Numerical benchmarking of fluid-rigid body interactions

Henry von Wahl[a,*], Thomas Richter[a], Christoph Lehrenfeld[b], Jan Heiland[c,a], Piotr Minakowski[a]

[a]*Institute for Analysis and Numerics, Otto-von-Guericke-University, Universitätsplatz 2, 39106 Magdeburg, Germany*
[b]*Institute for Numerical and Applied Mathematics, Georg-August-University, Lotzestr. 16-18, 37083 Göttingen, Germany*
[c]*Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstraße, 39106 Magdeburg, Germany*

Abstract

We propose a fluid-rigid body interaction benchmark problem, consisting of a solid spherical obstacle in a Newtonian fluid, whose centre of mass is fixed but is free to rotate. A number of different problems are defined for both two and three spatial dimensions. The geometry is chosen specifically, such that the fluid-solid partition does not change over time and classical fluid solvers are able to solve the fluid-structure interaction problem. We summarise the different approaches used to handle the fluid-solid coupling and numerical methods used to solve the arising problems. The results obtained by the described methods are presented and we give reference intervals for the relevant quantities of interest.

Keywords: Benchmarking, Computational fluid dynamics, Fluid–structure interaction, Finite Elements, Code validation, Reference values

1. Introduction

The interaction between a fluid flow and rigid bodies appears in many physical applications. The flow around a free rigid body causes both displacement and rotation of that body, via the forces and torque exerted from the fluid onto the body. Conversely, the motion of the body causes changes in the flow. In this work we focus on pure rotational effects of a single rigid sphere in two and three spatial dimensions by fixing its centre of mass.

We consider the fluid-structure interaction between an incompressible Newtonian fluid and a rigid body, which is free to rotate around it's centre of mass. The geometric set-up is based on the well known computational fluid dynamics (CFD) benchmark *Flow around a cylinder* [1], while adding the coupling between a freely rotating body and the fluid. The main advantage of this set-up is the constant fluid-solid partition over time, which allows the use of well established CFD methods to compute this fluid-structure interaction (FSI) problem, placing the focus on the coupling/decoupling approach used between the solid and the fluid.

Well posed benchmark problems are vital within computational mathematics. They are important for code validation and comparison of methods. For example the benchmark problem *Flow around a cylinder* from Schäfer and Turek [1] is one of the most widely used benchmarks within the computational fluid

---

*Corresponding author
   Email address: henry.vonwahl@ovgu.de (Henry von Wahl)
[1]ORCID: https://orcid.org/0000-0002-0793-1647
[2]ORCID: https://orcid.org/0000-0003-0206-3606
[3]ORCID: https://orcid.org/0000-0003-0170-8468
[4]ORCID: https://orcid.org/0000-0003-0228-8522
[5]ORCID: https://orcid.org/0000-0001-5154-2967

dynamics community. More recently Turek and Hron [2] proposed an FSI benchmark between an elastic obstacle and a laminar flow, while Hysing et al [3] proposed a benchmark for two dimensional bubbles rising in a fluid. In this work, we propose a benchmark problem in the setting of fluid-rigid body interactions.

The main contribution of this paper is to present a set of benchmark problems and relevant benchmark quantities for a fluid-rigid body interaction problem which can be used to quantitatively establish the quality of methods for fluid-particle interactions. This benchmark is more relevant in this situation than standard CFD benchmarks, since we incorporate the coupling between the solid and the fluid while keeping the set-up simple to implement.

Rotating spheres and cylinders in a channel flow have been studied widely in the literature, e.g. Badr et al [4], Fabre et al [5], Housiadas and Tanner [6], Juarez et al [7], Kong et al[8], Mittal and Kumar [9], Shaafin et al [10], Stojkovic et al [11] and Xia et al [12]. The main focus has been the effect of different configurations and parameter choices on the resulting flow. Computations have usually been done using a single method, thus reliable reference results for a given set-up are not available. The flow past a cylinder in two dimensions, rotating at a given prescribed rate, has been studied (see for example [8, 9, 11]) with the emphasis on the appearance of vortex shedding at different rotation rates, dependent on the Reynolds number. Furthermore, the effects of rotation speed and eccentricity (distance between the wall and cylinder) of the cylinder in two dimensions was studied [10] and the dynamics of a rotating cylinder moving at a constant speed [4]. The case of a two dimensional cylinder which is free to rotate was studied by Juarez et al[7], looking at the effects of eccentricity and Reynolds number on the rate and direction of rotation of the cylinder, while Xia et al [12] also took the blockage ratio into account. Three dimensional situations have also been considered in the literature [5, 6]. In Fabre et al [5] a sphere rotating freely around a transverse axis in a Newtonian fluid was considered and numerical results were compared to the case of a fixed sphere, while Housiadas and Tanner [6] studied a freely rotating sphere in a viscoelastic fluid and presented analytical results for the angular velocity.

The layout of this paper is as follows: In Section 2 we begin by describing the governing equations of this benchmark. We then describe the two dimensional set-up in Section 2.2 and the three dimensional set-up in Section 2.3, as well as defining the reference quantities which should be computed in each test case. In Section 3, we describe the different numerical methods and coupling/decoupling methods used to compute the described problems. The computed quantities of interest for the different benchmark set-ups and different methods are presented in Section 4.

2. Configurations of the benchmark problem

2.1. Governing equations

Consider a finite time-interval $I = [0, t_{\text{end}}]$ and a bounded domain $\Omega \in \mathbb{R}^d$ for $d \in \{2, 3\}$. We split the domain into a d-dimensional fluid region $\mathcal{F}$, a d-dimensional solid region $\mathcal{S}$ and a $(d-1)$-dimensional interface $\mathcal{I}$, such that $\Omega = \mathcal{F} \cup \mathcal{S} \cup \mathcal{I}$. In $\mathcal{F}$ we prescribe the incompressible Navier-Stokes equations

$$\rho_f \left(\partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u} \right) - \operatorname{div} \boldsymbol{\sigma}(\mathbf{u}, p) = \rho_f \mathbf{f}_f$$
$$\operatorname{div}(\mathbf{u}) = 0 \tag{1}$$

where $\mathbf{u} : \mathcal{F} \to \mathbb{R}^d$ denotes the fluid velocity field (note that we will denote vector valued functions with bold symbols), $p : \mathcal{F} \to \mathbb{R}$ the pressure, ρ_f the fluid's density, $\mathbf{f} : \Omega \to \mathbb{R}^d$ an external forcing vector acting on the fluid and

$$\boldsymbol{\sigma}(\mathbf{u}, p) = \rho_f \nu \left(\nabla \mathbf{u} + \nabla \mathbf{u}^T \right) - p \operatorname{Id}$$

denotes the Cauchy stress-tensor, with the kinematic viscosity $\nu > 0$. The fluid boundary is split into an inflow boundary Γ_{in}, an outflow boundary Γ_{out}, rigid no-slip wall boundaries Γ_{wall} and the fluid-solid interface $\mathcal{I}$. On the inflow, wall and interface boundaries we impose Dirichlet boundary conditions while on

the outflow condition we apply the do-nothing condition (see e.g. [13])

$$
\begin{aligned}
\mathbf{u} &= \mathbf{u}_{\text{in}} && \text{on } \Gamma_{\text{in}} \times I \\
\mathbf{u} &= 0 && \text{on } \Gamma_{\text{wall}} \times I \\
\mathbf{u} &= \mathbf{v}_s && \text{on } \mathcal{I} \times I \\
\rho_f \nu \left(\vec{\mathbf{n}} \cdot \nabla \right) \mathbf{u} - p\vec{\mathbf{n}} &= 0 && \text{on } \Gamma_{\text{out}} \times I
\end{aligned}
\tag{2}
$$

where $\mathbf{u}_{\text{in}}$ is a prescribed inflow-profile, $\mathbf{v}_s$ is the solid's velocity at the fluid-solid interface and $\vec{\mathbf{n}}$ is the outward pointing unit normal vector.

We assume that the rigid body $\mathcal{S}$ has uniform density and the motion of $\mathcal{S}$ is restricted to free rotation around its centre of mass. The motion can therefore be described by the object angular velocity $\boldsymbol{\omega}$ (in a counter-clockwise sense), which is governed by Newton's second law of motion. In our case, this can be stated as an ordinary differential equation for the angular velocity, given by

$$
J\partial_t\boldsymbol{\omega} = \mathbf{T}
\tag{3}
$$

where J is the body's moment of inertia and $\mathbf{T}$ is the total torque exerted onto the body by the fluid. Note that in two dimensions (3) is a scalar equation. The total torque $\mathbf{T}$ is then given by

$$
\mathbf{T} = \int_{\mathcal{I}} (\mathbf{x} - \mathbf{c}_{\mathcal{S}}) \times \left(\boldsymbol{\sigma}(\mathbf{u}, p)\vec{\mathbf{n}} \right) \mathrm{d}s
\tag{4}
$$

with the body's centre of mass $\mathbf{c}_{\mathcal{S}}$.

Remark. *Equations (3) and (4) are valid in three dimensions. In order to extend this formulation into two dimensions, we embed the the two dimensional domain trivially into three dimensions and take the scalar torque as the non-trivial component of the cross product in (3). A simpler formulation for this will be given below. Similarly, the angular velocity is then also taken as the one non-trivial component of the three dimensional angular velocity.*

2.2. Two-dimensional benchmark set-up

The set-up is based on the 2d-benchmark problems defined by Schäfer and Turek[1] and is chosen such that the partition of $\Omega = \mathcal{F} \cup \mathcal{S} \cup \mathcal{I}$ does not change over time in a geometric sense. This allows the use of standard CFD codes to compute these FSI problems.

