scienceOPENresearch SOR-MATH

Time-dependent Dirichlet conditions in finite element discretizations

Peter Benner and Jan Heiland*

Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstraße 1, 39106 Magdeburg, Germany
*Corresponding author's e-mail address: heiland@mpi-magdeburg.mpg.de

Published online: 6 October 2015 (version 1)
Cite as: P. Benner et al. ScienceOpen Research 2015 (DOI: 10.14293/S2199-1006.1.SOR-MATH.AV2JW3.v1)
Reviewing status: Please note that this article is under continuous review. For the current reviewing status and the latest referee's comments please click **here** or scan the QR code at the end of this article.

Primary discipline: Numerical methods
Secondary discipline: Numerical & Computational mathematics
Keywords: Finite Element Method, Boundary Conditions, Dirichlet Boundary Conditions, Variational Formulation, Boundary Control, FEM

ABSTRACT

For the modeling and 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.

INTRODUCTION

In practical applications, see [1,2] for examples on 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 modeling of control setups, cf. [3, 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., Refs. [4, Ch. 2] and [5, Ch. 2.3]. Another rather obvious obstacle is that a standard choice of trial and test function formulations implies a certain smoothness of the boundary data which may be impractical [5, Ch. 2.3].

For a general overview of the functional analysis for parabolic systems with Dirichlet boundary control, we refer to Refs. [4,6]. 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 Refs. [7–9], and time-dependent formulations are considered in Refs. [10,11]. The existence and the approximation of *very weak* solutions are well understood [7]. The key ingredient is the proper approximation of functions at the boundary via a projection [7,8,10].

An alternative approach of relaxing the boundary constraint via a penalization term in Robin boundary conditions has been investigated in Refs. [12,13].

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 Ref. [14] for an application example. The main criterion is 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 into 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 the 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 than what 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 Ref. [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 forthcoming paper.

This paper is organized as follows. In the section *Generic problem formulations*, we state the type of problems that we will consider both in an abstract setting and after a spatial discretization. In the section *Rewriting the spatially discretized equations*, we consider approaches that reformulate the spatially discretized equations into the desired form. In the section *Incorporation via variational formulations and their discretizations*, we discuss methods that reformulate the abstract equations such that a spatial discretization is a system of *distributed type* (1). In the section *Numerical tests*, we report on numerical tests concerning the approximation properties of the introduced methods. We conclude the paper by summarizing remarks and an outlook.

GENERIC PROBLEM FORMULATIONS

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

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 Ref. [16, 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. [17, Ch. 1.1], via

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

the space of bounded linear functionals on $\mathcal{Q}'$.
Let

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

be the trace operator as defined, e.g., in Ref. [17, Thm. 1.5].
We state the prototype of the continuous problem.

Problem 2.1. *Let $T > 0$ and consider $\mathcal{A} : (0,T) \times \mathcal{V} \to \mathcal{V}'$. For $\mathcal{F} \in L^2(0,T;\mathcal{V}')$, for $v_0 \in \mathcal{H}$, and $\mathcal{U} \in L^2(0,T;\mathcal{Q}')$, find v with $v(t) \in \mathcal{V}$ and $\dot{v}(t) \in \mathcal{V}'$, a.e. on $(0, T)$, so that*:

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

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

holds for almost all $t \in (0,T)$, and so that $v(0) = v_0$ in $\mathcal{H}$.

The system of abstract Equations (1) contains common weak formulations of PDEs that model physical phenomena, cf. [18]. We will not address time regularity here and, thus, leave the properties of the mappings $t \mapsto \mathcal{A}(t, v(t))$ and, e.g., $t \mapsto \dot{v}(t)$ 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 ρ due 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 $g(t) : \Gamma \to \mathbb{R}$ prescribing the boundary conditions, find a function ρ of space and time that satisfies*:

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

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

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

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

$$\begin{aligned}\langle \mathcal{A}(t, v(t)), \phi \rangle_{\mathcal{V}',\mathcal{V}} = &\int_\Omega (w \cdot \nabla v(t), \phi) + \nu(\nabla v(t), \nabla\phi) \, \mathrm{d}\omega \\ &- \int_\Gamma \nu\left(\frac{\partial v}{\partial n}(t), \phi\right) \mathrm{d}\gamma,\end{aligned} \tag{6}$$

for all $\phi \in \mathcal{V}$ and with $\partial/\partial n$ denoting the normal derivative. Here and in what follows, the pairing $(\cdot,\cdot)$ denotes the inner

product in the spaces under consideration. The boundary integral in Equation (6) is only well defined for sufficiently regular solutions and test functions. For $v(t)$, $\phi \in V = W^{1,2}(\Omega)$, it holds that $\frac{\partial v}{\partial n}(t)|_\Gamma \in W^{-\frac{1}{2},2}(\Gamma)$ and $\phi|_\Gamma \in W^{\frac{1}{2},2}(\Gamma)$, so that the term $\int_\Gamma \nu\left(\frac{\partial v}{\partial n}(t), \phi\right) \mathrm{d}\gamma$ is well defined as the continuous extension of the inner product in $L^2(\Gamma)$, see Ref. [17, Ch. 1.1]. Note that there are other possible choices for a weak formulation [11].

The boundary condition in Equation (4), viewed as a constraint, can also be incorporated using the dual operator of $\gamma : \mathcal{V} \to \mathcal{Q}'$ and a so-called *Lagrange multiplier*. Then, under certain smoothness and consistency conditions [19], Problem 2.1 is equivalent to:

Problem 2.3. *Let $T > 0$ and consider $\mathcal{A} : (0,T) \times \mathcal{V} \to \mathcal{V}'$. For $\mathcal{F} \in L^2(0,T;\mathcal{V}')$, $v_0 \in \mathcal{H}$, and $\mathcal{U} \in L^2(0,T;\mathcal{Q}')$, find v with $v(t) \in \mathcal{V}$ and $\dot{v}(t) \in \mathcal{V}'$ and Λ with $\Lambda(t) \in \mathcal{Q}$, a.e. on $(0, T)$, so that*:

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

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

hold for almost all $t \in (0,T)$, and so that $v(0) = v_0$ in $\mathcal{H}$.

Note that, in general, the Lagrangian multiplier resides in the dual space of the constraint. In the considered case, where $Q' = W^{\frac{1}{2},2}(\Gamma)$ is a Hilbert space, we can deliberately identify $Q'' = Q$.

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 $V_I = \operatorname{span}\{\psi_i\}_{i=1}^{n_I}$ is the space spanned by the basis functions that are associated with nodes in the inner and that are zero at the boundary. Accordingly, n_I is the number of nodes in the inner and $V_\Gamma \subset V$ is spanned by the basis functions $\{\psi_i\}_{i=n_I+1}^{n_v}$ that have nonzero values at the boundary. 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 Equation (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 is V_I. Otherwise, all boundary dofs that are not fixed by a Dirichlet condition are included in the test space.

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:

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

with respect to the test space,

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

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

$$M_I = [\,M_{II} \quad M_{I\Gamma}\,], \tag{9}$$

where

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

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.

Similarly, we define the discrete approximation $A : (0,T) \times \mathbb{R}^{n_v} \to \mathbb{R}^{n_v}$ to $\mathcal{A}$ as:

$$A(t, v) = [\,\langle \mathcal{A}(t, v), \psi_i\rangle_{\mathcal{V}',\mathcal{V}}\,]_{i=1,\dots,n_v}$$

and $A_I : (0,T) \times \mathbb{R}^{n_v} \to \mathbb{R}^{n_I}$ as its restriction to the test functions of the inner nodes, where, again, we have associated a vector $v \in \mathbb{R}^{n_v}$ with a function in V via (8). If $\mathcal{A}$ is linear, then its approximation on $A : (0,T) \times \mathbb{R}^{n_v} \to \mathbb{R}^{n_v}$ can be assembled as a matrix-valued function via:

$$A(t) = [\,\langle \mathcal{A}(t, \psi_j), \psi_i\rangle_{\mathcal{V}',\mathcal{V}}\,]_{i,j=1,\dots,n_v},$$

with the partitions A_I, A_{II}, and $A_{I\Gamma}$ as they were defined for M in Equation (9).

