v}}, \quad y_I(0) = \alpha_I - \tilde{v}_I(0).$$

We use the abbreviation $\bar{A}_{II} = M_{II}^{-1}A_{II}$ and, with the well-known solution representation, we obtain that

$$y_I(t) = e^{\bar{A}_{II}t}(\alpha_I - \tilde{v}_I(0)) + \int_0^t e^{\bar{A}_{II}(t-s)}M_{II}^{-1}(A_I\tilde{v} - M_I\dot{\tilde{v}}(s))\,\mathrm{d}s.$$

After an integration by parts, we find that

$$y_I(t) = e^{\bar{A}_{II}t}(\alpha_I - \tilde{v}_I(0)) + \int_0^t e^{\bar{A}_{II}(t-s)}(A_I\tilde{v}(s) - \bar{A}_{II}M_{II}^{-1}M_I\tilde{v}(s))\,\mathrm{d}s \\ - M_{II}^{-1}M_I\tilde{v}(t) + e^{\bar{A}_{II}t}M_{II}^{-1}M\tilde{v}(0).$$

Using that $M_I\tilde{v} = M_{II}\tilde{v}_I + M_{I\Gamma}u$ and having regrouped the terms, we conclude that $\hat{v}_I := y_I + M_{II}^{-1}M\tilde{v} = v_I + M_{II}^{-1}M_{I\Gamma}u$ fulfills the ordinary differential equation

$$M_{II}\dot{\hat{v}}_I = A_{II}\hat{v}_I + Bu, \quad \hat{v}_I(0) = \alpha_I + M_{II}^{-1}M_{I\Gamma}u(0), \tag{16}$$

with

$$B = [A_{II}M_I^{-1}M_{I\Gamma} - A_{I\Gamma}].$$

The actual solution is easily retrieved from $\hat{v} = v_I + M_{II}^{-1}M_{I\Gamma}u$. However, the dependency of the initial value on u in (16) is indeed an issue, cf. Remark 2.6.

Remark 3.1. We find it worth pointing out, that the system (16) does not depend on the choice of the lifting (15) and, thus, includes in particular the lifting by means of the *harmonic extension* of the boundary values into the inner.

Split mass matrix lifting

For the particular choice of the lifting

$$\tilde{v}(t) = \begin{bmatrix} -M_{II}^{-1}M_{I\Gamma}u(t) \\ u(t) \end{bmatrix}$$

which leads to $M_I\dot{\tilde{v}}(t) = 0$ for all time t, the application for nonlinear systems is straight forward. Considering again, $y = v - \tilde{v}$, and the nonlinear case of Problem 2.4, one arrives at the ODE

$$M_{II}\dot{y}_I = A_I(y_I + \tilde{v}(u)) + f, \quad y_I(0) = \alpha_I + M_{II}^{-1}M_{I\Gamma}u(0).$$

Again, the actual solution is easily obtained by a backwards substitution $v_I = y_I + \tilde{v}_I = y_I - M_{II}^{-1}M_{I\Gamma}u$, but the initial value depends on the possibly unknown input u.

Remark 3.2. A lifting as defined in this chapter leads to an ODE of the desired form. In a forthcoming work, we will investigate similar manipulations on the abstract equations. If the proposed algebraic splitting has a counterpart in infinite dimensions, then one can expect well-posedness of the transformed system also for ever finer discretizations.

Remark 3.3. For linear time-dependent cases, similar formulas can be derived using fundamental solution matrices or transition matrices. Also, the split mass matrix approach is readily applicable and gives a system of type (2).

3.3 Incorporation of the Boundary Data via Lagrange Multiplier

We consider the formulation of Problem 2.5:

$$M\dot{v}(t) - A(t, v(t)) - G^T\lambda(t) = f(t), \tag{17a}$$

$$Gv(t) = u(t), \tag{17b}$$

with the Lagrangian multiplier λ.

The saddle point structure is similar to the velocity-pressure formulation of Navier-Stokes equations where the pressure can be interpreted as the multiplier that couples the divergence constraint to the momentum equation. In particular, it is a special case of semi-explicit index-2 DAEs as they were considered, e.g., in [14]. Thus, the formulations for the treatment of the boundary conditions that we propose in this section are adaptions of algorithms for the numerical time integration of Navier-Stokes equations or, more general, DAEs of index 2.

3.3.1 Decoupling by Projection

In the considered case, G has the form $G = \begin{bmatrix} 0 & I_I \end{bmatrix}$ and M is symmetric positive definite. Thus, we can define

$$P := I - M^{-1}G^TS^{-1}G, \quad S := GM^{-1}G^T, \quad \text{and} \quad Q^- := S^{-1}GM^{-1}.$$

With this, system (17) can be equivalently [15] reformulated as $(v, \lambda) = (v_i + v_g, \lambda)$, where the transformed variables are the solutions of

$$v_g = M^{-1}G^TS^{-1}u, \tag{18}$$

$$\lambda = -Q^-A(v_g + v_i) - Q^-f - Q^-M\dot{v}_\Gamma, \tag{19}$$

and

$$\dot{v}_i - PM^{-1}A(v_i + M^{-1}G^TS^{-1}u) = PM^{-1}f. \tag{20}$$

Note that the differential equation (20) is of type (1).

Noting that $MP = P^TM$, in the linear case, we can write the differential equation for v_i as

$$M\dot{v}_i - P^TAv_i = P^Tf + P^TBu,$$

with $B := AM^{-1}G^TS^{-1}$. In the nonlinear case, the input appears inside the nonlinearity.

Remark 3.4. Since $n_d \ll n_v$, i.e. the number of dofs associated with the boundary is small if compared to the number of inner nodes, an explicit realization of the projection P is feasible. This is different from the analogue for the Navier-Stokes equation, where the dimension of the subspace of the divergence free functions equals the dimension of the pressure space and, thus, can be large.

Remark 3.5. The variable v_i has zero values at the boundary at all time. Thus, if one only considers the ODE (20) for v_i, there is no problem of possibly inconsistent initial values due to the chosen control, cf. Remark 2.6. However, a given initial value has to fulfill also (18).

3.3.2 Regularization via Penalization

If one adds the term $\alpha\lambda(t)$, $0 < \alpha \ll 1$, to the left hand side of Equation (17b), one can solve for the multiplier and eliminate it from the differential part:

$$M\dot{v}(t) - A(t, v(t)) + \frac{1}{\alpha} G^T G v = f(t) + \frac{1}{\alpha} G^T u.$$

This approach is known as *penalty scheme* and *pressure penalization* in the numerical integration of multibody and Navier-Stokes systems, respectively, cf., e.g., [10, 30]. The method is straight forward to implement but comes with the need of a proper choice of the penalization parameter. The main difficulty is that a small parameter α increases the quality of the approximation of the constraints but also increases the stiffness of the resulting ODE.