We choose $\Omega = [0, 2.2] \times [0, 0.41]$ while the solid $\mathcal{S}$ is a circular obstacle placed just below the vertical center of the domain $\mathcal{S} = \{ \left(\begin{smallmatrix} x \\ y \end{smallmatrix} \right) \in \Omega : \left\| \left(\begin{smallmatrix} x-0.2 \\ y-0.2 \end{smallmatrix} \right) \right\|_2 < R \}$ with $R = 0.05$. The inflow data is defined by a parabolic inflow profile

$$
\mathbf{u}_{\text{in}}(t) := \frac{4U(t)y(0.41 - y)}{0.41^2} \begin{pmatrix} 1 \\ 0 \end{pmatrix}
$$

for some inflow speed U. The velocity at the interface is given by $\mathbf{v}_s = \omega R\vec{\mathbf{t}}$, where $\vec{\mathbf{t}}$ is the unit tangential vector (pointing in the anti-clockwise direction). The fluid viscosity is set to $\nu = 0.001$, the fluid density $\rho_f = 1$ and the solid density $\rho_s = 10$. In the case of a circle, the moment of inertia is then given by

$$
J = \rho_s \int_{\mathcal{S}} |\mathbf{x} - \mathbf{c}_{\mathcal{S}}|^2 \,\mathrm{d}\mathbf{x} = \rho_s \int_0^{2\pi} \int_0^R r^2 r \,\mathrm{d}r \,\mathrm{d}\theta = \rho_s \frac{\pi}{2} R^4.
$$

In the two dimensional case, (4) reduces to the scalar equation

$$
T = \int_{\mathcal{I}} R\vec{\mathbf{t}} \cdot \left(\boldsymbol{\sigma}(\mathbf{u}, p)\vec{\mathbf{n}} \right) \,\mathrm{d}s
$$

and as a result, the pressure does not contribute to changes in angular velocity.
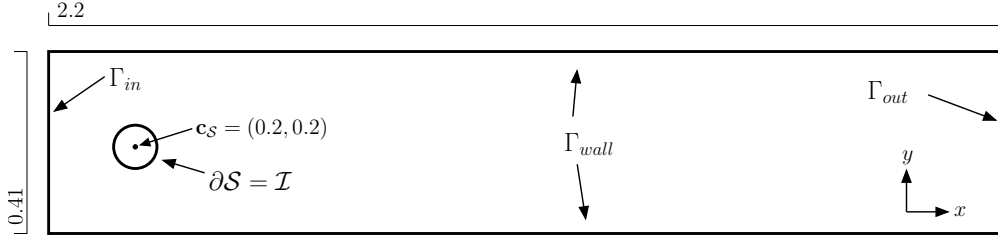
Figure 1: Spatial configuration of the two-dimensional benchmark problems.

As benchmark quantities, we are interested in the forces acting on the solid $\mathcal{S}$. The forces acting on the solid in the horizontal and vertical directions are

$$\mathbf{F} = \int_{\mathcal{I}} \boldsymbol{\sigma}(\mathbf{u}, p)\vec{\mathbf{n}}\, \mathrm{d}s. \tag{5}$$

As reference values, we then take the dimensionless drag (horizontal force, x-direction) and lift (vertical force, y-direction) coefficients which are defined as

$$C_D = \frac{2}{U_m^2 \rho_f L}\mathbf{F}_1 \quad \text{and} \quad C_L = \frac{2}{U_m^2 \rho_f L}\mathbf{F}_2$$

with the mean inflow speed U_m and characteristic length L, which we take to be the diameter of the circle, i.e., $L = 0.1$. To characterise the rotational force we take the dimensionless torque coefficient

$$C_T = \frac{4}{U_m^2 \rho_f L^2}T.$$

Furthermore, we compute the dimensionless angular velocity

$$\omega^* = \frac{\omega L}{2U_m}$$

and the pressure difference between the front and the back of the solid

$$\Delta p = p((0.15, 0.2)) - p((0.25, 0.2)).$$

To characterise the fluid we take the Strouhal number, defined as $St = {}^{Lf}/_{U_m}$, with the frequency of vortex shedding f, and the Reynolds number which we take as $Re = {}^{U_m L}/_{\nu}$.

2.2.1. Two-dimensional problem description

As in Schäfer and Turek[1], we consider three different two-dimensional test cases.

Rot2d-1 (Stationary). The inflow speed is set to $U = 0.3$. This results in a mean inflow speed of $U_m = {}^{2 \cdot 0.3}/_3 = 0.2$ which in turn gives the Reynolds number $Re = 20$.

The quantities to be computed are the drag, lift and torque coefficients C_D, C_L and C_T, the pressure difference Δp and the dimensionless angular velocity ω^*. The reference values are the stationary limit values of these quantities.

Rot2d-2 (Periodic). The inflow speed is $U = 1.5$, which gives a mean inflow speed of $U_m = 1$ and the Reynolds number $Re = 100$.

To be computed are the drag, lift and torque coefficients C_D, C_L and C_T, the pressure difference Δp and the angular velocity ω^* over one period $[t_0, t_0 + {}^1/_f]$ where $f = f(C_D)$ is the frequency of the drag coefficient. The reference values here are the maximum and minimum drag, lift and torque coefficients $C_{D,\max}$, $C_{D,\min}$, $C_{L,\max}$, $C_{L,\min}$, $C_{T,\max}$ and $C_{T,\min}$, the maximum and minimum dimensionless angular velocity $\omega^*_{\max}$ and $\omega^*_{\min}$, the Strouhal number and the pressure difference $\Delta p(t^*)$ at $t^* = t_0 + {}^1/_{2f}$, the midpoint of one period. The "initial" time t_0 corresponds to the time at which the maximum of the lift coefficient is realised.

Rot2d-3 (Unsteady & fixed time). We consider the fixed time interval $[0, 8]$ and the inflow speed $U(t) = 1.5 \sin(\pi t / 8)$. At the maximum in time, the resulting mean velocity (in space) is $U_m = 1$ and Reynolds number is again $Re = 100$. The initial state is $\mathbf{u}(0) = \mathbf{0}$.

To be computed are the drag, lift and torque coefficients C_D, C_L and C_T, the pressure difference Δp and the angular velocity ω^*. The reference values are the maximum drag, lift and torque coefficients $C_{D,\max}$, $C_{L,\max}$ and $C_{T,\max}$ and the respective times at which these are realised, the maximum dimensionless angular velocity $\omega^*_{\max}$ and the time at which it is realised as well as the pressure difference at time $t = 8$.

2.3. 3D Set-up

We consider the channel domain $\Omega = [0, 2.5] \times [0, 0.41] \times [0, 0.41]$ with a spherical obstacle $\mathcal{S} = \{\mathbf{x} \in \Omega : \|\mathbf{x} - \mathbf{c}_{\mathcal{S}}\|_2 < R\}$ with $\mathbf{c}_{\mathcal{S}} = (0.5, 0.2, 0.18)^T$ and $R = 0.05$, which is again free to rotate around it's centre of mass, $\mathbf{c}_{\mathcal{S}}$. This geometry has again been chosen such that the fluid-solid partition does not change over time and standard CFD codes can be used to compute this FSI problem. The inflow data for this problem is given by

$$\mathbf{u}_{\text{in}}(t) = \frac{16 U(t) y(y - 0.41) z(z - 0.41)}{0.41^4} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

with a maximum inflow speed $U(t)$. In the three dimensional case, the angular velocity $\boldsymbol{\omega}$ has three components, each of which represents the angular velocity around the corresponding axis. The coupling of the fluid and solid is then given by $\mathbf{u} = \mathbf{v}_s = \boldsymbol{\omega} \times (\mathbf{x} - \mathbf{c}_{\mathcal{S}})$ on $\mathcal{I}$. The fluid viscosity is set to $\nu = 0.001$, the fluid density is taken as $\rho_f = 1$ and for the solid we choose $\rho_s = 10$. In the case of a sphere, the moment of inertia is

$$J = \frac{2}{5} m R^2$$

with the mass of the sphere being $m = \rho_s 4/3 \pi R^3$. The torque is computed as in (4).

As reference quantities for this benchmark, we again take the forces acting on the obstacle. The forces acting in the coordinate axis are again defined by (5). As reference values we take the dimensionless coefficients, defined as

$$C_{\mathbf{F}_i} = \frac{2}{U_m^2 \rho_f L^2} \mathbf{F}_i \quad \text{and} \quad C_{\mathbf{T}_i} = \frac{4}{U_m^2 \rho_f L^3} \mathbf{T}_i$$

for $i \in \{1, 2, 3\}$, where we take the sphere diameter $L = 0.1$ as our reference length. The dimensionless angular velocity around each coordinate axis is given by $\boldsymbol{\omega}_i^* = \omega_i L / 2 U_m$. The pressure difference is defined as

$$\Delta p = p((0.45, 0.2, 0.18)) - p((0.55, 0.2, 0.18))$$

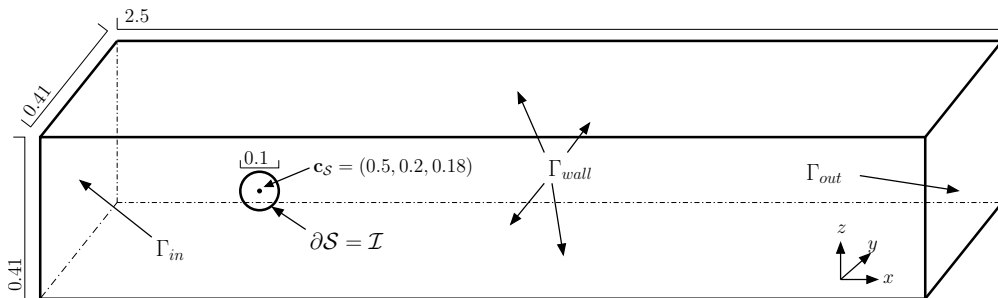while the Strouhal and Reynolds numbers are defined as in the two-dimensional case.

Figure 2: Spatial configuration of the three-dimensional benchmark problems.

5

2.3.1. Three-dimensional problem description

We only consider one stationary case, as we were unable to find a suitable stable configuration at higher Reynolds numbers.

Rot3d-1 (Stationary). We choose $U = 0.45$ giving a mean speed of $U_m = {}^{4\cdot0.45}/_9 = 0.2$. As a result of this choice we get the Reynolds number $Re = 20$.