The discrete approximation $f : (0,T) \to \mathbb{R}^{n_v}$ to the right-hand side $\mathcal{F} : (0,T) \to \mathcal{V}'$ is given as:

$$f(t) = [\,\langle \mathcal{F}(t), \psi_i\rangle_{\mathcal{V}',\mathcal{V}}\,]_{i=1,\dots,n_v}.$$

We will not distinguish notationally between f and its restriction to the inner test functions.
To assign the boundary values, we simply assign the dofs associated with the corresponding boundaries via:

$$G\upsilon = u, \quad \text{where} \quad G = \begin{bmatrix} 0 & I \end{bmatrix} \in \mathbb{R}^{n_\upsilon - n_I, n_\upsilon} \tag{10}$$

and where $u \in \mathbb{R}^{n_\upsilon - n_I}$ is a vector that will contain the current value of the boundary control at the given locations. As defined in Equation (10), the operator G picks out the boundary dofs of a function $\upsilon \in V$ and assigns the control values. Note, however, that there are other discrete approximations to the trace operator γ that consider, e.g., the inner product in Q' or include suitable projections [7,8].
Thus, if we assume $\upsilon(t) \in V$ 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_υ, $n_I \in \mathbb{N}$, and $n_d := n_\upsilon - n_I$ and consider $A_I : (0,T) \times \mathbb{R}^{n_\upsilon} \to \mathbb{R}^{n_I}$, $G \in \mathbb{R}^{n_d, n_\upsilon}$, and $M_I \in \mathbb{R}^{n_I, n_\upsilon}$ as defined in the beginning of the section, Spatially discretized equations. For $f \in L^2(0,T;\mathbb{R}^{n_I})$, $\alpha \in \mathbb{R}^{n_\upsilon}$, and $u \in L^2(0,T;\mathbb{R}^{n_d})$ find υ with $\upsilon(t) : (0,T) \to \mathbb{R}^{n_\upsilon}$, so that*:

$$M_I \dot{\upsilon}(t) - A_I(t, \upsilon(t)) = f(t), \tag{11a}$$

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

hold for almost all $t \in (0,T)$ and $\upsilon(0) = \alpha$.
For the system of Problem 2.3 with the multiplier, a possible spatial discretization defines a differential equation considering also the boundary parts, cf. [19,20]. It generically takes the form:

Problem 2.5. *Let $T > 0$, n_υ, $n_I \in \mathbb{N}$, and $n_d := n_\upsilon - n_I$ and consider $A : (0,T) \times \mathbb{R}^{n_\upsilon} \to \mathbb{R}^{n_\upsilon}$, $G \in \mathbb{R}^{n_d, n_\upsilon}$, and $M \in \mathbb{R}^{n_\upsilon, n_\upsilon}$ as defined in the beginning of the section Spatially discretized equations. For $f \in L^2(0,T;\mathbb{R}^{n_\upsilon})$, $\alpha \in \mathbb{R}^{n_\upsilon}$, and $u \in L^2(0,T;\mathbb{R}^{n_d})$ find $\upsilon : (0,T) \to \mathbb{R}^{n_\upsilon}$ and $\lambda : (0,T) \to \mathbb{R}^{n_d}$, so that*:

$$M\dot{\upsilon}(t) - A(t, \upsilon(t)) - G^{\mathrm{T}}\lambda(t) = f(t), \tag{12a}$$

$$G\upsilon(t) = u(t), \tag{12b}$$

hold for almost all $t \in (0,T)$ and $\upsilon(0) = \alpha$.
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 $A_I \in \mathbb{R}^{n_I, n_\upsilon}$ and write (4) as:

$$M_I \dot{\upsilon}(t) - A_I \upsilon(t) = f(t) \tag{13a}$$

$$Gv(t) = u(t). \tag{13b}$$

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. Only the values at the boundaries may have a jump at $t = 0$, since consistency with the boundary conditions is not possible for an arbitrary input. This is in line with the infinite dimensional setting, where the solution is typically only continuous in $(t \to \mathcal{H})$, with $\mathcal{H} = L^2(\Omega)$, where boundary conditions do not play a role.

REWRITING THE SPATIALLY DISCRETIZED EQUATIONS

In this section, we consider the spatially discretized equations introduced in the section *Spatially discretized equations*. 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.

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 [21], does not simply lead to a system of the form (1).
Consider the linear formulation (6) of Problem the 2.4 with the assignment of the boundary conditions as in Equation (13b):

$$M_I \dot{v} - A_I \upsilon = f, \tag{14a}$$

$$G\upsilon = \upsilon_\Gamma = u, \tag{14b}$$

$$\upsilon(0) = \alpha. \tag{14c}$$

Then, with the partitioning of M_I and A_I as in Equation (9), the state equation reads:

$$\begin{bmatrix} M_{II} & M_{I\Gamma} \end{bmatrix} \begin{bmatrix} \dot{\upsilon}_I \\ \dot{\upsilon}_\Gamma \end{bmatrix} = A_{II}\upsilon_I + A_{I\Gamma}\upsilon_\Gamma + f$$

which, having inserted Equation (14b), gives:

$$M_{II}\dot{v}_I = A_{II}v_I + f + A_{I\Gamma}u - M_{I\Gamma}\dot{u}. \tag{15}$$

System (15) is not of the form (2) because of the appearance of $\dot{u}$.
Remark 3.1. One can define a new input as $\tilde{u} := u$ and consider the system:

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

This approach uses a so-called *dynamical controller* that is defined via a differential relation. As pointed out in Ref. [22], for a dynamical controller one can set the initial value to

zero to circumvent the expected inconsistencies mentioned in Remark 2.6.

Although *Rothe*'s method is out of consideration, since it leads to a sequence of algebraic equations rather than to a differential equation, it is worth mentioning that it implicitly approximates the time-derivative of the Dirichlet conditions as they appear in Equation (15).

Remark 3.2. Using *Rothe*'s method of discretizing Problem (2) in time, first, e.g., via using an explicit Euler scheme with a time step length τ, and subsequently discretizing in space, leads to systems of type:

$$M_I v^{k+1} = M_I v^k + \tau(A_I v^k + f^k),$$
$$G v^{k+1} = v_\Gamma^{k+1} = u^{k+1},$$

where the superscript k relates to the values of the functions at the kth time instance. If at the previous time instance $v_\Gamma^k = u^k$, then the direct assignment at the current instance gives:

$$M_{II} v_I^{k+1} = M_{II} v_I^k + \tau(A_{II} v_I^k + f^k) + \tau A_{I\Gamma} u^k - M_{I\Gamma}[u^{k+1} - u^k],$$

which can be seen as time-discrete approximation to Equation (15).

Lifting of the boundary conditions

These approaches base on a lifting $\tilde{v}$ that fulfills the boundary conditions for all time and the decoupling of the solution $v = y + \tilde{v}$, see [23] for an example with linearized Navier–Stokes equations.

We consider the linear time-invariant case (7) and assume that $f = 0$.

At time $t \in [0, T]$, we define a lifting as:

$$\tilde{v}(t) = \begin{bmatrix} \tilde{v}_I(t) \\ u(t) \end{bmatrix}. \tag{16}$$

Then, considering Equation (8) with $v = y + \tilde{v}$ and splitting A_I and M_I as in Equation (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:

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

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 (ODE):

$$M_{II}\dot{\hat{v}}_I = A_{II}\hat{v}_I + Bu, \quad \hat{v}_I(0) = \alpha_I + M_{II}^{-1} M_{I\Gamma} u(0), \tag{17}$$

with

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

The actual solution is easily retrieved from $\hat{v} = v_I + M_{II}^{-1} M_{I\Gamma} u$. Note, however, the dependency of the initial value on $u(0)$ in Equation (17).

Remark 3.3. The dependency of the initial value on $u(0)$ is due to the ansatz that assumes smoothness of $\tilde{v}$, which then extends to the boundary nodes. Accordingly, at the boundary, the initial value needs to be consistent with the control at time $t = 0$, cf. Remark 2.6.

This is not an issue in practical applications where the determination of a control law does not depend on the initial value for the state like it is the case in linear-quadratic optimal control.

Remark 3.4. We find it worth pointing out, that the system (17) does not depend on the choice of the lifting (16) 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 straightforward. 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 3.5. 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 every finer discretizations.

Remark 3.6. 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).

Incorporation of the boundary data via Lagrange multiplier

We consider the formulation of Problem 2.5:

$$M\dot{v}(t) - A(t, v(t)) - G^{\mathrm{T}}\lambda(t) = f(t), \tag{18a}$$

$$Gv(t) = u(t), \tag{18b}$$

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 Ref. [24]. 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.

