

# Example Setups of Navier-Stokes Equations with Control and Observation: Spatial Discretization and Representation via Linear-quadratic Matrix Coefficients

Maximilian Behr      Peter Benner      Jan Heiland

July 28, 2017

We provide spatial discretizations of nonlinear incompressible Navier-Stokes equations with inputs and outputs in the form of matrices ready to use in any numerical linear algebra package. We discuss the assembling of the system operators and the realization of boundary conditions and inputs and outputs. We describe the two benchmark problems *driven cavity* and *cylinder wake* and provide the corresponding data. The use of the data is illustrated by numerous example setups. The test cases are provided as plain Python or Octave/MATLAB script files for immediate replication.

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Spatial Discretization of Navier-Stokes Equations</b>	<b>3</b>
<b>3</b>	<b>Assembling the Linear Operators</b>	<b>4</b>
<b>4</b>	<b>Assembling of the Trilinear Form</b>	<b>4</b>
<b>5</b>	<b>Incorporation of Boundary Conditions</b>	<b>5</b>
5.1	Dirichlet Boundary Conditions . . . . .	5
5.2	Natural Boundary Conditions . . . . .	6
<b>6</b>	<b>The Spatially Discretized System</b>	<b>7</b>
<b>7</b>	<b>Simulating the System</b>	<b>7</b>
7.1	Steady-state Solutions . . . . .	8

7.2	Linearizations . . . . .	8
7.3	Time Integration . . . . .	9
<b>8</b>	<b>Setups</b>	<b>9</b>
8.1	Lid Driven Cavity . . . . .	9
8.2	Cylinder Wake . . . . .	10
<b>9</b>	<b>Modelling and Approximation of Control and Observation</b>	<b>11</b>
9.1	Distributed Control and Observation . . . . .	12
9.2	Assembling of the Distributed Control and Observation Operators . . . . .	13
9.3	Boundary Control . . . . .	15
9.4	Assembling of the Boundary Control Operator . . . . .	16
<b>10</b>	<b>Control Setups</b>	<b>16</b>
10.1	Driven Cavity with Distributed Control and Observation . . . . .	17
10.2	Cylinder Wake with Distributed Control and Observation . . . . .	17
10.3	Cylinder Wake with Boundary Control . . . . .	17
<b>11</b>	<b>The Data, Plotting, and Auxiliary Functions</b>	<b>18</b>
11.1	The System Matrices Files . . . . .	18
11.2	Files for Visualizations . . . . .	19
11.3	Auxiliary Files . . . . .	19
<b>12</b>	<b>Numerical Examples</b>	<b>19</b>
12.1	Cylinder Wake . . . . .	19
12.1.1	Steady State with Re=40 . . . . .	20
12.1.2	Steady State with Re=40 with Boundary Control Impact . . . . .	20
12.1.3	Transient Case with Re=90 . . . . .	20
12.1.4	Transient Case with Re=40 and Boundary Actuation . . . . .	23
12.2	Driven Cavity . . . . .	23
12.2.1	Steady State with Re=1200 . . . . .	23
12.2.2	Transient Case with Re=800 and Distributed Actuation . . . . .	25

# 1 Introduction

This work considers spatial discretizations of incompressible Navier-Stokes equations

$$\frac{\partial v}{\partial t} + (v \cdot \nabla)v - \frac{1}{\text{Re}}\Delta v + \nabla p = f \text{ in } (0, t) \times \Omega, \quad (1a)$$

$$\nabla \cdot v = 0 \text{ in } (0, t) \times \Omega, \quad (1b)$$

with controls and observations, which we assume to be parametrized by the *Reynolds number*  $\text{Re}$ , posed on a bounded domain  $\Omega \subseteq \mathbb{R}^2$  for time  $t > 0$ , and equipped with an initial condition and boundary conditions.

This manuscript is accompanied by data that represents spatial discretizations of (1) in various flow setups. The modelling of the flow problems, the discretizations, and the data representation comes with the following features and advantages

1. The discretized nonlinearity is provided as an unfolded tensor  $H$ , that realizes  $H(v, v)$  as  $H \cdot v \otimes v$ . On the one hand side, this leads to a pure matrix representation. On the other hand, the tensor representation is directly accessible to recent developments of model reduction for bilinear quadratic systems.
2. The terms that are multiplied with  $\frac{1}{\text{Re}}$  are kept separate, so that simulations with different *Reynolds numbers* can be realized through a scaling of the corresponding matrices.
3. Nonhomogeneous boundary conditions are incorporated via a *penalized Robin* approach. This allows us to model the control action via an linear input operator that appears in the right hand side of the momentum equation (1a) as in the case of distributed control; cf. [2].

Altogether, these approaches provide a discrete representation of nonlinear Navier-Stokes equations with control and observations that can be readily used for simulations and system theoretical experiments not resorting to any finite element toolbox whatsoever.

The manuscript is structured as follows. First, we explain how the system coefficient matrices of the semi-discrete Navier-Stokes equations are assembled and explain the basic usage of the matrices in simulations. Then, we introduce the test setups for simulation or control of incompressible flows. In the last part, we document the data files and the setup and results of some example configurations.

#### **Code and Data Availability**

The source code and the data of the implementations used to compute the presented results is available from:

`doi:10.5281/zenodo.834940`

under the *MIT* license.

Figure 1: Link to code and data.

## **2 Spatial Discretization of Navier-Stokes Equations**

We consider a general mixed finite element scheme to illustrate the derivation. For mathematical and practical details on finite element approximations of flows, see, e.g., [3, 4].

Let  $V$  and  $Q$  be the finite element space, namely finite-dimensional function spaces defined on a tessellation of  $\Omega$ , for the velocity and the pressure spanned by nodal bases

like

$$V = \text{span}\{\phi_i\}_{i=1}^n \quad \text{and} \quad Q = \text{span}\{\psi_k\}_{k=1}^m, \quad n, m \in \mathbb{N}. \quad (2)$$

Thus, we look for approximations  $v$  and  $p$  to the velocity and pressure solutions of (1) that are linear combinations of the basis functions, namely

$$v(t, x) = \sum_{j=1}^n v_j(t) \phi_j(x) \quad \text{and} \quad p(t, x) = \sum_{k=1}^m p_k(t) \psi_k(x). \quad (3)$$

In what follows, we will often omit the time dependency of the coefficients and the space dependency of the basis functions. Also, we will freely identify the discrete functions  $v$  and  $p$  with their coefficient vectors

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} \quad \text{and} \quad \mathbf{p} = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_m \end{bmatrix}$$

defined through the expansion in (3).

### 3 Assembling the Linear Operators

For the finite element spaces (2), we assemble the finite dimensional approximations  $A$  and  $J$  to the diffusion and divergence operators  $\Delta$  and  $\nabla \cdot$  in the standard way:

$$A = \left[ \int_{\Omega} \nabla \phi_i : \nabla \phi_j \, dx \right]_{\substack{i=1, \dots, n \\ j=1, \dots, n}} \in \mathbb{R}^{n \times n} \quad \text{and} \quad J = \left[ \int_{\Omega} \psi_k \nabla \cdot \phi_j \, dx \right]_{\substack{k=1, \dots, m \\ j=1, \dots, n}} \in \mathbb{R}^{n \times n}, \quad (4)$$

where  $:$  stands for the Frobenius inner product.

The mass matrix  $M$  and the right hand side  $f$  are assembled as follows:

$$M = \left[ \int_{\Omega} \phi_i \cdot \phi_j \, dx \right]_{\substack{i=1, \dots, n \\ j=1, \dots, n}} \in \mathbb{R}^{n \times n} \quad \text{and} \quad f_n = \left[ \int_{\Omega} \phi_i \cdot f \, dx \right]_{i=1, \dots, n} \in \mathbb{R}^{n \times 1}. \quad (5)$$

By now, we haven't taken into account the boundary conditions. This we will discuss in Section 5.

