

A non-conforming composite quadrilateral finite element pair for feedback stabilization of the Stokes equations

P. BENNER^{*,†}, J. SAAK^{*,†}, F. SCHIEWECK[‡],
P. SKRZYPACZ^{*}, and H. K. WEICHEL[†]

Received April 2, 2013

Abstract — In this contribution, we show a method for the boundary feedback stabilization of the Stokes problem around a stationary trajectory. We derive a formal low-rank algorithm for solving the stabilization problem in operator notation. The appearing operator equations are formulated in terms of stationary partial differential equations (PDEs) instead of using their finite dimensional representations in terms of matrices. A Galerkin method, satisfying the divergence constraint pointwise locally is especially appealing since it represents appropriately the action of the Helmholtz projection.

The main advantages of the composite technique are the efficient assembly of element matrices, the reduction of computational costs using static condensation, and the diagonal mass matrix. The non-conforming character of the composite element guarantees a better sparsity pattern, compared to conforming elements, due to the lower number of couplings between basis functions corresponding to neighboring cells. We also achieve the pointwise mass conservation on sub-triangles of each element.

Keywords: non-conforming finite elements, Stokes equations, feedback stabilization

^{*}Research Group Computational Methods in Systems and Control Theory (CSC), Max Planck Institute for Dynamics of Complex Technical Systems Magdeburg, Sandtorstr. 1, 39106 Magdeburg, Germany

[†]Research Group Mathematics in Industry and Technology (MiIT), Chemnitz University of Technology, Reichenhainer Str. 39/41, 09126 Chemnitz, Germany

[‡]Institute for Analysis und Numerics, Otto-von-Guericke-University of Magdeburg, Postfach 4120, 39016 Magdeburg, Germany

1. Introduction

Feedback stabilization can be seen as the task of making a given working trajectory, for example from an *open-loop* controller [10], more robust with respect to disturbances. We want to apply a *linear quadratic regulator* (LQR) approach to stabilize this open-loop trajectory [17]. Raymond [19] deduces boundary feedback stabilization of 2D incompressible flow problems and extends this approach to finite dimensional controllers in [20]. Bansch/Benner investigate several ideas for the numerical realization in [2], where they apply standard Taylor–Hood finite elements [11]. This type of discretization naturally leads to a discrete differential algebraic system of equations of differentiation index 2 [22]. The differential algebraic character of the equations causes several technical difficulties in the regulator approach. Most of these have been solved by the approach given in [4], which is following ideas from [9]. Still, when solving the appearing linear systems approximately, i.e., by an iterative linear solver, one can not guarantee the validity of the divergence freeness condition.

This paper is devoted to two major subjects. On the one hand, we want to show how the numerical methods for the feedback computation can be reformulated independently of the actual spatial discretization. On the other hand, we present a new type of finite element that overcomes the difficulties regarding the algebraic constraints by guaranteeing the divergence freeness of the discrete solutions computed in each step of our algorithm.

To be independent of the numerical discretization, we first formulate the problem as an *ordinary differential equation* (ODE) defined in a *Hilbert space* and apply the *Newton-ADI* process [3] to compute the optimal control to the resulting operator equations in a formal way. We identify the main steps of the algorithm, i.e., the shifted linear operator equations in terms of stationary linear PDEs containing an additional reaction term resulting from the shift. These PDEs can be solved by any spatial discretization method. One efficient realization of the solver for the PDEs is given by the use of our quadrilateral finite elements.

An advantage of the usage of quadrilateral finite elements is that, in order to decompose a two-dimensional domain into simple cells, one needs approximately half the number of quadrilateral cells compared to triangular cells. In the three-dimensional case at least about 5 times more tetrahedra compared to hexahedra are required for decomposing a domain. On the other hand, the reference transformation for quadrilateral or hexahedral

elements is more complicated than for triangular or tetrahedral elements where it is simply an affine mapping. For an affine mapping the Jacobian is constant and has to be computed only once for all integration points when assembling the local element matrices.

Therefore, we work as a compromise with *composite quadrilateral elements* where the reference transformation is continuous and piecewise affine. This is realized by subdividing each quadrilateral mesh cell into 4 *son-triangles* and getting profit from the affine mapping between each son-triangle of the original cell and its corresponding son-triangle of the reference cell. One can regard this approach also as a blocking of 4 triangular finite elements to one quadrilateral element. The advantage is that one can eliminate all interior degrees of freedom of the quadrilateral cells by means of the well-known static condensation technique which leads to a much smaller linear system that has to be solved.

Our motivation for using non-conforming instead of conforming finite elements is the following:

- it allows us to use the low order element pair of Crouzeix–Raviart (see [7] and [6, pp. 107-109]) on the son-triangles which is inf-sup stable and has low computational costs;
- it guarantees pointwise mass-conservation within the son-triangles;
- the basis functions for the velocity are L^2 -orthogonal which leads to a diagonal mass matrix;
- after eliminating the interior degrees of freedom of the quadrilateral elements, the remaining basis functions have a much smaller number of couplings compared to the conforming case.

The remaining paper is structured as follows. In Section 2, we give a brief overview of the LQR idea in the operator setting and identify all PDEs we have to solve to compute the optimal control. Afterwards, we introduce our new composite non-conforming quadrilateral elements in Section 3. We show numerical results in Section 4 and conclude the paper at the end.

Finally, before getting to the main part of the paper, we fix some notation. Let $\Omega \subset \mathbb{R}^2$ be a bounded domain in \mathbb{R}^2 with boundary $\Gamma := \partial\Omega$. We denote the outer unit normal vector on Γ by n and use the standard Sobolev spaces $W^{k,p}(G)$, $H^k(G) = W^{k,2}(G)$, $H_0^k(G)$, and $L^p(G) = W^{0,p}(G)$ for a measurable one- or two-dimensional set $G \subset \Omega$ with its measure $|G|$,

where $1 \leq p \leq \infty$. The norms, semi-norms in the scalar and vector-valued versions in $W^{k,p}(G)$ are denoted by $\|\cdot\|_{k,p,G}$ and $|\cdot|_{k,p,G}$, respectively. The L^2 inner product on G is denoted by $(\cdot, \cdot)_G$. The broken norms and inner products defined over some partitions \mathcal{T}_h are indicated by the additional subscript h . To simplify the notation, we will drop G if $G = \Omega$ and p if $p = 2$. $\mathcal{L}(X, Y)$ stands for the set of linear operators $A : X \rightarrow Y$ between Hilbert spaces X, Y . The adjoint operator of A will be denoted by A^* whereas X' stands for the dual space of X .

2. Feedback stabilization of Stokes flow

In this section, we consider the feedback stabilization of flow problems by means of boundary control. Though Stokes flow is stable, feedback can be used in order to achieve steady state faster than in the uncontrolled situation. Moreover, external disturbances can be attenuated. In this paper, we consider the Stokes flow as the first simple candidate for a non-stationary incompressible flow problem. This will be a prime step on the way to the treatment of non-self-adjoint and non-linear incompressible Navier–Stokes equations.

2.1. Non-stationary Stokes problem

Let Ω be a two-dimensional domain with the boundary $\Gamma = \Gamma_D \cup \Gamma_n$ consisting of the Dirichlet part Γ_D and the *do-nothing* part Γ_n . The Dirichlet part $\Gamma_D = \Gamma_c \cup \Gamma_d$ is further decomposed into the control part Γ_c and the remaining Dirichlet part Γ_d . We consider the following non-stationary Stokes problem:

Find a velocity field $v(t) : \Omega \rightarrow \mathbb{R}^2$ and a pressure field $p(t) : \Omega \rightarrow \mathbb{R}$ such that for all $t \in (0, \infty)$ it holds

$$\frac{\partial}{\partial t} v(t) - \nu \Delta v(t) + \nabla p(t) = f \quad \text{in } \Omega \quad (2.1a)$$

$$\operatorname{div} v(t) = 0 \quad \text{in } \Omega \quad (2.1b)$$

$$v(t) = 0 \quad \text{on } \Gamma_c \quad (2.1c)$$

$$v(t) = g \quad \text{on } \Gamma_d \quad (2.1d)$$

$$-\nu \nabla v(t) \cdot n + p(t)n = 0 \quad \text{on } \Gamma_n \quad (2.1e)$$

$$v(0) = v_0 \quad \text{in } \Omega. \quad (2.1f)$$

Here, $\nu > 0$ denotes the constant dynamic viscosity, Δ, ∇ , as well as div , represent the usual differential operators with respect to the space variable x ; f denotes an external force influencing the system, g some given Dirichlet data, n the unit outer normal vector on Γ , and v_0 the initial velocity. Based on the ideas of Raymond [19, 20], we want to apply a feedback boundary stabilization technique. The arising linear-quadratic control problem will be presented in the next subsection.