Decoupling by projection. In the considered case, G has the form $G = [\,0 \quad I\,]$ and M is symmetric positive definite. Thus, we can define:

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

With this, system (8) can be equivalently [15] reformulated as $(v, \lambda) = (v_i + v_g, \lambda)$, where the transformed variables are the solutions of:

$$v_g = M^{-1}G^{\mathrm{T}}S^{-1}u, \tag{19}$$

$$\lambda = -Q^{-}A(v_g + v_i) - Q^{-}f - Q^{-}M\dot{v}_{\Gamma}, \tag{20}$$

and

$$\dot{v}_i - PM^{-1}A(v_i + M^{-1}G^{T}S^{-1}u) = PM^{-1}f. \tag{21}$$

Note that the differential Equation (21) is of type (1).

With $MP = P^{T}M$, in the linear case, we can write the differential equation for v_i as:

$$M\dot{v}_i - P^{\mathrm{T}}Av_i = P^{\mathrm{T}}f + P^{\mathrm{T}}Bu,$$

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

Remark 3.7. 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 3.8. The variable v_i has zero values at the boundary at all time. Thus, if one only considers the ODE (21) 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 (19).

Regularization via penalization. If one adds the term $\alpha\lambda(t)$, $0 < \alpha \ll 1$, to the left-hand side of Equation (18b), one can solve for the multiplier and eliminate it from the differential part:

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

This approach is known as *penalty scheme* and *pressure penalization* in the numerical integration of multibody and Navier–Stokes systems, respectively, cf., e.g., [25,26]. The method is straightforward to implement but comes with the need of a proper choice of the penalization parameter. The main difficulty is that a small parameter α not only increases the quality of the approximation of the constraints but also increases the stiffness of the resulting ODE.

INCORPORATION VIA VARIATIONAL FORMULATIONS AND THEIR DISCRETIZATIONS

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

$$\dot{v}(t) - \mathcal{A}(t, v(t)) + \frac{1}{\alpha}\lambda'(\gamma v(t) - \mathcal{U}(t)) = \mathcal{F}(t), \tag{22}$$

where $\lambda' : \mathcal{Q} \to \mathcal{V}'$ 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 the section *Penalized Robin* in this paper.

Ultra weak formulations

The basic concept of the *ultra weak* variational formulation is the transfer of smoothness requirements from the test space to the trial space. In numerical experiments, in a conforming discretization, this concept will require special finite element spaces that are not part of common finite element libraries. We will introduce the formulation and a nonconforming discretization suitable for a direct implementation.

Let $\Phi = W^{2,2}(\Omega) \cap W_0^{1,2}(\Omega)$ and consider the diffusion Equation (5) with $\beta = 0$. We call v an *ultra weak* solution if:

$$\int_{\Omega} (\dot{v}, \phi)\, \mathrm{d}\omega - \int_{\Omega} \nu(v, \Delta\phi)\ \mathrm{d}\omega = \langle f, \phi \rangle_{\Phi',\Phi} - \int_{\Gamma} \nu\left(g, \frac{\partial \phi}{\partial n}\right) \mathrm{d}\gamma \tag{23}$$

for all $\phi \in \Phi$, cf. [8]. The abstract Equation (23) indicates that a spatial discretization may lead to a system in the form of (2). The difficulty with a conforming approach, however, lies in the definition of matching test functions of high regularity with zero boundary values and suitable ansatz functions.

A possible approach is to drop the requirement that the test functions have zero boundary conditions and to consider:

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

which can be numerically approximated with standard finite element spaces of sufficiently high regularity.

With the nonconforming ansatz spaces $V \subset W_0^{1,2}(\Omega)$, an approximation to (23) that is readily realizable is given through $v \in V$ that satisfies:

$$\int_\Omega (\dot{v},\phi)\ \mathrm{d}\omega + \int_\Omega \nu(\nabla v,\nabla\phi)\ \mathrm{d}\omega = \int_\Omega (f,\phi)\ \mathrm{d}\omega - \nu\int_\Gamma \left(g,\frac{\partial\phi}{\partial n}\right)\ \mathrm{d}\gamma \tag{24}$$

for all $\phi \in V$, cf. [27, Ch. 5.2.1]. This approximation of the solution of the boundary value problem through functions with zero boundaries necessarily leads to a solution of L^2 regularity regardless of possibly higher regularity of the problem. We have observed this low regularity in experiments using explicit schemes for time integration. However, in the reported numerical tests that use implicit schemes, the discretization (24) leads to decent approximations.

The numerical approximation to *ultra weak* solutions of elliptic problems with boundary control as proposed in Ref. [7] uses a finite dimensional ansatz space $V \subset H^1(\Omega)$ and as the test space $W := V \cap H_0^1(\Omega)$, see Refs. [8,10] for the extension to parabolic problems. The elliptic case then reads, find $v \in V$, such that:

$$\int_\Omega \nu(\nabla v,\nabla\phi)\quad \mathrm{d}\omega = \int_\Omega (f,\phi)\quad \mathrm{d}\omega, \qquad \text{for all}\quad \phi \in V \cap H_0^1(\Omega), \tag{25a}$$

$$v = \Pi_V(u) \qquad \text{on}\quad \Gamma, \tag{25b}$$

where Π_V is the L^2 projection in $L^2(\Gamma)$ onto the grid induced by the triangulation that defines V. Using the spaces and formulations of (9) for a spatial discretization of a parabolic problem, one obtains a system that is the same as (7) apart from the appearance of the projector Π_V in the boundary term (14b). Anyways, the elimination of the boundary nodes will lead to a system like (15) that includes the time derivative $\dot{u}$ of the control. A discontinuous Galerkin ansatz for the time discretization as used in Ref. [10] includes $\dot{u}$ implicitly in the same way as *Rothe's* method, cf. Remark 3.2.

Remark 4.1. Since the known numerical approaches that base on (9) do not lead to systems of type (2) or (1), we did not consider them in the numerical experiments in this manuscript. However, the lifting, cf. the section *Lifting of the boundary conditions*, and the projection approach, cf. the section *Decoupling by projection*, readily apply to the formulation of the boundary term that includes the projector Π_V. The inclusion of the projection is necessary for well posedness of the Dirichlet control problem for the case of less regular boundary controls [7,8,10].

Nitsche variational formulation

A variant of the standard weak formulation of the pure diffusion, cf. (2) with $\beta = 0$, as proposed in Ref. [28] for the stationary Poisson equation reads:

$$\begin{aligned}\int_\Omega (\dot{v},\phi)\ \mathrm{d}\omega + \int_\Omega \nu(\nabla v,\nabla\phi)\ \mathrm{d}\omega - \int_\Gamma \nu\left(\frac{\partial v}{\partial n},\phi\right)\ \mathrm{d}\gamma \\ - \int_\Gamma \nu\left(v,\frac{\partial\phi}{\partial n}\right)\ \mathrm{d}\gamma + c_\gamma\int_\Gamma (v,\phi)\ \mathrm{d}\gamma \\ = \langle f,\phi\rangle_{\Phi',\Phi} - \int_\Gamma \nu\left(g,\frac{\partial\phi}{\partial n}\right)\ \mathrm{d}\gamma + c_\gamma\int_\Gamma (g,\phi)\ \mathrm{d}\gamma\end{aligned} \tag{26}$$

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

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

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 [28, Equation (12)].

For (26), a standard discrete formulation leads to an equation of type (2) with A and B explicitly given, see Ref. [29]. Cf. also [27, Ch. 5.2.2] where nonzero boundary values of y have been assumed.

Penalized Robin

If one approximates the Dirichlet conditions by a Robin-type condition:

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

with a parameter α that is intended to go to zero, then the boundary conditions are incorporated *naturally* in the weak

formulation of 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. [12] and the references therein.

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 modeling 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 (15) 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. the section *Direct assignment of the boundary dofs*
- lift – lifting of the boundary conditions via split mass matrix – cf. the section *Lifting of the boundary conditions*
- proj – incorporation of the constraint via Lagrange multiplier and projections – cf. the section *Decoupling by projection*
- pena – penalization of the constraint – cf. the section *Regularization via penalization*
- ncul – nonconforming approximation of *ultra week* solutions – cf. the section *Ultra weak formulations*
- nits – approximation via the *Nitsche* variational formulation – cf. the section *Nitsche variational formulation*
- pero – relaxation via Robin approximation – cf. the section *Penalized Robin*

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. Furthermore, we will investigate how the schemes perform if an iterative solver is applied.

