Optimal Control-Based Feedback Stabilization of Multi-field Flow Problems

Eberhard Bänsch, Peter Benner, Jens Saak, and Heiko K. Weichelt

Abstract We discuss the numerical solution of the feedback stabilization problem for multi-field flow problems. Our approach is based on an analytical Riccati feedback concept derived by Raymond which allows to steer a perturbed flow back to its desired state, assumed to be a stationary, possibly unstable, flow profile. This concept, originally derived for incompressible flow fields described by the Navier-Stokes equations, uses a linear-quadratic regulator (LQR) approach for the linearized Navier-Stokes equations formulated on the space of divergence-free velocity fields. We extend this approach to a setting where the Navier-Stokes equations are coupled to a diffusion-convection equation describing the transport of a reactive species in a fluid. The stabilizing feedback control resulting from the LQR problem is obtained via solving the associated operator Riccati equation. We describe a numerical procedure to solve this Riccati equation numerically. This involves several technical difficulties on the algebraic level that we address in this report. We illustrate the performance of our method by a numerical example.

Keywords Coupled flow control • Navier-Stokes equations • Diffusion-convection equation • Riccati-based feedback

E. Bänsch
Lehrstuhl für Angewandte Mathematik 3, Friedrich-Alexander-Universität Erlangen-Nürnberg, Cauerstr. 11, 91058 Erlangen, Germany
e-mail: baensch@am.uni-erlangen.de

P. Benner • J. Saak
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

Faculty of Mathematics, Research Group Mathematics in Industry and Technology (MiIT), Technische Universität Chemnitz, Reichenhainer Str. 39/41, 09126 Chemnitz, Germany
e-mail: benner@mpi-magdeburg.mpg.de; saak@mpi-magdeburg.mpg.de

H. K. Weichelt (✉)
Faculty of Mathematics, Research Group Mathematics in Industry and Technology (MiIT), Technische Universität Chemnitz, Reichenhainer Str. 39/41, 09126 Chemnitz, Germany
e-mail: weichelt@mpi-magdeburg.mpg.de

G. Leugering et al. (eds.), *Trends in PDE Constrained Optimization*, International Series of Numerical Mathematics 165, DOI 10.1007/978-3-319-05083-6_11

Mathematics Subject Classification (2010). 76D55,93D15,93C20,15A24.

1 Introduction

In this article we extend the ideas from [4, 5, 8, 18] for a closed-loop (boundary) control of the linearized Navier-Stokes equations to a coupled flow problem consisting of the Navier-Stokes equations and a diffusion-convection problem. The latter models passive transport of a reactive species by the flow field. A homogeneous Dirichlet condition on part of the boundary of the domain is used as a toy model for surface reaction. Although this setting is rather academic, it may serve as a paradigm to solve more involved problems.

In the present paper we focus on the computational realization of the feedback control for the coupled flow problem. More details about the underlying mathematical basis may be found in [5, 18]. The present article builds upon [4] and demonstrates how the numerical solution concept outlined there is realized for a multi-field flow problem.

Let us mention some related work. The numerical realization of a linear-quadratic regularization process applied to the Stokes equations is discussed in [8], where the focus lies on efficient solution strategies for the arising saddle point systems. These ideas are extended to the more general Navier-Stokes equations in [5]. Furthermore, a different approach to stabilize Navier-Stokes flow problems via boundary influence is shown in [1]. Extending these ideas and numerical techniques to a more general coupled multi-field flow problem is the main issue of this paper.

The rest of this article is organized as follows. In Sect. 2 the coupled flow problem is stated. Section 2.1 outlines the necessary concept of linearization. Discretization by finite elements leads to a finite dimensional system of differential-algebraic equations (DAEs), which is described in Sect. 2.2. Before we introduce the problem setting used for numerical testing in the Sect. 3, we discuss some details about the block structured saddle point systems in Sect. 2.4. The paper is concluded in Section "Conclusions and Outlook", where we discuss some further ideas that are part of ongoing research.

2 The Coupled Flow Problem

The basis for the coupled flow problems is described by the incompressible Navier-Stokes equations defined on a spatial inflow/outflow domain $\Omega \subset \mathbb{R}^2$ and the time interval of interest $\mathcal{I} \subset [0, \infty)$. For $x \in \Omega$, $t \in \mathcal{I}$, the velocity $\vec{v}(t, \vec{x}) \in \mathbb{R}^2$ and pressure $p(t, \vec{x}) \in \mathbb{R}$ satisfy

$$\frac{\partial}{\partial t}\vec{v} - \frac{1}{\mathrm{Re}}\Delta\vec{v} + (\vec{v}\cdot\nabla)\vec{v} + \nabla\chi = \vec{f}, \tag{2.1a}$$

$$\mathrm{div}\vec{v} = 0 \tag{2.1b}$$

with the Reynolds number Re, see [5].

Moreover, we consider the passive transport of some concentration field $c(t, \vec{x}) \in \mathbb{R}$ described by a diffusion-convection equation (DCE):

$$\frac{\partial}{\partial t} c - \frac{1}{\mathrm{ReSc}} \Delta c + (\vec{v} \cdot \nabla) c = 0 \tag{2.2}$$

with the Schmidt number Sc.

The joint evolution for both systems takes place within $\Omega \times \mathcal{I}$ with the boundary

$$\partial\Omega =: \Gamma = \Gamma_{in} \cup \Gamma_{wall} \cup \Gamma_{out} \cup \Gamma_r .$$

In this decomposition of the boundary, Γ_r denotes the boundary of an obstacle within the domain. For (2.1), we prescribe a parabolic inflow at Γ_{in}, "no-slip" boundary conditions at Γ_{wall} and Γ_r, and "do-nothing" conditions for the outflow. The initial condition is given by a stationary solution to (2.1). We impose the following boundary and initial conditions for the concentration:

$$c(t, \vec{x}) = h_{in}(\vec{x}) \quad \text{on } \Gamma_{in}, \tag{2.3a}$$

$$\frac{\partial c(t, \vec{x})}{\partial \vec{n}} = 0 \quad \text{on } \Gamma_{wall} \cup \Gamma_{out}, \tag{2.3b}$$

$$c(t, \vec{x}) = 0 \quad \text{on } \Gamma_r, \tag{2.3c}$$

$$c(0, \vec{x}) = 0 \quad \text{in } \Omega \tag{2.3d}$$

with $\vec{n}$ the outward normal to Γ_{out} and Γ_{wall}. The initial and boundary conditions can be interpreted as follows. The concentration enters the domain through Γ_{in} (2.3a) and leaves the domain, only convection driven, via Γ_{out} (2.3b). As soon as the concentration reaches the obstacle, a fast reaction is assumed absorbing the species. In the case considered here, the reaction is much faster than the transport in Ω and can thus be modeled by a homogeneous Dirichlet condition for $c(t, \vec{x})$ (2.3c). We will omit the arguments $t, \vec{x}$ hereafter for better readableness.