2.2. Riccati-based boundary feedback stabilization

Let (w, χ) denote the velocity and pressure solution of the stationary Stokes equations

$$-\nu \Delta w + \nabla \chi = f \quad \text{in } \Omega \tag{2.2a}$$

$$\text{div } w = 0 \quad \text{in } \Omega \tag{2.2b}$$

$$w = 0 \quad \text{on } \Gamma_c \tag{2.2c}$$

$$w = g \quad \text{on } \Gamma_d \tag{2.2d}$$

$$-\nu \nabla w \cdot n + \chi n = 0 \quad \text{on } \Gamma_n \tag{2.2e}$$

with some special properties we want to achieve. Such properties could be, for example, that (w, χ) solves a possibly constrained open loop (boundary) control problem. That means, $w(x)$ is a stationary solution of (2.1), see [19]. Our aim is to stabilize the solution (w, χ) by means of a feedback control driven by a time dependent control vector $u(t) = (u_k(t)) \in \mathbb{R}^{N_c}$. The vector $u(t)$ determines the velocity at the control boundary $\Gamma_c = \bigcup_{k=1}^{N_c} \Gamma_c^{(k)}$ in the way that the component $u_k(t)$ is responsible for the Dirichlet data on the part $\Gamma_c^{(k)}$. To be more specific, we introduce a boundary control operator $b_c : \mathbb{R}^{N_c} \rightarrow (H^1(\Gamma))^2$ which assigns the control vector $u(t)$ to some Dirichlet data on Γ_c defined as

$$(b_c u(t))(x) = \sum_{k=1}^{N_c} u_k(t) \xi^{(k)}(x) \quad \forall x \in \Gamma. \tag{2.3}$$

Here, $\xi^{(k)} \in (H^1(\Gamma))^2$ with $\xi^{(k)}|_{\Gamma \setminus \Gamma_c^{(k)}} = 0$ denotes the prescribed shape function associated with the control boundary part $\Gamma_c^{(k)}$ for all $k = 1, \dots, N_c$.

Using the splitting

$$(v, p) = (w + \tilde{v}, \chi + \tilde{p})$$

we define (\tilde{v}, \tilde{p}) as the perturbation in the solution (v, p) of (2.1) with respect to the desired solution (w, χ) of (2.2). Then, the control problem for (\tilde{v}, \tilde{p}) reads:

For $t \in (0, \infty)$, find a velocity field $\tilde{v}(t) : \Omega \rightarrow \mathbb{R}^2$ and a pressure field $\tilde{p}(t) : \Omega \rightarrow \mathbb{R}$ such that for all $t \in (0, \infty)$ it holds

$$\frac{\partial}{\partial t} \tilde{v}(t) - \nu \Delta \tilde{v}(t) + \nabla \tilde{p}(t) = 0 \quad \text{in } \Omega \quad (2.4a)$$

$$\operatorname{div} \tilde{v}(t) = 0 \quad \text{in } \Omega \quad (2.4b)$$

$$\tilde{v}(t) = b_c u(t) \quad \text{on } \Gamma_c \quad (2.4c)$$

$$\tilde{v}(t) = 0 \quad \text{on } \Gamma_d \quad (2.4d)$$

$$-\nu \nabla \tilde{v}(t) \cdot n + \tilde{p}(t) n = 0 \quad \text{on } \Gamma_n \quad (2.4e)$$

$$\tilde{v}(0) = \tilde{v}_0 \quad \text{in } \Omega. \quad (2.4f)$$

The boundary control $b_c u$ from (2.3) describes the influence of the feedback via boundary stabilization. The feedback stabilization computed via (2.4) then forces the instationary solution (v, p) of (2.1) to the stationary solution (w, χ) of (2.2) by means of the modified boundary condition $v(t) = b_c u(t)$ on Γ_c for any $t \in (0, \infty)$.

Raymond [19] determines a linear feedback law to stabilize (2.4). Based on his ideas, Bänsch/Benner show in [2] initial steps for the numerical realization. Benner et al. use a mixed finite element method [11] to show first numerical realizations based on a matrix approximation of (2.4) in [4]. Thereby, it is shown that one can use the projection idea presented in [9] to have a matrix approximation of the *Leray* projection.

In the following, we combine the matrix based approach in [4] with the operator formulation in [19, 20] in a formal way, using some notations of [21] and avoid the weak formulation of operators in $L^2(\Omega)$.

From now on we skip the arguments (t, x) where they are obvious. Furthermore, we use the following spaces

$$V_n^0(\Omega) = \{v \in (L^2(\Omega))^2 : \operatorname{div} v = 0 \text{ in } \Omega, v \cdot n = 0 \text{ on } \Gamma\}$$

$$V_0^1(\Omega) = \{v \in (H^1(\Omega))^2 : \operatorname{div} v = 0 \text{ in } \Omega, v = 0 \text{ on } \Gamma_D\}.$$

Let $\mathcal{P} : (L^2(\Omega))^2 \rightarrow V_n^0(\Omega)$ denote the so called *Leray* or *Helmholtz* projector and let \mathcal{A} be the Stokes operator defined as

$$\mathcal{A} \tilde{v}(t) = \nu \mathcal{P} \Delta \tilde{v}(t) \quad \forall \tilde{v}(t) \in (H^2(\Omega))^2 \cap V_0^1(\Omega).$$

Recasting the influence of the Dirichlet control $b_c u(t)$ as the action of a control operator $\mathcal{B} : \mathbb{R}^{N_c} \rightarrow V_n^0(\Omega)$, (2.4a)–(2.4b) can be reformulated (using the property $\tilde{v} = \mathcal{P}\tilde{v}$ for the solution \tilde{v}) in the spirit of [19] as

$$\mathcal{P} \frac{\partial}{\partial t} \tilde{v}(t) = \mathcal{A} \mathcal{P} \tilde{v}(t) + \mathcal{B}u(t). \tag{2.5}$$

The computed control $u(t) \in \mathbb{R}^{N_c}$ can be used independently of the approximation we applied for the underlying PDE. Note, however, that we may introduce a certain suboptimality when applying the control computed with respect to one approximation, to a much finer approximation, or even the original system.

We introduce the observation variable

$$y(t) = \mathcal{C} \mathcal{P} \tilde{v}(t) \tag{2.6}$$

where the output operator \mathcal{C} maps the velocity field onto our observation space $\mathbb{R}^{N_{\text{obs}}}$ and can be chosen in different ways, which is explained for our model problem in Subsection 4.1. The linear-quadratic regulator (LQR) problem reads as follows:

Minimize the cost functional

$$\mathcal{J}(y, u) = \frac{1}{2} \int_0^\infty \|y(t)\|^2 + \|u(t)\|^2 dt \tag{2.7a}$$

subject to

$$\mathcal{P} \frac{\partial}{\partial t} \tilde{v}(t) = \mathcal{A} \mathcal{P} \tilde{v}(t) + \mathcal{B}u(t) \tag{2.7b}$$

$$y(t) = \mathcal{C} \mathcal{P} \tilde{v}(t). \tag{2.7c}$$

It is known (e.g., [14, 16]) that the optimal control $t \mapsto \hat{u}(t) \in \mathbb{R}^{N_c}$ that solves the LQR problem (2.7) can be represented as

$$\hat{u} = -\mathcal{K} \tilde{v}$$

with the feedback operator

$$\mathcal{K} : V_n^0(\Omega) \rightarrow \mathbb{R}^{N_c}, \quad \mathcal{K} = \mathcal{B}^* \widehat{\mathcal{X}}$$

where $\widehat{\mathcal{X}} = \widehat{\mathcal{X}}^* \in \mathcal{L}(V_n^0(\Omega), V_n^0(\Omega))$ is the unique stabilizing weak solution of the Riccati operator equation

$$0 = \mathcal{C}^* \mathcal{C} + \mathcal{A}^* \mathcal{X} + \mathcal{X} \mathcal{A} - \mathcal{X} \mathcal{B} \mathcal{B}^* \mathcal{X} =: \mathcal{R}(\mathcal{X}). \quad (2.8)$$

A common way to solve the non-linear operator equation (2.8) is a Newton-type iteration as described in [1, 13]. Using the *Kleinman* reformulation [12], it turns out that we have to determine $\mathcal{X}^{(m+1)}$ from the equation

$$(\mathcal{A}^{(m)})^* \mathcal{X}^{(m+1)} + \mathcal{X}^{(m+1)} \mathcal{A}^{(m)} = -\mathcal{W}^{(m)} (\mathcal{W}^{(m)})^* \quad (2.9)$$

where $\mathcal{A}^{(m)} = \mathcal{A} - \mathcal{B} \mathcal{B}^* \mathcal{X}^{(m)}$ and $\mathcal{W}^{(m)} = [\mathcal{C}^*, (\mathcal{B}^* \mathcal{X}^{(m)})^*]$ in each Newton step m . Thereby, $\mathcal{W}^{(m)}$ is the operator column matrix [21] defined as

$$\mathcal{W}^{(m)} : \mathbb{R}^{N_{\text{obs}}} \times \mathbb{R}^{N_c} \rightarrow V_n^0(\Omega)' \times V_n^0(\Omega)'.$$

A solution strategy for the matrix version of equation (2.9) is the *low-rank ADI iteration* [3, 15], where low-rank factors of the solution are computed. The operator version of the low-rank ADI iteration is presented in [21]. Combing this with the Newton iteration, we end up with the *Operator Newton-ADI iteration* presented in formal operator notation in Algorithm 2.1 to determine the desired feedback operator \mathcal{K} . In practice, we need finite dimensional approximations of the infinite dimensional operators. A common way is to use a finite element method to explicitly create the matrix representations for a fixed finite element basis. The main difficulty is to fulfill the algebraic constraints given by equation (2.1b), which means the finite element space has to fulfill this property by default or we need a numerical realization of the *Leray* projector. As mentioned before, Benner et al. show in [4] a realization using mixed finite elements, like *Taylor–Hood* elements [11]. Here, we consider a possibly matrix free approach, where the crucial steps of Algorithm 2.1 are handled as *black-box-functions*, solving the underlying generalized Stokes problems with the algebraic constraints. In the next subsection, we give a brief overview about the required *black-box-functions* in Algorithm 2.1. The convergence of the ADI iteration depends on the shift parameters $\{\mu_i\}_{i=1}^{n_{\text{ADI}}}$. In the numerical experiments we apply the heuristic *Penzl* shifts [18].