4 Incorporation via Variational Formulations and their Discretizations

In its most general form, the *variational* or *weak* incorporation of Dirichlet boundary conditions is derived from Problem 2.1 as follows. Instead of considering the constraint (4b) one adds a penalty term to variational formulation of the dynamic equation (4a)

$$\dot{v}(t) - \mathcal{A}(t, v(t)) + \frac{1}{\alpha}\lambda'(\gamma v(t) - \mathcal{U}(t)) = \mathcal{F}(t), \tag{21}$$

where $\lambda' \colon \mathcal{Q} \to \mathcal{V}'$ and α is a small penalization parameter. Then, for various choices of λ and $\mathcal{Q}$, various weak incorporations of the Dirichlet conditions are realized. For example, defining λ' through

$$\langle \lambda' q, \phi \rangle_{\mathcal{V}',\mathcal{V}} = \int_\Gamma (q, \phi) \, \mathrm{d}\gamma$$

for a $q \in \mathcal{Q}$ and for any $\phi \in \mathcal{V}$, one obtains the penalized Robin approximation described in Section 4.3 below.

4.1 Ultra Weak Formulations

For completeness, we mention the *ultra week* variational formulation. However, we do not consider it in the numerical experiments since the *ultra weak* formulation requires special trial and test spaces that are not part of common finite element libraries. Let $\Phi = W^{2,2}(\Omega) \cap W_0^{1,2}(\Omega)$ and consider the diffusion equation (5) with $\beta = 0$. We call v a solution if

$$\int_\Omega (\dot{v}, \phi) \, \mathrm{d}\omega - \nu \int_\Omega (v, \Delta\phi) \, \mathrm{d}\omega = \langle f, \phi \rangle_{\Phi',\Phi} - \nu \int_\Gamma (g, \frac{\partial \phi}{\partial \nu}) \, \mathrm{d}\gamma \tag{22}$$

for all $\phi \in \Phi$, cf. [17].

The abstract equations (22) indicate that a spatial discretization may lead to a system in the form of (10). Known approaches for the numerical approximation of very weak solutions, however, do not provide an explicit representation of the discrete operators A and B, cf. [17, Ch. 3.2] and [7]. The major difficulty lies in the proper choice of matching test functions of high regularity with zero boundary values and suitable ansatz functions.

The explicit discrete representation that is given in [24, Ch. 5.2.1] bases on the assumption that

$$\int_\Omega (\Delta y, \phi) \, \mathrm{d}\omega = \int_\Omega (y, \Delta\phi) \, \mathrm{d}\omega,$$

cf. the proof of [24, Lem. 3.1.1], and that the trial space is a subset of $W_0^{1,2}(\Omega)$. These requirements, however, necessarily lead to a solution of L^2 regularity regardless of possibly higher regularity of the problem.

4.2 Nitsche Variational Formulation

A variant of the standard weak formulation of the pure diffusion, cf. (5) with $\beta = 0$, as proposed in [29] for the stationary Poisson equation reads

$$\begin{aligned}\int_\Omega (\dot{v}, \phi) \, \mathrm{d}\omega + \nu \int_\Omega (\nabla v, \nabla\phi) \, \mathrm{d}\omega - \nu \int_\Gamma (\frac{\partial v}{\partial \nu}, \phi) \, \mathrm{d}\gamma - \nu \int_\Gamma (v, \frac{\partial \phi}{\partial \nu}) \, \mathrm{d}\gamma + c_\gamma \int_\Gamma (v, \phi) \, \mathrm{d}\gamma \\ = \langle f, \phi \rangle_{\Phi',\Phi} - \nu \int_\Gamma (g, \frac{\partial \phi}{\partial \nu}) \, \mathrm{d}\gamma + c_\gamma \int_\Gamma (g, \phi) \, \mathrm{d}\gamma\end{aligned} \tag{23}$$

for all $\phi \in \Phi = W^{1,2}(\Omega)$. The formulation is derived by considering the cost functional

$$\mathcal{J}(w) = \nu \int_\Omega (\nabla w, \nabla w) \, \mathrm{d}\omega - 2\nu \int_\Gamma (\frac{\partial w}{\partial \nu}, w) \, \mathrm{d}\gamma + c_\gamma \int_\Gamma (w, w) \, \mathrm{d}\gamma,$$

with a parameter c_γ and the first order optimality conditions for $\mathcal{J}(w - v) \to \min$, where v is the solution of the stationary Poisson problem with nonhomogeneous Dirichlet boundary conditions. If for a given mesh c_γ is chosen sufficiently large, namely $c_\gamma \approx h^{-1}$ where h is a characteristic length of the triangulation, then the discretized optimization problem is convex [29, Eq. (12)].

For (23), a standard discrete formulation leads to an equation of type (2) with A and B explicitly given, see [33]. Cf. also [24, Ch. 5.2.2] where nonzero boundary values of y have been assumed.

4.3 Penalized Robin

If one approximates the Dirichlet conditions by a Robin type condition

$$v \approx \alpha \frac{\partial v}{\partial \nu} + v = g \quad \text{or} \quad \frac{\partial v}{\partial \nu} \approx \frac{1}{\alpha}(g - v) \quad \text{on } \Gamma,$$

with a parameter α that is intended to go to zero, then the boundary conditions are incorporated *naturally* in the weak formulation the convection-diffusion operator (6) and a standard finite element discretization leads to a system of type (1). For the pure diffusion case, convergence of the solutions to the actual solution for $\alpha \to 0$ has been shown in several contexts, cf. [4] and the references therein.

5 Numerical Tests

We consider two-dimensional convection-diffusion-reaction problems. All setups are directed to actuation at the boundary. In particular, there are no source terms. This excludes the method of *manufactured solutions* for consistency and convergence checks, where one constructs a solution and derives the corresponding source term and boundary data. In any case, the method of *manufactured solution* seems not well suited to test the modelling of boundary actuation, since the numerical solution will depend almost exclusively on the volume force, see the test case at the end of this section.

Hence, in order to evaluate the convergence numerically, we compute a reference solution using the naive approach (14) of directly assigning the boundary nodes and a very fine grid in space and time.

We refer to the tested schemes as follows:

- dias – direct assignment of the boundary values – cf. Section 3.1
- lift – lifting of the boundary conditions via split mass matrix – cf. Section 3.2
- proj – incorporation of the constraint via Lagrange multiplier and projections – cf. Section 3.3.1
- pena – penalization of the constraint – cf. Section 3.3.2
- nits – approximation via the *Nitsche* variational-formulation – cf. Section 4.2
- pero – relaxation via Robin approximation – cf. Section 4.3

For all test setups, we will check the convergence of dias and that the theoretically equivalent formulations lift and proj give the same results. Also, we will investigate how the relaxed methods pena, nits, and pero perform for different choices of the penalization parameter and for inexact solves of the resulting linear systems.

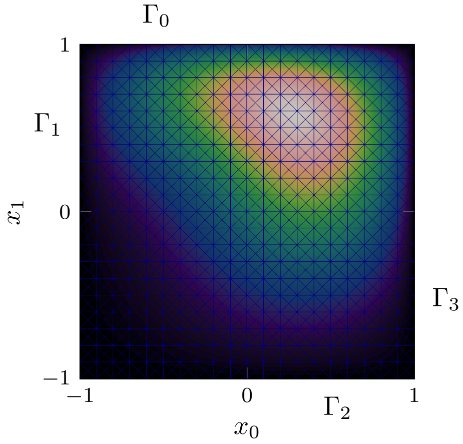


Figure 1: Illustration of the domain, the arrangement of the boundary segments, a triangulation with $N_h = 20$, and a snapshot of an approximation to the internal convection-diffusion as described in Test Case 1 at time $t = 3.0$.