2.1 Linearization

In this section we show how to linearize equations (2.1) and (2.2). Linearization is necessary for applying the Riccati based feedback stabilization approach to the coupled system. Using the linearization idea [5, Section 2.1], we define

$$\vec{z} := \vec{v} - \vec{w}, \tag{2.4a}$$

$$p := \chi - \chi_s \tag{2.4b}$$

that fulfill the linearized equations

$$\frac{\partial}{\partial t}\vec{z} - \frac{1}{\mathrm{Re}}\Delta\vec{z} + (\vec{w}\cdot\nabla)\vec{z} + (\vec{z}\cdot\nabla)\vec{w} + \nabla p = 0, \tag{2.5a}$$

$$\mathrm{div}\vec{z} = 0 \tag{2.5b}$$

with the same boundary and initial conditions, as in [5]. Furthermore, we define the stationary diffusion-convection equation

$$-\frac{1}{\mathrm{ReSc}}\Delta c_{\vec{w}} + (\vec{w}\cdot\nabla)c_{\vec{w}} = 0 \tag{2.6}$$

with similar boundary conditions like those in (2.3). The goal now is to stabilize $c_{\vec{w}}$, when this field is subject to (small) perturbations. We may for instance imagine the situation when the field arises from an open-loop controller [13] and thus is a desired state to be maintained. Let us define

$$c_{\vec{z}} = c - c_{\vec{w}} \tag{2.7}$$

as the difference between the actual concentration c and the stationary concentration $c_{\vec{w}}$. Using the linearization points (2.4), (2.7) together with (2.6) yields the linearized diffusion-convection equation

$$\frac{\partial}{\partial t}c_{\vec{z}} - \frac{1}{\mathrm{ReSc}}\Delta c_{\vec{z}} + (\vec{w}\cdot\nabla)c_{\vec{z}} + (\vec{z}\cdot\nabla)c_{\vec{w}} = 0, \tag{2.8}$$

defined for $t \in \mathcal{I}$ and $\vec{x} \in \Omega$ with boundary and initial conditions

$$\begin{aligned} c_{\vec{z}} &= 0 \quad \text{on } \Gamma_{in} \cup \Gamma_r, \\ \frac{\partial c_{\vec{z}}}{\partial \vec{n}} &= 0 \quad \text{on } \Gamma_{wall} \cup \Gamma_{out}, \\ c_{\vec{z}}(0,\vec{x}) &= 0 \quad \text{in } \Omega. \end{aligned}$$

The main goal of boundary feedback stabilization is the asymptotic stabilization of $\vec{z}$ and $c_{\vec{z}}$, which implies that the actual velocity field fulfills $\vec{v} \approx \vec{w}$ and the actual concentration $c \approx c_{\vec{w}}$, respectively. In the following we are going to apply a finite dimensional LQR approach, based on a spatial semi-discretization of the linearized Navier-Stokes (2.5) and diffusion-convection (2.8) equations. The discretization by finite elements is described in the following subsection.

2.2 Discretization

We use the same discretization idea for (2.5) as described in [5, Section 2.2] using the $\mathcal{P}_2 - \mathcal{P}_1$ *Taylor-Hood* [14] element and end up with the discretized linearized Navier-Stokes equations

$$M_{\mathbf{z}} \frac{d}{dt}\mathbf{z}(t) = A_{\mathbf{z}}\mathbf{z}(t) + G\mathbf{p}(t) + \mathbf{f}_{\mathbf{z}}(t), \tag{2.9a}$$

$$0 = G^T\mathbf{z}(t) \tag{2.9b}$$

with n_v degrees of freedom for the velocity space and n_p degrees of freedom for the pressure space. Equation (2.8) is discretized in space by linear finite elements yielding

$$M_{\mathbf{c}} \frac{d}{dt}\mathbf{c}(t) = A_{\mathbf{c}}\mathbf{c}(t) - R_{\vec{w}}\mathbf{z} + \mathbf{f}_{\mathbf{c}}(t) \tag{2.10}$$

with the nodal vector of discretized concentrations $\mathbf{c}(t) \in \mathbb{R}^{n_c}$, the concentration mass matrix $M_{\mathbf{c}} = M_{\mathbf{c}}^T \succ 0 \in \mathbb{R}^{n_c \times n_c}$, the concentration system matrix $A_{\mathbf{c}} \in \mathbb{R}^{n_c \times n_c}$, and the reaction term $R_{\vec{w}}$ that depends on the stationary velocity $\vec{w}$ and couples (2.9) and (2.10). Similar to the velocity field, the concentration field may be subject to a control $\mathbf{u}_{\mathbf{c}}$ acting through the source term $\mathbf{f}_{\mathbf{c}}(t) := B_{\mathbf{c}}\mathbf{u}_{\mathbf{c}}(t)$.