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:

$$\begin{aligned} \rho|_\Gamma &= g(x)u(t) : \\ &= \frac{1}{2}\left(\sin\left(\pi x_0 + \frac{\pi}{2}\right) + 1\right)(\cos(2t + \pi) + 1). \end{aligned} \tag{27}$$

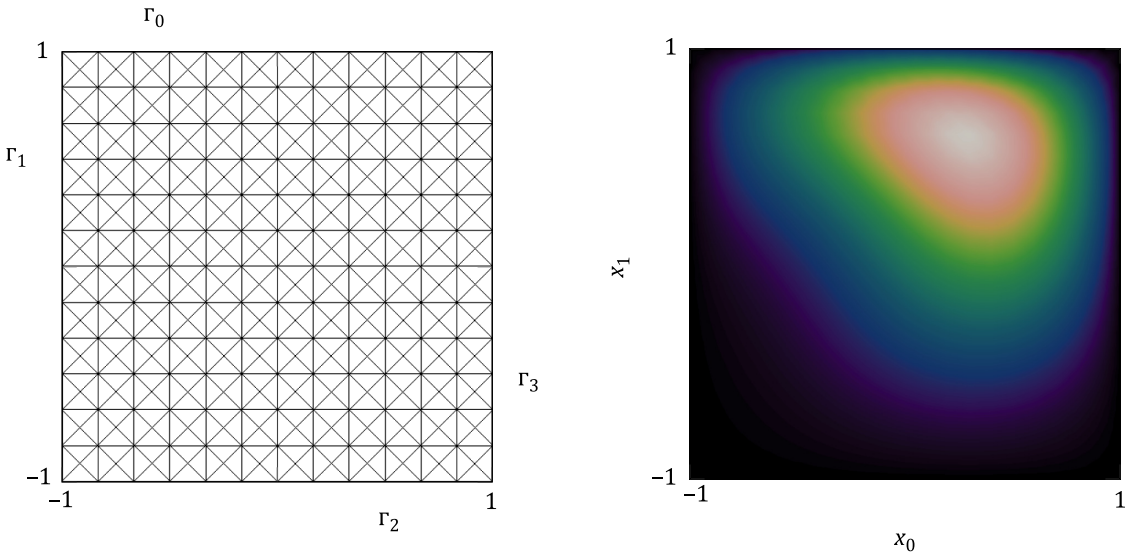

Figure 1. Illustration of the domain, the arrangement of the boundary segments, a triangulation with N_h = 12, and a snapshot of an approximation to the internal convection–diffusion as described in Test Case 1 at time t = 3.0.

Here, $g(x) := \frac{1}{2}\left(\sin\left(\pi x_0 + \frac{\pi}{2}\right) + 1\right)$ 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:

$$\dot{\rho}(t) + \beta_0 \cdot \nabla\rho(t) - \nu_0\Delta\rho(t) = 0, \tag{28a}$$

$$\rho|_{\Gamma_0}(t) = gu(t), \tag{28b}$$

$$\rho|_{\Gamma_1\cup\Gamma_2\Gamma_3}(t) = 0, \tag{28c}$$

$$\rho(0) = 0, \tag{28d}$$

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 2a for an illustration of the setup.

Test Case 2 (convection–diffusion). Given a convection wind and a diffusion parameter

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

find approximations to the scalar function ρ satisfying:

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

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 $0 \le \rho \le 1$ and negative elsewhere. Thus, for values of $\rho > 0$ the reaction pushes ρ towards $\rho = 1$, cf. Figure 2b.

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

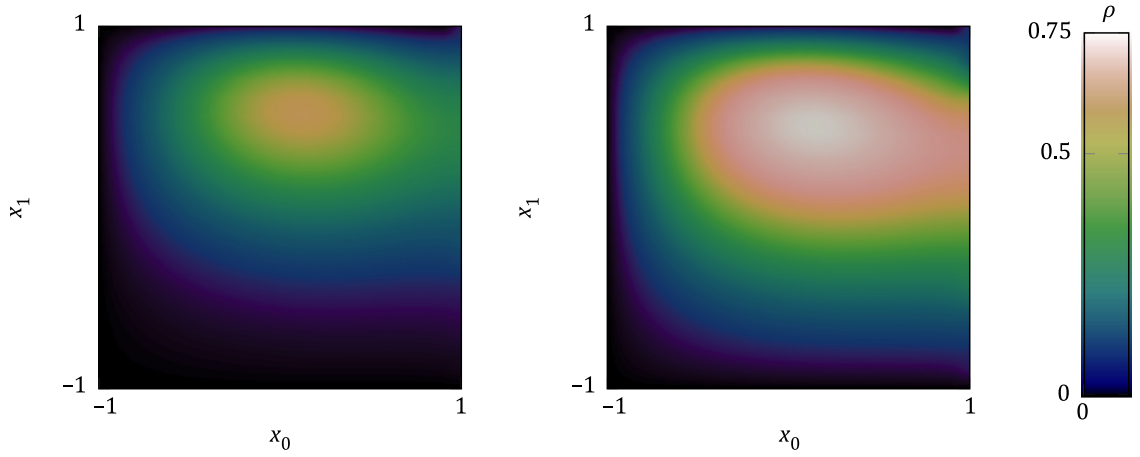

Figure 2. Illustration of the distribution of the scalar ρ seeded at the upper boundary after diffusion and convection (a) and additional reaction (b) as described in Test Cases 2 and 3 for $N_h = 15$ at time $t = 3.0$.

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

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

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

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 [30].

For the spatial discretization, we use the Python interface *dolfin* [31] to the finite element software suite *Fenics* [32]. 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 implicitly 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 error norm 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 generalized minimal residual method (*GMRES*) method using the implementation of the python module *krypy* [33]. The code used can be obtained from the author's public git repository [34].

By $\rho^{pcg}_{hN_h,\tau N_\tau}$, we denote the approximation to the solution of (10) with the discretization parameters N_h, N_τ, and *cg*. By $e^{pcg}_{hN_h,\tau N_\tau}$, we denote the approximation error

$$e^{pcg}_{hN_h,\tau N_\tau} := \rho^{pcg}_{hN_h,\tau N_\tau} - \rho_{\text{ref}}$$

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 $cg = 2$ scheme with $N_\tau = 240$ and $N_h = 96$.

Table 1. (Time space convergence of *dias* for linear elements, cf. the section *Convergence tests*) The approximation error $e^{pcg}_{hNh,\tau N\tau}$ with $\rho_{\text{ref}} = \rho^{p2}_{h96,\tau120}$ scaled by the inverse of $e^{p1}_{h6,\tau30} = 1.119\cdot10^{-1}$ (top) and $e^{p2}_{h6,\tau60} = 3.201\cdot10^{-2}$ (bottom) for varying space and time discretizations and for polynomial degree $cg = 1$ (top) and $cg = 2$ (bottom) for Test Case 1. Cf. also Figure 3a and 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	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

Convergence tests

In Tables 1 and 2, we list the approximation errors for dias 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 $e^{pcg}_{hNh,\tau120}$ for a fixed time discretization and varying space discretizations are plotted in Figure 3 for all three test cases. From the plots, one can see that the equivalent

Table 2. (Time space convergence of dias for quadratic elements, cf. the section *Convergence tests*) the approximation error $e^{pcg}_{hNh,\tau N\tau}$ with $\rho_{\text{ref}} = \rho^{p2}_{h96,\tau120}$ scaled by the inverse of $e^{p1}_{h6,\tau30} = 4.349\cdot10^{-4}$ (top) and $e^{p2}_{h6,\tau60} = 8.551\cdot10^{-05}$ (bottom) for varying space and time discretizations and for polynomial degree $cg = 1$ (top) and $cg = 2$ (bottom) for Test Case 2. Cf. also Figure 3c and d 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