5.1 Test Setups

We consider several convection-diffusion setups on a two-dimensional square domain. Let $\Omega = [-1, 1] \times [-1, 1] \subset \mathbb{R}^2$ be the computational domain with the spatial coordinates $x = (x_0, x_1)$. Let Γ be the boundary with parts Γ_0 to Γ_3 as depicted in Figure 1. All setups model the evolution in time and space of a scalar quantity ρ due to a convection wind β and diffusion with a diffusion coefficient ν, cf. Problem 2.2.

The quantity ρ is seeded into the domain at Γ_0, where we enforce the time-dependent Dirichlet conditions:

$$\rho\big|_{\Gamma_0} = g(x)u(t) := \frac{1}{2}(\sin(\pi x_0 + \frac{\pi}{2}) + 1)(\cos(2t + \pi) + 1). \tag{24}$$

Here, $g(x) := \frac{1}{2}(\sin(\pi x_0 + \frac{\pi}{2}) + 1)$ is the spatial shape function and $u(t) := \cos(2t + \pi) + 1$ is the scalar control function. At the remainder boundaries, Γ_1, Γ_2, and Γ_3, depending on the setup, homogeneous Dirichlet or homogeneous Neumann boundary conditions are applied. As the initial value, we set $\rho(0) = 0$, which is consistent with the control action at time $t = 0$.

As the first test case, we consider a setup with no convection at the boundary, so that the boundary control is propagated into the domain only by diffusion.

Test Case 1 (Internal Convection-Diffusion). Given a convection wind and a diffusion parameter

$$\beta_0(x) = \begin{bmatrix} -x_1(x_0-1)^2(x_0+1)^2(x_1-1)(x_1+1) \\ x_0(x_0-1)(x_0+1)(x_1-1)^2(x_2+1)^2 \end{bmatrix} \quad \text{and} \quad \nu_0 = 0.1,$$

find approximations to the scalar function ρ satisfying

$$\dot{\rho}(t) + \beta_0 \cdot \nabla\rho(t) - \nu_0\Delta\rho(t) = 0, \tag{25a}$$

$$\rho\big|_{\Gamma_0}(t) = gu(t), \tag{25b}$$

$$\rho\big|_{\Gamma_1\cup\Gamma_2\Gamma_3}(t) = 0, \tag{25c}$$

$$\rho(0) = 0, \tag{25d}$$

on given discretizations of the spatial domain $\Omega = [-1, 1]^2$ and of the time interval $[0, 4]$.

As a second test case, we consider a convection-diffusion problem with inflow and outflow, for which the boundary values are also transported into the domain via convection. See Figure 2 (a) for an illustration of the setup.

Test Case 2 (Convection-Diffusion). Given a convection wind and a diffusion parameter

$$\beta_1(x) = \tfrac{1}{10}\begin{bmatrix} x_0 + 1 \\ -(x_1 + 1)\end{bmatrix} \quad \text{and} \quad \nu_1 = 0.1,$$

find approximations to the scalar function ρ satisfying

$$\begin{aligned} \dot{\rho}(t) + \beta_1 \cdot \nabla\rho(t) - \nu_1\Delta\rho(t) &= 0, \\ \rho\big|_{\Gamma_0}(t) &= gu(t), \\ \rho\big|_{\Gamma_1\cup\Gamma_2}(t) &= 0, \\ \frac{\partial\rho}{\partial\nu}\big|_{\Gamma_3}(t) &= 0, \\ \rho(0) &= 0, \end{aligned}$$

on given discretizations of the spatial domain $\Omega = [-1, 1]^2$ and of the time interval $[0, 0.2]$.

The third test case is the same as the second but with an additional reaction source term $r(\rho) = \rho(1 - \rho)$ in the dynamical equation. This source term r is positive for values of $0 \leq \rho \leq 1$ and negative elsewhere. Thus, for values of $\rho > 0$ the reaction pushes ρ towards $\rho = 1$, cf. Figure 2 (b).

The considered system, for $t \in (0, 1]$, now reads

Test Case 3. Given the wind and the diffusion parameter defined in Test Case 2, find approximations to the scalar function ρ satisfying

$$\begin{aligned} \dot{\rho}(t) + \beta_1 \cdot \nabla\rho(t) - \nu_1\Delta\rho(t) &= \rho(t)(1 - \rho(t)), \\ \rho\big|_{\Gamma_0}(t) &= gu(t), \\ \rho\big|_{\Gamma_1\cup\Gamma_2}(t) &= 0, \\ \frac{\partial\rho}{\partial\nu}\big|_{\Gamma_3}(t) &= 0, \\ \rho(0) &= 0, \end{aligned}$$

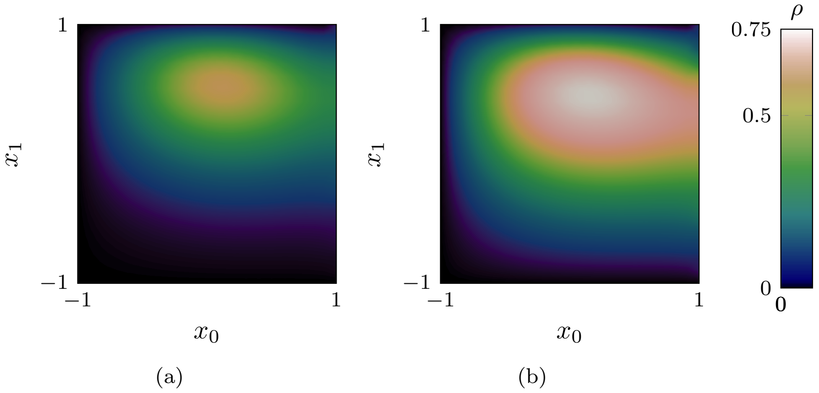


Figure 2: Illustration of the distribution of the scalar ρ seeded at the upper boundary after diffusion and convection (a) and additional reaction (b) as described in Test Case 2 and Test Case 3 for $N_h = 15$ at time $t = 3.0$.

on given discretizations of the spatial domain $\Omega = [-1, 1]^2$ and of the time interval $[0, 0.2]$.

For all test cases, the spatial discretization is done on a uniform *criss-cross* triangulation described by the parameter $N_h = \frac{2}{h}$ which is the length of the boundary parts divided by the length of the longest edge of the triangles, see Figure 1. For the discrete function space, we use the parameter `cg`, denoting the polynomial degree of the chosen standard *Lagrange* elements. For the time discretization, we use a uniform grid of size $N_\tau \approx \frac{1}{\tau}$ corresponding to the ratio of the length of the time interval versus the length of one time step. Here and in the following examples, already for the coarsest discretization, the local *Peclet number* $Pe := \|\beta(t)\|h/\nu$ is smaller than 1. Thus, we can expect reliable approximations without additional, e.g. upwind, stabilization [23].