Algorithm 2.1 Operator Newton-ADI method

Input: shift parameters $\{\mu_1, \dots, \mu_{n_{\text{ADI}}}\}$ with $\mu_i \in \mathbb{C}^-$
Output: feedback operator \mathcal{K}
1: $\mathcal{K}^{(0)} = 0$ 2: $m = 1$ 3: **while** not converged **do**4: $\mathcal{W}^{(m)} = [\mathcal{C}^*, (\mathcal{K}^{(m-1)})^*]$ 5: Get \mathcal{V}_1 by solving

$$(\mathcal{A}^* + \mu_1 \mathcal{I}) \mathcal{P} \mathcal{V}_1 = \mathcal{W}^{(m)}$$

6: $\mathcal{K}_1^{(m)} = -2 \operatorname{Re}(\mu_1) \cdot \mathcal{B}^* \mathcal{V}_1 \mathcal{V}_1^*$ 7: $i = 2$ 8: **while** not converged **do**9: Get $\tilde{\mathcal{V}}$ by solving (via Sherman–Morrison–Woodbury formula)

$$(\mathcal{A}^* - (\mathcal{K}^{(m-1)})^* \mathcal{B}^* + \mu_i \mathcal{I}) \mathcal{P} \tilde{\mathcal{V}} = \mathcal{P} \mathcal{V}_{i-1}$$

10: $\mathcal{V}_i = \mathcal{V}_{i-1} - (\mu_i + \bar{\mu}_{i-1}) \tilde{\mathcal{V}}$ 11: $\mathcal{K}_i^{(m)} = \mathcal{K}_{i-1}^{(m)} - 2 \operatorname{Re}(\mu_i) \mathcal{B}^* \mathcal{V}_i \mathcal{V}_i^*$ 12: $i = i + 1$ 13: **end while**14: $\mathcal{K}^{(m)} = \mathcal{K}_i^{(m)}$ 15: $m = m + 1$ 16: **end while**17: $\mathcal{K} = \mathcal{K}^m$

2.3. Practical realization of main steps in Algorithm 2.1

The main computational steps in Algorithm 2.1 are the lines 5 and 9. In line 5, we have to solve for \mathcal{V}_1 in the equation

$$(\mathcal{A}^* + \mu_1 \mathcal{I}) \mathcal{P} \mathcal{V}_1 = \mathcal{W}^{(m)}.$$

Therefore, one main *black-box-function* is to solve the *generalized* stationary Stokes problem

$$-v \Delta v - \mu_1 v + \nabla p = f_{\mathcal{W}^{(m)}} \quad \text{in } \Omega \quad (2.10a)$$

$$\operatorname{div} v = 0 \quad \text{in } \Omega \quad (2.10b)$$

$$v = 0 \quad \text{on } \Gamma_D \quad (2.10c)$$

$$-v \nabla v \cdot n + pn = 0 \quad \text{on } \Gamma_n \quad (2.10d)$$

for several right hand sides $f_{\mathcal{W}^{(m)}}$ depending on the realization of $\mathcal{W}^{(m)}$. This means every column in $\mathcal{W}^{(m)}$ creates a different function $f_{\mathcal{W}^{(m)}}$ and we arrange all solutions v again in columns of the operator \mathcal{V}_1 . In line 9 there is an additional term $(\mathcal{K}^{(m-1)})^* \mathcal{B}^* \mathcal{P} \tilde{\mathcal{V}}$ of two low-rank operators resulting from the formulation of the low-rank ADI. To avoid this term in (2.10), we use, analogous to the matrix version [4, Subsection 3.1], the *Sherman–Morrison–Woodbury* formula [8] and solve the operator equation

$$(\mathcal{A}^* + \mu_i \mathcal{I}) \mathcal{P} \tilde{\mathcal{V}} = [\mathcal{P} \mathcal{V}_{i-1}, (\mathcal{K}^{(m-1)})^*]$$

which means we have to solve the PDE (2.10) for the right hand sides $f_{\mathcal{V}}$ depending on $\mathcal{P} \mathcal{V}_{i-1}$ and $f_{\mathcal{K}^*}$ depending on $(\mathcal{K}^{(m-1)})^*$. We, thus, get different solutions from the different right hand sides which form the columns of the operator $\tilde{\mathcal{V}}$. Additionally, we need the computation of the action of the operators \mathcal{B}^* and \mathcal{K}^* as *black-box-functions*. In practice, we have to approximate the solution of (2.10) by means of discrete solution. We apply a finite element discretization which we explain in detail in Subsection 3.3.

3. Non-conforming composite quadrilateral finite elements

First, we introduce some notation concerning the used space grids. We denote by \mathcal{T}_h an admissible approximate decomposition of Ω into shape-regular quadrilaterals (with straight edges), where the curved boundary part is polygonally approximated. The mesh-size parameter h is given by

$h := \max\{h_K : K \in \mathcal{T}_h\}$, where h_K denotes the diameter of the cell $K \in \mathcal{T}_h$. From a given mesh \mathcal{T}_{h_ℓ} at grid level ℓ we create the next finer mesh at grid level $\ell + 1$ by subdividing each quadrilateral mesh cell $K \in \mathcal{T}_{h_\ell}$ into 4 quadrilaterals, where the midpoints of opposite edges of K are connected. For mesh cells $K \in \mathcal{T}_{h_\ell}$ having an edge E located at a curved boundary part of the domain, the midpoint of E is shifted onto Γ before being used as a new vertex in the new mesh $\mathcal{T}_{h_{\ell+1}}$. Thus, a domain Ω with curved boundary parts is approximated on each grid level by a polygonally bounded domain $\overline{\Omega}_h := \bigcup_{K \in \mathcal{T}_h} \overline{K}$. However, for the presentation of our theoretical basics, we will always assume that $\Omega = \Omega_h$.

In Subsection 3.1, we describe the decomposition of the reference element into son-triangles and the global space of non-conforming composite quadrilateral finite elements in Subsection 3.2.

3.1. Composite quadrilateral reference element

Let $\widehat{K} = (-1, 1)^2$ denote the reference cell and $\widehat{\mathcal{T}}$ its decomposition into (open) *son-triangles* \widehat{T}_i , i.e.,

$$\widehat{\mathcal{T}} = \{\widehat{T}_1, \dots, \widehat{T}_4\}, \quad \widehat{K} = \bigcup_{i=1}^4 \widehat{T}_i$$

such that all son-triangles have a common vertex at $\hat{x} = 0$ (see Fig. 1). The concept of composite elements consisting of triangular son-cells $T_{i,K}$ for all $i = 1, \dots, 4$ exploits the fact that one can profit from a continuous and piecewise affine reference transformation $F_K : \widehat{K} \rightarrow K$ such that

$$T_{i,K} = F_K|_{\widehat{T}_i}(\widehat{T}_i), \quad F_K|_{\widehat{T}_i} \in [\mathbb{P}_1(\widehat{T}_i)]^2 \quad \forall i = 1, \dots, 4 \quad (3.1)$$

where the son-triangle $T_{i,K} \subset K$ has the vertices a_i, a_{i+1} , and a_0 using the convention that a_0 denote the barycenter and a_1, \dots, a_4 the vertices of K , respectively, with $a_5 := a_1$. The advantage is that the Jacobian of the restriction $F_K|_{\widehat{T}_i}$ is constant which simplifies the generation of the element matrices since many terms can be pre-computed in advance on the reference element.

In order to define finite element spaces and basis functions on the reference element \widehat{K} and the original element K , we have to introduce some more notation about the decomposition of \widehat{K} and K . The decomposition of the original cell K into son-triangles will be denoted by $\mathcal{T}^K = \{T_{1,K}, \dots, T_{4,K}\}$. We define $\widehat{\mathcal{E}}$ as the set of all edges of the son-triangles of \widehat{K} . Furthermore,

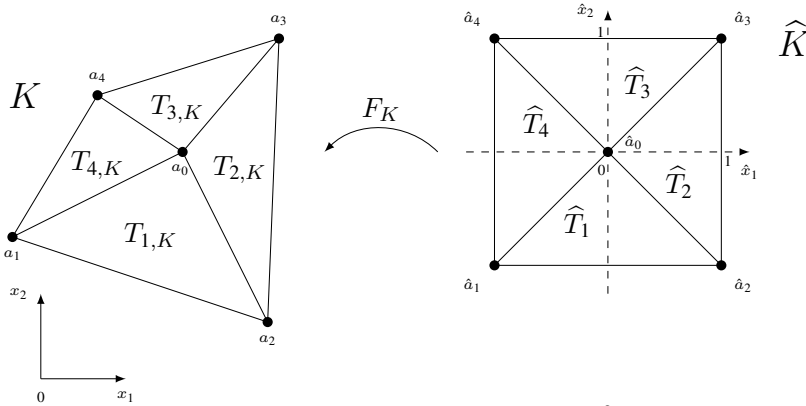


Figure 1. The composite cell $K = F_K(\widehat{K})$, where $F_K|_{\widehat{T}} \in [\mathbb{P}_1(\widehat{T})]^2$.