### 4 Assembling of the Trilinear Form

Consider two discrete velocities  $v = \sum_{j=1}^n v_j \phi_j$  and  $w = \sum_{k=1}^n w_k \phi_k$  with coefficient vectors  $\mathbf{v}$  and  $\mathbf{w}$  and the resulting convective term

$$(v \cdot \nabla)w = \sum_{j=1}^n \sum_{k=1}^n v_j w_k (\phi_j \cdot \nabla) \phi_k.$$

Testing this relation against the  $i$ -th basis function gives the relation

$$\sum_{j=1}^n \sum_{k=1}^n v_j w_k \int_{\Omega} ((\phi_j \cdot \nabla) \phi_k) \cdot \phi_i \, dx = H_i(\mathbf{v} \otimes \mathbf{w}),$$

with the *Kronecker product*  $\otimes: \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^{n^2}$  and where

$$H_i := \left[ \int_{\Omega} (\phi_j \cdot \nabla) \phi_k \cdot \phi_i \, dx \right]_{j=1, \dots, n; k=1, \dots, n} \in \mathbb{R}^{1 \times n^2} \quad (6)$$

Thus, the discrete approximation of the convection operator  $v \mapsto (v \cdot \nabla)v$  is given as

$$H: \mathbf{v} \mapsto H(\mathbf{v} \otimes \mathbf{v}), \quad \text{where} \quad H = \begin{bmatrix} H_1 \\ \vdots \\ H_n \end{bmatrix} \in \mathbb{R}^{n \times n^2} \quad (7)$$

and  $H_i$  is as defined in (6).

## 5 Incorporation of Boundary Conditions

We now explain how boundary conditions are treated in general and for the trilinear form in particular.

### 5.1 Dirichlet Boundary Conditions

To begin with, we assume that we only have Dirichlet conditions for the velocity. This means that the value of  $v$  at the boundary is known and, since we consider nodal bases, also the values of the coefficients of the discretized velocities associated with nodes at the boundary are known. In the theory of weak solutions, the test space is chosen to have a zero trace at the Dirichlet boundary. In practice, the basis functions  $\phi_i$  that are associated with a node at a Dirichlet boundary, are left out when assembling the coefficient matrices like in (4) and (5). Often it is more convenient to assemble the whole matrices and remove the lines corresponding to the boundary in a second step.

Let  $\mathcal{I}_{\Gamma}$  be the set of indices associated with the Dirichlet boundary nodes and let  $\mathcal{I}_I$  be the set that contains the indices of all other nodes, i.e. the nodes in the inner or at the boundary where Neumann conditions are posed. Assume that the entries of the solution vector are sorted such that

$$\mathbf{v} = \bar{\mathbf{v}}_I + \bar{\mathbf{v}}_{\Gamma} := \begin{bmatrix} \mathbf{v}_I \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \mathbf{v}_{\Gamma} \end{bmatrix}, \quad (8)$$

where  $\mathbf{v}_I$  and  $\mathbf{v}_{\Gamma}$  are the vectors of coefficients associated with  $\mathcal{I}_I$  and  $\mathcal{I}_{\Gamma}$ .

Let  $[\cdot]_I$  denote the operation of restricting a matrix or vector to the inner nodes with respect to degrees of freedom (columns) and the posed conditions (rows). For example,

assume that  $A$  is assembled without considering boundary conditions. Then, we resolve given Dirichlet boundary conditions via the relation

$$[A\mathbf{v}]_I = [A]_I \mathbf{v}_I + [A\bar{\mathbf{v}}_\Gamma]_I. \quad (9)$$

We will use the abbreviation  $A_I := [A]_I$ .

For the quadratic term, we first compute

$$H(\mathbf{v} \otimes \mathbf{v}) = H(\bar{\mathbf{v}}_I \otimes \bar{\mathbf{v}}_I) + H(\bar{\mathbf{v}}_I \otimes \bar{\mathbf{v}}_\Gamma) + H(\bar{\mathbf{v}}_\Gamma \otimes \bar{\mathbf{v}}_I) + H(\bar{\mathbf{v}}_\Gamma \otimes \bar{\mathbf{v}}_\Gamma).$$

Because of the zero entries in  $\bar{\mathbf{v}}_I$  and  $\bar{\mathbf{v}}_\Gamma$ , it holds that

$$[H(\mathbf{v} \otimes \mathbf{v})]_I = H_I(\mathbf{v}_I \otimes \mathbf{v}_I) + [L_1(\bar{\mathbf{v}}_\Gamma)]_I \mathbf{v}_I + [L_2(\bar{\mathbf{v}}_\Gamma)]_I \mathbf{v}_I + [H(\bar{\mathbf{v}}_\Gamma \otimes \bar{\mathbf{v}}_\Gamma)]_I, \quad (10)$$

where

$$L_1(\bar{\mathbf{v}}_\Gamma) = \left[ \sum_{k \in \mathcal{I}_\Gamma} [\mathbf{v}_\Gamma]_k \int_{\Omega} ((\phi_j \cdot \nabla) \phi_k) \cdot \phi_i \, dx \right]_{\substack{i=1, \dots, n \\ j=1, \dots, n}}, \quad (11a)$$

$$L_2(\bar{\mathbf{v}}_\Gamma) = \left[ \sum_{k \in \mathcal{I}_\Gamma} [\mathbf{v}_\Gamma]_k \int_{\Omega} ((\phi_k \cdot \nabla) \phi_j) \cdot \phi_i \, dx \right]_{\substack{i=1, \dots, n \\ j=1, \dots, n}}. \quad (11b)$$

Thus, the consideration of Dirichlet boundary conditions in the quadratic formulation can be done as follows

1. Assemble  $H = H_I$  like in (7) for the inner nodes both in test and trial space.
2. Assemble the linearized convection matrix  $L(\bar{\mathbf{v}}_\Gamma)$  with  $\bar{\mathbf{v}}_\Gamma$  as convection velocity like  $L_1(\bar{\mathbf{v}}_\Gamma)$  and  $L_2(\bar{\mathbf{v}}_\Gamma)$  in (11) and restrict it to the inner nodes.
3. Assemble the contribution  $H(\bar{\mathbf{v}}_\Gamma \otimes \bar{\mathbf{v}}_\Gamma)$  to the source term via

$$H(\bar{\mathbf{v}}_\Gamma \otimes \bar{\mathbf{v}}_\Gamma) = \left[ \sum_{j \in \mathcal{I}_\Gamma} \sum_{k \in \mathcal{I}_\Gamma} [\mathbf{v}_\Gamma]_j [\mathbf{v}_\Gamma]_k \int_{\Omega} ((\phi_j \cdot \nabla) \phi_k) \cdot \phi_i \, dx \right]_{i=1, \dots, n}$$

and restrict it to the inner nodes.

As the result, one has the coefficient for  $\mathbf{v}_I \otimes \mathbf{v}_I$ , an additional coefficient matrix for  $\mathbf{v}_I$ , and a correction for the source term.

## 5.2 Natural Boundary Conditions

Boundary conditions that contain the normal derivative of  $v$  are often called *natural boundary conditions* since they like naturally are resolved in the finite element discretization. In view of assembling the coefficients, nodes associated with natural boundary are

simply treated like inner nodes. However, one has to consider additional source terms, that derive from the partial integration that led to the forms in (4).

Let  $\Gamma_N$  denote the part of the boundary where boundary conditions other than of Dirichlet type are applied. Then the additional source term is assembled as

$$g_n = \left[ \int_{\Gamma_N} (p\vec{n} - \nu \frac{\partial v}{\partial \vec{n}}) \cdot \phi_i \, dx \right]_{i=1, \dots, n} = \left[ \int_{\Gamma_N} g \cdot \phi_i \, dx \right]_{i=1, \dots, n}, \quad (12)$$

where  $\vec{n}$  is the outward normal vector. We will frequently employ *do-nothing* conditions at the outflow boundary, i.e. we set

$$p\vec{n} - \nu \frac{\partial v}{\partial \vec{n}} = 0, \quad \text{on } \Gamma_N, \quad (13)$$

so that  $g_n$  is just zero.

## 6 The Spatially Discretized System

We consider the Navier-Stokes equations

$$\frac{\partial \mathbf{v}}{\partial t} - \frac{1}{\text{Re}} \Delta \mathbf{v} + (\mathbf{v} \cdot \nabla) \mathbf{v} + \nabla \mathbf{p} = \mathbf{f}_n, \quad (14a)$$

$$\nabla \cdot \mathbf{v} = 0, \quad (14b)$$

on the domain  $\Omega$  with Dirichlet or *do-nothing* boundary conditions that we write as

$$\gamma \mathbf{v} = \mathbf{v}_\Gamma. \quad (14c)$$

With the splitting of the velocity ansatz space into inner and boundary nodes as defined in (8), with the convention (9), and with the formula for the convective term (10), a standard finite element discretization of (14) for the velocity at the inner nodes  $\mathbf{v}_I$  and the discrete pressure  $\mathbf{p}$  can be written as

$$M_I \frac{d}{dt} \mathbf{v}_I + H_I (\mathbf{v}_I \otimes \mathbf{v}_I) + [L_1(\bar{\mathbf{v}}_\Gamma) + L_2(\bar{\mathbf{v}}_\Gamma) + \frac{1}{\text{Re}} A]_I \mathbf{v}_I - J_I^T \mathbf{p} = [f_n]_I - [g_n]_I - \frac{1}{\text{Re}} [A \bar{\mathbf{v}}_\Gamma]_I - [H(\bar{\mathbf{v}}_\Gamma \otimes \bar{\mathbf{v}}_\Gamma)]_I, \quad (15a)$$

$$J_I \mathbf{v}_I = -[J \bar{\mathbf{v}}_\Gamma]_I. \quad (15b)$$

For a given setup,  $\bar{\mathbf{v}}_\Gamma$  is defined through (14c), and, thus, we can define system (15) completely by means of the matrices and vectors

up to a scaling of  $\mathbf{A}$  and  $\mathbf{fv\_diff}$  by  $\frac{1}{\text{Re}}$ .