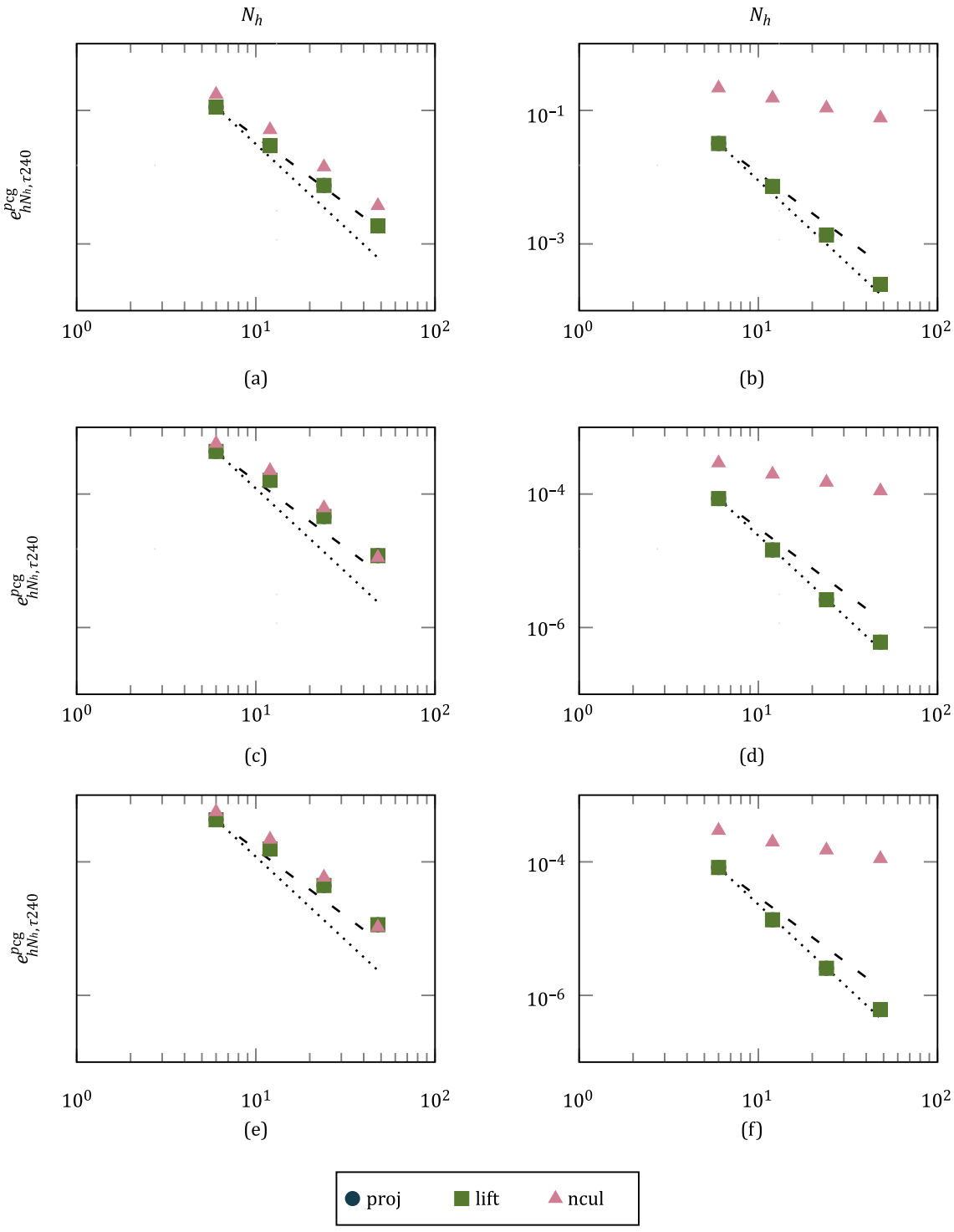

Figure 3. Convergence tests for the consistent implementations, cf. the section *Convergence tests*. The error $e^{pcg}_{hN_h,\tau120}$ for varying space discretizations N_h and for linear (left) and quadratic (right) shape functions. The first row of plots (a and b) corresponds to Test Case 1, the middle row (c and d) to Test Case 2, and the bottom line (e and 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.

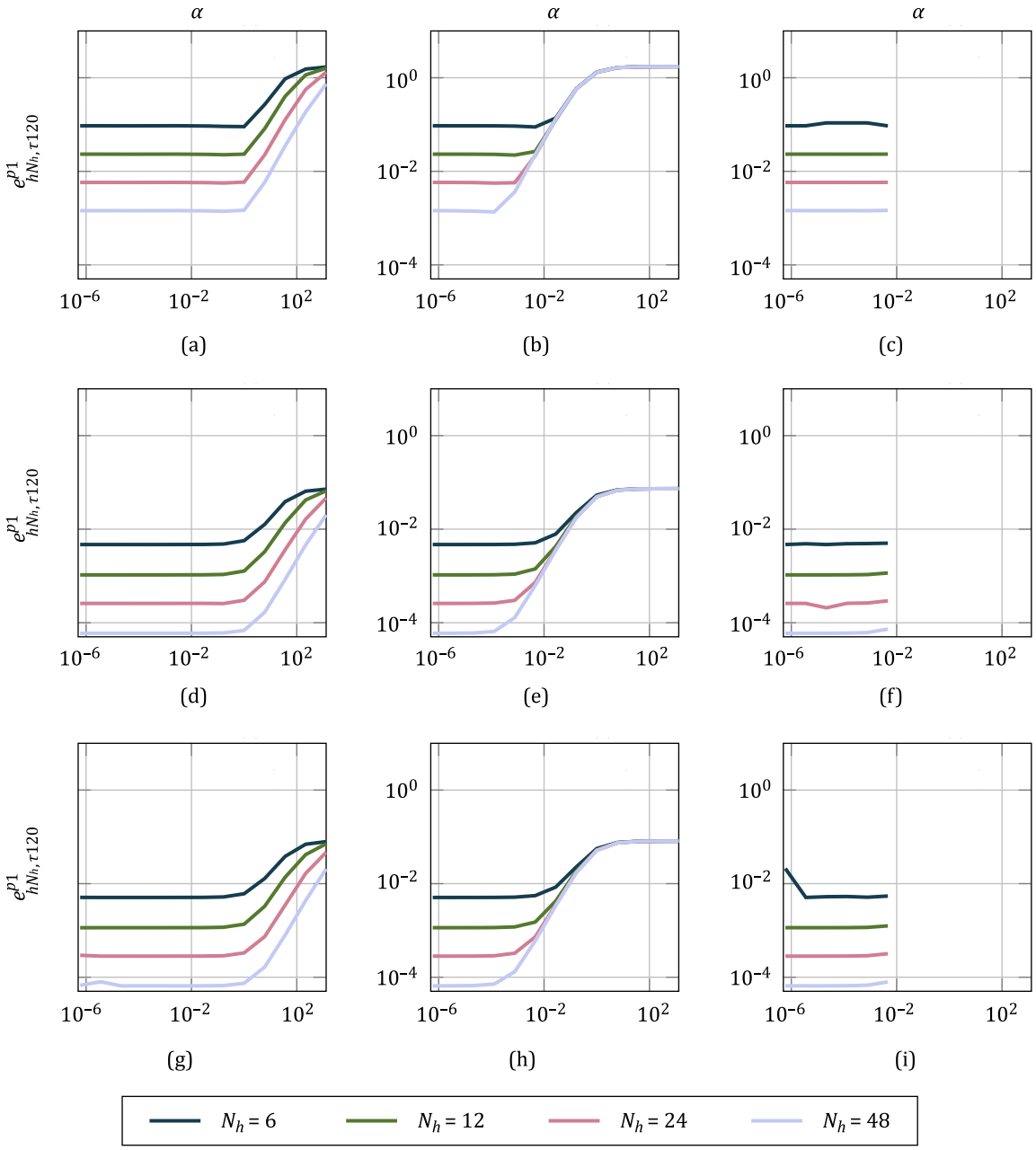

Figure 4. Penalization parameter study, cf. the section *Parameter studies for the penalty schemes*. The error $e^{p1}_{hN_h,\tau120}$ for varying space discretizations N_h versus the penalization parameter α for the schemes pena (left), pero (middle), and nits (right). The first row of plots (a–c) corresponds to Test Case 1, the middle row (d–f) to Test Case 2, and the bottom line (g and h) to Test Case 3.

formulations lift and proj give the same results and one can read off the numerically estimated order of spatial convergence (EOC). For the linear elements ($cg = 1$), one obtains $EOC = 2$ and for the quadratic elements ($cg = 2$), one obtains $EOC = 2.5$ at a lower error level. The observed order of convergence is not optimal as laid out in the section *Convergence tests with volume forcing*. For the linear elements, also ncul converges quadratically although with an error that is slightly larger than the one reported for proj and lift.