To be computed are the three force coefficients $C_{\mathbf{F}_i}$, the Euclidean norm of the torque coefficient vector $\|C_{\mathbf{T}}\|_2$, the pressure difference Δp and the three components of the dimensionless angular velocity $\boldsymbol{\omega}^*$. The reference values are the stationary limits of these quantities.

3. Numerical methods

We describe the different methods utilised here to compute the described problems.

3.1. Grad-div stabilised high-order Taylor-Hood ($TH_{gd,ho}$)

We consider the conforming, inf-sup stable family of Taylor-Hood finite element pairs $\mathbb{P}^k/\mathbb{P}^{k-1}$ for $k \geq 2$ [14, 15] together with grad-div stabilisation [16], i.e. we add the term

$$\gamma_{gd}(\text{div}(\mathbf{u}_h), \text{div}(\mathbf{v}_h))_\Omega$$

to the (Navier-)Stokes bilinear form. In the presented computations the grad-div parameter was chosen as $\gamma_{gd} = 0.1$. For the convective term we use the standard convective form. The computations are performed using the high-order finite element library `Netgen`/`NGSolve` [17, 18]. Local element unknowns of higher-order elements are eliminated from the global system via a Schur-compliment and the local mesh size on the obstacle ${}^{h_{max}}/_5$ in both two and three dimensional computations. In the two dimensional case we chose $k = 5$ and in three dimensions $k = 4$.

3.1.1. Details for the stationary computations

We use a Newton approach to directly compute the steady state solution in order to solve the problems **Rot2d-1** and **Rot3d-1**. The angular velocity is identified by using a Newton method so solve the problem

$$j(\omega) \coloneqq \mathbf{T}(\mathbf{u}(\omega)) \overset{!}{=} 0.$$

The Jacobian $j'(\omega)(\delta\omega)$ is computed using a finite difference approximation

$$j'(\omega)(\delta\omega) \approx \frac{\mathbf{T}(\mathbf{u}(\omega+\delta)) - \mathbf{T}(\mathbf{u}(\omega))}{\delta}.$$

For our computations we take $\delta = 10^{-4}$. This approximation of the Jacobian is only updated if the torque has not sufficiently decreased with the update of the angular velocity. The arising non-symmetric linear systems are solved using the sparse direct solver `Intel MKL PARDISO` [19]. The functionals for drag lift and torque are evaluated using the Babuška-Miller trick, cf. Babuška and Miller [20] or Richter[21, Remark 8.17].

3.1.2. Details on the non-stationary computations

To decouple the Navier-Stokes equations (1) from the rotational ODE (3) in the system (2), we use a (semi-) implicit scheme to solve the PDE and use an explicit scheme to solve the ODE.

For the temporal discretisation of the grad-div stabilised Taylor-Hood method we use the second order SBDF2 Implicit-Explicit (IMEX) time-stepping scheme[22]. This scheme uses the BDF2 scheme for the discretisation of the time derivative and the linear Stokes part while a second order extrapolation is used to treat the convective term explicitly. As a result, we solve the same linear system in each time-step with

changing right-hand sides. We denote this system as M^*. In this system we use the standard weak form of the Stokes term

$$\nu(\nabla \mathbf{u}_h, \nabla \mathbf{v}_h) - (p_h, \text{div}(\mathbf{u}_h)) - (q_h, \text{div}(\mathbf{u}_h))$$

rather than the full symmetric stress tensor which would result in the viscous term $\nu(\nabla \mathbf{u}_h + \nabla \mathbf{u}_h^T, \nabla \mathbf{v}_h)$. As a result of this, the outflow condition in (2) is the natural do-nothing condition, meaning that we do not have to correct for terms on Γ_{out}, which would otherwise appear from integration by parts of (1). This has the effect that the system M^* we have to solve in each time step is symmetric, which we take advantage of. The resulting linear system is solved using NGSolve's sparse direct solver `sparsecholesky`. Nevertheless, the full symmetric Cauchy stress tensor is used to compute the forces acting on the obstacle. Rather than computing (4) and (5) directly, we us an equivalent volume integral formulation to compute the forces, since this is more stable and accurate than the boundary integral formulation[23]. We further note that the explicit treatment of the convective term results in a CFL condition, restricting the size of the time-step.

In conjunction with this IMEX scheme, we use the explicit part of the schemes to advance the angular velocity, i.e. we use the BDF2 formula to discretise the time derivative and use a second order extrapolation for the right-hand side.

3.2. High-order, exactly divergence-free hybrid discontinuous Galerkin method ($HDG_{\boldsymbol{H}^{div}}$)

As another method, we consider a space discretisation using high-order, exactly divergence-free hybrid discontinuous Galerkin (HDG) method based on $H(\text{div})$-conforming finite elements as presented in [24]. In this discretisation an *only* $H(\text{div})$-conforming finite element space Σ_h is used for the discretization of the velocity field, i.e. BDM elements on simplices and Raviart-Thomas elements on quads. The pressure space Q_h is chosen as $Q_h = \text{div} \, \Sigma_h$ which renders the weak divergence constraint a strong divergence constraint yielding solenoidal discrete solutions. Elements in Σ_h are only normal-continuous and especially not H^1-conforming. To incorporate tangential continuity techniques from Discontinuous Galerkin (DG) methods are required. To eliviate the costs of these methods, hybrid versions of DG methods are applied here. In hybrid DG methods additional polynomial unknowns on the facets – here polynomials in tangential direction only – are introduced. These unknowns are used to break up the direct direct communication between element neighbours and allowing for static condensation.

The resulting spatial discretisation has several benefits such as energy stability and pressure robustness [25, 26] while allowing for efficient and high order accurate implementations [24, 27].

The discretisation of the viscous term is based on a hybrid version of the well-known symmetric interior penalty method (with stabilisation parameter $\alpha = 20$, cf. [28] for an introduction to HDG methods) whereas a standard Upwind technique is applied for the discretisation of the convection.

For the two-dimensional problems we use a triangular mesh with mesh size $h = 0.1 \cdot 2^{-L}$, for $L \in \{0, 1, .., 5\}$ with two layers of anisotropic quadrilaterals in the region with distance $\min\{0.02, h\}$ around the circle. Around the obstacle elements are curved up to the order k that is also used for the velocity space. For the three-dimensional problems we take the coarsest mesh which was also used for the computations in Section 3.1.1. For convergence studies we consider mesh refinements and refinements of the polynomial degree resulting in different pairs of mesh levels and polynomial orders (L, k). All the examples in this study with the $H(\text{div})$-conforming finite element method are computed with the high-order finite element library `Netgen/NGSolve` [17, 18].

3.2.1. Details for the stationary computations

For the stationary benchmarks problems **Rot2d-1** and **Rot3d-1** we use a Newton's method with a sparse direct solver `Intel MKL PARDISO` [19] for the linear systems as in Section 3.1.1. The Jacobi matrix is reused several times until the residual reduction becomes insufficient. As starting values for the Newton iteration we use the solution of the Stokes problem.

3.2.2. Details for the unsteady computations

Again, we use the same approach as in Section 3.1.2, i.e. an operator splitted time integration with the same SBDF2 scheme. Let us note that due to the explicit treatment the Upwinding of the convection

discretisation does not enter the linear systems that need to be solved. As a consequence the linear systems that need to be solved for are symmetric which allows to apply superconvergence strategies from HDG methods, cf. [24, Section 2.2]. Here, we compute the functionals (4) and (5) directly from boundary integrals. For the time step size we use the heuristic $\Delta t = \left(\left\lceil 1250 \cdot 2^L \cdot \left(\frac{k}{5} \right)^{3/2} \right\rceil \right)^{-1}$. For accessing the minima and maxima of the time dependent quantities of interest, we use a cubic spline interpolation of the equidistantly sampled data obtain after every time step.

3.3. Newton-multigrid for equal order finite elements with local projection stabilisation (EO_{LPS})

Space discretisation is based on biquadratic (triquadratic in 3d) finite elements on quadrilateral and hexahedral meshes for velocity and pressure unknowns. To stabilise the inf-sup condition a local projection method based penalisation of the divergence condition with respect to the space of bilinear finite elements is employed, i.e. we add

$$\sum_{K \in \Omega_h} \alpha_K \left(\nabla(p_h - i_h^{(1)} p_h), \nabla(q_h - i_h^{(1)} q_h) \right)_T$$

to the weak discrete Navier-Stokes equation. By $i_h^{(1)} : V_h^{(2)} \to V_h^{(1)}$ we denote the discrete interpolation from the space of biquadratic functions $V_h^{(2)}$ to the space of bilinear functions $V_h^{(1)}$ on the same mesh Ω_h. The stabilisation parameter is chosen as $\alpha_K = 0.1 h_K \cdot (\nu h_K^{-1} + \|\mathbf{u}_h\|_{\infty, K})^{-1}$ and depends on the local mesh size h_K on element $K \in \Omega_h$ and the local velocity. Details are given in [29]. For temporal discretisation of the Navier Stokes equations and the rigid body problem we use BDF methods of order 3. Surface integrals like drag or torque are evaluated in variational formulation based on the Babuška-Miller trick, see [20] or [21, Remark 8.17]. Given sufficient regularity of the solution we obtain fourth order convergence in these functional values [30].

The arising algebraic systems are approximated by Newton's method, where the Jacobian is computed analytically. Linear systems of equations are solved by a GMRES method, preconditioned with a geometric multigrid solver. As smoother we either employ an incomplete lower upper decomposition or a block-Jacobi iteration to employ some parallelisation. Details are described in [31, 32].

3.3.1. Details on the stationary benchmark problems

The approach as described in Section 3.1.1 is used to compute the stationary states in **Rot2d-1** and **Rot3d-1** directly. For the finite difference approximation of the Jacobian, $\delta = 10^{-3}$ is taken. An alternative to determine the Jacobian is to solve tangent problems, however, we do not take this approach to avoid costly matrix assemblies that would be required. For a finite difference approximation the available Jacobian can be reused.