After reordering (2.9) and (2.10), we obtain the system of DAEs [15]

$$\begin{bmatrix} M_{\mathbf{z}} & 0 & 0 \\ 0 & M_{\mathbf{c}} & 0 \\ 0 & 0 & 0 \end{bmatrix} \frac{d}{dt} \begin{bmatrix} \mathbf{z} \\ \mathbf{c} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} A_{\mathbf{z}} & 0 & G \\ -R_{\vec{w}} & A_{\mathbf{c}} & 0 \\ G^T & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{z} \\ \mathbf{c} \\ \mathbf{p} \end{bmatrix} + \begin{bmatrix} B_{\mathbf{z}} & 0 \\ 0 & B_{\mathbf{c}} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}_{\mathbf{z}} \\ \mathbf{u}_{\mathbf{c}} \end{bmatrix}. \tag{2.11a}$$

We assume that only parts of the states $\mathbf{z}$ and $\mathbf{c}$ are observed. Therefore, we add the observation equations

$$\begin{bmatrix} \mathbf{y}_{\mathbf{z}} \\ \mathbf{y}_{\mathbf{c}} \end{bmatrix} = \begin{bmatrix} C_{\mathbf{z}} & 0 \\ 0 & C_{\mathbf{c}} \end{bmatrix} \begin{bmatrix} \mathbf{z} \\ \mathbf{c} \end{bmatrix}. \tag{2.11b}$$

The DAE system (2.11) is of differential index 2 if G has full rank [19]. Since we use the inf-sup stable Taylor–Hood element, the latter condition is fulfilled. Defining the block matrices

$$\begin{aligned} M &= \begin{bmatrix} M_{\mathbf{z}} & 0 \\ 0 & M_{\mathbf{c}} \end{bmatrix}, \quad A = \begin{bmatrix} A_{\mathbf{z}} & 0 \\ -R_{\vec{w}} & A_{\mathbf{c}} \end{bmatrix}, \quad \tilde{G} = \begin{bmatrix} G \\ 0 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} \mathbf{z} \\ \mathbf{c} \end{bmatrix}, \\ B &= \begin{bmatrix} B_{\mathbf{z}} & 0 \\ 0 & B_{\mathbf{c}} \end{bmatrix}, \quad C = \begin{bmatrix} C_{\mathbf{z}} & 0 \\ 0 & C_{\mathbf{c}} \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} \mathbf{u}_{\mathbf{z}} \\ \mathbf{u}_{\mathbf{c}} \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} \mathbf{y}_{\mathbf{z}} \\ \mathbf{y}_{\mathbf{c}} \end{bmatrix}, \end{aligned} \tag{2.12}$$

(2.9) can be written as

$$\begin{bmatrix} M & 0 \\ 0 & 0 \end{bmatrix} \frac{d}{dt} \begin{bmatrix} \mathbf{x} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} A & \tilde{G} \\ \tilde{G}^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{p} \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} \mathbf{u}, \tag{2.13a}$$

$$\mathbf{y} = C\mathbf{x}. \tag{2.13b}$$

The matrix pencil

$$\left(\begin{bmatrix} A & \tilde{G} \\ \tilde{G}^T & 0 \end{bmatrix}, \begin{bmatrix} M & 0 \\ 0 & 0 \end{bmatrix} \right)$$

of the DAE (2.13) has $n_v + n_c - n_p$ finite eigenvalues $\lambda_i \in \mathbb{C}$ and $2n_p$ infinite eigenvalues $\lambda_\infty = \infty$ [10].

The DAE system (2.13) has the same structure as the DAE system arising from the Navier-Stokes equations in [5]; there, the projection approach of [12] is applied to transform the DAE [5, Equation (6)] into a generalized state space system. The main difficulty here is that we only want to project the velocity part $\mathbf{z}$ of the state variable $\mathbf{x}$. In the next subsection we show that this is indeed possible by adapting the projection idea from [12] to our block structured DAE system (2.13).

2.3 Projection Method

In order to adapt the projector definition of [12] to the case of our block matrices (2.12), we define

$$\begin{aligned} \tilde{\Pi} &:= I_{\mathbf{x}} - \tilde{G}(\tilde{G}^T M^{-1} \tilde{G})^{-1} \tilde{G}^T M^{-1} \\ &= \begin{bmatrix} I_{\mathbf{v}} & 0 \\ 0 & I_{\mathbf{c}} \end{bmatrix} - \begin{bmatrix} G \\ 0 \end{bmatrix} \left(\begin{bmatrix} G^T & 0 \end{bmatrix} \begin{bmatrix} M_{\mathbf{z}}^{-1} & 0 \\ 0 & M_{\mathbf{c}}^{-1} \end{bmatrix} \begin{bmatrix} G \\ 0 \end{bmatrix} \right)^{-1} \begin{bmatrix} G^T & 0 \end{bmatrix} \begin{bmatrix} M_{\mathbf{z}}^{-1} & 0 \\ 0 & M_{\mathbf{c}}^{-1} \end{bmatrix} \\ &= \begin{bmatrix} I_{\mathbf{v}} & 0 \\ 0 & I_{\mathbf{c}} \end{bmatrix} - \begin{bmatrix} G(G^T M_{\mathbf{z}}^{-1} G)^{-1} G^T M_{\mathbf{z}}^{-1} & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} \Pi & 0 \\ 0 & I_{\mathbf{c}} \end{bmatrix} \end{aligned}$$

with the discrete *Helmholtz* projector $\Pi \in \mathbb{R}^{n_v \times n_v}$ as defined in [5]. Note that we have the following equivalences:

$$\Pi^T \mathbf{z} = \mathbf{z} \ \wedge \ \mathbf{c} = \mathbf{c} \quad \Leftrightarrow \quad \begin{bmatrix} \Pi & 0 \\ 0 & I_{\mathbf{c}} \end{bmatrix}^T \begin{bmatrix} \mathbf{z} \\ \mathbf{c} \end{bmatrix} = \begin{bmatrix} \mathbf{z} \\ \mathbf{c} \end{bmatrix} \quad \Leftrightarrow \quad \tilde{\Pi}^T \mathbf{x} = \mathbf{x}.$$

Using (formally) the decomposition $\Pi = \Theta_l \Theta_r^T$ as in [12, Section 3], the projection matrix $\tilde{\Pi}$ can be decomposed into

$$\tilde{\Pi} = \tilde{\Theta}_l \tilde{\Theta}_r^T \quad \Leftrightarrow \quad \begin{bmatrix} \Pi & 0 \\ 0 & I_{\mathbf{c}} \end{bmatrix} = \underbrace{\begin{bmatrix} \Theta_l & 0 \\ 0 & I_{\mathbf{c}} \end{bmatrix}}_{=:\tilde{\Theta}_l} \underbrace{\begin{bmatrix} \Theta_r^T & 0 \\ 0 & I_{\mathbf{c}} \end{bmatrix}}_{=:\tilde{\Theta}_r^T}$$

with $\tilde{\Theta}_l^T \tilde{\Theta}_r = I_{\mathbf{x}}$. This decomposition is used to project the discretized velocity $\mathbf{z}$ onto the $n_v - n_p$ dimensional subspace of discretely divergence-free functions as in [12], without changing the discrete concentration $\mathbf{c}$. Substituting

