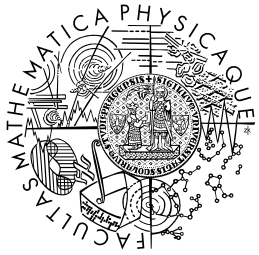

FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

**MASTER THESIS**

Bc. Ondřej Outrata

# Numerical Methods for Vortex Dynamics

The Mathematical Institute of Charles University

Supervisor of the master thesis: RNDr. Jaroslav Hron Ph.D

Study programme: Physics

Study branch: Mathematical and Computational Modelling in Physics

Prague 2020

Dedication.

The author thanks to

- RNDr. Jaroslav Hron, Ph.D. for his guidance.
- Dr. Marco La Mantia and Mgr. Patrik Švančara for their consultation.
- Family and friends who supported him throughout the studies.

Numerical Methods for Vortex Dynamics

Author: Bc. Ondřej Outrata

Department: The Mathematical Institute of Charles University

Supervisor: RNDr. Jaroslav Hron Ph.D, The Mathematical Institute of Charles University

Abstract: Two aspects of solving the incompressible Navier-Stokes equations are described in the thesis. The preconditioning of the algebraic systems arising from the Finite Element Method discretization of the Navier-Stokes equations is complex due to the saddle point structure of the resulting algebraic problems. The Pressure Convection Diffusion Reaction and the Least Squares Commutator preconditioners constitute two possible choices studied in the thesis. Solving the flow problems in time-dependent domains requires special numerical methods, such as the Fictitious Boundary method and the Arbitrary Lagrangian Eulerian formulation of Navier-Stokes equations which are used in the thesis. The problems examined in the thesis are simulations of experiments conducted in liquid Helium at low temperatures. These simulations can be used to establish a relationship between vorticity and new quantity pseudovorticity in an experiment-like setting.

Keywords: Incompressible Navier-Stokes, Finite Element Method, Projection Method, Preconditioning, ALE, Fictitious Boundary

# Contents

# Introduction

The main purpose of the thesis is to describe and test two aspects of simulating the incompressible Navier-Stokes flows with significant vortex dynamics. Firstly, the incompressible Navier-Stokes equations are a saddle point problem and the resulting discrete algebraic problems arising from numerical discretization require specific solution approaches to be solved efficiently. Secondly, the incompressible flow problems become difficult to solve in time-dependent domains as special formulation of the equations or treatment of boundary conditions is needed. This thesis describes these problems and their solutions and applies them to numerical simulations of experiments conducted by the Superfluidity research group based at Charles University.

The experiments are conducted in liquid Helium at low temperatures (1.5K-2K). Some simulations of liquid Helium below the lambda temperature (the temperature at which Helium transitions to superfluid Helium II state, approximately 2.17K at $1atm.$) have been carried out using the two fluid model [Landau, 1941] in simpler geometries [Bruce et al., 2017]. One goal of the thesis is to investigate whether the experiments can be simulated by the simpler incompressible Navier-Stokes equations under the assumption that the temperature is constant. As the velocity and vorticity data cannot be directly measured in the experiments, it is useful to employ the numerical simulations where the data are available. The simulations are used to establish a relationship between a new quantity called pseudovorticity and vorticity. Pseudovorticity is a Lagrangian quantity obtained by tracking the velocity of (ideally massless) particles in a fluid. Pseudovorticity can be used to identify various properties of vortices such as position and velocity.

The first chapter of the thesis includes a description of the Navier-Stokes equations which express general conservation laws, specifically, the conservation of mass and the conservation of linear momentum. The weak formulation of the equations is stated in the first chapter as the starting point of the thesis which is followed by a short introduction to a numerical solution of Navier-Stokes equation and spatial (using the Finite Element Method) and time discretization of the equations.

Quick, efficient, and scalable solvers are needed in order to solve flow problems in detail for large systems. This thesis describes two preconditioners and a projection method as options to replace the direct solver MUMPS with scalable iterative solvers. These iterative solvers scale much better than MUMPS [Amestoy et al., 2001] which makes them suitable for computation of large problems on computational clusters. The Pressure Convection Diffusion Reaction (PCDR) preconditioning [Elman et al., 2014] implemented in FENaPack [Blechta and Řehoř, 2019] and the Least Squares Commutator preconditioner [Elman et al., 2006] implemented in PETSc [Balay et al., 2015] are explained in the second chapter. These preconditioners are compared with the direct solver MUMPS and the projection method Incremental Pressure Correction Scheme (IPCS) [Guermond et al., 2006]. A numerical simulation of the vortex rings experiment [Švančara et al., 2020] is carried out at the end of the second chapter using previously mentioned methods. The simulation is used as a performance benchmark for a comparison of the methods. The ability of pseudovorticity to capture the position and velocity of

the vortex ring is demonstrated at the end of the chapter.