3.3.2. Details on the non-stationary benchmark problems

Implicit discretisation methods (BDF-3) are used for the Navier-Stokes equations and the rigid body problem. Coupling of the two problems is achieved by a simple outer iteration. A first prediction for the angular velocity is obtained by the explicit second order Heun's formula. In every time step $t_n \to t_{n+1}$ we first predict the angular velocity $\omega_n \xrightarrow[(\mathbf{u}_n, p_n)]{} \omega_{n+1}^{(0)}$ and set $\mathbf{u}_{n+1}^{(0)} = \mathbf{u}_n$. For $l = 1, 2, \ldots$ we iterate

1. Solve the Navier-Stokes equations $\mathbf{u}_{n+1}^{(l-1)} \xrightarrow[\omega_{n+1}^{(l-1)}]{} (\mathbf{u}_{n+1}^{(l)}, p_{n+1}^{(l)})$ with BDF-3

2. Solve the rigid body problem $\omega_{n+1}^{(l-1)} \xrightarrow[(\mathbf{u}_{n+1}^{(l)}, \mathbf{u}_n^{(l)})]{} \omega_{n+1}^{(l),*}$ with BDF-3 and relax the update $\omega_{n+1}^{(l)} :=$

   $(1 - \gamma)\omega_{n+1}^{(l-1)} + \gamma \omega_{n+1}^{(l),*}$
3. Stop if $|\omega_{n+1}^{(l)} - \omega_{n+1}^{(l-1)}| \leq 10^{-10} \cdot k$ where k is the time step.

The relaxation parameter is mostly set to $\gamma = 1$. Usually only two or three iterations are required and in most cases, the Navier-Stokes equations have to be solved only once as the initial Newton residual is still below the tolerance after updating $\omega_{n+1}^{(l)}$.

Since implicit time stepping formulas allow for large time step sizes, the periodic **Rot2d-2** benchmark problem requires a good capturing of the period length $1/f$. After each approximate period $t_0 \mapsto t_n$ we therefore update the time-step to minimise

$$\sum_{i=1}^{n-1} |\mathcal{P}^2[\omega](t_i + s) - \omega(t_{i-n})|^2 \to \min,$$

where $\mathcal{P}^2[\omega]$ is a quadratic reconstruction of the discrete angular velocities. Expecting fourth order convergence in space, we combine each mesh refinement $h \mapsto 2^{-1} \cdot h$ with a time step refinement of $k \mapsto 2^{-4/3}k$ to globally achieve fourth order.

3.4. Standard Taylor–Hood elements (TH)

We consider standard Taylor–Hood finite elements[14] consisting of continuous $\mathbb{P}^2$ Lagrange elements for the velocity component and continuous $\mathbb{P}^1$ Lagrange elements for the pressure field. As mentioned above this pair is inf-sup stable.

The implementation is done in `FEniCS (2018.1.0)` [33, 34]. The simulation are run on the prebuilt Anaconda Python packages from [35]. In 2D, the meshes where made using the built-in FEniCS mesh generation component `mshr` and the circular obstacle was resolved with a local mesh size $h_{max}/8$. In 3D the meshes where made using `Netgen`[17] with the local mesh size at the sphere also of $h_{max}/8$ and converted with a `Python` script `dolfin-convert`.

The forces are computed based on the equivalent volume integral formulation which is more accurate than the boundary formulation as well as more stable and robust with respect to the boundary approximation [23].

3.4.1. Details on the stationary benchmark problem

Stationary computations follow the approach described in Section 3.1.1. However, the parallel sparse direct solver MUMPS[36] was used to solve the arising linear systems.

3.4.2. Details on the non-stationary benchmark problems

Temporal discretisation of the Navier-Stokes system (1) is realised using the second order Crank-Nicolson scheme. The resulting non-linear systems are then solved using Newtons method. For time discretisation of the ordinary differential equation for the angular velocity (3) we employ the appropriate explicit linear multistep scheme, the second order Adams-Bashforth method.

3.5. Taylor-Hood and Optimisation or Method of Lines (TH_{scipy})

The spatial discretisation was done with Taylor-Hood $\mathbb{P}^2/\mathbb{P}^1$ elements in `FEniCS` [33] via its `Python` interface `dolfin`. With the help of the `Python` module `dolfin_navier_scipy` [37], all discrete operators were interfaced to `SciPy`[38] for the solution of the steady-state problem, the optimisation, and the numerical time integration.

The Dirichlet boundary values were included by directly assigning the relevant values to the corresponding *degrees of freedom*; see [39, Rem. 3.2] on how to adjust a time stepping scheme accordingly.

The boundary integrals in the functionals for drag lift and torque were approximated using the Babuška-Miller trick, cf. [20] or [21, Remark 8.17].

3.5.1. Details on the stationary benchmark problem

As mentioned in Section 3.1.1, the stationary limit can be computed via finding the ω for which the resulting torque force is zero:

$$\mathbf{T}(\mathbf{u}(\omega)) \overset{!}{=} 0. \tag{6}$$

We use the builtin function `scipy.optimize.brent` which employs the *brent* algorithm [40] to minimise the scalar univariate function $\omega \mapsto \mathbf{T}(\mathbf{u}(\omega))^2$.

3.5.2. Details on the non-stationary benchmark problem

For the time integration, we use the Crank-Nicolson scheme on the linear part of the momentum equation and second order Adams-Bashforth for the nonlinear part, while enforcing the incompressibility constraint explicitly in every time step. This implict-explicit scheme is of order two and was already used in [41] and analysed in [42]. Because of the explicit treatment of the convection and because of the equidistant time grid, in every time step, the same linear system is to be solved. For this, an *LU factorisation* of the corresponding coefficient matrix was precomputed. The ODE for the angular velocity (3) was discretised with the second order Adams-Bashforth method.

To reduce the time until the system has reached the limit cycle, we started from the corresponding steady-state solution with angular velocity ω such that $\mathbf{T}(\mathbf{u}(\omega)) \approx 0$ as computed with the optimisation method described above in Section 3.5.1. For higher Reynolds numbers the Newton iterations for the steady state converge only locally. We obtained convergence with the steady states solutions that correspond to $\nu = 800$ or $Re = 80$ as initial guess.

4. Results

We present the numerical results for the benchmark problems obtained by the various methods described in Section 3. Next to the reference values, we also provide further information on the discretisation through the number of degrees of freedom and the number on non-zero entries of the resulting linear systems which needed to be solved. The non-zero entries indicate the sparsity of the resulting matrix and therefore the effort needed to solving the system.

For $\mathrm{TH}_{\mathrm{gd,ho}}$ and $\mathrm{HDG}_{H^{div}}$ we also provide the number of degrees of freedom after removing internal degrees of freedom from the global system using a Schur complement, known as static condensation. As a result the non-zero entries are also only counted for the system after static condensation, i.e., the System which has to be solved in the method. Furthermore, we provide hardware information and computation times for the simulations. This is simply to indicate how challenging the proposed benchmark problems are.

Rot-2d1. The results obtained for the stationary *Rot-2d1* benchmark, computed using the methods described in Section 3, can be seen in Table 1. We summarise these results in Table 2, where we give reference intervals for the quantities of interest based on the full results in Table 1.

We see that the reverence intervals have sizes between the order of magnitude 10^{-5} for C_D and 10^{-8} for ω^*. This in combination with the relatively short computation times shows that the problem *Rot-2d1* can be seen as easily computable. Furthermore, we see that higher-order methods are particularly suited to this problem and that the *direct* quasi Newton approach as described in Section 3.1.1 is particularly efficient for this problem.

Rot-2d2. The results for the problem *Rot-2d2* are summarised in Table 3. Ranges for the reference values for the quantities of interest for this problem are given in Table 4. A plot of the quantities of interest over two periods can be seen in Figure 3.

We can see here that the non-stationary problem *Rot-2d2* is more challenging, because the interval lengths for the reference values is between the order 10^{-3} and the order 10^{-5}. We also note that the angular velocity seem to be comparably easy quantities to compute as the reference intervals are smallest for this quantity.

Rot-2d3. The quantities of interest in the problem *Rot-2d3* are presented in Table 5 and the resulting reference intervals are given in Table 6. The quantities of interest are also plotted over the entire time interval in Figure 3.

With respect to the spatial discretisation, high-order methods have performed very well here. With respect to the temporal integration, we note that this problem seems to be challenging. On the one hand, the forth order (by the choice of time-steps) time-discretisation in TR gave good results with relatively large time steps while the second order SBDF2 IMEX time stepping with $\mathrm{TH}_{\mathrm{gd,ho}}$ also gave good results

on (relatively) coarser meshes in combination with very small time-steps. This problem therefore seems particularly suited to assess the performance of time-integration methods for Navier-Stokes.

With respect to accuracy of the individual values, we note the most difficult to compute value with respect to the absolute error is $C_{L,\max}$ while we can give the smallest reference interval for $\omega^*_{\max}$. However, the times at which the extreme values are realised are all only accurate up to the order 10^{-3}.

Looking at the computational times we note that the problem is significantly more challenging than the stationary problem, however, it is also still a reasonable problem to compute.

Rot-3d1. The values computed for the problem Rot-3d1 can be seen in Table 7. We summarise these results in reference intervals given in Table 8.

From the reference intervals we see that ω^*_1 is especially challenging to compute accurately since it is relatively small in comparison to the other values. The change of sign between different computations is therefore not surprising. The force coefficient $C_{\mathbf{F}_2}$ seems to be the easiest to compute while the remaining values seem equally challenging.