$$\tilde{\mathbf{x}} = \tilde{\Theta}_l^T \mathbf{x} = \begin{bmatrix} \Theta_l^T & 0 \\ 0 & I_{\mathbf{c}} \end{bmatrix} \begin{bmatrix} \mathbf{z} \\ \mathbf{c} \end{bmatrix} = \begin{bmatrix} \Theta_l^T \mathbf{z} \\ \mathbf{c} \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{z}} \\ \mathbf{c} \end{bmatrix} \in \mathbb{R}^{(n_v - n_p) + n_c}$$

in (2.13) yields

$$\tilde{\Theta}_r^T M \tilde{\Theta}_r \frac{d}{dt} \tilde{\mathbf{x}} = \tilde{\Theta}_r^T A \tilde{\Theta}_r \tilde{\mathbf{x}} + \tilde{\Theta}_r^T B \mathbf{u},$$
$$\mathbf{y} = C \tilde{\Theta}_r \tilde{\mathbf{x}}.$$

We define the projected block structured matrices

$$\begin{aligned} \mathcal{M} &= \begin{bmatrix} \Theta_r^T M_{\mathbf{z}} \Theta_r & 0 \\ 0 & M_{\mathbf{c}} \end{bmatrix}, \quad \mathcal{A} = \begin{bmatrix} \Theta_r^T A_{\mathbf{z}} \Theta_r & 0 \\ -R_{\vec{w}} \Theta_r & A_{\mathbf{c}} \end{bmatrix}, \\ \mathcal{B} &= \begin{bmatrix} \Theta_r^T B_{\mathbf{z}} & 0 \\ 0 & B_{\mathbf{c}} \end{bmatrix}, \quad \mathcal{C} = \begin{bmatrix} C_{\mathbf{z}} \Theta_r & 0 \\ 0 & C_{\mathbf{c}} \end{bmatrix} \end{aligned} \tag{2.14}$$

and end up with the generalized state space system

$$\mathcal{M} \frac{d}{dt} \tilde{\mathbf{x}} = \mathcal{A} \tilde{\mathbf{x}} + \mathcal{B} \mathbf{u}, \tag{2.15a}$$
$$\mathbf{y} = \mathcal{C} \tilde{\mathbf{x}} \tag{2.15b}$$

with $\mathcal{M} = \mathcal{M}^T \succ 0 \in \mathbb{R}^{(n_v - n_p) + n_c}$.

2.4 The Linear-Quadratic Regulator Approach

To test the feedback stabilization approach for a coupled flow problem let us define

$$q := \int_{\Gamma_r} \partial_{\vec{n}} c_{\vec{w}} \, \mathrm{ds} \tag{2.16}$$

as the total flux of the stationary concentration $c_{\vec{w}}$ through the obstacle boundary Γ_r. Analogously to the LQR approach in [5] we define the cost functional

$$\mathcal{J}(c, \mathbf{u}(t)) := \frac{1}{2} \int_0^\infty \lambda \left| \int_{\Gamma_r} \partial_{\vec{n}} c \ \mathrm{ds} \ - q \right|^2 + |\mathbf{u}(t)|^2 \ \mathrm{dt} \tag{2.17}$$

measuring the difference of the actual flux of c through Γ_r and q, as well as the control costs $\mathbf{u}$, in the square of the Euclidean norm. Using the definition (2.16) in (2.17) we obtain

$$\int_{\Gamma_r} \partial_{\vec{n}} c \ \mathrm{ds} \ - q = \int_{\Gamma_r} \partial_{\vec{n}} (c - c_{\vec{w}}) \ \mathrm{ds} = \int_{\Gamma_r} \partial_{\vec{n}} c_{\vec{z}} \ \mathrm{ds}.$$

After discretization, this yields

$$C_{\mathbf{c}} \mathbf{c} = \mathbf{y}_{\mathbf{c}}$$

as the observation equation in (2.11b). We do not consider any observation of the velocity field such that we set $C_{\mathbf{z}} = 0$ and reduce the output Equation (2.11b) to $\mathbf{y} = C_{\mathbf{c}} \mathbf{c}$. Using this output equation, the minimization problem for the LQR approach can be written as:

Minimize

$$\mathcal{J}(\mathbf{c}(t), \mathbf{u}(t)) := \frac{1}{2} \int_0^\infty \lambda \left(\mathbf{c}(t)^T C_{\mathbf{c}}^T C_{\mathbf{c}} \mathbf{c}(t) \right) + \mathbf{u}(t)^T \mathbf{u}(t) \ \mathrm{dt}, \tag{2.18}$$

subject to (2.15a).

Minimizing this cost functional subject to (2.15a) forces the discrete velocity field $\mathbf{z}$ and concentration $\mathbf{c}$ asymptotically to zero for $t \to \infty$ so that the actual flow field $\vec{v}$ and concentration c are expected to approach the stationary velocity field $\vec{w}$ and the concentration $c_{\vec{w}}$, respectively. Introducing the regularization parameter λ in the first term of (2.18) provides the possibility to achieve qualitatively different results.

On the one hand, we observe only parts of the concentration $\mathbf{c}$. On the other hand, we want to influence the whole system only via a control influence on the velocity field $\mathbf{z}$; that means we define $B_{\mathbf{c}} = 0$ and reduce the control input to $\begin{bmatrix} B_{\mathbf{z}}^T & 0 & 0 \end{bmatrix}^T \mathbf{u}_z$ in (2.11). We will skip details about the realization of $B_{\mathbf{z}}$.