For the spatial discretization we use the Python interface *dolfin* [27] to the finite element software suite *Fenics* [1]. Our investigation focusses on the space discretization errors but we will make sure that the time integration error is sufficiently small. For the linear cases, the time integration is done by means of the *implicit trapezoidal* rule. The nonlinear case is treated implicit in the linear part and with the *Method of Heun* in the nonlinear part. The norms are computed using the piecewise trapezoidal rule for the time integration and *dolfin*'s built-in function `errornorm` that evaluates the L^2 norm in the discrete function spaces. In general, we solve the occurring linear equation systems via a direct solver that makes use of the python module *scipy*'s built-in sparse LU decomposition method. In some tests we employ the *GMRES* method using the implementation of the python module *krypy* [11]. The code used can be obtained from the author's git repository [16].

By $\rho^{p\mathtt{cg}}_{hN_h,\tau N_\tau}$ we denote the approximation to the solution of (25) with the discretization parameters N_h, N_τ, and `cg`. By $e^{p\mathtt{cg}}_{hN_h,\tau N_\tau}$ we denote the approximation error

$$e^{p\mathsf{cg}}_{hN_h,\tau N_\tau} := \rho^{p\mathsf{cg}}_{hN_h,\tau N_\tau} - \rho_{\mathrm{ref}}$$

measured in a numerical approximation of the $L^2(0,1;L^2([-1,1]^2))$ norm, where ρ_{ref} is a reference computed with the $\mathsf{cg} = 2$ scheme with $N_\tau = 240$ and $N_h = 96$.

5.2 Convergence Tests

In Tables 1 and 2, we list the approximation errors for increasingly fine space and time discretizations for Test Cases 1 and 2. One can see, that the spatial discretization error is dominating, i.e. convergence in the time discretization is only observed for larger values of N_τ. This justifies the choice of $N_\tau := 240$ as the reference discretization for further error comparisons.

The errors $e^{p\mathsf{cg}}_{hN_h,\tau 120}$ for a fixed time discretization and varying space discretizations are plotted Figure 3 for all three test cases. From the plots, one can see that the equivalent formulations lift and proj coincide with the naive implementation dias. Also, one can read off the numerically estimated order of spatial convergence EOC. For the linear elements ($\mathsf{cg} = 1$), one obtains $\mathtt{EOC} = 2$ and for the quadratic elements ($\mathsf{cg} = 2$), one obtains $\mathtt{EOC} = 2.5$ at a lower error level. The observed order of convergence is not optimal as laid out in Section 5.4.

5.3 Parameter Studies for the Penalty Schemes

For the schemes pena, nits, and pero that depend on a parameter, we investigate the accuracy of the approximation versus the choice of the penalization parameter α, where we have defined the relation $c_\gamma = \frac{\nu}{\alpha}$ to fit in Nitsche's method (23). Judging from the results depicted in Figure 4, for large penalization parameters, the approximation is bad, while for small parameters the accuracy of the consistent approximations is obtained. Using the *Nitsche* method nits, for large values of α, we didn't find reasonable approximations.

The necessity to properly choose the penalization parameter is evident in the errors that are reported for inexact solutions of the resulting linear systems. If one applies *GMRES*, to solve the algebraic equations in every time step, the approximation error of the penalization schemes increases with smaller penalization parameters α. The plots in Figure 5 show this phenomenon. For this investigation, we allowed a relative residual of atmost $\mathsf{tol} = 10^{-5}$, which is already less than the overall error which is of magnitude 10^{-4}.

The increase in the approximation error is mainly due to the increase of the magnitude of the right hand side that scales with $\frac{1}{\alpha}$. In fact, having solved the exemplary linear system $\mathsf{A}\mathsf{x} = \mathsf{f}$ up to a relative residual of tol, one has that

$$\frac{\|\mathsf{A}\mathsf{x} - \mathsf{f}\|}{\|\mathsf{f}\|} = \mathsf{tol} \quad \text{or} \quad \|\mathsf{A}\mathsf{x} - \mathsf{f}\| = \mathsf{tol} \cdot \|\mathsf{f}\|,$$

which means that for larger right hand sides f, the absolute residual $\|\mathsf{A}\mathsf{x} - \mathsf{f}\|$ can be larger. A remedy is to control the absolute residual which can be done by correcting

$N_h \backslash N_\tau$	30	60	120
6	1.0000	1.0026	1.0033
12	0.2608	0.2641	0.2651
24	0.0654	0.0661	0.0671
48	0.0244	0.0163	0.0166
96	0.0215	0.0059	0.0041

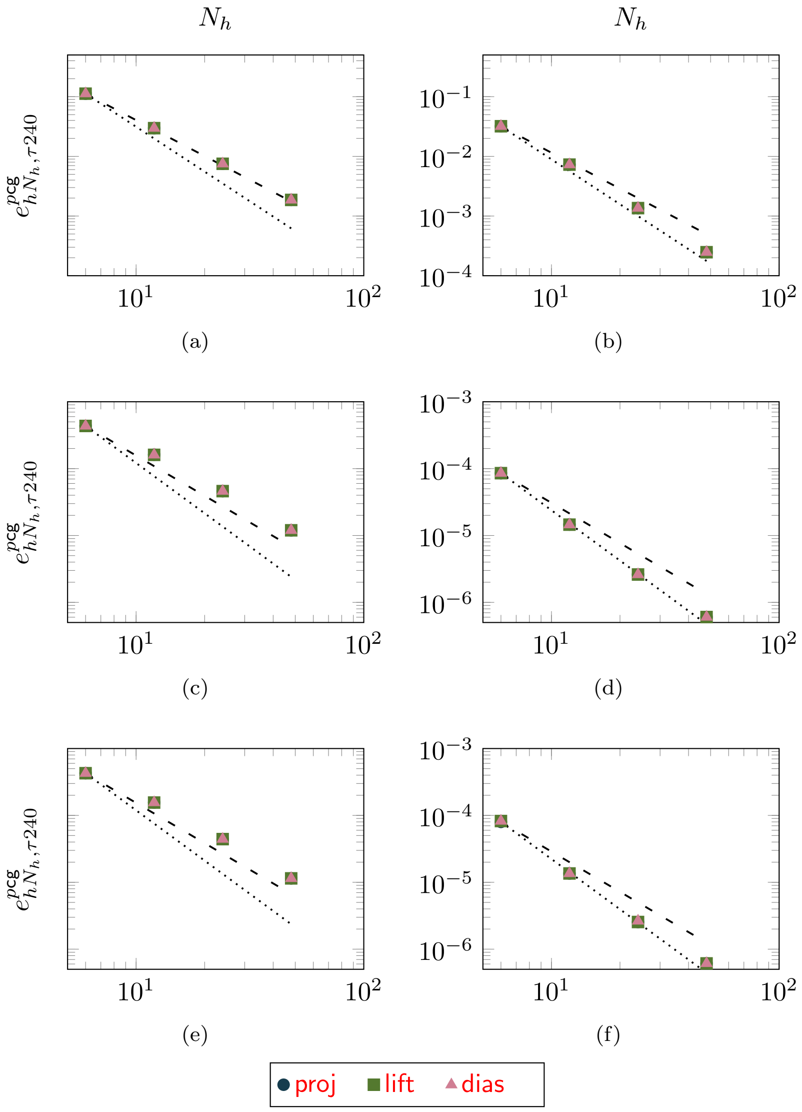
$N_h \backslash N_\tau$	60	120	240
6	1.0000	0.9982	0.9978
12	0.2295	0.2272	0.2269
24	0.0482	0.0424	0.0419
48	0.0201	0.0077	0.0063