Discretisation			Results					
Method	#dof [K]	#nze [M]	C_D	C_L	C_T	Δp	ω^*	Comp.T. [s]
$\mathrm{TH_{gd,ho}}$[1]	15.6(8.0)	0.6	5.5795523091	0.0047151364	-4.10e-16	0.1174948237	0.0012629527	4.1
$k=5$	51.1(26.8)	2.0	5.5795587061	0.0047141285	-3.03e-16	0.1175213881	0.0012629334	6.2
	186.5(99.7)	7.5	5.5795588114	0.0047141929	6.03e-16	0.1175205313	0.0012629345	19.6
	693.6(374.2)	28.0	5.5795588133	0.0047141930	7.28e-16	0.1175202728	0.0012629346	68.1
$\mathrm{EO_{LPS}}$[2,3]	33.4	0.5	5.579526747	0.004727828	1.67e-12	0.117518677	0.00126296256	5.7
	131.9	2.1	5.579556532	0.004715150	-2.15e-15	0.117519524	0.00126292998	31.6
	523.9	8.3	5.579558664	0.004714255	-2.17e-15	0.117520022	0.00126293418	137.2
	2088.1	33.3	5.579558804	0.004714197	-4.47e-16	0.117520193	0.00126293459	604.3
	extrapolated		5.579558814	0.004714193	-	0.117520282	0.00126293463	
	(order)		(3.93)	(3.95)	-	(2.00)	(3.36)	
TH[4]	9.8	0.3	5.5908135746	0.0034359338	0	0.1175346984	0.0011602113	0.4
	29.7	0.8	5.5829138831	0.0054926020	-2.71e-16	0.1174672688	0.0014131732	1.2
	99.7	2.9	5.5801404856	0.0048778267	2.37e-16	0.1175411687	0.0012546495	4.6
	362.4	10.6	5.5795742677	0.0047489983	0	0.1175218581	0.0012574215	20.3
	1379.1	40.6	5.5795553554	0.0047154167	-1.07e-13	0.1175158081	0.0012628280	127.0
	5370.2	158.7	5.5795581671	0.0047141845	1.22e-14	0.1175193123	0.0012629381	657.0
$\mathrm{HDG}_{H^{div}}$[5] $(L,k)=(0,\ 5)$	8.9(5.2)	0.4	5.5795057887	0.0047518775	-9.99e-13	0.1175238245	0.0012598766	6.5
$(L,k)=(0,\ 6)$	25.4(6.1)	0.5	5.5795547031	0.0047188391	-7.19e-14	0.1175209652	0.0012621510	8.3
$(L,k)=(0,\ 7)$	33.0(7.0)	0.6	5.5795584283	0.0047140877	-7.44e-13	0.1175200945	0.0012629298	9.9
$(L,k)=(0,\ 8)$	41.5(7.8)	0.8	5.5795588381	0.0047141851	-7.81e-13	0.1175202550	0.0012629342	19.1
$(L,k)=(0,\ 9)$	51.0(8.7)	1.0	5.5795588122	0.0047141915	-9.51e-16	0.1175202619	0.0012629337	16.3
$(L,k)=(0,10)$	61.5(9.6)	1.2	5.5795588016	0.0047141905	-3.12e-13	0.1175202597	0.0012629346	67.5
$\mathrm{HDG}_{H^{div}}$[5] $(L,k)=(0,\ 5)$	18.9(5.2)	0.4	5.5795057887	0.0047518775	-9.99e-13	0.1175238245	0.0012598766	6.5
$(L,k)=(1,\ 5)$	50.1(14.7)	0.9	5.5795397082	0.0046822197	-8.23e-13	0.1175243136	0.0012676502	12.5
$(L,k)=(2,\ 5)$	181.1(55.4)	3.4	5.5795402179	0.0047150139	7.35e-13	0.1175244061	0.0012628701	23.4
$(L,k)=(3,\ 5)$	745.5(231.8)	14.1	5.5795566498	0.0047142079	2.89e-14	0.1175205506	0.0012629366	126.0
$(L,k)=(4,\ 5)$	2949.9(924.2)	55.8	5.5795587277	0.0047141940	-7.16e-13	0.1175202662	0.0012629347	615.3
$\mathrm{TH_{scipy}}$[6]	30.9	0.6	5.5787704610	0.0047091588	-1.84e-13	0.1174850027	0.0012628449	68.1
	77.9	1.6	5.5792205094	0.0047130497	-2.29e-13	0.1175153993	0.0012627522	284.2
	273.0	5.7	5.5794655316	0.0047136722	-2.04e-13	0.1175255316	0.0012628650	2668.0
	696.4	14.5	5.5795218195	0.0047139943	-2.90e-14	0.1175224057	0.0012629083	13806.5
	1931.4	40.2	5.5795454842	0.0047141221	-1.32e-12	0.1175200148	0.0012629253	68856.1

Table 1: Results for Rot-2D1. The number of degrees of freedom (dof) is given in $K = 10^3$. Dofs in brackets are the number of unconstrained dofs after static condensation. The number of non-zero entries (nze) of the condensed (if applicable) system matrix are given in $M = 10^6$. The computational run-time is given in seconds s.

---

[1]Computed using AMD Ryzen 5 2400G, 3.6GHz using 4 cores.
[2]Computed on MacBookPro using Intel i7-4870HQ, 2.5 GHz using 4 cores.
[3]Extrapolated values are indicated if fitting to $a(h) = a + ch^q$ is possible.
[4]Computed using Intel Core E5620, 2.4GHz with 4 cores and 8 threads.
[5]Computed using Intel Core i7-8650U, 4.2GHz with 4 cores and 8 threads.
[6]Computed using Intel Xeon E7-8837, 2.67GHz, using 1 (physical and virtual) core.

11

Coefficient	C_D	C_L	Δp	ω^*
Value range	$[5.57954, 5.579559]$	$[0.0047141, 0.0047142]$	$[0.117519, 0.117521]$	$[0.001262925, 0.00126294]$

Table 2: Reference value ranges for the quantities of interest for the problem Rot-2d1.

Discretisation				Results									
Method	#dof $[K]$	#nze $[M]$	τ	$C_{D,\max}$	$C_{D,\min}$	$C_{L,\max}$	$C_{L,\min}$	$C_{T,\max}$	$C_{T,\min}$	$\Delta p(t^*)$	$\omega^*_{\max}$	$\omega^*_{\min}$	St
$\text{TH}_{\text{gd,ho}}$	15.6(8.0)	0.3	$^1/_{2500}$	3.2257143516	3.1623995194	0.9682189506	-1.0272488814	0.0177090000	-0.0177300000	2.4832272754	0.0045764494	0.0021991719	0.3019323671
$k=5$	51.1(26.8)	1.1	$^1/_{5000}$	3.2257765946	3.1624660264	0.9676789054	-1.0273532513	0.0176960000	-0.0177170000	2.4834909048	0.0045289959	0.0021540270	0.3019323671
	186.5(99.7)	3.9	$^1/_{10000}$	3.2257012749	3.1624311702	0.9674937697	-1.0271414624	0.0176920000	-0.0177140000	2.4840487990	0.0045308702	0.0021563680	0.3018412315
	693.6(374.2)	14.5	$^1/_{20000}$	3.2257003973	3.1624161918	0.9674952132	-1.0271424650	0.0176920000	-0.0177140000	2.4841879091	0.0045307692	0.0021562635	0.3018412315
EO_{LPS}^3	22.5	1.0	$t_P/20$	3.265238	3.184170	1.092590	-1.152795	0.0199021	-0.01992426	2.516449	0.00468163	0.00199515	0.299292
	88.0	4.1	$t_P/50$	3.229858	3.164937	0.980084	-1.039833	0.0179042	-0.01792674	2.487680	0.00454846	0.00214495	0.301792
	348.0	16.5	$t_P/126$	3.225959	3.162583	0.968258	-1.027915	0.0177047	-0.01772654	2.484521	0.00453230	0.00215602	0.301871
	1 348.1	66.1	$t_P/320$	3.225713	3.162425	0.967522	-1.027180	0.0176923	-0.01771480	2.484473	0.00453212	0.00215744	0.301869
	extrapolated			3.225697	3.162414	0.967473	-1.027131	0.0176915	-0.01771407	2.484318	0.00453120	0.00215765	c.o.s.
	(order)			(4.00)	(3.90)	(4.09)	(4.02)	(4.01)	(4.09)	(3.27)	(3.14)	(2.96)	-
TH	9.8	0.3	$^1/_{40}$	3.1140915806	3.0835435234	0.6942111384	-0.7604045006	0.0140683780	-0.0140959295	2.4312708845	0.0055445513	0.0034041747	0.2884975288
	29.7	0.8	$^1/_{80}$	3.2150998204	3.1562985910	0.9535596576	-1.0168443659	0.0176195464	-0.0176358975	2.4690328925	0.0048579972	0.0024203675	0.2995190345
	99.7	2.9	$^1/_{160}$	3.2245565079	3.1614311213	0.9674861840	-1.0274123288	0.0176662978	-0.0176716112	2.4739243898	0.0045337913	0.0021466233	0.3014554428
	362.4	10.6	$^1/_{320}$	3.2256301394	3.1621590399	0.9687958092	-1.0285405038	0.0178384619	-0.0178564492	2.4826673308	0.0045427458	0.0021447040	0.3017710165
	1379.1	40.6	$^1/_{640}$	3.2256835394	3.1622852649	0.9685324763	-1.0281714901	0.0178273180	-0.0178503246	2.4837365764	0.0045386903	0.0021448458	0.3018441883
TH_{scipy}	30.9	0.6	2^{-12}	3.2258901997	3.1644408460	0.9719117446	-1.0316255447	0.0177891954	-0.0178119760	2.4836587332	0.0045491972	0.0021580693	0.3014232379
	77.9	1.6	2^{-12}	3.2253415537	3.1645119141	0.9678401829	-1.0274808411	0.0177006474	-0.0177221160	2.4835061003	0.0045297258	0.0021538441	0.3018393315
	145.4	3.0	2^{-13}	3.2256469196	3.1646546774	0.9674822148	-1.0271283545	0.0176931937	-0.0177146006	2.4839169565	0.0045306268	0.0021560077	0.3018705247
	273.0	5.7	2^{-13}										
$\text{HDG}_{H^{\text{div}}}$ $(L,k)=(0,\ 5)$	19.1	0.2	$^1/_{1250}$	3.2250894217	3.1631970522	0.9554878894	-1.0167407708	0.0174498580	-0.0174774565	2.4821304744	0.0046180909	0.0022830661	0.3026634383
$(L,k)=(1,\ 5)$	50.3	0.5	$^1/_{2500}$	3.2255461452	3.1623559873	0.9675410784	-1.0266478889	0.0176809137	-0.0177017106	2.4841689387	0.0044381232	0.0020651124	0.3019323671
$(L,k)=(2,\ 5)$	181.5	2.0	$^1/_{5000}$	3.2256839724	3.1623930523	0.9675089999	-1.0272149646	0.0176928223	-0.0177140904	2.4841763850	0.0045347470	0.0021601730	0.3019323671
$(L,k)=(3,\ 5)$	746.2	8.2	$^1/_{10000}$	3.2256935844	3.1624101505	0.9674881509	-1.0271362492	0.0176921047	-0.0177135376	2.4841005356	0.0045308638	0.0021563879	0.3018412315
$\text{HDG}_{H^{\text{div}}}$ $(L,k)=(0,\ 5)$	19.1	0.2	$^1/_{1250}$	3.2250894217	3.1631970522	0.9554878894	-1.0167407708	0.0174498580	-0.0174774565	2.4821304744	0.0046180909	0.0022830661	0.3026634383
$(L,k)=(0,\ 6)$	25.7	0.3	$^1/_{1644}$	3.2268543979	3.1635587023	0.9671329198	-1.0273404878	0.0176726465	-0.0176938112	2.4843767829	0.0046129318	0.0022434678	0.3022051913
$(L,k)=(0,\ 7)$	33.3	0.4	$^1/_{2071}$	3.2263007223	3.1629779600	0.9679161173	-1.0276532855	0.0176944680	-0.0177167087	2.4840834627	0.0045702262	0.0021963030	0.3018949949
$(L,k)=(0,\ 8)$	41.9	0.5	$^1/_{2530}$	3.2262827883	3.1628681950	0.9686648558	-1.0283599930	0.0177079714	-0.0177294765	2.4845121998	0.0045563934	0.0021801191	0.3019095781
$(L,k)=(0,\ 9)$	51.4	0.6	$^1/_{3019}$	3.2259541386	3.1626013428	0.9681203559	-1.0277726888	0.0177017718	-0.0177233126	2.4842577357	0.0045419773	0.0021662651	0.3019004634
$(L,k)=(0,10)$	62.0	0.7	$^1/_{3536}$	3.2258239459	3.1625056339	0.9678453674	-1.0274854052	0.0176975287	-0.0177189699	2.4843132451	0.0045359375	0.0021607464	0.3017064517