For piecewise quadratic shape functions, the scheme ncul delivered good approximations but its convergence rate was estimated as $EOC = 0.5$, see Figure 3 (right column), which is much less than expected from theory. A possible explanation for this breakdown is the oscillation that occurs when approximating a step function by a quadratic polynomial. Recall that the scheme ncul enforces a zero value at the boundary, while in the inner it approximates a solution which is not zero at the boundary. The inevitable jump in the

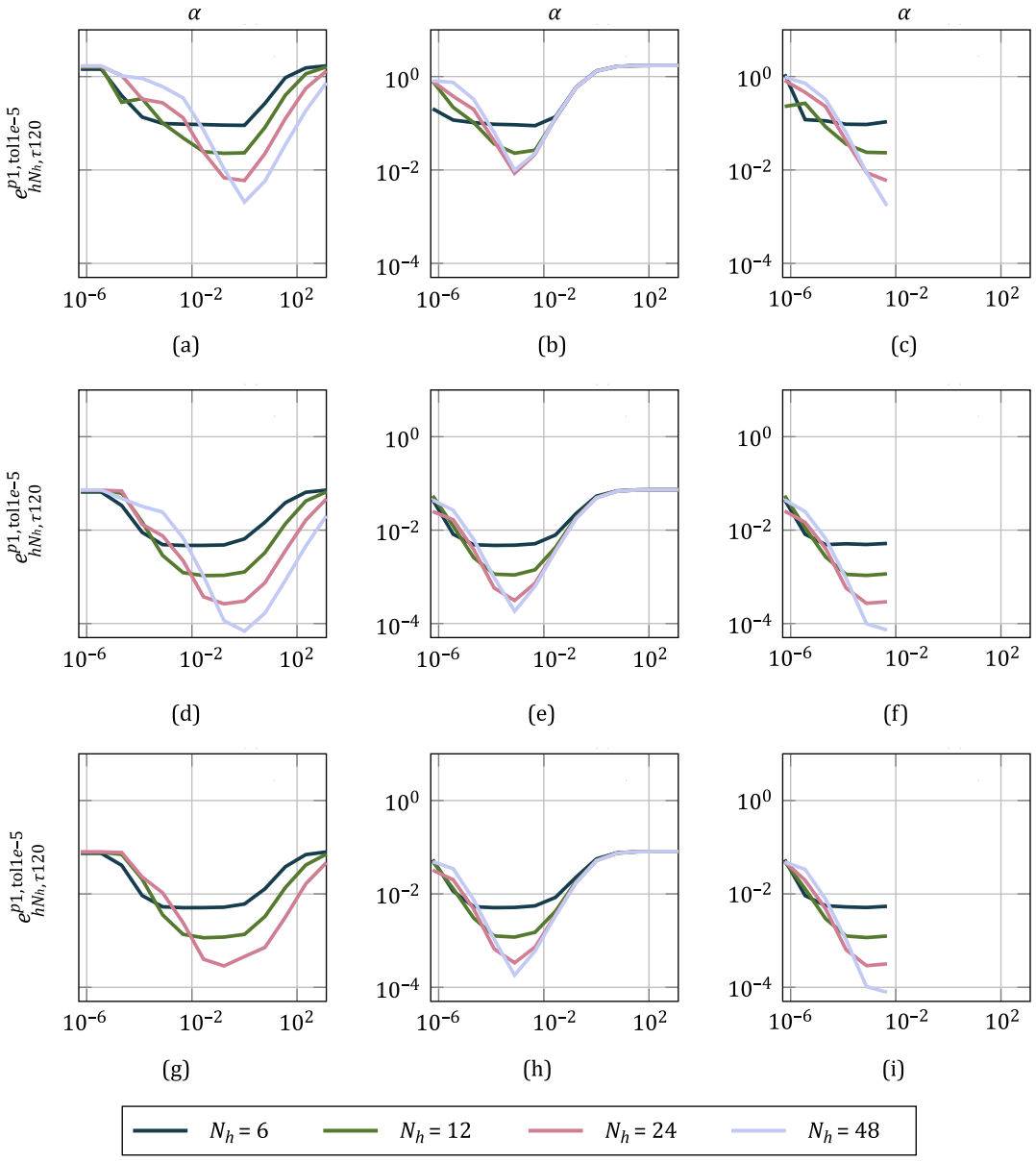

Figure 5. Penalty schemes and inexact solves, cf. the section *Parameter studies for the penalty schemes*. The error $e^{p1,tol1e-5}_{hN_h,\tau 120}$ for varying space discretizations N_h versus the penalization parameter α for the schemes *pena* (left), *pero* (middle), and *nits* (right), where the 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.

solution approximation is seemingly well captured by linear but not by quadratic elements.

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 (26).

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. The *Nitsche* method *nits* did not lead to reasonable approximations for large values of a.

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* preconditioned with the inverse of the mass matrix, to solve the

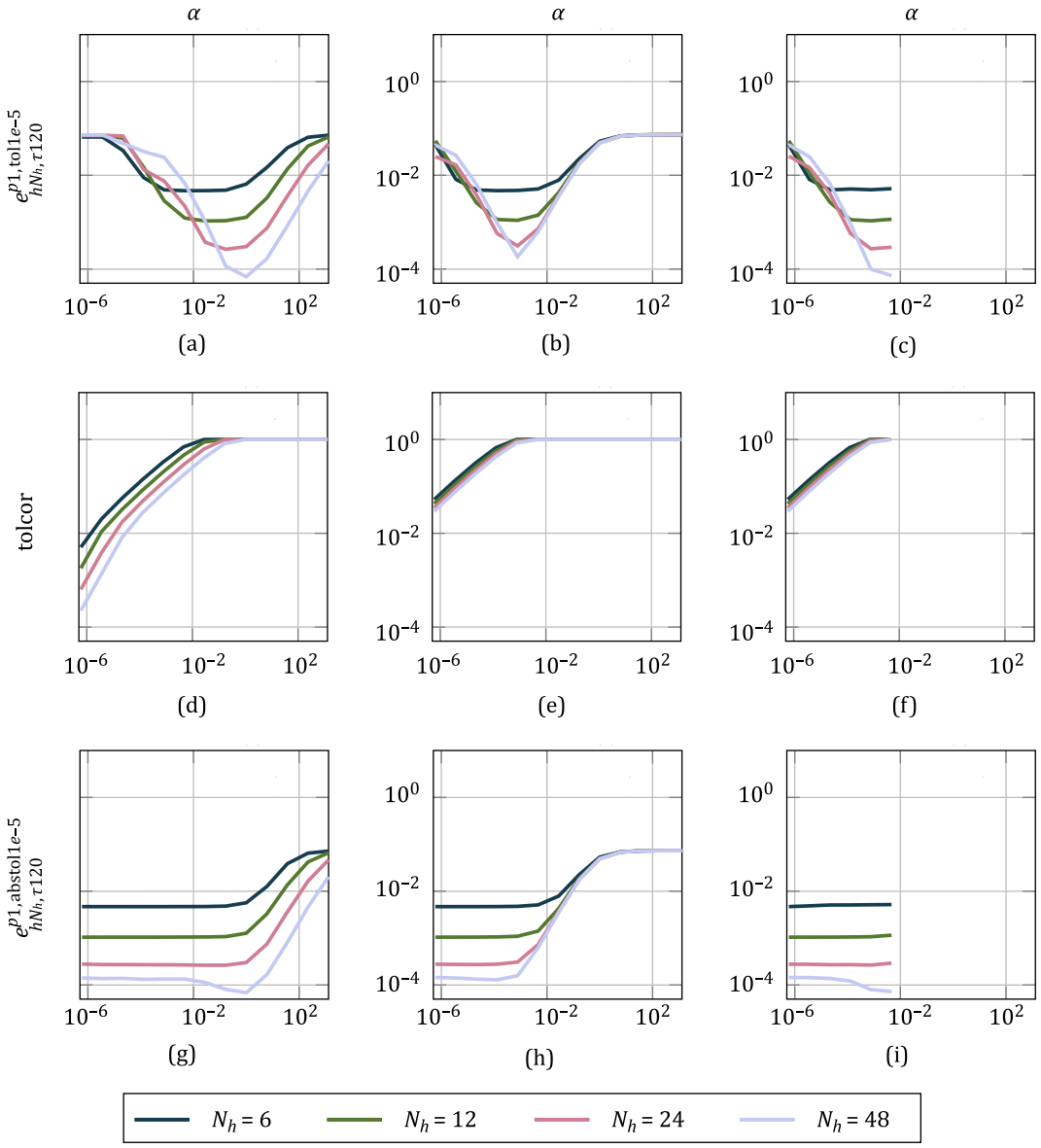

Figure 6. Penalty schemes and absolute tolerances, cf. the section *Parameter studies for the penalty schemes*. The error $e^{p2,tol1e-5}_{hN_h,\tau 120}$ for a fixed relative residual $tol = 10^{-5}$ (top row), the correction of the residual tolcor (middle row), and for a fixed absolute residual $abstol = 10^{-5}$ (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.