Table 1: (Time space convergence of dias for linear elements, cf. Section 5.2) The approximation error $e^{p\mathsf{cg}}_{hN_h,\tau N_\tau}$ with $\rho_{\mathrm{ref}}{=}\rho^{p2}_{h96,\tau 120}$ scaled by the inverse of $e^{p1}_{h6,\tau 30} = 1.119 \cdot 10^{-1}$ (left) and $e^{p2}_{h6,\tau 60} = 3.201 \cdot 10^{-2}$ (right) for varying space and time discretizations and for polynomial degree $\mathsf{cg} = 1$ (left) and $\mathsf{cg} = 2$ (right) for Test Case 1. Cf. also Figure 3(a,b) illustrating the convergence in space for the finest time discretization (the rightmost columns in the tables).

$N_h \backslash N_\tau$	30	60	120
6	1.0000	0.9997	0.9996
12	0.3696	0.3695	0.3694
24	0.1060	0.1059	0.1059
48	0.0276	0.0275	0.0275
96	0.0071	0.0070	0.0069

$N_h \backslash N_\tau$	60	120	240
6	1.0000	1.0000	0.9999
12	0.1699	0.1699	0.1699
24	0.0316	0.0330	0.0305
48	0.0085	0.0071	0.0071

Table 2: (Time space convergence of dias for quadratic elements, cf. Section 5.2) The approximation error $e^{p\mathsf{cg}}_{hN_h,\tau N_\tau}$ with $\rho_{\mathrm{ref}}{=}\rho^{p2}_{h96,\tau 120}$ scaled by the inverse of $e^{p1}_{h6,\tau 30} = 4.349 \cdot 10^{-4}$ (left) and $e^{p2}_{h6,\tau 60} = 8.551 \cdot 10^{-05}$ (right) for varying space and time discretizations and for polynomial degree $\mathsf{cg} = 1$ (left) and $\mathsf{cg} = 2$ (right) for Test Case 2. Cf. also Figure 3(c,d) illustrating the convergence in space for the finest time discretization (the rightmost columns in the tables).



(a)

(b)

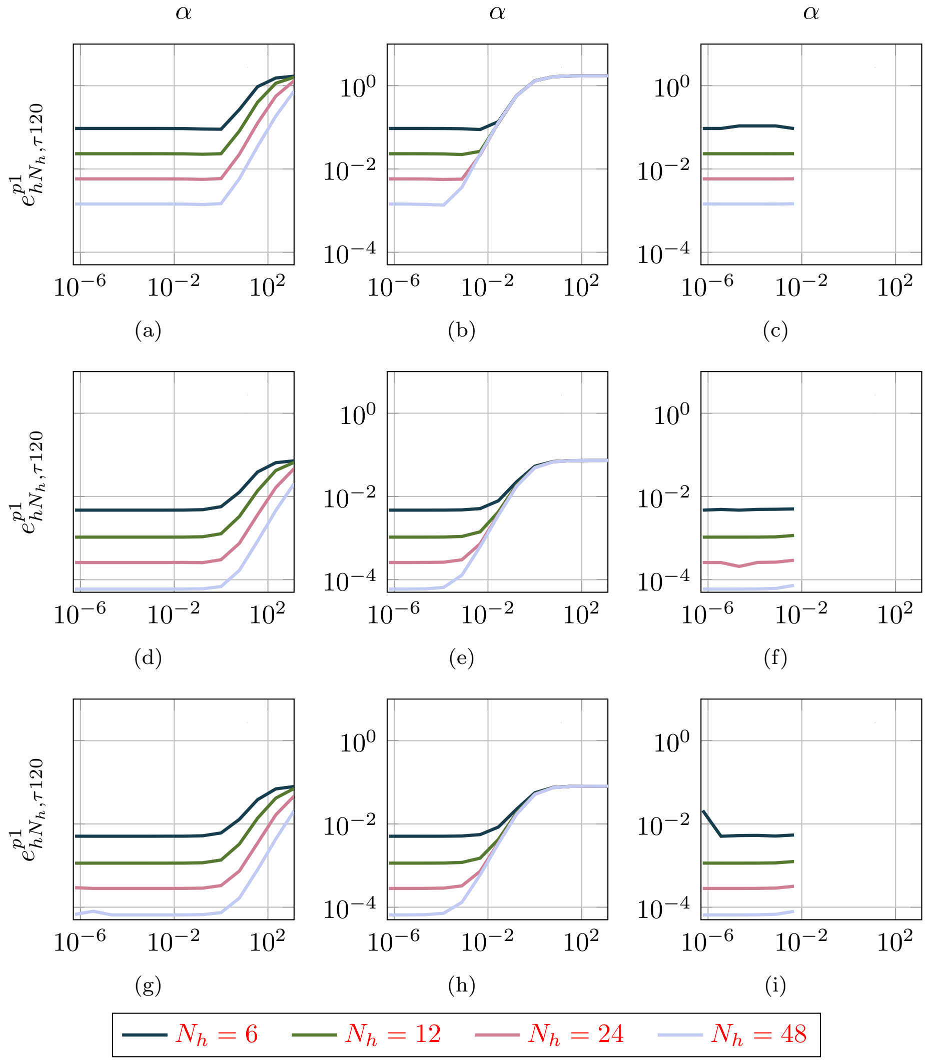
(c)

(d)

(e)

(f)

Figure 3: (Convergence tests for the consistent implementations, cf. Section 5.2) The error $e^{pcg}_{hN_h,\tau 120}$ for varying space discretizations N_h and for linear (left) and quadratic (right) shape functions. The first row of plots (a-b) corresponds to Test Case 1, the middle row (c-d) to Test Case 2, and the bottom line (e-f) to Test Case 3. The dashed lines indicate the slope of a quadratic convergence the dotted lines indicate a convergence of order 2.5.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 4: (Penalization parameter study, cf. Section 5.3) The error $e^{p1}_{hN_h,\tau 120}$ for varying space discretizations N_h versus the penalization parameter α for the schemes pena (left), pero (middle), and nits (right). The first row of plots (a-c) corresponds to Test Case 1, the middle row (d-f) to Test Case 2, and the bottom line (g-h) to Test Case 3.

the provided relative residual by a factor $\mathsf{tolcor} = \min\{\frac{1}{\|\mathsf{f}\|}, 1\}$, where f denotes the current right hand side. In Figure (6d-f) we have reported the discrete $L^2(0, 0.2)$ norm of tolcor for Test Case 2. Applying this correction, that scales with $\frac{1}{\alpha}$, recovers the approximation properties of exact solves over the whole range of α, cf. Figure (6g-i) and (4d-f).