Starting from the setting to minimize (2.18) subject to (2.15a), the whole process to compute the optimal control $\mathbf{u}_*(t) = -\mathcal{K} \tilde{x}_*(t)$ with the feedback $\mathcal{K}$ via a generalized Newton-ADI iteration analogous to [5] is used. In short, this method consists in applying Newton's method to the algebraic Riccati equation obtained from the LQR problem after (implicitly) projecting onto the space of discretely divergence-free functions. In each step of Newton's method applied to an algebraic Riccati equation, a Lyapunov equation has to be solved. This is a linear system of equations having tensor structure. As suggested in [6], we employ the alternating

directions implicit (ADI) method for this purpose. This requires the solution of a linear system of equations involving the projected system matrices (2.14) in each ADI step, see also [9]. How to avoid the explicit formation of the projected matrices following the approach from [12] is discussed in detail in [5] for the Navier-Stokes case. In the following, we will adapt this to the case of the multi-field flow problem discussed here. Therefore, we consider a projected system of the form

$$\left((\mathcal{A}^{(m)})^T + q_i \mathcal{M}\right) \Lambda = \mathcal{Y} \tag{2.19}$$

in the innermost step of the nested Newton-ADI iteration. Equation (2.19) is of the same structure as in the Navier-Stokes case and the approach in [12] to avoid this explicit projection can be applied in a similar way for the coupled flow problem. To this end we observe that the solution Λ is determined by solving the linear system

$$\tilde{\Pi} \left((A - BK^{(m)})^T + q_i M\right) \tilde{\Pi}^T \Lambda = \tilde{\Pi} Y,$$

which is equivalent to solving the saddle point system

$$\begin{bmatrix} (A - BK^{(m)})^T + q_i M & \tilde{G} \\ \tilde{G}^T & 0 \end{bmatrix} \begin{bmatrix} \Lambda \\ * \end{bmatrix} = \begin{bmatrix} Y \\ 0 \end{bmatrix} \tag{2.20}$$

with the feedback matrix $K^{(m)}$ in the m-th Newton step and q_i the ADI shift in the i-th ADI step. ("$*$" denotes an auxiliary quantity not further used.)

The feedback matrix K can then be computed via the generalized low-rank Cholesky factor Newton method as it is shown in Algorithm 1. The whole algorithm uses the original large-scale and sparse matrices from (2.11). We will skip details about shift selection in this paper and refer to [5, 16].

The Newton iteration consists of a number of Newton steps, each of which requires a certain amount of ADI steps to determine the update for the Newton iteration [5] and, in our formulation, to directly update the feedback matrix K. In turn, the saddle point system (2.20) has to be solved for different ADI shifts q_i in every ADI step for a couple of right hand sides during the Newton-ADI iteration. Solving the large-scale saddle point system efficiently is crucial for a suitable computation time.

Following the algebra in [5], where the *Sherman-Morrison-Woodbury* formula is exploited, one eventually ends up with a system to be solved having the form

$$\begin{bmatrix} A^T + q_i M & \tilde{G} \\ \tilde{G}^T & 0 \end{bmatrix} \begin{bmatrix} \Lambda \\ * \end{bmatrix} = \begin{bmatrix} \tilde{Y} \\ 0 \end{bmatrix}.$$

Using the block matrix definitions (2.12) yields

$$\underbrace{\begin{bmatrix} A_{\mathbf{z}}^T + q_i M_{\mathbf{z}} & -R_{\tilde{w}}^T & G \\ 0 & A_{\mathbf{c}}^T + q_i M_{\mathbf{c}} & 0 \\ G^T & 0 & 0 \end{bmatrix}}_{=:\mathbf{A}} \begin{bmatrix} \Lambda_{\mathbf{z}} \\ \Lambda_{\mathbf{c}} \\ * \end{bmatrix} = \begin{bmatrix} \tilde{Y}_{\mathbf{z}} \\ \tilde{Y}_{\mathbf{c}} \\ 0 \end{bmatrix}$$

Algorithm 1 Generalized low-rank Cholesky factor Newton method for coupled flow problems

Input: $M_{\mathbf{z}}, M_{\mathbf{c}}, A_{\mathbf{z}}, A_{\mathbf{c}}, G, R_{\vec{w}}, B_{\mathbf{z}}, C_{\mathbf{c}}$, initial feedback $K_{\mathbf{z}}^{(0)}$, ADI shift parameters $q_i \in \mathbb{C}^- : i = 1, \ldots, n_{\text{ADI}}$, tol_{ADI}, tol_{Newton}, and regularization parameter λ

Output: feedback operator K

for $m = 1, 2, \ldots, n_{\text{Newton}}$ **do**

$$W^{(m)} = \begin{bmatrix} 0 & \sqrt{\lambda}\, C_{\mathbf{c}} & 0 \\ (K_{\mathbf{z}}^{(m-1)}) & 0 & 0 \end{bmatrix}$$

Get $\begin{bmatrix} V_{1,\mathbf{z}}^T & V_{1,\mathbf{c}}^T \end{bmatrix}^T$ by solving

$$\begin{bmatrix} A_{\mathbf{z}}^T - (K_{\mathbf{z}}^{(m-1)})^T B_{\mathbf{z}}^T + q_1 M_{\mathbf{z}} & -R_{\vec{w}}^T & G \\ 0 & A_{\mathbf{c}}^T + q_1 M_{\mathbf{c}} & 0 \\ G^T & 0 & 0 \end{bmatrix} \begin{bmatrix} V_{1,\mathbf{z}} \\ V_{1,\mathbf{c}} \\ * \end{bmatrix} = \sqrt{-2\text{Re}\,(q_1)}\,(W^{(m)})^T$$

$K_{1,\mathbf{z}}^{(m)} = \begin{bmatrix} B_{\mathbf{z}}^T V_{1,\mathbf{z}} \overline{V}_{1,\mathbf{z}}^T M_{\mathbf{z}} & B_{\mathbf{z}}^T V_{1,\mathbf{z}} \overline{V}_{1,\mathbf{c}}^T M_{\mathbf{c}} \end{bmatrix}$

for $i = 2, 3, \ldots, n_{\text{ADI}}$ **do**

Get $\begin{bmatrix} \tilde{V}_{\mathbf{z}}^T & \tilde{V}_{\mathbf{c}}^T \end{bmatrix}^T$ by solving

$$\begin{bmatrix} A_{\mathbf{z}}^T - (K_{\mathbf{z}}^{(m-1)})^T B_{\mathbf{z}}^T + q_i M_{\mathbf{z}} & -R_{\vec{w}}^T & G \\ 0 & A_{\mathbf{c}}^T + q_i M_{\mathbf{c}} & 0 \\ G^T & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{V}_{\mathbf{z}} \\ \tilde{V}_{\mathbf{c}} \\ * \end{bmatrix} = \begin{bmatrix} M_{\mathbf{z}}\, V_{i-1,\mathbf{z}} \\ M_{\mathbf{c}}\, V_{i-1,\mathbf{c}} \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} V_{i,\mathbf{z}} \\ V_{i,\mathbf{c}} \end{bmatrix} = \sqrt{\text{Re}\,(q_i)/\text{Re}\,(q_{i-1})} \left(\begin{bmatrix} V_{i-1,\mathbf{z}} \\ V_{i-1,\mathbf{c}} \end{bmatrix} - (q_i + \overline{q_{i-1}}) \begin{bmatrix} \tilde{V}_{\mathbf{z}} \\ \tilde{V}_{\mathbf{c}} \end{bmatrix} \right)$$