## 7 Simulating the System

Once one has the coefficients listed in Table 1 at hand, the implementation of a simulation can be done in any numerical linear algebra package. To illustrate the usage, we address a few important points of a simulation.

M	$\leftrightarrow$	$M_I$			
A	$\leftrightarrow$	$A_I$		fv	$\leftrightarrow$ $[f_n]_I$
H	$\leftrightarrow$	$H_I$		gv	$\leftrightarrow$ $[g_n]_I$
L1	$\leftrightarrow$	$[L_1(\bar{v}_\Gamma)]_I$	and	fv_diff	$\leftrightarrow$ $[A\bar{v}_\Gamma]_I$
L2	$\leftrightarrow$	$[L_2(\bar{v}_\Gamma)]_I$		fv_conv	$\leftrightarrow$ $[N(\bar{v}_\Gamma)\bar{v}_\Gamma]_I$
J	$\leftrightarrow$	$J_I$		fp_div	$\leftrightarrow$ $[J(\bar{v}_\Gamma)]_I$
J.T	$\leftrightarrow$	$J_I^T$			

Table 1: Parameters defining the spatially discretized Navier-Stokes equations (15).

## 7.1 Steady-state Solutions

For a *Reynolds number* RE, the steady-state Stokes solution  $[v, p]$  is defined through

$$\begin{bmatrix} 1./RE*A & -J.T \\ J & 0 \end{bmatrix} * \begin{bmatrix} v \\ p \end{bmatrix} \stackrel{!}{=} \begin{bmatrix} fv - 1./RE*fv\_diff \\ -fp\_div \end{bmatrix} \quad (16)$$

and the steady-state Navier-Stokes solution is given as the solution of

$$\begin{bmatrix} 1./RE*A+L1+L2 & -J.T \\ J & 0 \end{bmatrix} * \begin{bmatrix} v \\ p \end{bmatrix} \stackrel{!}{=} \begin{bmatrix} fv - H*kron(v, v) - 1./RE*fv\_diff - fv\_conv \\ -fp\_div \end{bmatrix},$$

where  $\stackrel{!}{=}$  means that a solver has to be applied and where `kron` is the function that computes the Kronecker product. Note that  $v$  represents the velocity solution only at the inner nodes, while the values of the boundary nodes are defined by the boundary conditions.

## 7.2 Linearizations

We consider linearizations as they occur in the modelling of optimal control systems or during the numerical solution of the nonlinear state equations.

In the nonlinear setting the convection is modelled through

$$L1*v + L2*v + H*kron(v, v) + fv\_conv$$

If  $a$  is a linearization point that in particular fulfills the boundary conditions and if  $\mathbf{a}$  is its approximation, then the *Oseen* or *Picard* linearization  $(u \cdot \nabla)u \leftarrow (a \cdot \nabla)u$  is realized as

$$H*kron(\mathbf{a}, v) + L1*v + L2*\mathbf{a} + fv\_conv$$

and the *Newton* linearization  $(u \cdot \nabla)u \leftarrow (a \cdot \nabla)u + (u \cdot \nabla)a - (a \cdot \nabla)a$  as

$$H*kron(\mathbf{a}, v) + H*kron(v, \mathbf{a}) - H*kron(\mathbf{a}, \mathbf{a}) + L1*v + L2*v + fv\_conv$$

In the frequent case, that one decomposes  $v \stackrel{!}{=} v_s + v_d$ , where  $v_s$  is the steady-state Navier-Stokes solution, the linearized system for the difference state  $v_d$  to be solved reads



```

M*vd.dt + A*vd + H*kron(vs, vd) + H*kron(vd, vs) - J.T*p != 0
# and, simultaneously,
J*vd != 0

```

where  $\text{vd.dt}$  denotes the time derivative. Note that, because the difference state is zero at the boundary, in particular the terms  $L1+L2$  and  $\mathbf{f}_{\text{conv}}$  do not appear.

The matrices  $H1$  and  $H2$  representing the linear functions  $H*\text{kron}(\mathbf{a}, \mathbf{v}) =: H1*\mathbf{v}$  and  $H*\text{kron}(\mathbf{v}, \mathbf{a}) =: H2*\mathbf{v}$  can be computed columnwise:

```

H1 = [H*kron(e_i, a) for i in range(Nv)]
H2 = [H*kron(a, e_i) for i in range(Nv)]

```

where  $\mathbf{a}[i]$  is the  $i$ -th component of  $\mathbf{a}$  and  $\mathbf{e}_i$  is the  $i$ -th unit basis vector.

### 7.3 Time Integration

Assume that the current velocity approximation  $\mathbf{v}_{\text{old}}$  is given. Then one can apply the implicit-explicit (IMEX) Euler scheme to advance the velocity by a time step of size  $\text{dt}$  by solving

$$\begin{bmatrix} M + \text{dt}*(A+L1+L2) & -\text{dt}*J.T \\ J & 0 \end{bmatrix} * \begin{bmatrix} \mathbf{v}_{\text{new}} \\ p \end{bmatrix} != \begin{bmatrix} M*\mathbf{v}_{\text{old}} + \text{dt}*(\mathbf{f}_v - H*\text{kron}(\mathbf{v}_{\text{old}}, \mathbf{v}_{\text{old}}) - \mathbf{f}_{v,\text{diff}} - \mathbf{f}_{v,\text{conv}}) \\ -\mathbf{f}_{p,\text{div}} \end{bmatrix}.$$

## 8 Setups

We describe the setups for which the coefficient matrices can be downloaded.

For all considered setups we describe the geometrical properties, the boundary conditions. The provided matrices are obtained by finite element discretizations as provided by *FEniCS* [6]. Depending on the geometry, we use uniform and nonuniform grids of different coarseness. As finite element pair we choose *Taylor-Hood*  $P_2 - P_1$  finite elements [10].

### 8.1 Lid Driven Cavity

The two-dimensional driven cavity is a well investigated and common benchmark problem [8]. As the model, one considers the incompressible Navier-Stokes equations on a square with homogeneous Dirichlet boundary conditions for the velocity except from the upper boundary  $\Gamma_0$  – the lid – where the tangential velocity component is set to 1. In the setup of the regularized driven cavity the tangential velocity set to smoothly approach

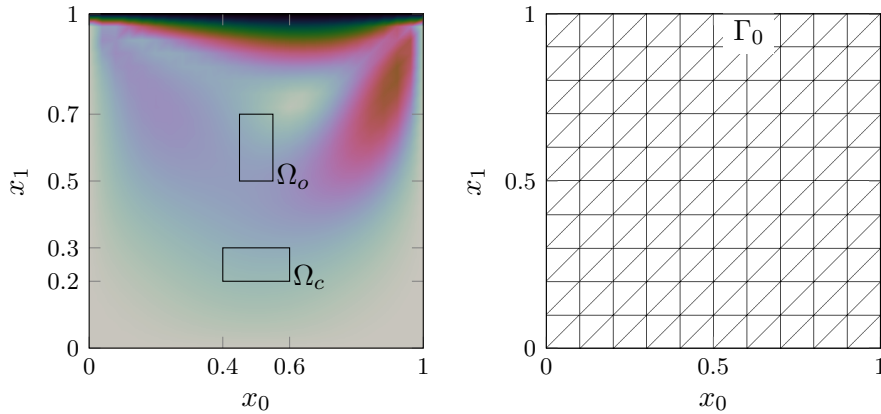


Figure 2: Illustration of the velocity magnitude of the driven cavity problem and the domain of control and observation  $\Omega_c = [0.4, 0.6] \times [0.2, 0.3]$  and  $\Omega_o = [0.45, 0.55] \times [0.5, 0.7]$ . The second plot shows a triangulation of the domain for  $N=10$ .

zero towards the edges in order to comply with the zero conditions on the other walls. The regularized and standard driven cavity is a test problem of the Octave/MATLAB FE package IFISS [9].