we denote by $\widehat{\mathcal{E}}^{\text{int}} \subset \widehat{\mathcal{E}}$ the subset of all edges in the interior of \widehat{K} and by $\widehat{\mathcal{E}}^{\text{bnd}}$ the subset of all edges on the boundary $\partial\widehat{K}$. To an edge $\widehat{E} \in \widehat{\mathcal{E}}$, we assign a unit normal vector $\widehat{n}_{\widehat{E}}$ on \widehat{E} with an arbitrary, but fixed orientation. For a given possibly discontinuous function $\widehat{\phi} : \widehat{K} \rightarrow \mathbb{R}^n$, we define for each interior edge $\widehat{E} \in \widehat{\mathcal{E}}^{\text{int}}$ the *jump* $[[\widehat{\phi}]]_{\widehat{E}}$ as the function

$$[[\widehat{\phi}]]_{\widehat{E}}(\widehat{x}) := \lim_{\tau \rightarrow +0} \widehat{\phi}(\widehat{x} + \tau \widehat{n}_{\widehat{E}}) - \lim_{\tau \rightarrow +0} \widehat{\phi}(\widehat{x} - \tau \widehat{n}_{\widehat{E}}) \quad \forall \widehat{x} \in \widehat{E}.$$

In the following, let $\langle \cdot, \cdot \rangle_{\widehat{E}}$ denote the inner product in $L^2(\widehat{E})$. Now, the local non-conforming composite polynomial space on \widehat{K} can be described as

$$\begin{aligned} \mathbb{P}_1^{\text{nc}}(\widehat{K}) := \left\{ \widehat{v} \in L^2(\widehat{K}) : \widehat{v}|_{\widehat{T}} \in \mathbb{P}_1(\widehat{T}) \quad \forall \widehat{T} \in \widehat{\mathcal{T}}, \right. \\ \left. \langle [[\widehat{v}]]_{\widehat{E}}, 1 \rangle_{\widehat{E}} = 0 \quad \forall \widehat{E} \in \widehat{\mathcal{E}}^{\text{int}} \right\}. \end{aligned} \tag{3.2}$$

To ensure the existence of traces of a function on the edges $\widehat{E} \in \widehat{\mathcal{E}}$, we introduce the broken Sobolev space

$$H_b^1(\widehat{\mathcal{T}}) := \{ \widehat{v} \in L^2(\widehat{K}) : \widehat{v}|_{\widehat{T}} \in H^1(\widehat{T}) \quad \forall \widehat{T} \in \widehat{\mathcal{T}} \}.$$

The degrees of freedom of the non-conforming composite \mathbb{P}_1 -element are associated with the edges of the son-triangles. Therefore, we number the edges in $\widehat{\mathcal{E}}$ (see Fig. 2) with the convention that

$$\widehat{\mathcal{E}}^{\text{bnd}} = \{ \widehat{E}_1, \dots, \widehat{E}_4 \}, \quad \widehat{\mathcal{E}}^{\text{int}} = \{ \widehat{E}_5, \dots, \widehat{E}_8 \}$$

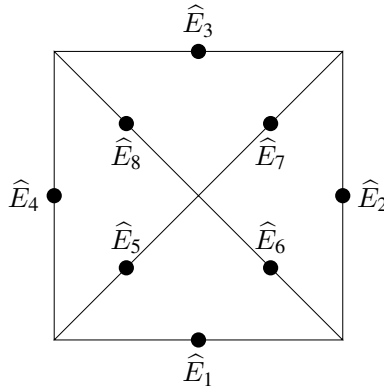


Figure 2. Numbering of the edges of the son-triangles on the reference element and local degrees of freedom (marked by ●) of the composite $\mathbb{P}_1^{\text{nc}}(\widehat{K})$ -element.

and assign to each son-edge \widehat{E}_i the nodal functional

$$\widehat{N}_i : H_b^1(\widehat{\mathcal{T}}) \rightarrow \mathbb{R}, \quad \widehat{N}_i(\widehat{v}) = |\widehat{E}_i|^{-1} \langle \widehat{v}, 1 \rangle_{\widehat{E}_i}, \quad i = 1, \dots, 8$$

where $|\widehat{E}_i|$ denotes the length of the edge \widehat{E}_i . These nodal functionals are unisolvent with respect to the polynomial space $\mathbb{P}_1^{\text{nc}}(\widehat{K})$ and define uniquely the reference basis functions $\widehat{\phi}_i \in \mathbb{P}_1^{\text{nc}}(\widehat{K})$ by means of the conditions

$$\widehat{N}_j(\widehat{\phi}_i) = \delta_{ij}, \quad i, j = 1, \dots, 8 \tag{3.3}$$

where δ_{ij} denotes the Kronecker delta. In order to show the approximation properties of the discrete velocity space on the reference element we introduce the velocity interpolation operator $\widehat{I}_v : H^1(\widehat{K}) \rightarrow \mathbb{P}_1^{\text{nc}}(\widehat{K})$ as

$$\widehat{I}_v \widehat{u}(\widehat{x}) := \sum_{i=1}^8 \widehat{N}_i(\widehat{u}) \widehat{\phi}_i(\widehat{x}) \quad \forall \widehat{u} \in H^1(\widehat{K}).$$

Applying the Bramble–Hilbert lemma [5] on each son-triangle $\widehat{T} \in \widehat{\mathcal{T}}$, we can show the estimate

$$\|\widehat{u} - \widehat{I}_v \widehat{u}\|_{1,\widehat{T}} \leq C \|\widehat{u}\|_{2,\widehat{T}} \quad \forall \widehat{T} \in \widehat{\mathcal{T}}, \quad \widehat{u} \in H^2(\widehat{K}) \tag{3.4}$$

which is the basis to prove optimal approximation properties on the original mesh \mathcal{T}_h . For a vector-valued function $\widehat{v} \in (H^1(\widehat{K}))^2$, the operator \widehat{I}_v is applied to each component of \widehat{v} .

3.2. Global finite element spaces for velocity and pressure

In this section, we describe the construction of the global finite element spaces V_h and Q_h for approximating velocity and pressure, respectively. We start with the definition of V_h which is based on the piecewise affine reference transformations $F_K : \widehat{K} \rightarrow K$ and the previously defined polynomial space $\mathbb{P}_1^{\text{nc}}(\widehat{K})$ on the reference element. In order to keep the consistency error of the global non-conforming discretization within the right asymptotic order, we have to ensure for each discrete function $v_h \in V_h$ that the jumps of v_h across the edges of the son-triangles of all quadrilateral cells are zero in the integral mean value. Therefore, we need some notation on edges and jumps analogously to the previous section for the reference element. For a given quadrilateral cell $K \in \mathcal{T}_h$, the symbol $\mathcal{E}(K)$ denotes the set of all boundary edges of the quadrilateral cell K which are the images of the edges $\widehat{E} \in \widehat{\mathcal{E}}^{\text{bnd}}$ via the mapping $F_K : \widehat{K} \rightarrow K$. Furthermore, let \mathcal{E}_h denote the set of all edges in the union of the $\mathcal{E}(K)$ over all cells $K \in \mathcal{T}_h$ and $\mathcal{E}_h^{\text{int}}, \mathcal{E}_h^{\text{bnd}}$ the subsets of the edges $E \in \mathcal{E}_h$ in the interior and on the boundary of the domain Ω . The set $\mathcal{E}_h^{\text{bnd}}$ is split as

$$\mathcal{E}_h^{\text{bnd}} = \mathcal{E}_h^{\text{D}} \cup \mathcal{E}_h^{\text{N}}$$

where \mathcal{E}_h^{D} denotes the set of boundary edges with prescribed Dirichlet boundary conditions, while \mathcal{E}_h^{N} denotes the set where ‘do-nothing’ boundary conditions are imposed. To each edge $E \in \mathcal{E}_h$, we assign a unit vector n_E normal to E which is assumed to point outwards with respect to Ω if $E \in \mathcal{E}_h^{\text{bnd}}$. For a given edge $E \in \mathcal{E}_h$ and a possibly discontinuous function $\varphi : \Omega \rightarrow \mathbb{R}$, which is continuous on each sub-triangle of the cells $K \in \mathcal{T}_h$, we define for all $x \in E$ the *jump* $[[\varphi]]_E$ as the function

$$[[\varphi]]_E(x) := \begin{cases} \lim_{\tau \rightarrow +0} \varphi(x + \tau n_E) - \lim_{\tau \rightarrow +0} \varphi(x - \tau n_E), & E \in \mathcal{E}_h^{\text{int}} \\ \lim_{\tau \rightarrow +0} \varphi(x - \tau n_E), & E \in \mathcal{E}_h^{\text{bnd}}. \end{cases}$$

Now, we are in the position to define the scalar discrete velocity space S_h . It can be described via the piecewise affine reference mappings $F_K : \widehat{K} \rightarrow K$ as follows:

$$S_h := \left\{ \varphi \in L^2(\Omega) : \varphi|_K \circ F_K \in \mathbb{P}_1^{\text{nc}}(\widehat{K}) \quad \forall K \in \mathcal{T}_h, \right. \\ \left. \langle [[\varphi]]_E, 1 \rangle_E = 0 \quad \forall E \in \mathcal{E}_h^{\text{int}} \right\}.$$

For the test functions in the discrete problem, we need also the subspace $S_h^0 \subset S_h$ with discretely homogeneous Dirichlet boundary conditions, i.e.,

$$S_h^0 := \{ \varphi \in S_h : \langle [[\varphi]]_E, 1 \rangle_E = 0 \quad \forall E \in \mathcal{E}_h^D \}.$$

The associated vector-valued discrete velocity spaces are $V_h := (S_h)^2$ and $V_h^0 := (S_h^0)^2$. The jump conditions in the definition of S_h are realized by the following choice of the global basis functions. One subset of basis functions is associated with the set of the interior edges and the set of the non-Dirichlet boundary edges. Therefore, we number these edges as

$$\mathcal{E}_h^{\text{int}} \cup \mathcal{E}_h^{\text{N}} = \{E_1, \dots, E_{N_{\text{EDG}}}\}.$$