5.4 Convergence Tests with Volume Forcing

In the beginning of Section 5, we have mentioned that the method of *manufactured solutions* is not suitable for boundary controlled processes. This is intuitively clear since for ever finer discretizations the weight of a boundary tends to zero if compared to a surface or volume patch. More concretely, in two spatial dimensions, the number of nodes at the boundary grows linearly while the number of nodes in the inner grows at least quadratically. Thus, if the boundary conditions are merely an extension of a volume force, the volume force will dominate over what happens at the boundary.

To back this assertion by a numerical experiment, we consider Test Case 1 and Test Case 2 (see Section 5.1) but with an additional volume force in (25a) corresponding to the constructed solution

$$\rho_{\mathrm{ref}} = \frac{1}{8}\big(\sin(x_0\pi + \frac{\pi}{2}) + 1)\big)\big(\sin(\frac{x_1}{2}\pi) + 1\big)\big(1 + x_1\big)u(t),$$

with u as in (24). The solution ρ_{ref} is constructed such that at Γ_0 it coincides with the boundary control function defined in (24) and such that it is zero at the remaining boundaries. Also, $\frac{\partial \rho_{\mathrm{ref}}}{\partial \nu}\big|_{\Gamma_3} = 0$ as required for the setup of Test Case 2.

Taking the method lift and tabulating the approximation errors for varying time and space discretization, for linear elements, we find spatial convergence orders $\mathtt{EOC} = 2$, i.e. doubling N_h reduces the error by a factor of 2^{-2}. For quadratic elements we find $\mathtt{EOC} = 3$, i.e. doubling N_h reduces the error by a factor of 2^{-3}, cf. Table 3, Table 4, and Figure 7. The convergence order is as expected for stationary problems and, for the quadratic ansatz functions, significantly better than in the previous experiments, cf., in particular, Table 1 and Figure 3(b,d). This indicates that the boundary conditions are not optimally considered by standard discretization schemes. Moreover, this insufficiency is not captured by numerical tests with systems that are driven by a volume force.

6 Conclusion

We have listed common numerical schemes and introduced a projection based method for problems with time dependent Dirichlet boundary conditions. We have made the distinction between consistent schemes and relaxed schemes that depend on a penalization parameter.

The analysis has shown, that the considered schemes all come with the necessity that an initial value is consistent with the boundary actuation at the initial time. This is a severe issue in control applications where the boundary actuation is unknown a

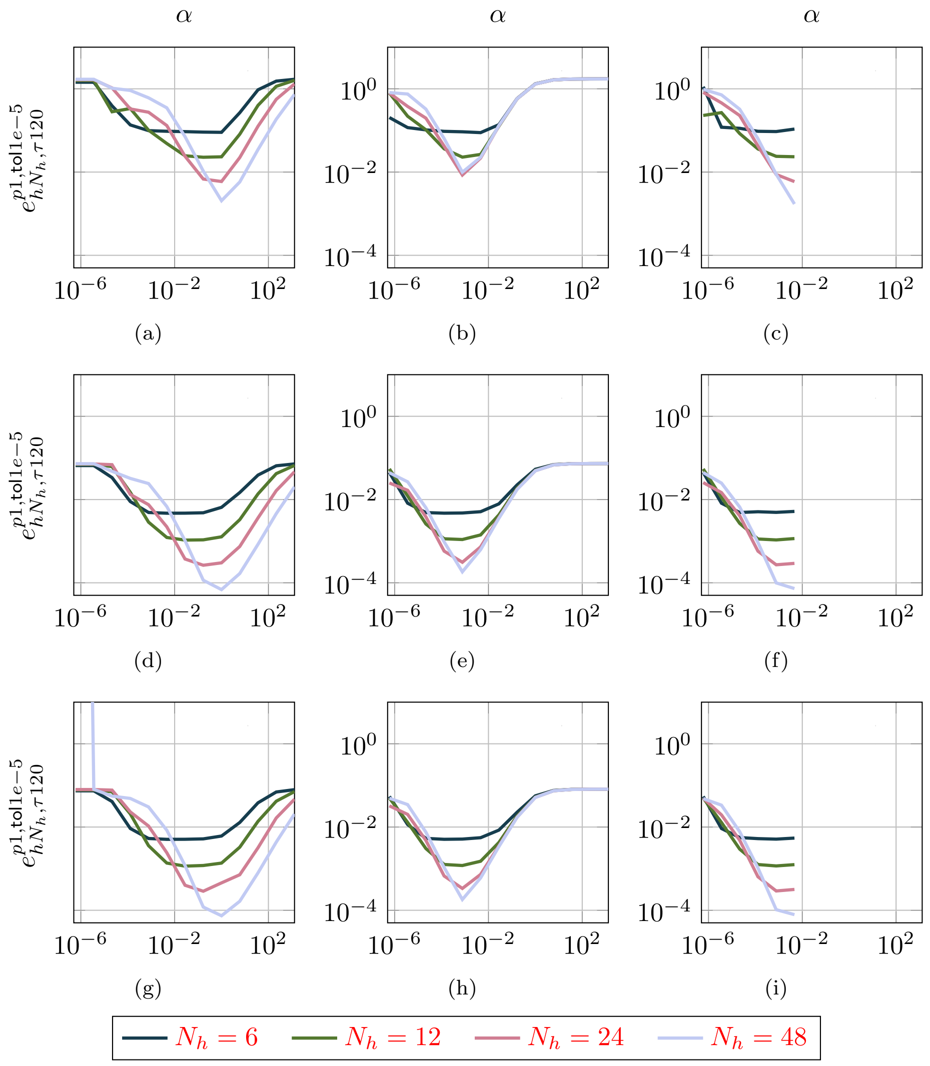
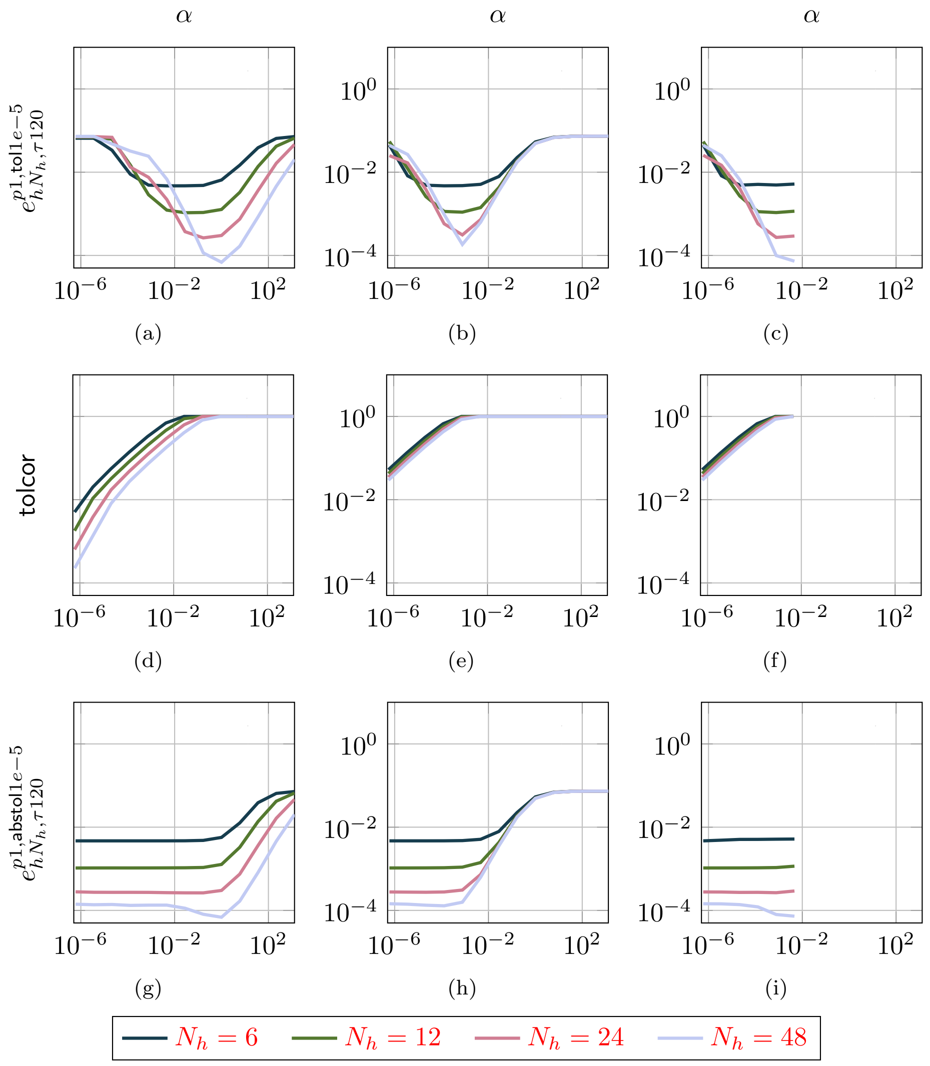