We consider the non regularized driven cavity on the unit square  $\Omega = (0, 1) \times (0, 1)$  and its boundary  $\Gamma$  with boundary conditions

$$\gamma v : \begin{cases} v = [1, 0]^T, & \text{on } \Gamma_0, \\ v = [0, 0]^T, & \text{elsewhere,} \end{cases}$$

cf. Fig. 2. For the spatial discretization, we use a uniform grid as illustrated in Fig. 2 with a parameter  $N$  that describes the number of segments that one boundary is divided into.

Note that for flow setups with only Dirichlet boundary conditions for the velocity, like the driven cavity, the pressure is defined up to a constant. In the numerical approximation this leads to a rank deficiency of the divergence and the gradient matrices  $J$  and  $JT$ . For iterative schemes this does not pose a problem. However, in order to have the coefficient matrix in (16) invertible, one can *pin* the pressure to zero at one node of the discretization by simply removing the corresponding row and column of  $J$  and  $JT$ .

## 8.2 Cylinder Wake

We consider the two dimensional wake of a cylinder which is a popular flow benchmark problem, a subject for simulations, and a test field for flow control [11, 7, 1].

The computational domain is as depicted in Fig. 3.

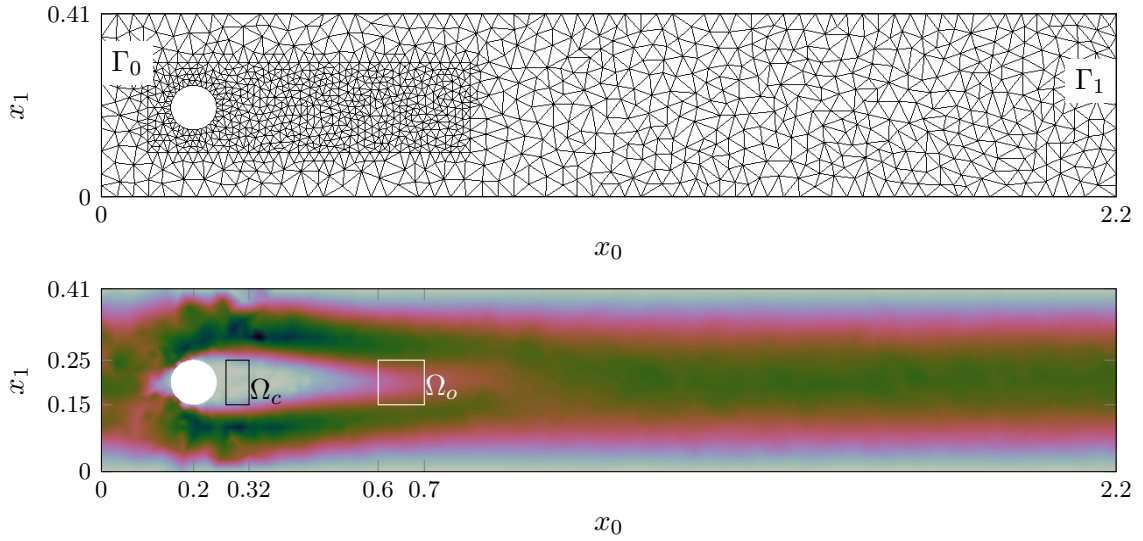


Figure 3: Illustration of geometrical setup including the domains of distributed control and observation and of the velocity magnitude for the cylinder wake.

We consider the incompressible Navier-Stokes equations on the domain as illustrated in Fig. 3 with boundary  $\Gamma$  with boundary conditions as follows, at the inflow  $\Gamma_0$  we prescribe a parabolic velocity profile through the function

$$g(s) = 4\left(1 - \frac{s}{0.41}\right) \frac{s}{0.41}.$$

We impose *do-nothing* conditions, cf. (13), at the outflow  $\Gamma_1$  and *no-slip*, i.e. zero Dirichlet conditions, at the upper and the lower wall of the channel and at the cylinder periphery.

$$\gamma(v, p) : \begin{cases} v = [g(x_1), 0]^T, & \text{on } \Gamma_0, \\ p\vec{n} - \frac{1}{\text{Re}} \frac{\partial v}{\partial \vec{n}} = [0, 0]^T, & \text{on } \Gamma_1, \\ v = [0, 0]^T, & \text{elsewhere on the boundary.} \end{cases}$$

## 9 Modelling and Approximation of Control and Observation

In flow control, one tries to act on the flow in such a way that certain flow pattern are stabilized or enforced by means of actuators. Also, sensors are used to monitor the process and to provide necessary information for the actuation. For the simulation, actuation and measuring are modelled through input and output operators  $B$  and  $C$  and corresponding spaces for the input and output signals  $u$  and  $y$  that are included in the equations as follows:

$$\frac{\partial v}{\partial t} + (v \cdot \nabla)v - \frac{1}{\text{Re}}\Delta v + \nabla p = f + Bu, \quad (17a)$$

$$\nabla \cdot v = 0, \quad (17b)$$

$$y = C(v, p). \quad (17c)$$

Distributed control, where the control acts as a volume force distributed in a domain of control, is hardly realizable in practice. Nevertheless, we provide models for distributed control, because the modelling and spatial discretization naturally leads to a standard state-space system, which is the basis for typical control or model reduction algorithms.

Boundary control through Dirichlet boundary conditions for the velocity, which is the setup for many flow control problems, does not simply lead to a standard state space representation, cf. [2]. We provide boundary control models in state space form by means of Robin approximations of the Dirichlet boundary conditions.

## 9.1 Distributed Control and Observation

We define input and output state spaces  $U$  and  $Y$  as subspaces of  $L^2([0, 1])$ .

Let  $\Omega$  be the computational domain and  $\Omega_c = [x_{0,min}^c, x_{0,max}^c] \times [x_{1,min}^c, x_{1,max}^c]$  a rectangular subdomain where the control acts. We define the input operator  $B: U \times U \rightarrow [L^2(\Omega)]^2$  such that the control acts distributed in the domain of control  $\Omega_c$  with two spatial components.

In order to save some degrees of freedom, we assume that  $Bu$  is constant in one spatial direction – say  $x_0$ .

Concretely, for  $\mathcal{U} := L^2(0, T; U \times U)$  we define the  $\mathcal{B}: \mathcal{U} \rightarrow L^2(0, T; L^2(\Omega; \mathbb{R}^2))$  via

$$\mathcal{B}u(t; x_0, x_1) = \begin{cases} u(t; \theta(x_1)), & \text{for } (x_0, x_1) \in \Omega_c, \\ [0, 0]^T, & \text{elsewhere,} \end{cases} \quad (18)$$

for  $u \in \mathcal{U}$  and with the affine linear function  $\theta_c$  mapping the  $x_1$ -extension of  $\Omega_c$ ,  $[x_{1,min}^c, x_{1,max}^c]$ , onto  $[0, 1]$ . Note that the roles of  $x_0$  and  $x_1$  can be interchanged depending on the setup.

The output operator  $\mathcal{P}_Y \mathcal{C}_v: L^2(0, T; L^2(\Omega; \mathbb{R}^s)) \rightarrow L^2(0, T; Y \times Y)$  for the velocity is defined as follows

1. For  $v(t) \in L^2(\Omega, \mathbb{R}^2)$  compute  $\mathcal{C}_v v(t) \in L^2(0, 1; \mathbb{R}^2)$  through

$$\mathcal{C}_v v(t)(\eta) = \frac{1}{x_{0,max}^o - x_{0,min}^o} \int_{x_{0,min}^o}^{x_{0,max}^o} v(t; x_0, \theta_o(\eta)) \, dx_0, \quad (19)$$

which, basically, is the velocity in the domain of observation  $\Omega_o$  averaged in  $x_0$  direction. Here,  $\theta_o$  is an affine linear mapping adjusting  $[0, 1]$  to  $[x_{1,min}^o, x_{1,max}^o]$ .

2. Define the corresponding discrete signal  $y(t) \in L^2(0, T; Y \times Y)$  as  $y(t) = \mathcal{P}_Y \mathcal{C}_v v(t)$ , where  $\mathcal{P}_Y: [L^2(0, 1)]^2 \rightarrow Y \times Y$  is the  $L^2$ -orthogonal projection.