$K_{i,\mathbf{z}}^{(m)} = K_{i-1,\mathbf{z}}^{(m)} + \begin{bmatrix} B_{\mathbf{z}}^T V_{i,\mathbf{z}} \overline{V}_{i,\mathbf{z}}^T M_{\mathbf{z}} & B_{\mathbf{z}}^T V_{i,\mathbf{z}} \overline{V}_{i,\mathbf{c}}^T M_{\mathbf{c}} \end{bmatrix}$

if $\left(\frac{||K_{i,\mathbf{z}}^{(m)} - K_{i-1,\mathbf{z}}^{(m)}||_F}{||K_{i,\mathbf{z}}^{(m)}||_F} < tol_{\text{ADI}} \right)$ **then**

break

end if

end for

$K_{\mathbf{z}}^{(m)} = K_{n_{\text{ADI}},\mathbf{z}}^{(m)}$

if $\left(\frac{||K_{\mathbf{z}}^{(m)} - K_{\mathbf{z}}^{(m-1)}||_F}{||K_{\mathbf{z}}^{(m)}||_F} < tol_{\text{Newton}} \right)$ **then**

break

end if

end for

$K = \begin{bmatrix} K_{\mathbf{z}}^{(n_{\text{Newton}})} & 0 \end{bmatrix}$

with $M_{\mathbf{z}}, M_{\mathbf{c}}$ symmetric positive definite, $G, R_{\vec{w}}$ full rank, and $q_i \in \mathbb{C}^-$. The full system matrix $\mathbf{A}$ is indefinite $\forall q_i \in \mathbb{C}^-$.

Details about the solution strategy for such block structured indefinite saddle point systems are not part of this article. Note that the use of preconditioned iterative solvers is necessary if the dimension of the system grows. In this case the block

preconditioning ideas in [5, 8] can be extended. However, more details regarding this have to be postponed to future publications due to space limitations.

In the next section the configurations for the numerical tests used to illustrate our numerical procedure are introduced.

3 Numerical Examples

The main focus of this section is to verify the usability of the Newton-ADI iteration described in the previous section to compute the optimal control $\mathbf{u}(t)$ for the linearized version of a diffusion-convection equation coupled with the linearized Navier-Stokes equations. All computations are based on the finite element discretization of the reactor model shown in Fig. 1.

The reactor consists of an inflow channel on the left and an outflow channel on the right. Both have a diameter of 1.0 and a length of 3.0. Inside the reactor of dimension 5.0×7.0 there is a quadratic obstacle of dimension 1.5×1.5. The fluid flows around the obstacle and transports the concentration via the convection through the domain. Additionally, the concentration is spread due to a diffusion process. As described above, we assume a fast reaction of the concentration at the surface Γ_r of the obstacle, such that concentration that arrives at the obstacle is absorbed immediately.

The coarse discretization depicted in Fig. 1 is refined using a *Bänsch refinement* [2]. We apply this refinement strategy as a threefold bisection in the whole domain, ninefold bisection in the outflow channel and elevenfold bisection on the boundary of the obstacle, yielding the dimensions in Table 1b. Furthermore, we use heuristic Penzl ADI shifts for all configurations [16].

The FORTRAN90 based finite element software NAVIER [3] was used to assemble the matrices representing the finite element discretization. The computations for the resulting matrix equations were executed in MATLAB® R2012b on a a 64-bit server with Intel® Xeon® X5650 @2.67 GHz, with 2 CPUs, 12 Cores (6

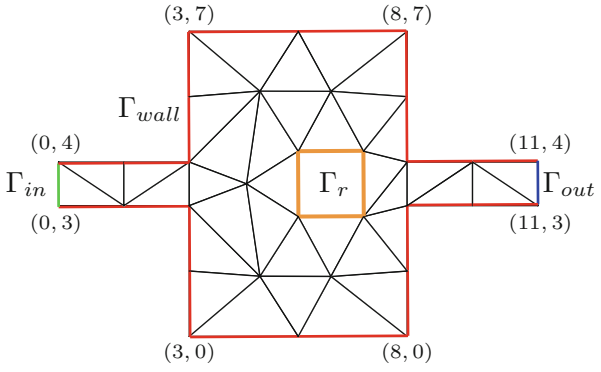


Fig. 1 Initial triangulation of the reactor model with coordinates and boundary conditions

Cores per CPU) and 48 GB main memory available. We refer the reader to [5] for more details regarding the interaction of both software packages.

3.1 Reynolds Number and Regularization Parameter λ

The Newton-ADI method is tested for five different combinations of Reynolds number Re and Schmidt number Sc, as given in Table 1a. Figure 2 depicts the convergence behavior of the Newton-ADI method clustered in subfigures for different regularization parameters. Each subfigure shows the evolution of the relative change

$$\frac{||K^{(m)} - K^{(m-1)}||_F}{||K^{(m)}||_F}$$

of the feedback matrix dependent on the Newton step m for all different sets in Table 1a. It shows that the graphs group for the different products of Re Sc $\in \{1, 10, 100\}$. If the product becomes larger, the Newton-ADI iteration needs more steps to converge. Note that the convergence for the set with larger Reynolds number is slightly slower within the group.

The regularization parameter λ penalizes the output $\mathbf{y}$ in the cost functional (2.18); that means the computed feedback K should stabilize our system more efficiently. This is reflected in the increasing number of Newton steps for increasing λ. Nevertheless, the Newton-ADI method computes the feedback matrix K for all settings to a suitable accuracy.

The quadratic convergence of the Newton iteration highly depends on the accuracy of the ADI method. In Fig. 2e stagnation appears for Set V during the iteration. In that case we would need to use a higher ADI accuracy. We analyze this phenomenon in more detail in the next subsection.