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 at most $tol = 10^{-5}$, 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 $Ax = f$ up to a relative residual of *tol*, one has that:

$$\frac{\|Ax - f\|}{\|f\|} = \mathrm{tol} \quad \text{or} \quad \|Ax - f\| = \mathrm{tol} \cdot \|f\|,$$

which means that for larger right-hand sides f, the absolute residual $\|Ax - f\|$ can be larger. A remedy is to control the absolute residual which can be done by correcting the provided relative residual by a factor $\mathrm{tolcor}=\min\left\{\frac{1}{\|f\|}, 1\right\}$, where f denotes the current right-hand side. In Figure 6d–f,

Table 3. Performance of *GMRES* within various formulations, see section *GMRES performance*. The averaged number of iterations per time-step *av.♯its* and the approximation error for several methods and all three test cases for $N_h = 48$, $N_\tau = 120$, and linear elements ($cg = 1$) in the case that the resulting linear equations are solved using *GMRES* up to a relative residual of $tol = 10^{-7}$.

	Test Case 1		Test Case 2		Test Case 3	
	$av.\sharp its$	$e^{p1,tol1e-7}_{h48,\tau120}$	$av.\sharp its$	$e^{p1,tol1e-7}_{h48,\tau120}$	$av.\sharp its$	$e^{p1,tol1e-7}_{h48,\tau120}$
proj	41.7	$1.9\cdot10^{-3}$	10.6	$1.2\cdot10^{-5}$	10.6	$1.1\cdot10^{-5}$
lift	41.7	$1.9\cdot10^{-3}$	10.6	$1.2\cdot10^{-5}$	10.6	$1.1\cdot10^{-5}$
pero	60.8	$4.0\cdot10^{-3}$	14.2	$7.8\cdot10^{-6}$	14.2	$7.6\cdot10^{-6}$
pena	48.9	$9.7\cdot10^{-3}$	15.5	$1.2\cdot10^{-5}$	15.5	$1.1\cdot10^{-5}$
ncul	43.2	$3.7\cdot10^{-3}$	10.6	$1.1\cdot10^{-5}$	10.6	$1.1\cdot10^{-5}$

The colored cells contain the lowest measured values.

Table 4. (Performance of *GMRES* within various formulations, see section *GMRES performance*) The averaged number of iterations per time-step *av.♯its* and the approximation error for several methods and all three test cases for $N_h = 48$, $N_\tau = 120$, and quadratic elements ($cg = 2$) in the case that the resulting linear equations are solved using *GMRES* up to a relative residual of $tol = 10^{-7}$.

	Test Case 1		Test Case 2		Test Case 3	
	$av.\sharp its$	$e^{p2,tol1e-7}_{h48,\tau120}$	$av.\sharp its$	$e^{p2,tol1e-7}_{h48,\tau120}$	$av.\sharp its$	$e^{p2,tol1e-7}_{h48,\tau120}$
proj	83.4	$2.5\cdot10^{-4}$	20.1	$6.3\cdot10^{-7}$	20.1	$6.1\cdot10^{-7}$
lift	83.4	$2.5\cdot10^{-4}$	20.1	$6.0\cdot10^{-7}$	20.1	$6.1\cdot10^{-7}$
pero	106.7	$4.6\cdot10^{-3}$	24.9	$1.1\cdot10^{-5}$	24.9	$1.1\cdot10^{-5}$
pena	71.2	$3.8\cdot10^{-2}$	20.6	$4.2\cdot10^{-6}$	20.6	$4.4\cdot10^{-6}$
ncul	84.9	$7.7\cdot10^{-2}$	20.3	$1.1\cdot10^{-4}$	20.3	$1.1\cdot10^{-4}$

The colored cells contain the lowest measured values.

we have reported the discrete $L^2(0, 0.2)$ norm of *tolcor* for Test Case 2. Applying this correction, that scales with $\frac{1}{\alpha'}$, one recovers the approximation properties of exact solves over the whole range of α, cf. Figures 6g– i and 4d–f.

GMRES performance

In this test setup, we investigated how the different but mainly equivalent formulations of the same problem affect the performance of an iterative solver. Therefore, we fixed the time and space discretization $N_h = 48$ and $N_\tau = 120$ and, for polynomial degrees $\mathrm{cg} = 1, 2$, we considered the simulations of Test Case 1, 2, and 3 if the resulting linear equations are solved using *GMRES* up to a relative residual smaller than $tol = 10^{-7}$. The results are listed in Tables 3 and 4.

The residuals were calculated in the inner product induced by the inverse of the mass matrices which was achieved by using the inverses as a preconditioner. At each timestep, as initial guesses, we took the values obtained by linear extrapolation on the bases of the two latest values. The parameter α was set to $\alpha = 1$ for pena and $\alpha = 10^{-3}$ for pero which corresponds to the optimal values for the $cg = 1$ case, cf. Figure 5.

As a performance measure that is comparatively independent of the sophistication of the implementation, we took the averaged numbers of iteration per timestep that were needed to obtain a residual below $tol = 10^{-7}$. A second quality measure was the resulting approximation error with respect to the reference solution. In all tests, the methods proj and lift took the least number of iterations. In some cases, in terms of approximation quality, they were outperformed by *pero*, but at the price of significantly more necessary iterations. The scheme ncul performs similar to proj and lift for $cg = 1$. For $cg = 2$ the approximation was much worse as it was already observed in Ref. [22]. At almost all tests, the penalization schemes needed more iterations and lead to worse approximations if compared to the consistent schemes. Note, however, that the choice of the penalization parameters was certainly not optimal for the $cg = 2$ cases.

Convergence tests with volume forcing

In the beginning of the section *Numerical tests*, we have mentioned that the method of *manufactured solutions* is not suitable for boundary controlled processes. This is intuitively clear since for every 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 Cases 1 and 2 (see the section *Test setups*) but with an additional volume force in Equation (10a) corresponding to the constructed solution:

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

with u as in Equation (27). The solution ρ_{ref} is constructed such that at Γ_0 it coincides with the boundary control function defined in Equation (27) and such that it is zero at the remaining boundaries. Also, it holds that $\frac{\partial \rho_{\mathrm{ref}}}{\partial \nu}|_{\Gamma_3} = 0$ 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 $EOC = 2$, i.e., doubling N_h reduces the error by a factor of 2^{-2}. For quadratic elements, we find $EOC = 3$, i.e., doubling N_h reduces the error by a factor of 2^{-3}, cf. Tables 5 and 6, 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 3b and d. This indicates that the boundary conditions are not

Table 5. (Time space convergence of lift with volume forcing, cf. section *Convergence tests*) The approximation error $e^{p1}_{hNh,\tau N\tau}$ scaled by the inverse of $e^{p1}_{h6,\tau 30} = 9.7149\cdot 10^{-2}$ for linear ansatz functions (top) and $e^{p1}_{hNh,\tau N\tau}$ scaled by the inverse of $e^{p2}_{h6,\tau 60} = 5.288\cdot 10^{-3}$ for quadratic ansatz functions (bottom) 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	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 6. (Time space convergence of lift with volume forcing, cf. section *Convergence tests*) The approximation error $e^{p1}_{hNh,\tau N\tau}$ scaled by the inverse of $e^{p1}_{h6,\tau 30} = 1.29\cdot 10^{-4}$ for linear ansatz functions (top) and $e^{p1}_{hNh,\tau N\tau}$ scaled by the inverse of $e^{p2}_{h6,\tau 60} = 7.234\cdot 10^{-6}$ for quadratic ansatz functions (bottom) with ρ_{ref} explicitly given for varying space and time discretizations for Test Case 2.

$N_h \backslash N_\tau$	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

optimally considered by standard discretization schemes. Moreover, this insufficiency is not captured by numerical tests with systems that are driven by a volume force.

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.