Namely, for a rectangular domain of observation  $\Omega_o = [x_{0,min}^o, x_{0,max}^o] \times [x_{1,min}^o, x_{1,max}^o]$ , for  $\mathcal{Y} := L^2(0, T; Y \times Y)$ , and for a  $v \in L^2(0, T; L^2(\Omega; \mathbb{R}^2))$ , we define the observation operator  $\mathcal{C}_v: v \mapsto y \in \mathcal{Y}$  via

As a pressure-based output we take the average pressure in a subdomain

$$\Omega_p = [x_{0,min}^p, x_{0,max}^p] \times [x_{1,min}^p, x_{1,max}^p],$$

i.e.

$$y_p(t) = \mathcal{C}_p p(t) := \frac{1}{|\Omega_p|} \int_{\Omega_p} p(t, x) dx \quad (20)$$

evaluated in the corresponding finite element space.

## 9.2 Assembling of the Distributed Control and Observation Operators

To setup a discrete representation of the input operator  $B$ , we impose several assumptions on the space-time structure of the input signals.

Firstly, for a given  $N_u \in 2\mathbb{N}$ , we choose the signal state space  $U \times U$  as  $U_0 \oplus U_1$ , where  $U_0 := \text{span}\{\vec{\nu}_l\}_{l=1, \dots, N_u/2}$  and  $U_1 := \text{span}\{\vec{\nu}_l\}_{l=N_u/2+1, \dots, N_u}$ , where

$$\vec{\nu}_l := \begin{bmatrix} \nu_l \\ 0 \end{bmatrix}, \quad \text{for } l = 1, \dots, N_u/2 \quad \text{and} \quad \vec{\nu}_l := \begin{bmatrix} 0 \\ \nu_l \end{bmatrix}, \quad \text{for } l = N_u/2 + 1, \dots, N_u, \quad (21)$$

with  $\nu_l$  being the hierarchical piecewise linear hat functions illustrated in Figure 4.

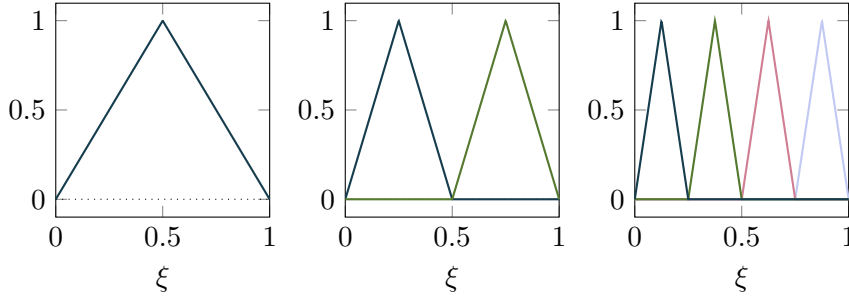


Figure 4: Three levels of a hierarchical basis for piecewise linear functions for nested  $L^2(0, 1)$ -subspaces.

Note that by defining  $U_0$  and  $U_1$  with only one spatial coordinate  $\xi$ , we anticipate that the input is to be defined depending only on one spatial coordinate, cf. (18), and note that bases of subspaces of  $L^2(0, 1)$  other than shown in Figure 4 can be used.

Secondly, we assume that the time and space component in the chosen discrete input space are separated, i.e., that we can write the inputs as

$$u(t; \xi) = \sum_{l=1}^{N_u} s_l(t) \vec{v}_l(\xi). \quad (22)$$

With  $u$  defined as (22) and the input to the system  $\mathcal{B}u$  as defined in (18) discretized like the right hand side as in (5), the discrete input operator  $B \in \mathbb{R}^{N_u} \rightarrow \mathbb{R}^n$ , that maps the vector  $\mathbf{u} = [s_1, \dots, s_{N_u}]^T$  of control coefficients, cf. (22), onto the control action  $B\mathbf{u}$  is defined as

$$B = \left[ \int_{\Omega} \phi_i : \mathcal{B} \vec{v}_l \, dx \right]_{i=1, \dots, n; l=1, \dots, N_u}. \quad (23)$$

For the output space for the velocity measurements, given  $q \in 2\mathbb{N}$ , we define  $Y \times Y = Y_0 \oplus Y_1$  similar to  $U \times U$  with basis functions for both spatial components as in (21). As basis functions we use  $q$  piecewise linear hat functions defined on  $[0, 1]$  partitioned into  $q/2 - 1$  equidistant segments. From the space-time separated ansatz for the discrete velocity (3) and from the definition of the output operator in (20) it follows that the measured output will always be of the form

$$\mathcal{P}_Y \mathcal{C}_v v(t)(\eta) = \sum_{j=1}^n v_j(t) \mathcal{P}_Y \mathcal{C}_v \phi_j(\eta).$$

Since for every  $j = 1, \dots, n$ , the term  $\mathcal{P}_Y \mathcal{C}_v \phi_j$  is in  $Y \times Y$  and, thus, can be expanded in the basis (21), the measurement from the discrete velocity can be written as

$$\mathcal{P}_Y \mathcal{C}_v v(t)(\eta) = \sum_{l=1}^q y_k(t) \vec{v}_l(\eta), \quad (24)$$

with a unique vector of coefficients  $\mathbf{y} := [y_1, \dots, y_q]^T$ .

These coefficients are obtained through the discrete approximation  $\mathcal{C}_v$  to  $\mathcal{P}_Y \mathcal{C}_v$  that maps the coefficient vector  $\mathbf{v}$  of the velocity approximation onto  $\mathbf{y}$  and that is defined through

$$\mathcal{C}_v = \mathcal{M}_Y^{-1} \left[ \int_0^1 \vec{v}_l(\eta) : \mathcal{C}_v \phi_j(\eta) \, d\eta \right]_{l=1, \dots, q; j=1, \dots, n}. \quad (25)$$

This assembling is basically taking the output for all basis functions, testing them against the basis functions of  $Y \times Y$ , and premultiplying the obtained matrix by the inverse of the mass matrix  $\mathcal{M}_Y := \left[ \int_0^1 \vec{v}_l(\eta) : \vec{v}_k(\eta) \, d\eta \right]_{k, l=1, \dots, q}$ .

The pressure based output as defined in (20) only depends on time, i.e.  $\mathcal{C}_p : L^2(0, T; \mathbb{R}^m) \rightarrow L^2(0, T)$ . Thus, the discrete representation  $\mathcal{C}_p$  of  $\mathcal{C}_p$  that maps the coefficient vector  $\mathbf{p}$  of a discrete pressure  $p$ , cf. (3), onto  $\mathcal{C}_p p \in L^2(0, T)$  is defined through

$$\mathcal{C}_p = [\mathcal{C}_p \psi_k]_{k=1, \dots, m}. \quad (26)$$

$$\begin{array}{llll}
\mathbf{B} & \leftrightarrow & [B]_I & \\
\mathbf{C}_v & \leftrightarrow & [\mathcal{C}_v]_I & \text{and} \\
\mathbf{C}_p & \leftrightarrow & \mathcal{C}_p & \\
\mathbf{M}_y & \leftrightarrow & \mathcal{M}_Y & \\
\mathbf{M}_u & \leftrightarrow & \mathcal{M}_U & 
\end{array}$$

Table 2: Parameters defining the discrete input and output operators and the mass matrices of the signal state spaces.

Depending on the setup, in the provided data files, the discrete input and output operators appear as listed in Table 2. In  $\mathbf{B}$ , the first  $N_u/2$  columns are associated with the control in  $x_0$  direction and the last  $N_u/2$  columns with control in  $x_1$  direction. Thus, to vary the number of control parameters one may discard a control direction or, thanks to the hierarchy of the basis functions, only use, e.g., the first three columns. Similarly, in  $\mathbf{C}$  the first  $q/2$  rows correspond to the  $x_0$  component of the output, while the last  $q/2$  rows correspond to the  $x_1$  component. To reduce the output dimension, one may also discard one component. Also one can discard single degrees of freedom in the output space. Note, however, that the basis in the output space is not hierarchical but the standard hat function basis.

### 9.3 Boundary Control

A common case of boundary control of a flow is defining a time-varying velocity profile on a part of the boundary. This models, for example, a valve at the periphery where one can inject a jet into the flow or exhaust certain amounts of the fluid.

For illustration, we assume that such a control takes place at a single part  $\Gamma_c$  of the boundary  $\Gamma$  and that we have constant Dirichlet or *do-nothing* conditions elsewhere. The generalization to more controlled segments of the boundary and to other boundary conditions elsewhere is straight forward.