Table 1 Test parameter settings

Set	Re	Sc
I	1	1
II	1	10
III	10	1
IV	1	100
V	10	10

(a) Different parameter settings.

Variable	Dimension
n_v	9 092
n_c	1 187
n_p	1 276
n_x	11 555

(b) Different dimensions of FE space.

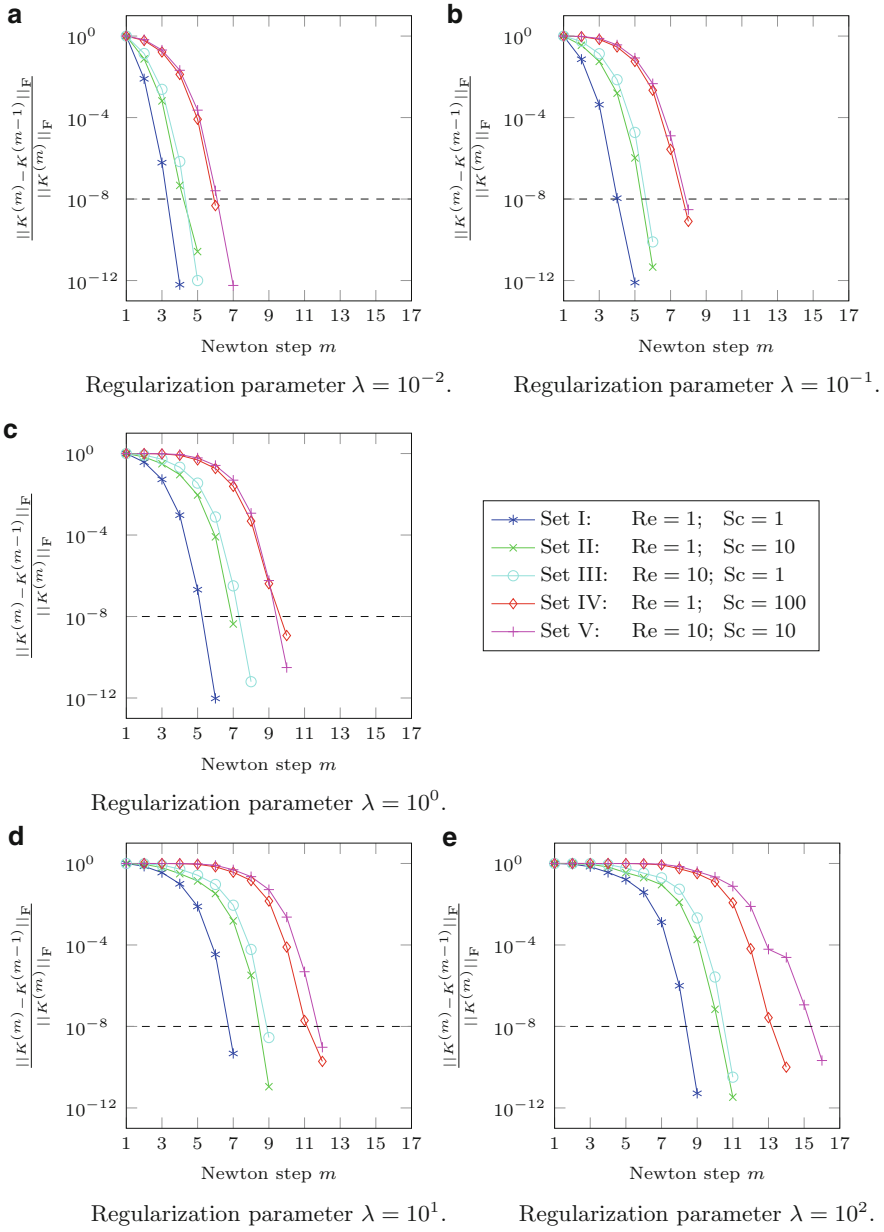
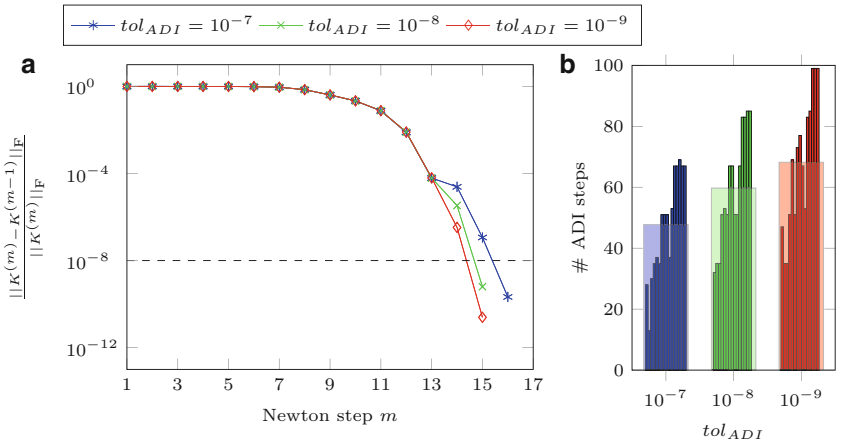


Fig. 2 Influence of Reynolds/Schmidt numbers and regularization parameter λ on the Newton-ADI convergence ($tol_{Newton} = 10^{-8}, tol_{ADI} = 10^{-7}$)



Convergence behavior of Newton iteration for different ADI accuracies.

Number of ADI steps per Newton step (small bars) and average over all Newton steps (wide bars).

Fig. 3 Influence of tol_{ADI} for Newton-ADI convergence ($tol_{Newton} = 10^{-8}$, Set V: Re = 10, Sc = 10)

3.2 ADI Tolerance vs. Newton Convergence

Figure 3 illustrates the influence of the ADI accuracy. We increase the ADI accuracy to avoid the observed stagnation for Set V. Figure 3a shows that for $tol_{ADI} = 10^{-8}$ stagnation still occurs. For $tol_{ADI} = 10^{-9}$ the Newton iteration converges quadratically. The higher ADI accuracy implies more ADI steps, as it is depicted in Fig. 3b. In total, the Newton-ADI with a higher ADI accuracy needs more time, although we can save one Newton step. To avoid these problems we will extend the idea of inexact Newton methods for the standard state space case worked out theoretically in [11] and for practical implementations in [7] to the structured DAE problems in the future. The difficulty here is the necessity for the projected ADI residuals in order to perform the accuracy control but also avoiding the explicit projection. A formulation of the index-2 ADI that has this capability is currently being investigated.