Figure 5: (Penalty schemes and inexact solves, cf. Section 5.3) The error $e^{p1,\mathsf{tol}1e-5}_{hN_h,\tau 120}$ for varying space discretizations N_h versus the penalization parameter α for the schemes pena (left), pero (middle), and nits (right), where the occurring algebraic equations are solved via *GMRES* up to an residual of 10^{-5}. The first row of plots (a-c) corresponds to Test Case 1, the middle row (d-f) to Test Case 2, and the bottom line (g-i) to Test Case 3.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 6: (Penalty schemes and absolute tolerances, cf. Section 5.3) The error $e^{p2,\mathsf{tol}1e-5}_{hN_h,\tau120}$ for a fixed relative residual $\mathsf{tol} = 1e-5$ (top row), the correction of the residual tolcor (middle row), and for a fixed absolute residual $\mathsf{abstol} = 10^{-5}$ (bottom row) for varying space discretizations N_h versus the penalization parameter α for the schemes pena (left), pero (middle), and nits (left) for Test Case 2.

$N_h \backslash N_\tau$	30	60	120
6	1.0000	0.9975	0.9975
12	0.2720	0.2579	0.2579
24	0.1064	0.0652	0.0651
48	0.0797	0.0172	0.0163
96	0.0766	0.0067	0.0041

$N_h \backslash N_\tau$	60	240	960
6	1.0000	0.9429	0.8810
12	0.3681	0.1049	0.1018
24	0.3488	0.0258	0.0124
48	0.3485	0.0218	0.0021

Table 3: (Time space convergence of lift with volume forcing, cf. Section 5.4) The approximation error $e^{p1}_{hN_h,\tau N_\tau}$ scaled by the inverse of $e^{p1}_{h6,\tau 30} = 9.7149 \cdot 10^{-2}$ for linear ansatz functions (left) and $e^{p1}_{hN_h,\tau N_\tau}$ scaled by the inverse of $e^{p2}_{h6,\tau 60} = 5.288 \cdot 10^{-3}$ for quadratic ansatz functions (right) with ρ_{ref} explicitly given for varying space and time discretizations for Test Case 1.

$N_h \backslash N_\tau$	30	60	120
6	1.0000	1.0773	0.9992
12	0.2744	0.2740	0.2740
24	0.0614	0.0610	0.0610
48	0.0153	0.0152	0.0152
96	0.0039	0.0038	0.0038

$N_h \backslash N_\tau$	60	120	240
6	1.0000	0.9998	0.9997
12	0.1175	0.1174	0.1174
24	0.0140	0.0139	0.0139
48	0.0022	0.0017	0.0017

Table 4: (Time space convergence of lift with volume forcing, cf. Section 5.4) The approximation error $e^{p1}_{hN_h,\tau N_\tau}$ scaled by the inverse of $e^{p1}_{h6,\tau 30} = 1.29 \cdot 10^{-4}$ for linear ansatz functions (left) and $e^{p1}_{hN_h,\tau N_\tau}$ scaled by the inverse of $e^{p2}_{h6,\tau 60} = 7.234 \cdot 10^{-6}$ for quadratic ansatz functions (right) with ρ_{ref} explicitly given for varying space and time discretizations for Test Case 2.

priori. In applications, however, the projection based scheme proj is not affected by such an inconsistency.

Having made the time discretization sufficiently accurate, we investigated the order of convergence of the space discretization for the different schemes. The estimated order of convergence was in between EOC = 2 and EOC = 2.5 which is not satisfactory. Similar tests but with a volume force led to an EOC = 3 the quadratic elements. This result suggests that boundary conditions are not treated optimally in the considered finite element schemes. Additionally, the results as a whole show that the *method of manufactured solutions* is not well suited for the numerical investigation of spatial convergence of boundary actuation driven setups.

The relaxed schemes showed the same accuracy as the consistent schemes, but only at certain ranges of the penalization parameter value. If one solves the algebraic equations with high accuracy, one only has to choose the penalization small enough. However, if the algebraic equations are solved iteratively up to a certain residual, then the approximation gets worse again for smaller penalization parameters. This effect might be partially due to an ill-conditioning of the system which might be cured by a suitable preconditioner. The main factor, however, is that for small penalization parameters α the residual is dominated by the penalization term. As a remedy one can consider absolute residuals as convergence criteria. Conversely, that means that one has to prescribe relative residuals that scale with α which is not practical for small α.

So far we have investigated the approximation quality but not the efficiency like the performance of iterative solvers applied within the various schemes.

A main motivation of the survey was that standard model reduction or optimal control approaches are readily applicable to systems of *distributed* type like (2). In a forthcoming paper, we will investigate how well the proposed formulations work in control setups. Also the consistency of the reformulations with the abstract equations is still open and subject to ongoing work.

References

[1] M. S. Alnaes, A. Logg, K.-A. Mardal, and O. Skavhaug. Unified framework for finite element assembly. *International Journal of Computational Science and Engineering*, 4(4):231–244, 2009.

[2] R. Altmann. Index reduction for operator differential-algebraic equations in elastodynamics. *Z. Angew. Math. Mech.*, 93(9):648–664, 2013.

[3] M. Badra and T. Takahashi. Stabilization of parabolic nonlinear systems with finite dimensional feedback or dynamical controllers: application to the Navier-Stokes system. *SIAM J. Cont. Optim.*, 49(2):420–463, 2011.

[4] F. Ben Belgacem, H. El Fekih, and J.-P. Raymond. A penalized Robin approach for solving a parabolic equation with nonsmooth Dirichlet boundary conditions. *Asymptotic Anal.*, 34(2):121–136, 2003.