Table 3: Results for Rot-2D2. The number of degrees of freedom (dof) is given in $K = 10^3$. Dofs in brackets are the number of unconstrained dofs after static condensation. The number of non-zero entries (nze) of the condensed (if applicable) system matrix are given in $M = 10^6$. The computational run-time is given in seconds s. Time step size τ for TR is based on a subdivision of each period $t_P \approx 0.33$.

Coefficient	$C_{D,\max}$	$C_{D,\min}$	$C_{L,\max}$	$C_{L,\min}$	$C_{T,\max}$
Value	$[3.2256, 3.22571]$	$[3.1622, 3.1647]$	$[0.9674, 0.9686]$	$[-1.0282, -1.0271]$	$[0.01769, 0.01783]$
Coefficient	$C_{T,\min}$	$\Delta p(t^*)$	$\omega^*_{\max}$	$\omega^*_{\min}$	St
Value	$[-0.01786, -0.01771]$	$[2.4837, 2.4844]$	$[0.004530, 0.004539]$	$[0.002144, 0.002161]$	$[0.30171, 0.30188]$

Table 4: Reference value ranges for the quantities of interest for the problem Rot-2d2.

Discretisation				Results									
Method	#dof [K]	#nze [M]	τ	$C_{D,\max}$	$t_{D,\max}$	$C_{L,\max}$	$t_{L,\max}$	$C_{T,\max}$	$t_{T,\max}$	$\Delta p(8)$	$\omega^*_{\max}$	$t_{\omega,\max}$	Comp.T. [s]
$\mathrm{TH}_{\mathrm{gd,ho}}$[1]	15.6(8.0)	0.3	$1/2500$	2.9509895176	3.93640	0.4680735702	5.69320	0.0097303511	5.84240	-0.1118627466	0.0033807682	5.95560	875
$k=5$	51.1(26.8)	1.1	$1/5000$	2.9509040487	3.93620	0.4657404192	5.69380	0.0096899195	5.84320	-0.1118754106	0.0033487422	5.95620	5597
	186.5(99.7)	3.9	$1/10000$	2.9508932851	3.93620	0.4656656345	5.69390	0.0096887696	5.84300	-0.1118750991	0.0033484815	5.95630	41244
	693.6(374.2)	14.5	$1/20000$	2.9508912256	3.93620	0.4656663341	5.69390	0.0096887894	5.84300	-0.1118751459	0.0033484327	5.95630	310842
$\mathrm{EO}_{\mathrm{LPS}}$[7,3]	15.6	0.7	$1/50$	2.95148758	3.93715	0.54559013	5.73098	0.01121033	5.88498	-0.11005404	0.00349536	6.00203	282
	46.5	2.1	$1/126$	2.95083751	3.93629	0.47297707	5.69516	0.00982570	5.84448	-0.11193809	0.00336110	5.95803	2090
	154.5	7.2	$1/330$	2.95093223	3.93620	0.46609980	5.69394	0.00969654	5.84300	-0.11187846	0.00334929	5.95639	17418
	554.7	26.2	$1/831$	2.95088595	3.93619	0.46567339	5.69391	0.00968898	5.84295	-0.11187501	0.00334838	5.95633	165886
extrapolated				change of sign		0.46564520		0.00968851		-0.11187479	0.00334830		
(order)				-	-	(4.01)		(4.09)		(4.07)	(3.70)		
TH[4]	3.8	0.1	$1/40$	3.0066353678	3.97770	0.0050600950	0.98152	0.0010600233	4.06587	-0.1208947707	0.0042513467	5.86591	17
	9.8	0.3	$1/80$	2.9647408765	3.94673	0.1848436023	6.86969	0.0046721653	6.37981	-0.1022916281	0.0028664803	7.68158	50
	29.7	0.8	$1/160$	2.9520976467	3.94280	0.4552154174	5.75891	0.0097338773	5.90943	-0.1059880107	0.0028602560	7.68262	225
	99.7	2.9	$1/320$	2.9518102419	3.94043	0.4637777077	5.70197	0.0096836137	5.84945	-0.1117757539	0.0035836382	5.95961	1380
	362.4	10.6	$1/640$	2.9513456900	3.93838	0.4659134118	5.69664	0.0097683463	5.84458	-0.1114409580	0.0033407595	5.95645	11890
	1379.1	40.6	$1/1280$	2.9509141480	3.93722	0.4658976721	5.69509	0.0097613315	5.84332	-0.1116494707	0.0033616140	5.95589	117083
$\mathrm{HDG}_{\mathrm{H}^{\mathrm{div}}}$[5] $(L,k)=(0,\ 5)$	19.1	0.2	$1/1250$	2.9511386439	3.93680	0.4756634788	5.69280	0.0098269810	5.84240	-0.1117569458	0.0034432916	5.95520	204.9
$(L,k)=(1,\ 5)$	50.3	0.5	$1/2500$	2.9509424810	3.93640	0.4660733580	5.69400	0.0096957993	5.84360	-0.1118916796	0.0032765928	5.95680	765.7
$(L,k)=(2,\ 5)$	181.5	2.0	$1/5000$	2.9508803634	3.93620	0.4656849766	5.69380	0.0096895862	5.84320	-0.1118754521	0.0033509689	5.95620	5083.0
[8] $(L,k)=(3,\ 5)$	746.0	8.1	$1/10000$	2.9508882652	3.93620	0.4656563266	5.69390	0.0096885678	5.84300	-0.1118742936	0.0033485024	5.95630	21829.4
$\mathrm{HDG}_{\mathrm{H}^{\mathrm{div}}}$[5] $(L,k)=(0,\ 5)$	19.1	0.2	$1/1250$	2.9511386439	3.93680	0.4756634788	5.69280	0.0098269810	5.84240	-0.1117569458	0.0034432916	5.95520	204.9
$(L,k)=(0,\ 6)$	25.7	0.3	$1/1644$	2.9509517602	3.93613	0.4697975801	5.69161	0.0097615210	5.84124	-0.1117081557	0.0033587325	5.95377	356.4
$(L,k)=(0,\ 7)$	33.3	0.4	$1/2071$	2.9508984404	3.93626	0.4668674486	5.69339	0.0097111637	5.84307	-0.1118381480	0.0033566456	5.95558	519.4
$(L,k)=(0,\ 8)$	41.9	0.5	$1/2530$	2.9508992999	3.93636	0.4661219409	5.69368	0.0096984067	5.84308	-0.1118513538	0.0033486328	5.95613	838.7
$(L,k)=(0,\ 9)$	51.4	0.6	$1/3019$	2.9508944978	3.93607	0.4658316869	5.69361	0.0096922436	5.84299	-0.1118662036	0.0033490102	5.95628	1425.9
$(L,k)=(0,10)$	62.0	0.7	$1/3536$	2.9508942137	3.93609	0.4657339952	5.69372	0.0096901571	5.84304	-0.1118688505	0.0033485887	5.95617	2305.8

Table 5: Results for Rot-2D3. The number of degrees of freedom (dof) is given in $K = 10^3$. Dofs in brackets are the number of unconstrained dofs after static condensation. The number of non-zero entries (nze) of the condensed (if applicable) system matrix are given in $M = 10^6$. The computational run-time is given in seconds s.

Coefficient	$C_{D,\max}$	$t_{D,\max}$	$C_{L,\max}$	$t_{L,\max}$	$C_{T,\max}$
Value	$[2.95085, 2.95092]$	$[3.9360, 3.9372]$	$[0.4656, 0.4659]$	$[5.6937, 5.6951]$	$[0.0096885, 0.009762]$
Coefficient	$t_{T,\max}$	$\Delta p(8)$	$\omega^*_{\max}$	$t_{\omega,\max}$	
Value	$[5.842, 5.845]$	$[-0.11189, -0.11186]$	$[0.003347, 0.003362]$	$[5.9558, 5.9567]$	

Table 6: Reference value ranges for the quantities of interest for the problem Rot-2d3.