The third chapter further extends the results of [Outrata, 2018] and outlines two methods of dealing with the time-dependent domains. These methods are: the Fictitious Boundary method (FIB) [Turek et al., 2003] which works in the physical domain of the problem and the Arbitrary Lagrangian Eulerian (ALE) [Lozovskiy et al., 2018] formulation which transforms the problem into a static domain where it is solved. These methods are then applied to two problems: a benchmark test [Schott, 2017] including a computation of the flow around an oscillating cylinder and a simulation of the experiment documented in [Duda et al., 2015]. Special outflow conditions and stabilization methods needed to simulate the experiment are introduced at the end of third chapter. The numerical comparison of the FIB and the ALE is the subject of the fourth chapter.

# 1. Navier-Stokes equations

This chapter will go over some basic equations and relations regarding incompressible Navier-Stokes equations.

## 1.1 Notation

Definitions and notations needed in the following text:

- Let $\Omega$,$\Omega_0$ be open bounded domains in $\mathbb{R}^d, d = 2, 3$ with piecewise smooth boundaries and unit outer normals $\mathbf{n}$ and $\mathbf{n}_0$.
- Vector spaces for vector functions are denoted by blackboard letters, such as $\mathbb{V}$ which denotes space

$$\mathbb{V}(\Omega) = \{\mathbf{f} : f_i \text{ is in } V \text{ for } i = 1, ..., d\}$$

- Sobolev spaces - spaces defined by

$$H^1(\Omega) = \{f \in L^2(\Omega) : \int_\Omega |\nabla f|^2 + |f|^2 \, \mathrm{d}V < \infty\}$$

$$H_0^1(\Omega) = \{f(\Omega) \in H^1 : \mathrm{Tr}(f) = 0 \text{ on } \partial\Omega\},$$

  where $\nabla f$ denotes vector of partial derivatives of $f$ in a weak sense and Tr() is the trace operator (see [Evans and Society, 1998]).

- Bochner spaces - spaces involving time, especially $L^2[(0;T);X]$, which is defined by norm

$$\|f\|_{L^2[(0,T);X]} = (\int_0^T \|f(t)\|_X^2 \mathrm{d}t)^{\frac{1}{2}} < \infty. \tag{1.1}$$

- Inner products - Let $u, v \in L^2(\Omega)$ and $\mathbf{u}, \mathbf{v} \in \mathbb{L}^2(\Omega)$ (and $\mathbf{u}, \mathbf{v} \in \mathbb{H}^1(\Omega)$ for product of gradients), then

$$(u, v) = \int_\Omega uv \, \mathrm{d}V$$

$$(\mathbf{u}, \mathbf{v}) = \int_\Omega \mathbf{u} \cdot \mathbf{v} \, \mathrm{d}V.$$

$$(\nabla\mathbf{u}, \nabla\mathbf{v}) = \int_\Omega \nabla\mathbf{u} : \nabla\mathbf{v} \, \mathrm{d}V$$

## 1.2 Navier-Stokes Equations

Let $\Omega$ be a domain filled with fluid. The incompressible Navier-Stokes equations can be written in the strong formulation as follows

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} &= \mathbf{f} - \frac{1}{\rho} \nabla p + \nu \Delta \mathbf{u}, \\ \nabla \cdot \mathbf{u} &= 0 \quad \text{in} \quad \Omega, \end{aligned} \tag{1.2}$$

where $\mathbf{u}$ denotes the velocity of the fluid, $p$ is the pressure, $\mathbf{f}$ are the external volume forces acting on the fluid, $\rho$ is the density of the fluid and $\nu$ denotes kinematic viscosity of the fluid. We only consider case where $\rho$ is constant so the following pressure notation is used in the thesis $\tilde{p} = p/\rho$ but we omit the tilde. Let us define $\mathbb{T} = -p\mathbb{I} + 2\nu\mathbb{D}$, where $\mathbb{D}$ is the symmetric part of the velocity gradient. For now, we consider two possible boundary conditions, the Dirichlet and traction free boundary conditions

$$\begin{aligned} \mathbf{u} &= \mathbf{u}_{\partial\Omega} \quad && \text{on} \quad \partial\Omega_d \\ \mathbb{T}\mathbf{n} &= \mathbf{0} && \text{on} \quad \partial\Omega_f \end{aligned} \tag{1.3}$$

under the condition that $\partial\Omega_d \cup \partial\Omega_f = \partial\Omega$ and $\partial\Omega_d \cap \partial\Omega_f = \emptyset$. The initial condition is $\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x})$.