So assume that the control action is given as

$$v(t)|_{\Gamma_c} = u(t). \quad (27)$$

One may enforce the Dirichlet conditions by simply setting the values of the discrete approximations at the boundary. However, this leads to the appearance of  $\dot{u}$ , cf. [2]. Therefore, we relax the Dirichlet to Robin conditions and approximate (27) via

$$u(t) \approx v(t)|_{\Gamma_c} + \alpha(p\vec{n} - \nu \frac{\partial v}{\partial \vec{n}})|_{\Gamma_c}, \quad (28)$$

with a penalization parameter  $\alpha$  that is intended to go to zero. This approximation of the Dirichlet control has been investigated in the context of optimal control of Navier–Stokes equations in [5].

$$\begin{aligned} \text{Bbc} &\leftrightarrow [B_{bc}]_I \\ \text{Abc} &\leftrightarrow [A_{bc}]_I \end{aligned}$$

Table 3: The linear operators realizing the boundary control through the Robin relaxation with  $\alpha = 1$ .

## 9.4 Assembling of the Boundary Control Operator

We assume that  $u$  is given as a sum of spatial shape functions with time dependent coefficients, i.e.  $u$  has the form  $u(x, t) = \sum_{l=1}^{N_u} g_l(x)u_l(t)$ .

Rearranging (28) to  $p\vec{n} - \nu \frac{\partial v}{\partial \vec{n}}|_{\Gamma_c} \approx \frac{1}{\alpha}u(t) - \frac{1}{\alpha}v(t)|_{\Gamma_c}$ , the control is numerically realized through the term  $f_N$ , cf. (12), which gives the linear coefficient

$$A_{bc} = \frac{1}{\alpha} \left[ \int_{\Gamma_N} \phi_i : \phi_j \, dx \right]_{i,j=1,\dots,n}, \quad (29)$$

and the source term,  $B_{bc}\mathbf{u}$ , where  $\mathbf{u} := [u_1, \dots, u_{N_u}]^T$  and

$$B_{bc} = \frac{1}{\alpha} \left[ \int_{\Gamma_N} \phi_i : g_l \, dx \right]_{i=1,\dots,n; l=1,\dots,N_u}.$$

Depending on the setup, in the provided data files, the discrete boundary control operators appear as listed in Table 3. There also  $\alpha$  is set to 1 so that different penalization parameter values can be realized through scaling **Abc** and **Bbc** correspondingly.

Note, that there is no constant contribution of other Dirichlet boundary values through  $A_{bc}\bar{\mathbf{v}}_\Gamma$ , cf. (9). A contribution may occur where a Dirichlet and a Robin node share some support on  $\Gamma_N$  (cf. (29)) which can only happen where Dirichlet and control boundaries meet. However, in the considered setups the control boundaries are in the neighborhood of zero Dirichlet so that the possible contribution would be zeroed out by the prescribed node value.

If one considers Robin control, the set  $\mathcal{I}_I$ , cf. (8), will also contain the Robin nodes as if they were inner nodes. Accordingly, these nodes are also solved for in the process of the simulations and they simply integrated as inner nodes in the assembling of the linear and nonlinear operators; cf. Sections 3 and 4.

## 10 Control Setups

We describe the particular setups that include control and observation.



## 10.1 Driven Cavity with Distributed Control and Observation

We consider the test case of the lid driven cavity described in Section 8.1 and add distributed control and observation that are defined and assembled as described in Section 9.

Specifically, we define the domain of control and observation of the velocity as  $\Omega_c := [0.4, 0.6] \times [0.2, 0.3]$  and  $\Omega_o := [0.45, 0.55] \times [0.5, 0.7]$ , see Figure 2. The pressure is observed in  $\Omega_p := [0.45, 0.55] \times [0.7, 0.8]$ .

Recall that the input and the velocity output were modelled with only one dimensional space dependencies, cf. (18) and (19). For the driven cavity the spatial components of the signal are modelled as follows

	$x_0$ -direction	$x_1$ -direction
input	space-varying	constant
output	constant (averaged)	space-varying

Table 4: Spatial components of the input and velocity output signals for the distributed control of the driven cavity problem.

## 10.2 Cylinder Wake with Distributed Control and Observation

The test case of the cylinder wake is described in Section 8.2. Again, we add distributed control and observation as defined in Section 9.

Specifically, we define the domain of control and observation of the velocity as  $\Omega_c := [0.27, 0.32] \times [0.15, 0.25]$  and  $\Omega_o := [0.6, 0.7] \times [0.15, 0.25]$ , see Figure 3. The pressure is observed in  $\Omega_p := [0.6, 0.64] \times [0.18, 0.22]$ .

Recall that the input and the velocity output were modelled with only one dimensional space dependencies, cf. (18) and (19). For the cylinder wake, the spatial components of the signal are modelled as follows

	$x_0$ -direction	$x_1$ -direction
input	constant	space-varying
output	constant (averaged)	space-varying

Table 5: Spatial components of the input and velocity output signals for the distributed control of the driven cavity problem.

## 10.3 Cylinder Wake with Boundary Control

The setup of the simulation and of the output definition is the same as described in Section 10.2. For the input we define and assemble a penalized Robin scheme as explained



Figure 5: Illustration of the control locations on the cylinder periphery.

in Section 9.4. Specifically, we define the control through injection and suction at outlets  $\Gamma_{c_1}$ ,  $\Gamma_{c_2}$  centered at the cylinder periphery at  $\pm\pi/3$  occupying  $\pi/6$  of the circumference. Then, we prescribe Dirichlet conditions for the velocity

$$v = n_1 g_1(x) u_1(t), \quad v = n_2 g_2(x) u_2(t),$$

at  $\Gamma_{c_1}$  and  $\Gamma_{c_2}$ , where  $n_{1/2}$  are the vectors that are normal to the outlets in their center and that point into the domain, where  $u_{1/2}$  are the magnitudes of the controls, and where  $g_{1/2}$  are the shape functions that parametrize the segments  $\Gamma_{c_1}/\Gamma_{c_2}$  via  $s \in [0, 1]$  and that return the value of

$$1 - 0.5(1 + \sin((2s + 0.5)\pi)),$$

thus realizing a shape function that is 1 in the center of the control outlets and that smoothly extends to the zero *no-slip* conditions at the adjacent boundaries.

## 11 The Data, Plotting, and Auxiliary Functions

### 11.1 The System Matrices Files

The provided data, see Table 1, are vectors and sparse matrices stored as `.mat` files in the *MATLAB 6* version.

The naming of the data is as

```
PROBLEMNAME__mats__NV*XX*_Re1[_bccontrol_alpha1].mat
```

where

- `PROBLEMNAME`={`cylinderwake`,`drivencavity`},
- `*XX*` denotes the size of the system, namely the dimension of the velocity approximation space, and
- `[_bccontrol_alpha1]` is present, if boundary control is realized.

drivencavity	cylinderwake	cylinderwake_bccontrol
N=10 ↔ NV=722	N=1 ↔ NV=5812	N=1 ↔ NV=5824
N=20 ↔ NV=3042	N=2 ↔ NV=9356	N=2 ↔ NV=9384
N=30 ↔ NV=6962	N=3 ↔ NV=19468	N=3 ↔ NV=19512

Table 6: Relation of the mesh parameters  $N$  and the discretization parameter  $NV$  for the considered problems and the provided data.

## 11.2 Files for Visualizations

Visualization is done through writing the values of the pressure or velocity approximation to `.vtu` files that can be viewed in *Paraview*. For this there are *json* files that, among others, contains the geometrical information of the mesh and the map of the coordinates of the approximating vectors to the mesh vertices and that are named as

`visualization_PROBLEMNAME_N*XX*.json`

with `PROBLEMNAME` as above and `*XX*` standing for the discretization level. Note that  $N$  is different from  $NV$  used for the naming of the system matrices, since on a given mesh topology one can define approximations of, e.g., different degrees of freedom. For the provided data, the relations between mesh parameters  $N$  and approximation scheme parameters  $NV$  are listed in Table 6.

## 11.3 Auxiliary Files

The data comes with two software modules, namely `conv_tensor_utils` that provides functionality for linearization and memory-efficient evaluation of the nonlinearity and `visualization_utils` that assists in writing output files for rendering in *Paraview*.

# 12 Numerical Examples

We illustrate the use of the data in some example setups.

## 12.1 Cylinder Wake

In this section, we consider example setups for the cylinder wake as described in Sections 8.2, 10.2, and 10.3.