[5] P. Benner and J. Heiland. LQG-Balanced Truncation low-order controller for stabilization of laminar flows. In R. King, editor, *Active Flow and Combustion Control 2014*, volume 127 of *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, pages 365–379. Springer, Berlin, 2015.

[6] A. Bensoussan, G. Da Prato, M. C. Delfour, and S. K. Mitter. *Representation and Control of Infinite-Dimensional Systems. Vol. I.* Birkhäuser, Basel, Switzerland, 1992.

[7] M. Berggren. Approximations of very weak solutions to boundary-value problems. *SIAM J. Numer. Anal.*, 42(2):860–877, 2004.

[8] D. Braess. *Finite Elemente. Theorie, schnelle Löser und Anwendungen in der Elastizitätstheorie.* Springer, Berlin, Germany, 4th revised and extended edition, 2007.

[9] E. Casas and J. Raymond. Error estimates for the numerical approximation of dirichlet boundary control for semilinear elliptic equations. *SIAM J. Cont. Optim.*, 45(5):1586–1611, 2006.

[10] J. C. García Orden and D. D. Dopico. On the stabilizing properties of energy-momentum integrators and coordinate projections for constrained mechanical systems. In J. C. García Orden, J. M. Goicolea, and J. Cuadrado, editors, *Multibody Dynamics*, volume 4 of *Computational Methods in Applied Sciences*, pages 49–67. Springer, Amsterdam, Netherlands, 2007.

[11] A. Gaul. Krypy – a Python toolbox of iterative solvers for linear systems, commit: 23500bc9. https://github.com/andrenarchy/krypy, 2014.

[12] V. Girault and P.-A. Raviart. *Finite Element Methods for Navier-Stokes Equations. Theory and Algorithms.* Springer, Berlin, Germany, 1986.

[13] M. D. Gunzburger and S. L. Hou. Treating inhomogeneous essential boundary conditions in finite element methods and the calculation of boundary stresses. *SIAM J. Numer. Anal.*, 29(2):390–424, 1992.

[14] E. Hairer, C. Lubich, and M. Roche. *The Numerical Solution of Differential-Algebraic Systems by Runge-Kutta Methods.* Springer, Berlin, Germany, 1989.

[15] J. Heiland. *Decoupling and optimization of differential-algebraic equations with application in flow control.* PhD thesis, TU Berlin, 2014. http://opus4.kobv.de/opus4-tuberlin/frontdoor/index/index/docId/5243.

[16] J. Heiland. tdpbcvals – Python module and test suite for convection-diffusion problems with time dependent dirichlet boundary conditions, v1.0. https://gitlab.mpi-magdeburg.mpg.de/heiland/timedp-bcvals, 2015.

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

[18] R. King (editor). *Active flow control. Papers contributed to the conference 'Active flow control 2006', Berlin, Germany, September 27–29, 2006.* Springer, Berlin, Germany, 2007.

[19] R. King (editor). *Active Flow Control II: Papers Contributed to the Conference 'Active Flow Control II 2010', Berlin, Germany, May 26–28, 2010.* Notes on Numerical Fluid Mechanics and Multidisciplinary Design. Springer, Berlin, Germany, 2010.

[20] M. Krstic and A. Smyshlyaev. *Boundary Control of PDEs. A Course on Backstepping Designs.* SIAM, Philadelphia, PA, 2008.

[21] K. Kunisch and B. Vexler. Constrained Dirichlet boundary control in L^2 for a class of evolution equations. *SIAM J. Cont. Optim.*, 46(5):1726–1753, 2007.

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

[23] D. Kuzmin and J. Hamalainen. *Finite Element Methods for Computational Fluid Dynamics: A Practical Guide*, volume 14. SIAM, Philadelphia, PA, 2014.

[24] I. Lasiecka and R. Triggiani. *Control Theory for Partial Differential Equations: Continuous and Approximation Theories I. Abstract Parabolic Systems.* Cambridge University Press, Cambridge, UK, 2000.

[25] J. L. Lions. *Optimal Control of Systems Governed by Partial Differential Equations.* Springer-Verlag, Berlin, Germany, 1971.

[26] A. Logg, K.-A. Mardal, and G. Wells, editors. *Automated Solution of Differential Equations by the Finite Element Method*, volume 84 of *Lect. Notes Comput. Sci. Eng.* Springer-Verlag, 1 edition, 2012.

[27] A. Logg and G. Wells. Dolfin: Automated finite element computing. *ACM Trans. Math. Softw.*, 37(2):417–444, 2010.

[28] S. May, R. Rannacher, and B. Vexler. Error analysis for a finite element approximation of elliptic Dirichlet boundary control problems. *SIAM J. Cont. Optim.*, 51(3):2585–2611, 2013.

[29] J. Nitsche. Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind. *Abh. Math. Semin. Univ. Hambg.*, 36(1):9–15, 1971.

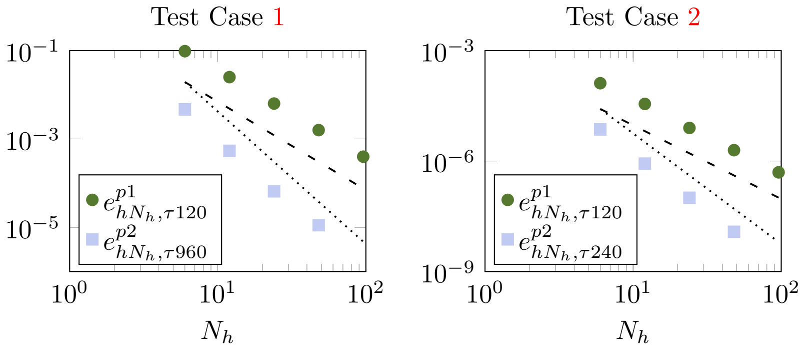
[30] R. Rannacher. On the numerical solution of the incompressible Navier-Stokes equations. *Z. Angew. Math. Mech.*, 73(9):203–216, 1993.

[31] J.-P. Raymond. Stokes and Navier-Stokes equations with nonhomogeneous boundary conditions. *Ann. Inst. H. Poincaré Anal. Non Linéaire*, 24(6):921 – 951, 2007.

[32] T. Roubíček. *Nonlinear Partial Differential Equations with Applications.* Birkhäuser, Basel, Switzerland, 2005.

[33] F. Schieweck. Uniformly stable mixed *hp*-finite elements on multilevel adaptive grids with hanging nodes. *ESAIM Math. Model. Numer. Anal.*, 42(3):493–505, 2008.

[34] F. Tröltzsch. *Optimale Steuerung partieller Differentialgleichungen.* Vieweg+Teubner, Wiesbaden, Germany, 2009.



(a)

(b)

Figure 7: (Spatial convergence with *manufactured solutions*, cf. Section 5.4) The error $e^{p\mathsf{cg}}_{hN_h,\tau N_\tau}$ for Test Case 1 (a) for Test Case 2 (b) for sufficiently fine time discretizations N_τ, for varying space discretizations N_h, and for linear and quadratic shape functions. The dashed lines indicate the slope of a quadratic convergence the dotted lines indicate a convergence of order 3.

Max Planck Institute Magdeburg Preprints