For each edge $E_j \in \mathcal{E}_h^{\text{int}} \cup \mathcal{E}_h^{\text{N}}$, we define an associated scalar basis function φ_j by its restriction to an arbitrary element $K \in \mathcal{T}_h$:

$$\varphi_j|_K(x) := \begin{cases} \hat{\varphi}_i(F_K^{-1}(x)), & E_j \in \mathcal{E}(K), E_j = F_K(\hat{E}_i), \\ 0, & E_j \notin \mathcal{E}(K), \end{cases} \quad x \in K.$$

Due to the Kronecker-delta property (3.3) of the reference basis functions $\hat{\varphi}_i$, we get the jump-property $\langle [[\varphi_j]]_E, 1 \rangle_E = 0$ for all edges $E \in \mathcal{E}_h^{\text{int}} \cup \mathcal{E}_h^{\text{D}}$ which implies $\varphi_j \in S_h$. The remaining basis functions of S_h are so-called *non-conforming bubble functions* since their support is just one cell. For each cell $K \in \mathcal{T}_h$, there are four of such bubble functions $\varphi_{K,i}^b$ defined for all $x \in \Omega$ as:

$$\varphi_{K,i}^b(x) := \begin{cases} \hat{\varphi}_{4+i}(F_K^{-1}(x)), & x \in K, \\ 0, & x \notin K, \end{cases} \quad \forall i = 1, \dots, 4.$$

The scalar velocity space in the case of homogeneous Dirichlet boundary conditions can be represented by means of its basis functions as

$$S_h^0 = \text{span} \{ \varphi_j, j = 1, \dots, N_{\text{EDG}}, \varphi_{K,i}^b, K \in \mathcal{T}_h, i = 1, \dots, 4 \}.$$

Remark 3.1. It is easy to verify that for each son-triangle \hat{T}_m , $m = 1, \dots, 4$, of \hat{K} it holds

$$(\hat{\varphi}_j, \hat{\varphi}_i)_{\hat{T}_m} = 0, \quad i, j \in \{1, \dots, 8\}, i \neq j.$$

Since the reference mapping $F_K : \hat{K} \rightarrow K$ is affine on each \hat{T}_m , we get that the local basis functions $\hat{\varphi}_i \circ F_K^{-1}$ are orthogonal in $L^2(K)$. This implies that the above constructed global basis functions of S_h and V_h are orthogonal with respect to the inner products in $L^2(\Omega)$ and $(L^2(\Omega))^2$, respectively.

In the following, we will describe the discrete pressure space Q_h . Since this space consists of piecewise constant functions, it is not necessary to define it via the reference mapping. We start with the definition of the local composite pressure space for an arbitrary quadrilateral cell $K \in \mathcal{T}_h$ which is given as

$$\mathbb{P}_0^{\text{dc}}(K) := \{q \in L^2(K) : q|_{T_{i,K}} \in \mathbb{P}_0(T_{i,K}), i = 1, \dots, 4\} \quad (3.5)$$

where the $T_{i,K} := F_K(\widehat{T}_i)$ denote the son-triangles of K . The natural basis functions of $\mathbb{P}_0^{\text{dc}}(K)$ are the characteristic functions $\chi_{i,K}, i = 1, \dots, 4$, which are defined to be 1 on the associated son-triangle $T_{i,K}$ and 0 on the remaining part of the cell K . Using the local composite spaces $\mathbb{P}_0^{\text{dc}}(K)$, the global pressure space Q_h can be described as

$$\begin{aligned} Q_h &:= \{q \in L^2(\Omega) : q|_K \in \mathbb{P}_0^{\text{dc}}(K), K \in \mathcal{T}_h\} \\ &= \text{span} \{\chi_{i,K}, i = 1, \dots, 4, K \in \mathcal{T}_h\}. \end{aligned} \quad (3.6)$$

3.3. Discretization of the generalized Stokes problem

In this subsection, we treat the discretization of the following weak formulation of the generalized Stokes problem (2.10), which is the key ingredient in performing the main steps in Algorithm 2.1:

Find $(v, p) \in (v_b + V) \times Q$ such that

$$v(\nabla v, \nabla \varphi) - \mu(v, \varphi) - (p, \text{div } \varphi) = (f, \varphi) \quad \forall \varphi \in V \quad (3.7)$$

$$(\text{div } v, q) = 0 \quad \forall q \in Q. \quad (3.8)$$

Here $\mu \leq 0$ denotes a given constant (shift parameter from Algorithm 2.1), the spaces Q and V are defined by $Q := L^2(\Omega)$,

$$V := \{\varphi \in (H^1(\Omega))^2 : \varphi|_{\Gamma_D} = 0\}$$

where $v_b \in (H^1(\Omega))^2$ is an extension to Ω of the given Dirichlet boundary data, and $f \in (L^2(\Omega))^2$ a given body force. The name *generalized* Stokes problem comes from the presence of the term with the factor μ which arises from our approach to compute the feedback control. This extra term does not cause any problems in theory and practice.

In the non-conforming case, the first derivatives of the discrete velocity function exist only locally on the son-triangles of the quadrilateral mesh

cells. Therefore, we introduce the following discrete bilinear forms

$$a_h(v, \varphi) := \sum_{K \in \mathcal{T}_h} \sum_{i=1}^4 \{v(\nabla v, \nabla \varphi)_{T_{i,K}} - \mu(v, \varphi)_{T_{i,K}}\}$$

$$b_h(v, q) := - \sum_{K \in \mathcal{T}_h} \sum_{i=1}^4 (\operatorname{div} v, q)_{T_{i,K}}$$

for all $v, \varphi \in V_h$ and $q \in Q_h$. Let $v_{bh} \in V_h$ be a discrete function which interpolates the Dirichlet data with optimal order. The discrete generalized Stokes problem reads as follows:

Find $(v_h, p_h) \in (v_{bh} + V_h^0) \times Q_h$ such that

$$\begin{aligned} a_h(v_h, \varphi_h) + b_h(\varphi_h, p_h) &= (f, \varphi_h) & \forall \varphi_h \in V_h^0 \\ b_h(v_h, q_h) &= 0 & \forall q_h \in Q_h. \end{aligned} \tag{3.9}$$

Since this discrete problem is equivalent to the standard discretization using the non-conforming Crouzeix–Raviart element pair on the triangular mesh consisting of all the son-triangles, we know that this problem admits a stable unique solution which is accurate of optimal order, see [7]. In Section 4, we will confirm this accuracy by means of a numerical test example.

3.4. Static condensation

Static condensation is an old principle in finite element technology and means that one eliminates all ‘interior’ degrees of freedom in each cell $K \in \mathcal{T}_h$ for which the support of the associated basis function is contained in \bar{K} . The idea behind this approach is that, on each cell K , the values of the solution components related to these interior degrees of freedom can be locally expressed by means of the values of the remaining *non-interior* degrees of freedom associated with K . In the following, we will describe this technique in more detail for the generalized Stokes problem (3.9).

For a fixed cell $K \in \mathcal{T}_h$, let $E_i^K := F_K(\hat{E}_i)$, $i = 1, \dots, 8$, denote the edges of all son-triangles of K and $\varphi_i^K := \hat{\varphi}_i \circ F_K^{-1}$ the associated scalar basis functions for the discrete velocity v_h on K . The corresponding vector-valued basis functions $\underline{\varphi}_j^K : K \rightarrow \mathbb{R}^2$, $j = 1, \dots, 16$, are defined as

$$\underline{\varphi}_{2i-1}^K := \begin{pmatrix} \varphi_i^K \\ 0 \end{pmatrix}, \quad \underline{\varphi}_{2i}^K := \begin{pmatrix} 0 \\ \varphi_i^K \end{pmatrix}, \quad i = 1, \dots, 8. \tag{3.10}$$

An arbitrary discrete velocity function $v_h \in V_h$ restricted to K can be represented as

$$v_h|_K(x) = \sum_{j=1}^{16} v_j^K \underline{\varphi}_j^K(x) \quad \forall x \in K$$

where the degrees of freedom v_j^K are given by

$$v_{2i-1}^K := |E_{K,i}|^{-1} \langle v_{h,1}|_K, 1 \rangle_{E_{K,i}}, \quad v_{2i}^K := |E_{K,i}|^{-1} \langle v_{h,2}|_K, 1 \rangle_{E_{K,i}} \quad (3.11)$$

for all $i = 1, \dots, 8$. Due to Remark 3.1, we get an L^2 -orthogonal decomposition of an arbitrary $v_h \in V_h$ into an *interior* part v_h^{int} (which can be condensed) and a *remaining* part v_h^r as follows

$$v_h(x) = v_h^{\text{int}}(x) + v_h^r(x) \quad \forall x \in \Omega$$

where for all $x \in K$ it holds together with (3.11)

$$v_h^{\text{int}}|_K(x) = \sum_{j=9}^{16} v_j^K \underline{\varphi}_j^K(x), \quad v_h^r|_K(x) = \sum_{j=1}^8 v_j^K \underline{\varphi}_j^K(x). \quad (3.12)$$

Similarly, we can define an L^2 -orthogonal decomposition of an arbitrary discrete pressure function $p_h \in V_h$ into an *interior* part p_h^{int} (which can be condensed) and a *remaining* part p_h^r as follows

$$p_h(x) = p_h^{\text{int}}(x) + p_h^r(x) \quad \forall x \in \Omega$$

where for all $x \in K$ it holds

$$p_h^{\text{int}}|_K(x) = \sum_{j=2}^4 p_j^K \psi_j^K(x), \quad p_h^r|_K(x) = p_1^K := |K|^{-1} (p_h, 1)_K. \quad (3.13)$$