Conclusions and Outlook

In this report, we have extended the Riccati feedback stabilization approach for incompressible Navier-Stokes flows developed by Raymond in [17, 18] to a multi-field flow setting. For this purpose, we have coupled the Navier-

(continued)

Stokes equations with a convection-diffusion equation modeling the passive transport of a reactive species in a fluid.

We have extended the numerical method detailed in [5] for stabilization of the perturbed Navier-Stokes equations to this setting. For a proof-of-concept, we have used a merely academic problem configuration and tested our algorithm on this setting. The numerical results indicate that the Newton-ADI framework from [5] extended to the coupled problem can be used to robustly solve for the Riccati feedback in a regime of modest Reynolds and Schmidt numbers.

Future work will include the extension of our approach in several directions. This includes the development of efficient preconditioners for the saddle point problems to be solved in the innermost step of the Newton-ADI method for the coupled setting, the extension to higher Reynolds/Schmidt numbers (though highly turbulent flow can probably not be tackled by this stabilization method), and the adaptation of our approach to more complicated multi-field flow problems. The latter also requires the extension of Raymond's functional analysis framework to coupled stabilization problems, as well as a convergence analysis for the computed finite-dimensional feedback operators and an investigation of their stabilization properties for the infinite-dimensional system.

Acknowledgements We would like to thank Stephan Weller for his helpful advices regarding the handling of the FEM software NAVIER and René Schneider for many useful discussion throughout the whole project time. In addition, we would like to thank Martin Stoll, Andy Wathen, Matthias Heinkenschloss, and Jan Heiland for various discussions that extended our knowledge about optimal control for fluid mechanics.

References

1. L. Amodei, J.-M. Buchot, A stabilization algorithm of the Navier-Stokes equations based on algebraic Bernoulli equation. Numer. Lin. Alg. Appl. **19**, 700–727 (2012)
2. E. Bänsch, Local mesh refinement in 2 and 3 dimensions. IMPACT Comput. Sci. Eng. **3**, 181–191 (1991)
3. E. Bänsch, Simulation of instationary, incompressible flows. Acta Math. Univ. Comenianae **67**, 101–114 (1998)
4. E. Bänsch, P. Benner, Stabilization of incompressible flow problems by Riccati-based feedback, in *Constrained Optimization and Optimal Control for Partial Differential Equations*, ed. by G. Leugering, S. Engell, A. Griewank, M. Hinze, R. Rannacher, V. Schulz, M. Ulbrich, S. Ulbrich. Vol. 160 of International Series of Numerical Mathematics (Birkhäuser, Basel, 2012), pp. 5–20
5. E. Bänsch, P. Benner, J. Saak, H. K. Weichelt, Riccati-based boundary feedback stabilization of incompressible Navier-Stokes flow, Preprint SPP1253-154, DFG-SPP1253, 2013. http://www.am.uni-erlangen.de/home/spp1253/wiki/images/b/ba/Preprint-SPP1253-154.pdf

6. P. Benner, J.-R. Li, T. Penzl, Numerical solution of large Lyapunov equations, Riccati equations, and linear-quadratic control problems. Numer. Lin. Alg. Appl. **15**, 755–777 (2008)
7. P. Benner, J. Saak, Numerical solution of large and sparse continuous time algebraic matrix Riccati and Lyapunov equations: a state of the art survey. GAMM Mitteilungen **36**, 32–52 (2013)
8. P. Benner, J. Saak, M. Stoll, H. K. Weichelt, Efficient solution of large-scale saddle point systems arising in Riccati-based boundary feedback stabilization of incompressible Stokes flow. SIAM J. Sci. Comput. **35**, S150–S170 (2013)
9. P. Benner, T. Stykel, Numerical solution of projected algebraic Riccati equations. SIAM J. Numer. Anal. **52**(2), 581–600 (2014)
10. K. A. Cliffe, T. J. Garratt, A. Spence, Eigenvalues of block matrices arising from problems in fluid mechanics. SIAM J. Matrix Anal. Appl. **15**, 1310–1318 (1994)
11. F. Feitzinger, T. Hylla, E. W. Sachs, Inexact Kleinman-Newton method for Riccati equations. SIAM J. Matrix Anal. Appl. **31**, 272–288 (2009)
12. M. Heinkenschloss, D. C. Sorensen, K. Sun, Balanced truncation model reduction for a class of descriptor systems with application to the Oseen equations. SIAM J. Sci. Comput. **30**, 1038–1063 (2008)
13. M. Hinze, K. Kunisch, Second order methods for boundary control of the instationary Navier-Stokes system. Z. Angew. Math. Mech. **84**, 171–187 (2004)
14. P. Hood, C. Taylor, Navier-Stokes equations using mixed interpolation, in *Finite Element Methods in Flow Problems*, ed. by J. T. Oden, R. H. Gallagher, C. Taylor, O. C. Zienkiewicz (University of Alabama in Huntsville Press, Huntsville, 1974), pp. 121–132
15. P. Kunkel, V. Mehrmann, *Differential-Algebraic Equations: Analysis and Numerical Solution*. Textbooks in Mathematics (EMS Publishing House, Zürich, 2006)
16. T. Penzl, LYAPACK *users guide*, Tech. Report SFB393/00-33, Sonderforschungsbereich 393 *Numerische Simulation auf massiv parallelen Rechnern*, TU Chemnitz, 09107 Chemnitz, 2000. Available from http://www.tu-chemnitz.de/sfb393/sfb00pr.html
17. J.-P. Raymond, Local boundary feedback stabilization of the Navier-Stokes equations, in Control Systems: Theory, Numerics and Applications, Rome, 30 March–1 April 2005, Proceedings of Science, SISSA, http://pos.sissa.it, 2005
18. J.-P. Raymond, *Feedback boundary stabilization of the two-dimensional Navier-Stokes equations*. SIAM J. Control Optim. **45**, 790–828 (2006)
19. J. Weickert, Navier-Stokes equations as a differential-algebraic system, Preprint SFB393/96-08, Preprint Series of the SFB 393 "Numerische Simulation auf massiv parallelen Rechnern", Chemnitz University of Technology, Aug 1996