5. Code availability

Full data sets for the results presented in Section 4 as well as code used to compute these benchmarks can be found at

DOI: `10.5281/zenodo.3253455`.

Here we provide both the scripts used to compute the respective benchmarks and instructions and links to the relevant external software used.

Acknowledgements

---

[7]Computed using Intel Xeon E5-2640 v4, 2.40GHz using 8 cores.

[8]$(L = 3, k = 5)$: Computed using 2x Intel Xeon E5-2650 v4, 2.20 GHz using 24 cores.

[9]Computed using 2x Intel Xeon E5-2650 v4, 2.20GHz using 24 cores.

[10]Computed using 2x Intel Xeon E5-2640 v4, 2.40GHz using 20 cores.

[11]$1 \leq k \leq 3$: Computed using Intel Xeon E5-2650 v4, 2.20 GHz using 12 cores,
$k = 4$: Computed using 2x Intel Xeon E5-2650 v4, 2.20 GHz using 24 cores,
$k = 5$: Computed using 4x Intel Xeon E5-4620 v3, 2.00 GHz using 18 cores,
$k = 6$: Computed using 4x Intel Xeon E5-4620 v3, 2.00 GHz using 24 cores.

Discretisation			Results								
Method	#dof [K]	#nze [M]	$C_{\mathbf{F}_1}$	$C_{\mathbf{F}_2}$	$C_{\mathbf{F}_3}$	$\|C_{\mathbf{T}}\|_2$	Δp	ω_1^*	ω_2^*	ω_3^*	Comp.T. [s]
$\mathrm{TH}_{\mathrm{gd,ho}}$[9]	212.9(167.8)	50.9	6.02155266	0.02502608	0.12755555	2.09e-12	0.13635030	-2.97751e-06	0.03465252	-0.00689595	284
$k=4$	652.1(536.3)	158.3	6.02151890	0.02496250	0.12753770	6.47e-15	0.13681163	-6.22459e-07	0.03464332	-0.00686038	1168
	1120.0(923.5)	272.1	6.02151253	0.02495675	0.12752276	1.96e-13	0.13662752	-1.03142e-06	0.03464001	-0.00686132	3671
	1831.0(1468.0)	440.5	6.02150624	0.02495834	0.12752303	3.38e-13	0.13661989	-1.01483e-06	0.03463992	-0.00686150	8036
$\mathrm{EO}_{\mathrm{LPS}}$[3,10]	189	41	6.0194099	0.027620778	0.1413845	5.59e-13	0.1457953	-1.853299e-06	0.03793676	-0.00750854	79
	1426	336	6.0209826	0.025185144	0.1286906	2.52e-09	0.1385932	-1.017284e-06	0.03489545	-0.00691099	685
	11061	2716	6.0214724	0.024973759	0.1276006	3.11e-11	0.1369655	-1.017598e-06	0.03465537	-0.00686465	6728
	87100	21836	6.0215013	0.024959810	0.1275276	6.34e-14	0.1366775	-1.018982e-06	0.03464069	-0.00686186	57331
	extrapolated		6.0215031	0.0249588	0.1275223	–	0.1366156	–	0.03463972	0.006861677	
	(order)		(4.08)	(3.97)	(3.92)	–	(2.50)	–	(4.03)	(4.05)	
TH[4]	37.2	3.4	5.97793	0.0276684	0.133363	1.77855e-12	0.134442	0.000106824	0.0390861	-0.00873872	26
	64.4	5.8	6.00204	0.0250739	0.126979	3.21458e-10	0.137206	1.47338e-05	0.035706	-0.00759	52
	173.2	15.9	6.01025	0.026336	0.128098	1.65002e-10	0.135844	-1.99197e-05	0.0353658	-0.00765735	195
	285.1	25.9	6.01655	0.0253422	0.128058	7.69239e-10	0.13658	1.35604e-05	0.0353014	-0.00720906	435
	990.2	93.1	6.01912	0.0249711	0.127594	1.85239e-10	0.136474	1.2935e-06	0.0348498	-0.00691854	3587
$\mathrm{HDG}_{H^{div}}$[11] $k=1$	133.4 (116.0)	7.7	5.26348	0.0557553	0.115768	7.77e-13	0.137416	0.0092711	0.0341869	-0.0268611	36
$k=2$	320.7 (232.0)	30.7	6.23837	0.0411259	0.143691	4.21e-12	0.139044	0.000612444	0.0384272	-0.0125745	118
$k=3$	624.3 (386.7)	85.4	6.00429	0.0283705	0.125754	4.61e-13	0.132603	-0.000164957	0.0345766	-0.00779181	545
$k=4$	1071.0 (580.1)	192.1	6.01982	0.0251	0.126563	1.39e-12	0.13611	5.29841e-06	0.0343653	-0.00686743	752
$k=5$	1688.0 (812.1)	376.5	6.02186	0.0248633	0.127461	8.03e-12	0.13656	-1.42517e-05	0.0346083	-0.00684754	2319
$k=6$	2501.9 (1082.8)	669.3	6.02153	0.0249391	0.127536	6.12e-12	0.136585	-1.25483e-06	0.0346432	-0.00685716	4122

Table 7: Results for Rot-3d1. The number of degrees of freedom (dof) is given in $K = 10^3$. Dofs in brackets are the number of unconstrained dofs after static condensation. The number of non-zero entries (nze) of the condensed (if applicable) system matrix are given in $M = 10^6$. The computational run-time is given in seconds s.

Coefficient	$C_{\mathbf{F}_1}$	$C_{\mathbf{F}_2}$	$C_{\mathbf{F}_3}$	Δp
Value	$[6.019, 6.0216]$	$[0.02493, 0.02498]$	$[0.12752, 0.1276]$	$[0.1364, 0.1367]$
Coefficient	ω_1^*	ω_2^*	ω_3^*	
Value	$[-1.3 \cdot 10^{-6}, 1.3 \cdot 10^{-6}]$	$[0.034639, 0.03485]$	$[-0.00692, -0.00685]$	

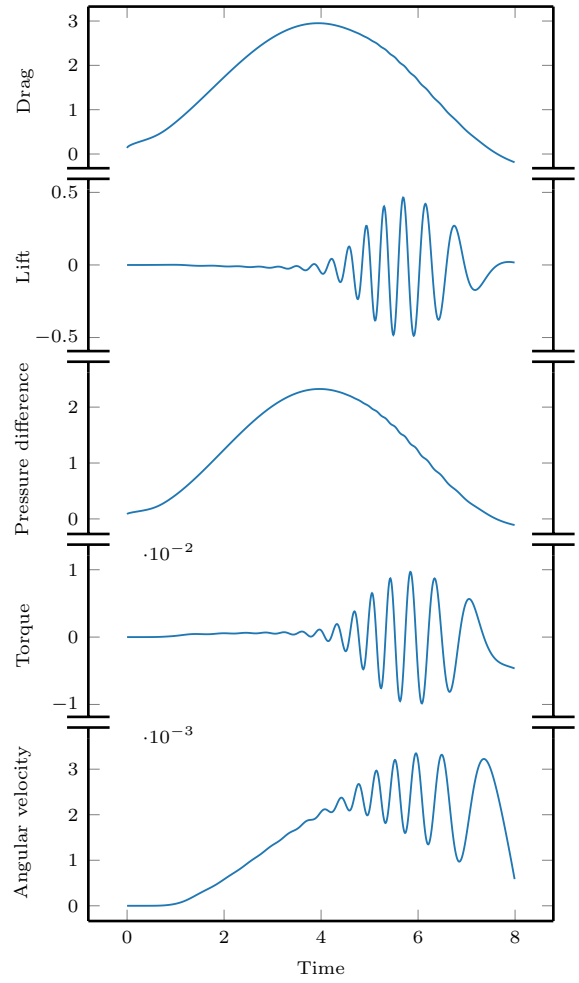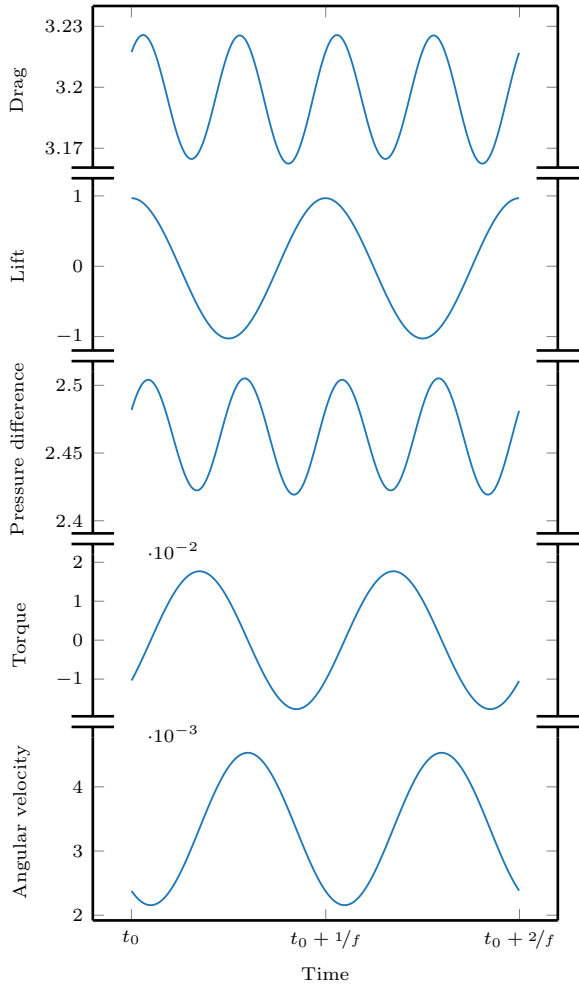Table 8: Reference Values for the problem Rot-3d1.

Figure 3: The quantities of interest for the problem Rot-2d2 over two periods (left) and the quantities of interest for the problem Rot-2d3 over the entire time-interval (right).