### 12.1.1 Steady State with Re=40

Setup:	cylinderwake	Simulation results:
Steady state simulation		v : Figure 6
Reynolds number: Re=40		p: Figure 6
Grid: N=1		
To replicate this simulation run		
<code>cylinderwake_steadystate.{m,py}</code>		

### 12.1.2 Steady State with Re=40 with Boundary Control Impact

Setup:	cylinderwake	Simulation results:
Steady state simulation		v : Figure 7
Reynolds number: Re=40		p: Figure 7
Grid: N=2		
Robin-Penalization: palpha=1e-3		
Input: uvec=[1,-1].T		
To replicate this simulation run		
<code>cylinderwake_steadystate_bccontrol.{m,py}</code>		

### 12.1.3 Transient Case with Re=90

Setup:	cylinderwake	Simulation results:
Transient simulation with IMEX-Euler		v[t=tE] : Figure 8
Reynolds number: Re=90		p[t=tE]: Figure 8
Grid: N=3		C*v: Figure 9
Time grid: t0, tE, Nts = 0., 4., 2**11		C*p: Figure 9
Initial value: Stokes solution		
To replicate this simulation run		
<code>cylinderwake_tdp_vout_pout.{m,py}</code>		

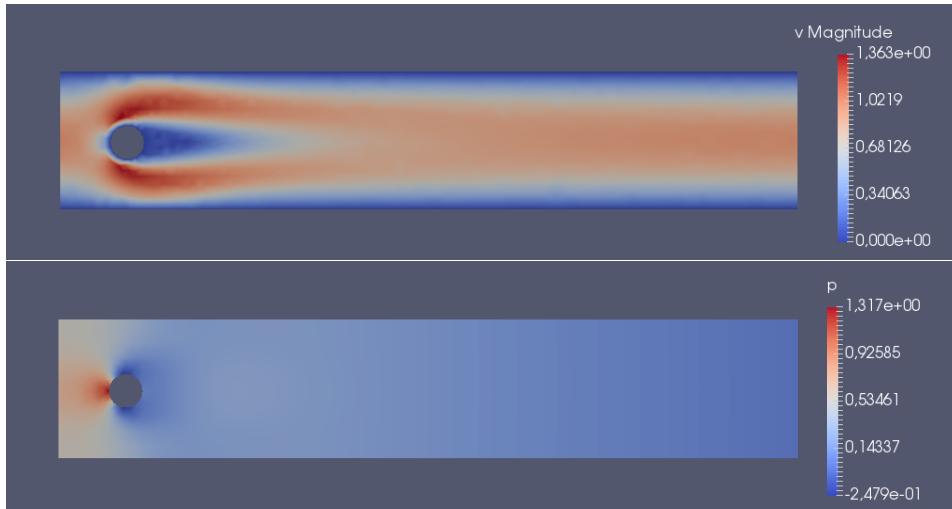


Figure 6: Illustration of the velocity magnitude and pressure field for the setup described in Section 12.1.1

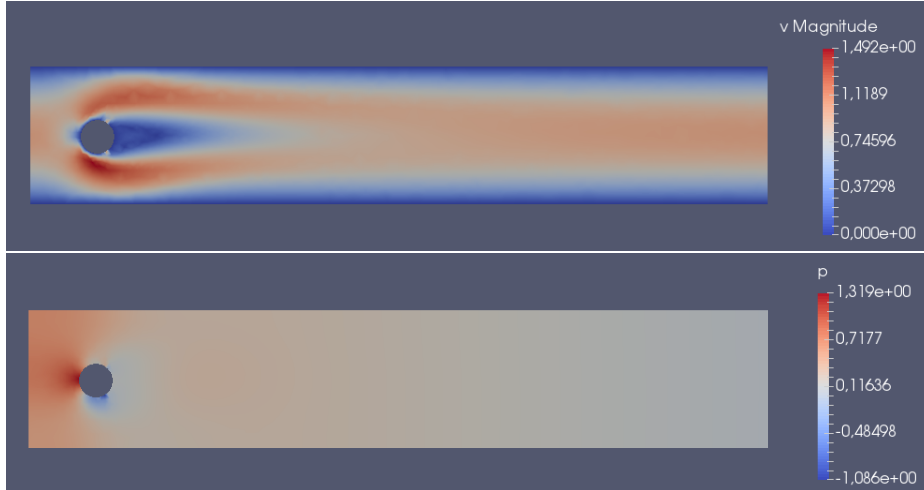


Figure 7: Illustration of the velocity magnitude and pressure field for the setup described in Section 12.1.2

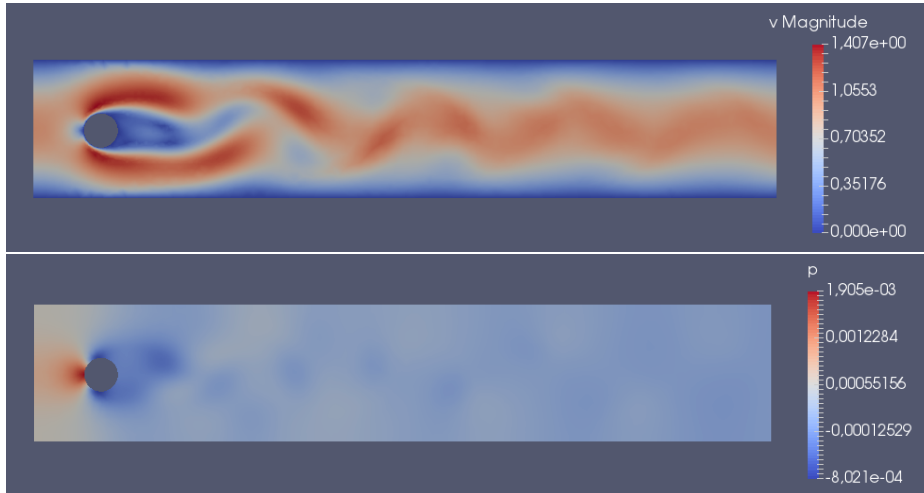


Figure 8: Illustration of the velocity magnitude and pressure field at  $t=4.0$  for the setup described in Section 12.1.3

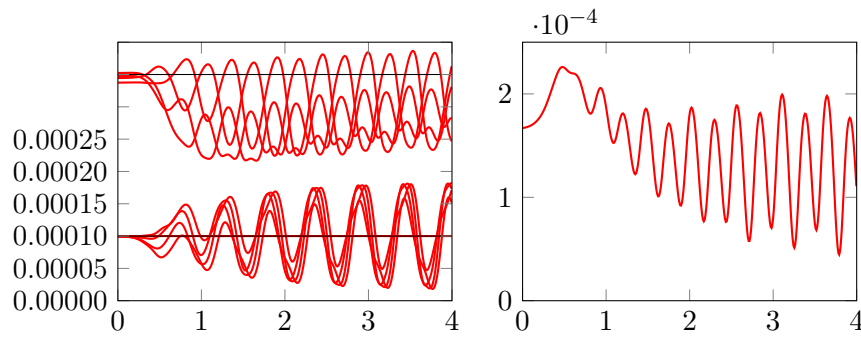


Figure 9: Illustration of the velocity and pressure output for the setup described in Section 12.1.3

### 12.1.4 Transient Case with $Re=40$ and Boundary Actuation

Setup:	<code>cylinderwake</code>	
Transient simulation with IMEX-Euler		
Reynolds number:	<code>Re=40</code>	
Grid:	<code>N=2</code>	Simulation results:
Time grid: $t_0, t_E, N_{ts}$	<code>= 0., 6., 2**10</code>	<code>C*v:</code> Figure 10
Initial value:	<code>Stokes solution</code>	<code>C*p:</code> Figure 10
Robin-Penalization:	<code>palpha=1e-3</code>	
Input frequency:	<code>omega=3*Pi/(tE-t0)</code>	
Input:	<code>uvec=sin(omega*t)*[1,-1].T</code>	
To replicate this simulation run		
<code>cylinderwake_tdp_pout_vout_bccontrol.{m,py}</code>		

## 12.2 Driven Cavity

In this section, we consider example setups for the driven cavity as described in Sections 8.1 and 10.1.