These equations can be also rewritten in the dimensionless form

$$\begin{aligned} \frac{\partial \mathbf{u}'}{\partial t'} + \mathbf{u}' \cdot \nabla' \mathbf{u}' &= \frac{1}{Fr^2} \mathbf{f}' - \nabla' p' + \frac{1}{Re} \Delta' \mathbf{u}', \\ \nabla' \cdot \mathbf{u}' &= 0 \end{aligned} \tag{1.4}$$

using the Reynolds number $Re = \frac{UL}{\nu}$, the Froude number $Fr = \frac{U}{\sqrt{f_0 L}}$ and the transformations $\mathbf{x}' = \frac{\mathbf{x}}{L}$, $\mathbf{u}' = \frac{\mathbf{u}}{U}, t' = \frac{tU}{L}, p' = \frac{p}{\rho U^2}$ and $\mathbf{f}' = \frac{\mathbf{f}}{f_0}$, where $U$ is the characteristic speed of the system, $L$ is the characteristic length of the system and $f_0$ is the characteristic force of the system. From the previous form of equations it can be seen that if the Reynolds number is low, then the viscous forces (corresponding to the Laplace operator) dominate the flow while the convection has smaller effects. Flows at low Reynolds numbers are often called laminar. These flows are easier to simulate because of the stability ensured by the viscous forces. If the Reynolds number is high, then the nonlinear term $\mathbf{u} \cdot \nabla \mathbf{u}$ corresponding to convection dominates the equations and the flows are named turbulent. This situation is typically much harder to simulate because the lack of viscous forces leads very small scale features, possible instabilities and chaotic behavior. Up to this day, the problem of existence and uniqueness of the classical solution to the Navier-Stokes equations remains unsolved as one of the 7 Millenium problems announced by the Clay Mathematics institute. The weak formulation of the Navier-Stokes equations is more suitable for the numerical purposes (from now on all spatial derivatives are considered as weak derivatives).

The definition of the weak solution (for simplicity with homogeneous Dirichlet boundary condition) can be specified as follows. Let $\mathbf{f} \in L^2[(0,T); \mathbb{L}^2(\Omega)]$ and initial condition $\mathbf{u}_0 \in \mathbb{H}_0^1(\Omega)$ be given. We say that pair $(\mathbf{u}, p)$ is a weak solution of the Navier-Stokes problem with homogeneous Dirichlet boundary conditions,

if $\mathbf{u} \in L^2[(0,T);\mathbb{H}^1_0(\Omega)] \cap L^\infty[(0,T);\mathbb{L}^2(\Omega)]$ , $p \in L^2[(0,T);L^2(\Omega)]$ and

$$\begin{aligned}(\frac{\partial \mathbf{u}}{\partial t}, \mathbf{v}) + (\mathbf{u}\cdot\nabla\mathbf{u}, \mathbf{v}) + \nu(2\mathbb{D}, \nabla\mathbf{v}) - (p, \nabla\cdot\mathbf{v}) &= (\mathbf{f}, \mathbf{v}) \\ (\nabla\cdot\mathbf{u}, q) &= 0 \\ \mathbf{u}(0) &= \mathbf{u}_0\end{aligned} \tag{1.5}$$

hold for every $\mathbf{v} \in \mathbb{H}^1_0(\Omega)$ and $q \in L^2(\Omega)$. Previous equations are meant in the sense of scalar distributions on interval $(0,T)$. More detailed information can be found in [Feistauer, 2006]. It is easy to see, that the classical solution to problem (1.2) is also a weak solution. The weak formulation of Navier-Stokes equations is usually formulated using the spaces of solenoidal functions (more information for example in [Feistauer, 2006]), however such a formulation is not suitable for the numerical purposes of this thesis. The external forces are omitted in the rest of the thesis.

### 1.2.1 Time Discretization

Consider time discretization $\{t_j\}_{j=0}^K, t_K = T$. Two main approaches to discretization are described in the thesis. The $\theta$ scheme which fully couples velocity and pressure computation together and the Incremental Pressure Correction Scheme (IPCS) where the computation of velocity and pressure is decoupled.