Here, $|K|$ denotes the area of the cell K and the local basis functions $\psi_j^K \in \mathbb{P}_0^{\text{dc}}(K) \cap L_0^2(K)$ with $L_0^2(K) := \{q \in L^2(K) : (q, 1)_K = 0\}$ are defined as the following linear combinations of the characteristic functions $\chi_{i,K}$ from Subsection 3.2:

$$\psi_j^K(x) := \chi_{j,K}(x) - \frac{|T_{j,K}|}{|T_{1,K}|} \chi_{1,K}(x), \quad j = 2, \dots, 4, \quad x \in K. \quad (3.14)$$

Now, we can eliminate, on each cell $K \in \mathcal{T}_h$, the *interior* parts v_h^{int} and p_h^{int} in dependence of the *remaining* parts v_h^r and p_h^r . To this end, we simply

choose in (3.9) K -local test functions and get the following local generalized Stokes problem:

For given $v_h^r|_K$ and $p_h^r|_K$, find $v_h^{\text{int}}|_K$ and $p_h^{\text{int}}|_K$ described in (3.12) and (3.13) such that

$$\begin{aligned} a_h \left(v_h^{\text{int}}|_K + v_h^r|_K, \underline{\varphi}_j^K \right) + b_h \left(\underline{\varphi}_j^K, p_h^{\text{int}}|_K + p_h^r|_K \right) &= \left(f, \underline{\varphi}_j^K \right), \quad j = 9, \dots, 16 \\ b_h \left(v_h^{\text{int}}|_K + v_h^r|_K, \underline{\psi}_j^K \right) &= 0, \quad j = 2, \dots, 4. \end{aligned} \tag{3.15}$$

It is simple to show that the two subspaces of the *interior* velocity and pressure functions are inf–sup-stable uniformly with respect to h . Therefore, the K -local problem (3.15) admits a unique and stable solution for each given pair (v_h^r, p_h^r) and each cell $K \in \mathcal{T}_h$. These local problems are equivalent to a linear 11×11 -system which can be solved directly by means of an optimized linear algebra subroutine. In the implementation of the static condensation, one solves on each mesh cell K simultaneously the corresponding local system for all possible pairs of basis functions for $(v_h^r|_K, p_h^r|_K)$ and stores the solutions. Thus, one has an efficient cell-wise matrix representation of the following solution operators

$$v_h^{\text{int}} = S_v(v_h^r, p_h^r, f), \quad p_h^{\text{int}} = S_p(v_h^r, p_h^r, f)$$

which assign to an arbitrary pair $(v_h^r, p_h^r) \in (v_{bh} + V_h^r) \times Q_h^r$ the corresponding *interior* parts $(v_h^{\text{int}}, p_h^{\text{int}})$, where the *remaining* spaces V_h^r and Q_h^r are given by

$$V_h^r := \text{span} \left\{ \begin{pmatrix} \varphi_j \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ \varphi_j \end{pmatrix}, \quad j = 1, \dots, N_{\text{EDG}} \right\}$$

and

$$Q_h^r := \{ q \in L^2(\Omega) : q|_K = \text{const} \quad \forall K \in \mathcal{T}_h \}.$$

Using the solution operators S_v and S_p , the *remaining* generalized Stokes system reads:

Find $(v_h^r, p_h^r) \in (v_{bh} + V_h^r) \times Q_h^r$ such that

$$\begin{aligned} a_h (S_v(v_h^r, p_h^r, f) + v_h^r, \varphi_h) + b_h (\varphi_h, S_p(v_h^r, p_h^r, f) + p_h^r) &= (f, \varphi_h) \\ &\quad \forall \varphi_h \in V_h^r \\ b_h (S_v(v_h^r, p_h^r, f) + v_h^r, q_h) &= 0 \\ &\quad \forall q_h \in Q_h^r. \end{aligned}$$

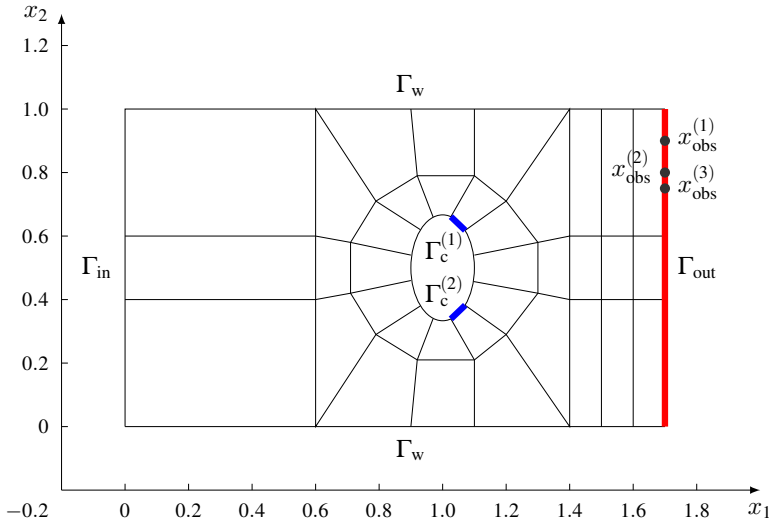


Figure 3. The coarse mesh for the feedback control flow problem with $L = 1.7$, $H = 1$, $N_c = 2$, $I_c^{(1)} = [7/12, 8/12]$ and $I_c^{(2)} = [11/12, 1]$. The straight-lined control boundary parts are marked with Γ_c .

The corresponding sparse linear system can be assembled efficiently and has a dimension of

$$N^r = \dim(V_h^r) + \dim(Q_h^r) = 2N_{\text{EDG}} + N_{\text{EL}}$$

where N_{EL} denotes the number of quadrilateral cells in the mesh \mathcal{T}_h . This dimension is about 3 times smaller than the dimension of the original system (3.9).

4. Numerical results

We have implemented a finite element package for quadrilaterals and programs for the *operator Newton-ADI method* in MATLAB. The resulting algebraic linear systems have been solved using the sparse direct solver provided by MATLAB. For the 2D-case, the computational efficiency of this solver is comparable to that of modern multigrid solvers.

4.1. Grid and data

In the following, we describe the problem data and the grid used for our test example of a channel flow around an obstacle. In this two dimensional example, the domain $\Omega = \Omega_r \setminus \overline{\Omega_e}$ is defined by a rectangular channel $\Omega_r = (0, L) \times (0, H)$ and a mostly elliptic-shaped obstacle Ω_e as depicted in Fig. 3. The boundary Γ_e of the obstacle Ω_e consists of N_c straight lines $\Gamma_c^{(k)}$ forming the control boundary $\Gamma_c = \bigcup_{k=1}^{N_c} \Gamma_c^{(k)}$ and the remaining wall part $\Gamma_w \cap \Gamma_e$ which has the elliptic-shaped parametrization

$$\gamma(s) = \begin{bmatrix} x_1^{e_0} \\ x_2^{e_0} \end{bmatrix} + \begin{bmatrix} r_1 \cos(2\pi(-s_0 - s)) \\ r_2 \sin(2\pi(-s_0 - s)) \end{bmatrix} \quad \forall s \in (0, 1] \setminus \bigcup_{k=1}^{N_c} I_c^{(k)}$$

where $I_c^{(k)}$ denote the parameter intervals of the control parts $\Gamma_c^{(k)}$ defined below. Furthermore, $(x_1^{e_0}, x_2^{e_0}) = (1, 1/2)$ denotes the center of the ellipse, $s_0 = 5/24$, and $r_1 = 1/10$, $r_2 = 1/6$ are the semi-axes of the ellipse. The parametrization of the control part $\Gamma_c^{(k)}$, $k = 1, \dots, N_c$, is defined as

$$\gamma(s) = \gamma(s_a^{(k)}) \frac{s_b^{(k)} - s}{s_b^{(k)} - s_a^{(k)}} + \gamma(s_b^{(k)}) \frac{s - s_a^{(k)}}{s_b^{(k)} - s_a^{(k)}} \quad \forall s \in I_c^{(k)} = [s_a^{(k)}, s_b^{(k)}].$$

On each part $\Gamma_c^{(k)}$ of the control boundary Γ_c , we prescribe at each time $t \in (0, \infty)$ a parabolic inflow profile of the velocity in normal direction to $\Gamma_c^{(k)}$ with the maximum value $u_k(t) \in \mathbb{R}$ (see Fig. 4). The case $u_k(t) < 0$ means that we prescribe an outflow done by exhausting fluid out of the model.