Using a reference solution on a fine discretization, 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-driven problems are not treated optimally in the considered finite element schemes. A numerical analysis would be needed to detect the source of the breakdown and to find remedies like, maybe, the *boundary concentrated* Finite Element approximation [35]. Apart from that, 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

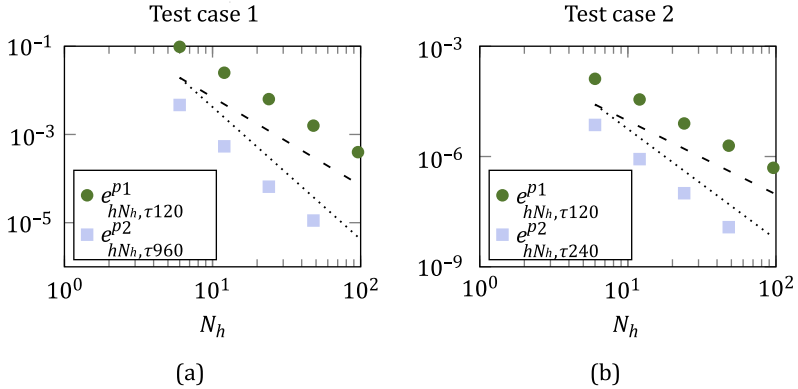

Figure 7. Spatial convergence with *manufactured solutions*, cf. section *Convergence tests*. The error $e^{pcg}_{hN_h,\tau N_\tau}$ for Test Case 1 (a) for Test Case 2 (b) for sufficiently fine time discretizations N_τ, for varying space discretizations N_h, and for linear and quadratic shape functions.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

prescribe relative residuals that scale with α which is not practical for small α.

In addition to the approximation quality, we have investigated the performance of *GMRES* applied within the various schemes. In these tests, as expected, the consistent schemes outperformed the schemes with a penalization.

Based on the results, depending on the situation, we speak out in favor of certain methods as follows. In view of minimal effort for implementation, pero and ncul are the methods of choice. If one wants to invest some time in implementation, proj and lift are better choices since they provide reliable approximations independent of parameters and for higher-order elements and they perform better in iterative schemes. If one can afford the incorporation of the projector, in particular for optimization, proj might be preferable over lift since a possibly inconsistent initial value is not an issue here. 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] King R, editor. Active flow control. Papers contributed to the conference 'Active flow control 2006'; 2006 Sep 27–29; Berlin (Germany): Springer; 2007.

[2] King R, editor. Active flow control II: Papers contributed to the conference 'Active Flow Control II 2010'; 2010 May 26–28. Notes on numerical fluid mechanics and multidisciplinary design. Berlin (Germany): Springer; 2010.

[3] Krstic M, Smyshlyaev A. Boundary control of PDEs. A course on Backstepping designs. Philadelphia (PA): SIAM; 2008.

[4] Bensoussan A, Da Prato G, Delfour MC, Mitter SK. Representation and control of infinite-dimensional systems. Vol. I. Basel (Switzerland): Birkhäuser; 1992.

[5] Tröltzsch F. Optimale Steuerung partieller Differentialgleichungen. Wiesbaden (Germany): Vieweg+Teubner; 2009.

[6] Lions JL. Optimal control of systems governed by partial differential equations. Berlin (Germany): Springer; 1971.

[7] Berggren M. Approximations of very weak solutions to boundary-value problems. SIAM J Numer Anal. 2004;42(2):860–77.

[8] Hinze M, Pinnau R, Ulbrich M, Ulbrich S. Optimization with PDE constraints. Dordrecht (Netherlands): Springer; 2009.

[9] May S, Rannacher R, Vexler B. Error analysis for a finite element approximation of elliptic Dirichlet boundary control problems. SIAM J Control Optim. 2013;51(3):2585–611.

[10] Gong W, Hinze M, Zhou Z. A finite element method for Dirichlet boundary control problems governed by parabolic PDEs. ArXiv e-prints, Oct. 2014. Available from: http://arxiv.org/abs/1410.0136

[11] Kunisch K, Vexler B. Constrained Dirichlet boundary control in L^2 for a class of evolution equations. SIAM J Control Optim. 2007;46(5):1726–53.

[12] Belgacem FB, Fekih HE, Raymond J-P. A penalized Robin approach for solving a parabolic equation with nonsmooth Dirichlet boundary conditions. Asymptotic Anal. 2003;34(2):121–36.

[13] Casas E, Raymond J. Error estimates for the numerical approximation of Dirichlet boundary control for semilinear elliptic equations. SIAM J Control Optim. 2006;45(5):1586–611.

[14] Benner P, Heiland J. LQG-balanced truncation low-order controller for stabilization of laminar flows. In: King R, editor. Active flow and combustion control 2014, Volume 127 of notes on numerical fluid mechanics and multidisciplinary design. Berlin: Springer; 2015. p. 365–379.

[15] Heiland J. Decoupling and optimization of differential-algebraic equations with application in flow control (PhD thesis). Technical University of Berlin; 2014. Available from: http://opus4.kobv.de/opus4-tuberlin/frontdoor/index/index/docId/5243

[16] Braess D. Finite Elemente. Theorie, schnelle Löser und Anwendungen in der Elastizitätstheorie. 4th revised and extended edition. Berlin (Germany): Springer; 2007.

[17] Girault V, Raviart P-A. Finite element methods for Navier–Stokes equations theory and algorithms. Berlin (Germany): Springer; 1986.

[18] Roubíček T. Nonlinear partial differential equations with applications. Basel (Switzerland): Birkhäuser; 2005.

[19] Gunzburger MD, Hou SL. Treating inhomogeneous essential boundary conditions in finite element methods and the calculation of boundary stresses. SIAM J Numer Anal. 1992;29(2):390–424.

[20] Altmann R. Index reduction for operator differential-algebraic equations in elastodynamics. Z Angew Math Mech. 2013;93(9):648–64.

[21] Logg A, Mardal K-A, Wells G, editors. Automated solution of differential equations by the finite element method, Volume 84 of Lect. Notes Comput Sci Eng. 1st edition. Springer-Verlag; 2012.

[22] Badra M, Takahashi T. Stabilization of parabolic nonlinear systems with finite dimensional feedback or dynamical controllers: application to the Navier–Stokes system. SIAM J Control Optim. 2011;49(2):420–63.

[23] Raymond J-P. Stokes and Navier–Stokes equations with nonhomogeneous boundary conditions. Ann Inst H Poincaré Anal Non Linéaire. 2007;24(6):921–51.

[24] Hairer E, Lubich C, Roche M. The numerical solution of differential-algebraic systems by Runge-Kutta methods. Berlin (Germany): Springer; 1989.

[25] García Orden JC, Dopico DD. On the stabilizing properties of energy-momentum integrators and coordinate projections for constrained mechanical systems. In: García Orden JC, Goicolea JM, Cuadrado J, editors. Multibody dynamics, Volume 4 of computational methods in applied sciences. Amsterdam (Netherlands): Springer; 2007. p. 49–67.

[26] Rannacher R. On the numerical solution of the incompressible Navier–Stokes equations. Z Angew Math Mech. 1993;73(9):203–16.

[27] Lasiecka I, Triggiani R. Control theory for partial differential equations: continuous and approximation theories I. Abstract parabolic systems. Cambridge (UK): Cambridge University Press; 2000.

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

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

[30] Kuzmin D, Hamalainen J. Finite element methods for computational fluid dynamics: a practical guide, Volume 14. Philadelphia (PA): SIAM; 2014.

[31] Logg A, Wells G. Dolfin: automated finite element computing. ACM Trans Math Softw. 2010;37(2):417–44.

[32] Alnaes MS, Logg A, Mardal K-A, Skavhaug O. Unified framework for finite element assembly. Int J Comput Sci Eng. 2009;4(4): 231–44.
[33] Gaul A. Krypy – a python toolbox of iterative solvers for linear systems, commit: 23500bc9; 2014. Available from: https://github.com/andrenarchy/krypy
[34] Heiland J. tdpbcvals – python module and test suite for convection-diffusion problems with time dependent Dirichlet boundary conditions, v2.0. 2015. Available from: https://gitlab.mpi-magdeburg.mpg.de/heiland/timedp-bcvals
[35] Khoromskij BN, Melenk JM. Boundary concentrated finite element methods. SIAM J Numer Anal. 2003;41(1):1–36.

COMPETING INTERESTS

The authors declare no competing interests.

PUBLISHING NOTES

© 2015 P. Benner and J. Heiland. This work has been published open access under Creative Commons Attribution License **CC BY 4.0**, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. Conditions, terms of use and publishing policy can be found at **www.scienceopen.com**.

Please note that this article may not have been peer reviewed yet and is under continuous post-publication peer review. For the current reviewing status please click **here** or scan the QR code on the right.

scienceOPEN.com
research+publishing network