15

References

[1] M. Schäfer, S. Turek, Benchmark computations of laminar flow around a cylinder. (With support by F. Durst, E. Krause and R. Rannacher), in: E. Hirschel (Ed.), Flow Simulation with High-Performance Computers II. DFG priority research program results 1993-1995, no. 52 in Notes Numer. Fluid Mech., Vieweg, Wiesbaden, 1996, pp. 547–566. `doi:10.1007/978-3-322-89849-4_39`.

[2] S. Turek, J. Hron, Proposal for numerical benchmarking of fluid-structure interaction between an elastic object and laminar incompressible flow, in: Lecture Notes in Computational Science and Engineering, Springer Berlin Heidelberg, 2006, pp. 371–385. `doi:10.1007/3-540-34596-5_15`.

[3] S. Hysing, S. Turek, D. Kuzmin, N. Parolini, E. Burman, S. Ganesan, L. Tobiska, Quantitative benchmark computations of two-dimensional bubble dynamics, Internat. J. Numer. Methods Fluids 60 (11) (2009) 1259–1288. `doi:10.1002/fld.1934`.

[4] H. M. Badr, S. C. R. Dennis, P. J. S. Young, Steady and unsteady flow past a rotating circular cylinder at low Reynolds numbers, Comput. & Fluids 17 (4) (1989) 579–609. `doi:10.1016/0045-7930(89)90030-3`.

[5] D. Fabre, J. Tchoufag, V. Citro, F. Giannetti, P. Luchini, The flow past a freely rotating sphere, Theor. Comput. Fluid Dyn. 31 (5-6) (2016) 475–482. `doi:10.1007/s00162-016-0405-x`.

[6] K. D. Housiadas, R. I. Tanner, The angular velocity of a freely rotating sphere in a weakly viscoelastic matrix fluid, Phys. Fluids 23 (5) (2011) 051702. `doi:10.1063/1.3583376`.

[7] H. Juárez, R. Scott, R. Metcalfe, B. Bagheri, Direct simulation of freely rotating cylinders in viscous flows by high-order finite element methods, Comput. & Fluids 29 (5) (2000) 547–582. `doi:10.1016/s0045-7930(99)00016-x`.

[8] S. Kang, H. Choi, S. Lee, Laminar flow past a rotating circular cylinder, Phys. Fluids 11 (11) (1999) 3312–3321. `doi:10.1063/1.870190`.

[9] S. Mittal, B. Kumar, Flow past a rotating cylinder, J. Fluid Mech. 476 (feb 2003). `doi:10.1017/s0022112002002938`.

[10] K. Shaafi, S. N. Naik, S. Vengadesan, Effect of rotating cylinder on the wake-wall interactions, Ocean Eng. 139 (2017) 275–286. `doi:10.1016/j.oceaneng.2017.04.044`.

[11] D. Stojković, M. Breuer, F. Durst, Effect of high rotation rates on the laminar flow around a circular cylinder, Phys. Fluids 14 (9) (2002) 3160–3178. `doi:10.1063/1.1492811`.

[12] Y. Xia, J. Lin, X. Ku, T. Chan, Shear-induced autorotation of freely rotatable cylinder in a channel flow at moderate Reynolds number, Phys. Fluids 30 (4) (2018) 043303. `doi:10.1063/1.5021877`.

[13] J. Heywood, R. Rannacher, S. Turek, Artificial boundaries and flux and pressure conditions for the incompressible Navier-Stokes equations, Int. J. Numer. Math. Fluids. 22 (1992) 325–352. `doi:10.1002/(SICI)1097-0363(19960315)22:5<325::AID-FLD307>3.0.CO;2-Y`.

[14] C. Taylor, P. Hood, A numerical solution of the Navier-Stokes equations using the finite element technique, Comput. & Fluids 1 (1) (1973) 73 – 100. `doi:10.1016/0045-7930(73)90027-3`.

[15] D. Boffi, Stabiliy of higher order triangular Hood-Taylor methods for the stationary Stokes equations, Math. Models Methods Appl. Sci. 04 (02) (1994) 223–235. `doi:10.1142/s0218202594000133`.

[16] L. Franca, T. Hughes, Two classes of mixed finite element methods, Comput. Methods Appl. Mech. Engrg. 69 (1988) 89–129. `doi:10.1016/0045-7825(88)90168-5`.

[17] J. Schöberl, NETGEN an advancing front 2d/3d-mesh generator based on abstract rules, Comput. Vis. Sci. 1 (1) (1997) 41–52. `doi:10.1007/s007910050004`.

[18] J. Schöberl, C++ 11 Implementation of Finite Elements in NGSolve, Preprint, ASC Report 30/2014, Institute for Analysis and Scientific Computing, Vienna University of Technology (2014).
URL `http://www.asc.tuwien.ac.at/~schoeberl/wiki/publications/ngs-cpp11.pdf`

[19] Intel math kernel library, `https://software.intel.com/en-us/mkl` (2019).

[20] I. Babuška, A. Miller, The post-processing approach in the finite element method. I. calculations of displacements, stresses and other higher derivatives of the displacements, Internat. J. Numer. Methods Engrg. 20 (1984) 1085–1109. `doi:10.1002/nme.1620200610`.

[21] T. Richter, Fluid-structure Interactions. Models, Analysis and Finite Elements, Vol. 118 of Lecture notes in computational science and engineering, Springer, 2017. `doi:10.1007/978-3-319-63970-3`.

[22] U. Ascher, S. Ruuth, B. Wetton, Implicit-explicit methods for time-dependent partial differential equations, SIAM J. Numer. Anal. 32 (3) (1995) 797–823. `doi:10.1137/0732037`.

[23] V. John, Reference values for drag and lift of a two-dimensional time-dependent flow around a cylinder, Internat. J. Numer. Methods Fluids 44 (7) (2004) 777–788. `doi:10.1002/fld.679`.

[24] C. Lehrenfeld, J. Schöberl, High order exactly divergence-free hybrid discontinuous galerkin methods for unsteady incompressible flows, Comput. Methods Appl. Mech. Engrg. 307 (2016) 339–361. `doi:10.1016/j.cma.2016.04.025`.

[25] V. John, A. Linke, C. Merdon, M. Neilan, L. G. Rebholz, On the divergence constraint in mixed finite element methods for incompressible flows, SIAM Rev. 59 (3) (2017) 492–544.

[26] P. W. Schroeder, A. Linke, C. Lehrenfeld, G. Lube, Towards computable flows and robust estimates for inf-sup stable FEM applied to the time-dependent incompressible Navier-Stokes equations, SeMA Journal (2018) 1–25`doi:10.1007/s40324-018-0157-1`.
URL `https://doi.org/10.1007/s40324-018-0157-1`

[27] P. W. Schroeder, V. John, P. L. Lederer, C. Lehrenfeld, G. Lube, J. Schöberl, On reference solutions and the sensitivity of the 2d kelvin–helmholtz instability problem, Comput. Mathem. Appl. 77 (4) (2019) 1010–1028.

[28] C. Lehrenfeld, Hybrid Discontinuous Galerkin Methods for Incompressible Flow Problems, Master's thesis, RWTH Aachen (May 2010).

[29] R. Becker, M. Braack, A finite element pressure gradient stabilization for the Stokes equations based on local projections, Calcolo 38 (4) (2001) 173–199. `doi:10.1007/s10092-001-8180-4`.

[30] M. Braack, T. Richter, Solutions of 3D Navier-Stokes benchmark problems with adaptive finite elements, Comput. & Fluids 35 (4) (2006) 372–392. `doi:10.1016/j.compfluid.2005.02.001`.

[31] L. Failer, T. Richter, A parallel Newton multigrid framework for monolithic fluid-structure interactions, arXiv Preprint (2019).
URL `https://arxiv.org/abs/1904.02401`

[32] M. Kimmritz, T. Richter, Parallel multigrid method for finite element simulations of complex flow problems on locally refined meshes, Numer. Linear Algebra Appl. 18 (4) (2010) 615–636. `doi:10.1002/nla.744`.

[33] A. Logg, K.-A. Mardal, G. N. Wells, et al., Automated Solution of Differential Equations by the Finite Element Method, Springer, 2012. `doi:10.1007/978-3-642-23099-8`.

[34] M. S. Alnæs, J. Blechta, J. Hake, A. Johansson, B. Kehlet, A. Logg, C. Richardson, J. Ring, M. E. Rognes, G. N. Wells, The fenics project version 1.5, Archive of Numerical Software 3 (100) (2015). `doi:10.11588/ans.2015.100.20553`.

[35] FEniCS project, `https://fenicsproject.org/download/`, accessed: 2018-04-01.

[36] P. R. Amestoy, A. Guermouche, J.-Y. L'Excellent, S. Pralet, Hybrid scheduling for the parallel solution of linear systems, Parallel Comput. 32 (2) (2006) 136–156. `doi:10.1016/j.parco.2005.07.004`.

[37] J. Heiland, dolfin_navier_scipy: a python SciPy FEniCS interface, `https://doi.org/10.5281/zenodo.3238622` (Jun 2019). `doi:10.5281/zenodo.3238622`.

[38] E. Jones, T. Oliphant, P. Peterson, et al., SciPy: Open source scientific tools for Python, online; www.scipy.org; accessed July 5th, 2019 (2001).
URL `http://www.scipy.org/`

[39] P. Benner, J. Heiland, Time-dependent Dirichlet conditions in finite element discretizations, ScienceOpen Res. (2015) 1–18`doi:10.14293/S2199-1006.1.SOR-MATH.AV2JW3.v1`.

[40] R. P. Brent, An algorithm with guaranteed convergence for finding a zero of a function, Comput. J. 14 (1971) 422–425. `doi:10.1093/comjnl/14.4.422`.

[41] J. Kim, P. Moin, Application of a fractional-step method to incompressible Navier-Stokes equations, J. Comput. Phys. 59 (1985) 308–323. `doi:10.1016/0021-9991(85)90148-2`.

[42] M. Marion, R. Temam, Navier-Stokes equations: Theory and approximation, in: P. G. Ciarlet, J. L. Lions (Eds.), Numerical Methods for Fluids., Handbook of Numerical Analysis, Elsevier, 1998, pp. 503–688. `doi:10.1016/s1570-8659(98)80010-0`.