Once the control vector $u(t) \in \mathbb{R}^{N_c}$ is given for some time $t \in (0, T)$, we know all Dirichlet boundary data for the velocity at this time on the control boundary Γ_c . In order to define the remaining boundary conditions we decompose the whole boundary $\Gamma = \partial\Omega$ into

$$\Gamma = \Gamma_D \cup \Gamma_{out}, \quad \Gamma_D := \Gamma_{in} \cup \Gamma_w \cup \Gamma_c$$

where Γ_D denotes the Dirichlet-part and

$$\Gamma_{in} := \{(x_1, x_2) \in \Gamma : x_1 = 0\}, \quad \Gamma_{out} := \{(x_1, x_2) \in \Gamma : x_1 = L\}$$

$$\Gamma_w := \Gamma \setminus (\Gamma_{in} \cup \Gamma_{out} \cup \Gamma_c)$$

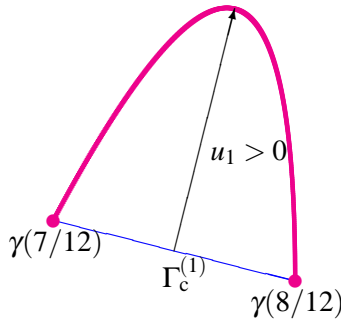


Figure 4. The parabolic inflow profile at the control boundary part $\Gamma_c^{(1)}$.

the inflow-, outflow- and wall-part of the boundary (see Fig. 3). On the outflow-part Γ_{out} , we impose the so-called *do-nothing* boundary condition

$$-v\nabla v(x,t) \cdot n(x) + p(t,x)n(x) = 0 \quad \forall (t,x) \in (0,\infty) \times \Gamma_{out} \quad (4.1)$$

where n denotes the outer unit normal vector on Γ_{out} . On the Dirichlet part Γ_D we prescribe

$$v(t,x) = \begin{cases} g_{in}(x), & (t,x) \in (0,\infty) \times \Gamma_{in} \\ 0, & (t,x) \in (0,\infty) \times \Gamma_w \\ g_c(t,x), & (t,x) \in (0,\infty) \times \Gamma_c. \end{cases} \quad (4.2)$$

The aim of our feedback control is to adjust the control vector $u(t) \in \mathbb{R}^{N_c}$ such that the resulting horizontal component of the velocity $v(t,x)$ is minimized at the observation points $x = x_{obs}^{(m)}$, $m = 1, \dots, N_{obs}$, located on the outflow boundary Γ_{out} (see Fig. 3) with $N_{obs} = 3$.

In the following, we choose the channel geometry with $L = 1.7$, $H = 1$ and set $N_c = 2$, $I_c^{(1)} = [7/12, 8/12]$, and $I_c^{(2)} = [11/12, 1]$ (see Fig. 3). In our computational tests, the coarse mesh at grid level $\ell = 1$ consists of $N_{EL} = 36$ cells, $N_{VT} = 52$ vertices, and $N_{EDG}^{all} = 88$ edges. The number of edges located at the Dirichlet boundary Γ_D at grid level $\ell = 1$ is $N_D = 29$ (see Fig. 3). The refined mesh at grid level $\ell = 4$ is depicted in Fig. 5.

4.2. Convergence study for non-conforming composite element

In this subsection, we study the *experimental order of convergence* (EOC) for our composite element pair. We consider the generalized Stokes problem

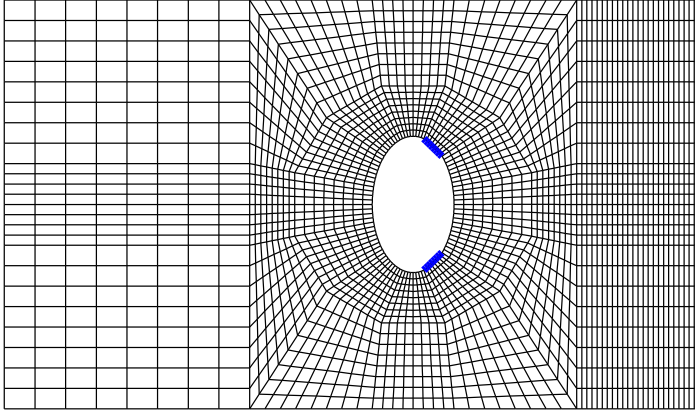


Figure 5. Refined mesh at level 4.

with $\nu = 1.0$ and $\mu = 0.5$ on a sequence of uniformly refined meshes starting from the level-1-mesh shown in Fig. 3. In order to be able to measure the actual error, we prescribe the exact velocity and pressure solution as

$$\begin{pmatrix} v_1(x_1, x_2) \\ v_2(x_1, x_2) \end{pmatrix} = \begin{pmatrix} \cos\left(\frac{\pi}{2L}x_1\right) \sin(\pi x_2) \\ -\frac{1}{2L} \sin\left(\frac{\pi}{2L}x_1\right) \cos(\pi x_2) \end{pmatrix}$$

$$p(x_1, x_2) = -\nu \frac{\pi}{2L} \sin\left(\frac{\pi}{2L}x_1\right) \sin(\pi x_2)$$

and choose the right-hand side f and the Dirichlet boundary conditions according to this solution.

In Table 1 we present, for different grid levels, the total numbers of the degrees of freedom for the velocity, pressure, and condensed spaces, respectively. We see that the reduction factor in the dimension of the condensed space compared to the usual velocity and pressure space is about 3.

The error of the discrete velocity v_h measured in the L^2 -norm as well as in the H^1 -semi-norm and the error of the pressure approximation p_h measured in the L^2 -norm are shown in Table 2. The achieved convergence rates are of optimal order.

In Table 3, we present the errors in the vertical velocity component $v_{h,2}$ evaluated at the observation points $x_{\text{obs}}^{(1)} = (L, 1/6)$ and $x_{\text{obs}}^{(2)} = (L, 2/3)$. We see that the computed errors $e_i := |v_2(x_{\text{obs}}^{(i)}) - v_{2,h}(x_{\text{obs}}^{(i)})|$, $i = 1, 2$, are of sec-

Table 1.

Degrees of freedom for the velocity, pressure, and condensed space.

Level	velocity	pressure	velocity & pressure	condensed
1	464	144	608	212
2	1792	576	2368	784
3	7040	2304	9344	3008
4	27904	9216	37120	11776
5	111104	36864	147968	46592

Table 2.Errors of velocity and pressure for the composite element pair $\mathbb{P}_1^{\text{nc}} / \mathbb{P}_0^{\text{dc}}$.

Level	$\ v - v_h\ _0$	EOC	$ v - v_h _1$	EOC	$\ p - p_h\ _0$	EOC
1	3.773e-2	—	4.307e-1	—	2.435e-1	—
2	1.096e-2	1.783	2.232e-1	0.948	9.278e-2	1.392
3	2.866e-3	1.935	1.128e-1	0.984	3.909e-2	1.247
4	7.252e-4	1.982	5.659e-2	0.996	1.839e-2	1.088
5	1.819e-5	1.995	2.832e-2	0.999	9.039e-3	1.025

ond order for sufficiently fine meshes. Note that the observation points have been chosen such that they do not coincide with any mesh point.

4.3. Feedback stabilization

In this subsection, we apply Algorithm 2.1. Therefore, we use the above described grid at refinement level 2, chose the observation points as

$$x_{\text{obs}}^{(1)} = (1.7, 0.9), \quad x_{\text{obs}}^{(2)} = (1.7, 0.8), \quad x_{\text{obs}}^{(3)} = (1.7, 0.75)$$

and compute the feedback operator for the Stokes equation with $\nu = 0.1$ via Algorithm 2.1. To show the influence of the feedback operator, we solve a discretized version of equation (2.7b) in MATLAB by assembling the matrices explicitly, using an initial non-steady-state condition, and compute a forward simulation of the resulting differential algebraic system with (*closed-loop*) and without (*open-loop*) the influence of the feedback. Because the Stokes equations are asymptotically stable, the compute feedback does not influence the evolution of the output, measured via the first summand $\|y\|^2$ in (2.7a), too much. To show the effect of feedback stabilization, we add two regularization parameters in the cost functional (2.7a) to penalize the output with λ and the control costs with $1/\rho$. This means we computed the

Table 3.
Errors at observation points for the vertical velocity component.

Level	e_1	EOC	e_2	EOC
1	6.861e-1	—	1.042e-2	—
2	3.675e-3	0.901	5.481e-4	4.250
3	1.842e-4	4.318	8.768e-5	2.644
4	4.288e-5	2.103	2.800e-5	1.647
5	1.121e-5	1.935	6.240e-6	2.166

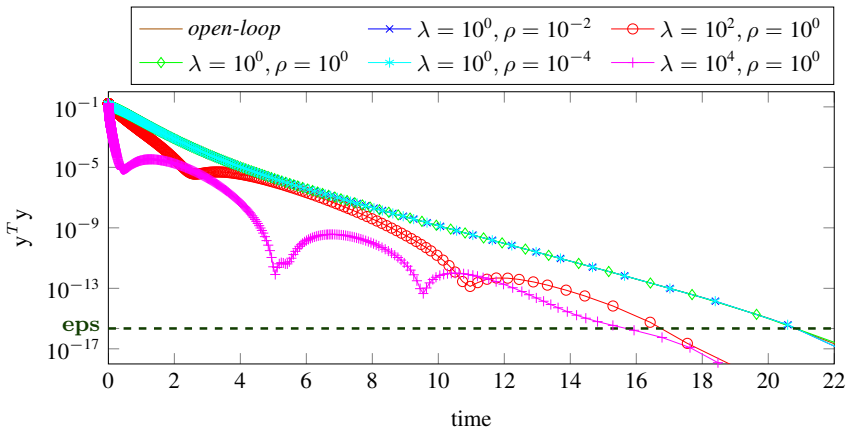


Figure 6. Evolution of output for different regularization parameters.

feedback for the adapted cost functional

$$\mathcal{J}(y, u) = \frac{1}{2} \int_0^\infty \lambda \|y\|^2 + \frac{1}{\rho} \|u\|^2 dt.$$

In Fig. 6 we compare the evolution of the output for different settings of (λ, ρ) . It shows that the trajectories of the *open-loop* simulation (without any feedback), for $\lambda = \rho = 1$, and for penalize the control costs are nearly the same. This means the controller notices that not doing anything is good enough to minimize the cost functional, especially if a large controller influence is penalized.