The $\theta$ scheme can be stated as follows. Let us suppose, that we know a solution $\mathbf{u}^k, p^k$ for known time $t_k, k \in \mathbb{N}$ and the goal is to find a solution $\mathbf{u}^{k+1}, p^{k+1}$ at time $t_{k+1}$ after a time step $\mathrm{d}t$. The problem can be rewritten as

$$(\frac{\partial \mathbf{u}^{k+1}}{\partial t}, \mathbf{v}) + (p^{k+1}, \nabla\cdot\mathbf{v}) + (\nabla\cdot\mathbf{u}^{k+1}, q) + F(t_{k+1}) = 0, \tag{1.6}$$

for every $\mathbf{v} \in \mathbb{H}^1_0(\Omega)$ and $q \in L^2(\Omega)$, where function $F(t)$ stands for rest of the terms in (1.5). After the use of *one step theta time stepping scheme* as described in [Turek et al., 2006], which states that the previous equation can be approximated as

$$(\frac{\mathbf{u}^{k+1} - \mathbf{u}^k}{\mathrm{d}t}, \mathbf{v}) + (p^{k+1}, \nabla\cdot\mathbf{v}) + (\nabla\cdot\mathbf{u}^{\mathbf{k+1}}, q) + \theta F(t_{k+1}) + (1-\theta)F(t_k) = 0, \tag{1.7}$$

for $\theta \in (0,1]$ for every $\mathbf{v} \in \mathbb{H}^1_0(\Omega)$ and $q \in L^2(\Omega)$. Two sensible choices are $\theta = 0.5$ and $\theta = 1$. The scheme is called the Crank-Nicholson scheme for $\theta = 0.5$. The scheme is called the implicit Euler scheme for $\theta = 1$ (hereinafter only Euler scheme). The convergence order of the Crank-Nicholson scheme is $\mathcal{O}(\mathrm{d}t^2)$ while the order of Euler scheme is only $\mathcal{O}(\mathrm{d}t)$. The disadvantage of the Crank-Nicholson scheme is its lesser stability.

### 1.2.2 The Finite Element Method

The Finite Element Method is used for the spatial discretization. This thesis contains only a short intro on the topic, more information can be found in [Feistauer, 2006] or [Glowinski, 2003], Let us have a triangular (tetrahedral in 3D) discretization of $\Omega$ called $\Omega_h$, where $h$ is a discretization parameter. It can be the biggest

side of the triangle in 2D or radius of the inscribed or circumscribed ball in 3D for example. Instead of infinite dimensional spaces, a finite dimensional subspaces will be used for finding the solution. We consider two finite dimensional spaces of piece-wise polynomial functions, one for approximating velocity $\mathbb{X}_h$, other for approximating pressure $Q_h$. Spaces $\mathbb{X}_h, Q_h$ have to satisfy the Babuška-Brezzi condition

$$\inf_{\substack{0\neq q\in Q \\ \int_\Omega q\,\mathrm{d}V=0}} \sup_{\mathbf{v}\in\mathbb{X}} \frac{(\nabla\cdot\mathbf{v},q)}{\|q\|_{L^2}\|\mathbf{v}\|_{H^1}} \geq \beta \tag{1.8}$$

for $\beta > 0$ to obtain a stable discretization. The *Taylor Hood* $P_2/P_1$ [Feistauer, 2006] and *mini* [Brezzi and Fortin, 2012] elements will be used in this thesis as both sastisfy the previous condition. One can then state fully discretized problem for both time stepping schemes by interchanging the infinite dimensional space by the finite dimensional subspaces. That is find $\mathbf{u}_h^{k+1} \in \mathbb{X}_h, p_h^{k+1} \in Q_h$ such that

$$\begin{aligned}(\frac{\mathbf{u_h^{k+1}}-\mathbf{u}_h^k}{\mathrm{d}t},\mathbf{v})+(p_h^{k+1},\nabla\cdot\mathbf{v})+(\nabla\cdot\mathbf{u_h^{k+1}},q)+\theta((\nabla\mathbf{u}_h^{k+1}\mathbf{u}_h^{k+1},\mathbf{v})+(2\nu\mathbb{D}_h^{k+1},\nabla\mathbf{v}))\\ +(1-\theta)((\nabla\mathbf{u}_h^k\mathbf{u}_h^k,\mathbf{v})+(2\nu\mathbb{D}_h^k,\nabla\mathbf{v}))=0\end{aligned} \tag{1.9}$$

for every $\mathbf{v}\in\mathbb{X}_h$ and $q\in Q_h$.