### 12.2.1 Steady State with $Re=1200$

Setup:	<code>drivencavity</code>	Simulation results:
Steady state simulation		
Reynolds number: $Re$	<code>=1200</code>	<code> v :</code> Figure 11
Grid:	<code>N=30</code>	<code>p:</code> Figure 11
To replicate this simulation run		
<code>drivencavity_steadystate.{m,py}</code>		

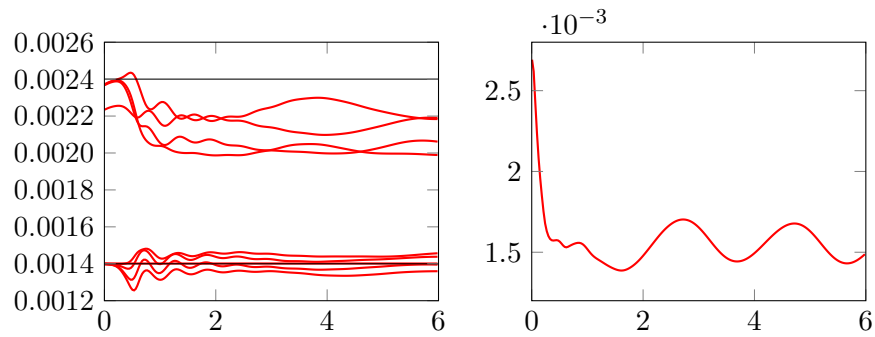


Figure 10: Illustration of the velocity and pressure output for the setup described in Section 12.1.4

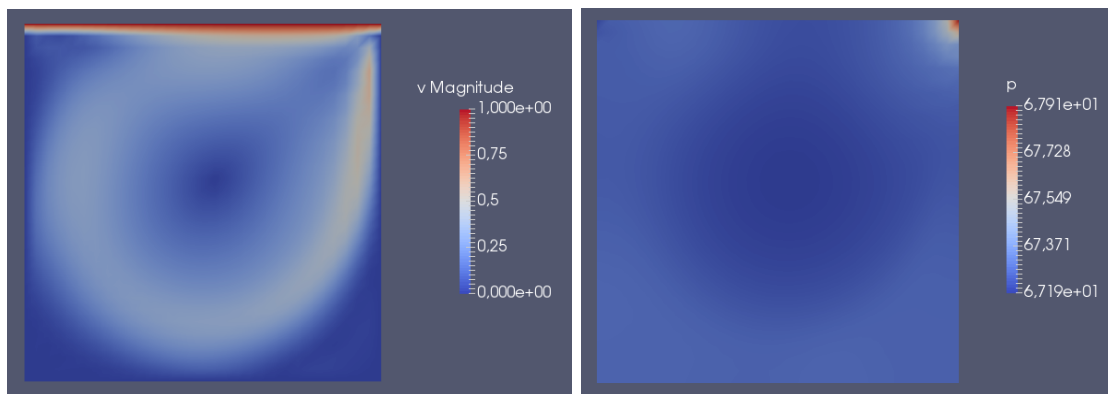


Figure 11: Illustration of the velocity magnitude and pressure field for the setup described in Section 12.2.1



### 12.2.2 Transient Case with $Re=800$ and Distributed Actuation

Setup:	drivencavity	
Transient simulation with IMEX-Euler		
Reynolds number:	Re=800	Simulation results:
Grid:	N=20	v(tk): Figure 12
Time grid: t0, tE, Nts = 0., 20., 2**12		C*v: Figure 13
Initial value:	Stokes solution	C*p: Figure 13
Input frequency:	omega=4*Pi/(tE-t0)	
Input: uvec=[sin(omega*t), cos(omega*t)].T		
To replicate this simulation run		
drivencavity_tdp_pout_vout_distcontrol.{m,py}		

## References

- [1] P. Benner and J. Heiland. LQG-Balanced Truncation low-order controller for stabilization of laminar flows. In R. King, editor, *Active Flow and Combustion Control 2014*, volume 127 of *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, pages 365–379. Springer, Berlin, 2015.
- [2] P. Benner and J. Heiland. Time-dependent Dirichlet conditions in finite element discretizations. *ScienceOpen Research*, 2015.
- [3] H. C. Elman, D. J. Silvester, and A. J. Wathen. *Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics*. Oxford University Press, Oxford, UK, 2005.
- [4] P. M. Gresho and R. L. Sani. *Incompressible Flow and the Finite Element Method. Vol. 2: Isothermal Laminar Flow*. Wiley, Chichester, UK, 2000.
- [5] L. S. Hou and S. S. Ravindran. Numerical approximation of optimal flow control problems by a penalty method: Error estimates and numerical results. *SIAM J. Sci. Comput.*, 20(5):1753–1777, 1999.
- [6] A. Logg, K. B. Ølgaard, M. E. Rognes, and G. N. Wells. FFC: the FEniCS form compiler. In *Automated Solution of Differential Equations by the Finite Element Method*, pages 227–238. Springer, Berlin, Germany, 2012.
- [7] B. R. Noack, K. Afanasiev, M. Morzyński, G. Tadmor, and F. Thiele. A hierarchy of low-dimensional models for the transient and post-transient cylinder wake. *Journal of Fluid Mechanics*, 497:335–363, 2003.
- [8] P. N. Shankar and M. D. Deshpande. Fluid mechanics in the driven cavity. In J. L. e. a. Lumley, editor, *Annual review of fluid mechanics*, volume 32, pages 93–136. Annual Reviews Inc., 2000.

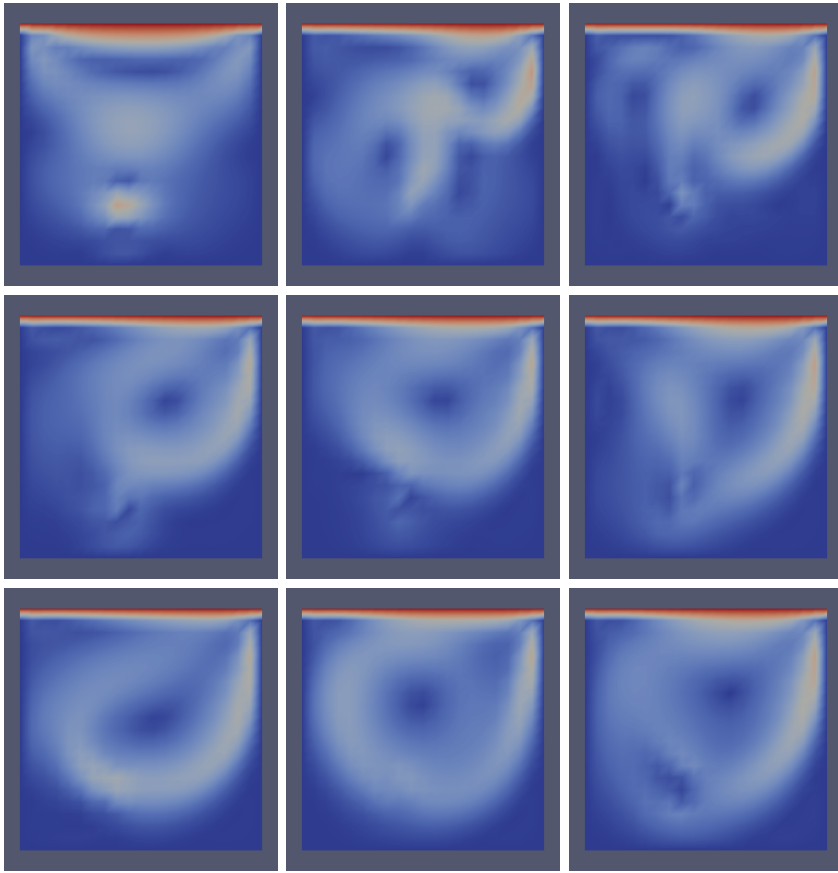


Figure 12: Illustration of the velocity magnitude field for the setup described in Section 12.2.2 at  $t=0.0, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0$ .

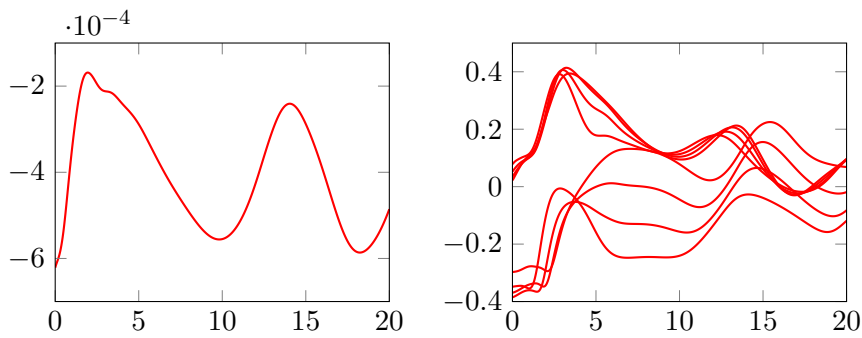


Figure 13: Illustration of the velocity and pressure output for the setup described in Section 12.2.2

- [9] D. Silvester, H. Elman, and A. Ramage. Incompressible Flow and Iterative Solver Software (IFISS) version 3.3, October 2013. <http://www.manchester.ac.uk/ifiss/>.
- [10] C. Taylor and P. Hood. A numerical solution of the Navier-Stokes equations using the finite element technique. *Internat. J. Comput. & Fluids*, 1(1):73–100, 1973.
- [11] C. H. K. Williamson. Vortex dynamics in the cylinder wake. *Annual Review of Fluid Mechanics*, 28(1):477–539, 1996.