But if we set $\rho = 1$ and increase λ , we see that the controller forces the output faster to zero.

Of course, this behavior involves higher control costs. To show this, we compare the evolution of the optimal control in Fig. 7 measured via the second summand $\|u\|^2$ in (2.7a). We plotted the evolution for the same regularization parameter settings as above. The expected behavior can be observed.

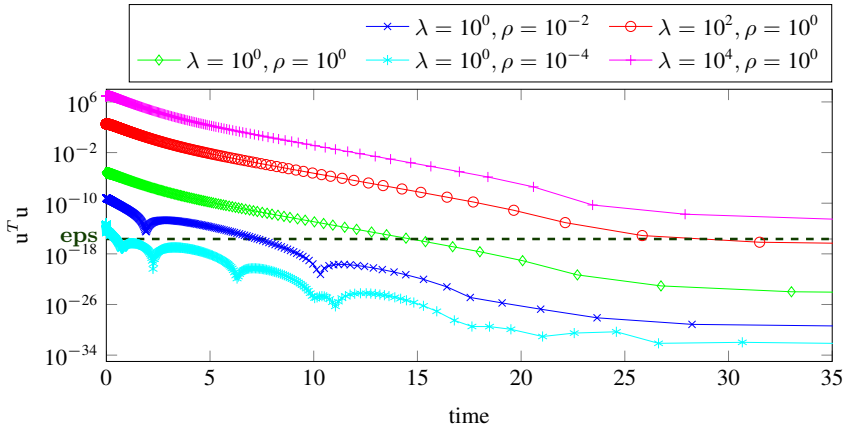


Figure 7. Evolution of control for different regularization parameters.

Table 4.

Newton-ADI convergence results for refinement level 2.

λ	ρ	matrix-version			operator-version		
		$\varnothing n_{\text{ADI}}$	n_{Newton}	$\text{error}_{\text{Newton}}$	$\varnothing n_{\text{ADI}}$	n_{Newton}	$\text{error}_{\text{Newton}}$
10^0	10^0	17.0	3	$2.76 \cdot 10^{-13}$	17.0	3	$2.78 \cdot 10^{-13}$
10^0	10^{-2}	17.0	2	$1.49 \cdot 10^{-6}$	17.0	2	$1.49 \cdot 10^{-6}$
10^0	10^{-4}	17.0	2	$1.99 \cdot 10^{-6}$	17.0	2	$1.99 \cdot 10^{-6}$
10^2	10^0	17.2	5	$1.01 \cdot 10^{-8}$	17.2	5	$1.01 \cdot 10^{-8}$
10^4	10^0	20.8	11	$1.64 \cdot 10^{-5}$	20.9	11	$1.64 \cdot 10^{-5}$

If we penalized the control costs, the controller becomes inactive quite fast, because the system is going to zero fast enough without any control. But if we force the system to endeavor a zero output faster, by increasing λ , we need significant higher control costs at the beginning.

These results will become more interesting if we consider unstable Navier–Stokes flows in the future. Additionally, we want to implement the feedback in a closed-loop simulation of the flow to see the stabilizing influence of the optimal control within a visualized simulation.

In the second part of this subsection we want to compare the convergence results of the operator Newton-ADI method with the matrix version presented in [4]. We assembled all matrices for the forward simulation explicitly. Now, we use these matrices to run the matrix based version of Algorithm 2.1 presented as Algorithm 2 in [4]. In both methods, we use the heuristic *Penzl* shifts and a stopping criteria of $\text{tol}_{\text{ADI}} = 2.5 \cdot 10^{-7}$ for the

ADI iteration and $\text{tol}_{\text{Newton}} = 2.5 \cdot 10^{-5}$ for the Newton iteration. The stopping criteria refer to the relative change of the computed feedback.

Table 4 shows that both methods converge within the same number of steps and achieve nearly the same relative error. Furthermore, we see the influence of the regularization parameters described above. If we penalize the output costs with ρ , the algorithm converges faster, because it knows that it does not need any control as described above. Otherwise, if we penalize the output with λ , the computation of the feedback needs more iterations to be the optimal for the chosen cost functional.

Summarizing, we see that computing the optimal control is useful in combination with the correct cost functional. We also could verify the matrix free approach to compute the optimal control for this kind of problems independent of the numerical discretization. This means we can use special discretizations, to avoid the drawbacks of standard mixed finite elements.

5. Conclusions and outlook

The proposed non-conforming composite element provides a good compromise of the advantages of quadrilateral and triangular elements. The numerical experiments show that the expected asymptotic orders of convergence are observed already for moderately sized meshes. The element is especially well suited for solving the linear quadratic regulator problem for the Stokes flow by the (operator) Newton-ADI method. This is due to both the preservation of the divergence constraint on the discrete level and the L^2 -orthogonality of the corresponding basis functions. The latter does not only increase the efficiency due to the diagonal structure of the resulting mass matrix, but also eliminates some term in the weak form of the equations that need extra treatment when standard finite elements are used.

Our approach allows a natural extension to the case of the Navier–Stokes equations and time-dependent working trajectory which will be the subjects of forthcoming research.

References

1. W.F. Arnold and A. J. Laub, Generalized eigenproblem algorithms and software for algebraic Riccati equations, *Proc. IEEE* (1984) **72**, 1746–1754.
2. E. Bänsch and P. Benner, Stabilization of Incompressible Flow Problems by Riccati-Based Feedback. In: *Constrained Optimization and Optimal Control for Partial Differ-*

- ential Equations* (Eds. G. Leugering, S. Engell, A. Griewank, M. Hinze, R. Rannacher, V. Schulz, M. Ulbrich, and S. Ulbrich), Int. Series of Numerical Mathematics, Vol. 160, Birkhäuser, 2012, pp. 5–20.
3. P. Benner, J.-R. Li, and T. Penzl, Numerical solution of large Lyapunov equations, Riccati equations, and linear-quadratic control problems, *Numer. Lin. Alg. Appl.* (2008) **15**, 755–777.
 4. P. Benner, J. Saak, M. Stoll, and H. K. Weichelt, Efficient solution of large-scale saddle point systems arising in Riccati-based boundary feedback stabilization of incompressible Stokes flow, DFG-SPP1253, *Preprint No. SPP1253-130*, 2012 (submitted to SISC Copper Mountain Special Section, 2012).
 5. J. H. Bramble and S. R. Hilbert, Estimation of linear functionals on Sobolev spaces with application to Fourier transforms and spline interpolation, *SIAM J. Numer. Anal.* (1970) **7**, 112–124.
 6. F. Brezzi and M. Fortin, *Mixed and Hybrid Finite Element Methods*, Springer Series in Computational Mathematics, Vol. 15, Springer-Verlag, New York, 1991.
 7. M. Crouzeix and P.-A. Raviart, Conforming and nonconforming finite element methods for solving the stationary Stokes equations, I. *Rev. Française Automat. Informat. Recherche Opérationnelle Sér. Rouge* (1973) **7**, 33–75.
 8. G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, 1996.
 9. M. Heinkenschloss, D. C. Sorensen, and K. Sun, Balanced truncation model reduction for a class of descriptor systems with application to the Oseen equations, *SIAM J. Sci. Comp.* (2008) **30**, 1038–1063.
 10. M. Hinze and K. Kunisch, Second order methods for boundary control of the instationary Navier–Stokes system, *ZAMM Z. Angew. Math. Mech.* (2004) **84**, 171–187.
 11. P. Hood and C. Taylor, Navier–Stokes equations using mixed interpolation. In: *Finite Element Methods in Flow Problems* (Eds. J. T. Oden, R. H. Gallagher, C. Taylor, and O. C. Zienkiewicz), University of Alabama in Huntsville Press, 1974, pp. 121–132.
 12. D. L. Kleinman, On an Iterative Technique for Riccati Equation Computations, *IEEE Trans. Automat. Control* (1968) **AC-13**, 114–115.
 13. P. Lancaster and L. Rodman, *The Algebraic Riccati Equation*, Oxford University Press, Oxford, 1995.
 14. 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.
 15. J.-R. Li and J. White, Low rank solution of Lyapunov equations, *SIAM J. Matrix Anal. Appl.* (2002) **24**, 260–280.
 16. J. L. Lions, *Optimal Control of Systems Governed by Partial Differential Equations*, Berlin, FRG, 1971.
 17. A. Locatelli, *Optimal Control: An Introduction*, Birkhäuser Verlag, Basel–Boston–

- Berlin, 2001.
18. T. Penzl, *Lyapack Users Guide*, Sonderforschungsbereich, 393, Numerische Simulation auf massiv parallelen Rechnern, TU Chemnitz, *Report No. SFB393/00-33*, 09107 Chemnitz, Germany, 2000.
 19. J.-P. Raymond, Feedback boundary stabilization of the two-dimensional Navier–Stokes equations, *SIAM J. Control Optim.* (2006) **45**, 790–828.
 20. J.-P. Raymond and L. Thevenet, Boundary feedback stabilization of the two dimensional Navier–Stokes equations with finite dimensional controllers, *Discrete Contin. Dyn. Syst.* (2010) **27**, 1159–1187.
 21. T. Reis and W. Wollner, Finite-rank ADI iteration for operator Lyapunov equations, *Hamburger Beiträge zur Angewandten Mathematik*, *Preprint No. 2012-09*, 2012.
 22. J. Weickert, Navier–Stokes equations as a differential-algebraic system, Department of Mathematics, Chemnitz University of Technology, *Preprint No. SFB393/96-08*, August 1996.