By expansions of velocity and pressure into the basis of finite dimensional spaces ($N_u$ is the number of dof - degrees of freedoms of the velocity space,$N_p$ number of dof of the pressure space)

$$\begin{aligned}\mathbb{X}_h &= \{\boldsymbol{\phi}_1,...,\boldsymbol{\phi}_{N_u}\}\\ Q_h &= \{\varphi_1,...,\varphi_{N_p}\}\\ \mathbf{u}_h^k(\mathbf{x}) &= \sum_{i=1}^{N_u}\hat{u}_i^k\boldsymbol{\phi}_i(\mathbf{x}).\\ p_h^k(\mathbf{x}) &= \sum_{i=1}^{N_p}\hat{p}_i^k\varphi_i(\mathbf{x}).\end{aligned} \tag{1.10}$$

One obtains a set of nonlinear (linear for IPCS) equations for each time step. These equations can be solved via Newton's method, or other methods for solving set of nonlinear equations such as Picard iteration. The hat notation $\hat{\mathbf{u}}_h = \{\hat{u}_i\}_{i=1}^{N_u}, \hat{p}_h = \{\hat{p}_i\}_{i=1}^{N_p}$ is used to distinguish algebraic vectors from functions in the following chapter.

# 2. Vortex Rings

This chapter is focused on a preconditioning of the incompressible Navier-Stokes equations and the comparison of a fully coupled $\theta$ discretization and the projection IPCS method. The tests are carried out on the vortex rings simulation of an experiment [Švančara et al., 2020]. The experiment studies behavior of vortex rings in a liquid $^4He$ at low temperatures $(1.3K - 1.9K)$. The aim of the simulations is to compare new quantity pseudovorticity with vorticity on the vortex rings. The pseudovorticity could serve as a new tool for tracking and identifying properties of vortices in the experiments and for this reason it is useful to study its relationship with vorticity in the simulations.

## 2.1 Incremental Pressure Correction Scheme

The basic idea of the IPCS scheme [Guermond et al., 2006] is to transform the nonlinear problem into three linear problems. The boundary conditions for the IPCS scheme will be discussed at the end of this section.

Consider the velocity equation in (1.5). Let suppose we know the solution $\mathbf{u}^k, p^k$ at time $t_k$ and we want to find solution $\mathbf{u}^{k+1}, p^{k+1}$ at time $t_{k+1}$. Consider using $\theta$ discretization scheme on the velocity equation, but replacing pressure unknown pressure $p^{k+1}$ by known pressure $p^k$ and linearizing the inertia term by

$$\mathbf{u}^{k+1} \cdot \nabla \mathbf{u}^{k+1} \approx \mathbf{u}^k \cdot \nabla \mathbf{u}^{k+1} \tag{2.1}$$

one obtains the equation for so called *tentative* velocity $\mathbf{u}^t$

$$\begin{aligned}(\frac{\mathbf{u}^t - \mathbf{u}^k}{\mathrm{d}t}, \mathbf{v}) + (\mathbf{u}^k \cdot \nabla(\theta \mathbf{u}^t + (1-\theta)\mathbf{u}^k), \mathbf{v}) \\ + 2\nu((\theta \mathbb{D}^t + (1-\theta)\mathbb{D}^k), \nabla \mathbf{v}) = (p^k, \nabla \cdot \mathbf{v}) \quad \forall \mathbf{v} \in \mathbb{H}^1_0(\Omega).\end{aligned} \tag{2.2}$$

The inertia term is often linearized using the Adam-Bashfort scheme as

$$\mathbf{u}^{k+1} \cdot \nabla \mathbf{u}^{k+1} \approx \frac{3\mathbf{u}^k - \mathbf{u}^{k-1}}{2} \cdot \nabla \mathbf{u}^{k+1}$$

however it seemed to make no difference in the simulation of the vortex rings so the simpler option was chosen instead. In the second step of the IPCS scheme, one wants find $\mathbf{u}^{k+1}, p^{k+1}$ such that $\nabla p^{k+1}$ is correction to $\mathbf{u}^t$ to ensure incompressibility of $\mathbf{u}^{k+1}$. This can be achieved by the following steps. Compute the new pressure $p^{k+1}$ by solving

$$(\nabla p^{k+1}, \nabla q) = (\frac{\nabla \cdot \mathbf{u}^t}{\mathrm{d}t}, q) + (\nabla p^k, \nabla q) \quad \forall q \in L^2(\Omega) \tag{2.3}$$

and then update the tentative velocity to obtain the final velocity $\mathbf{u}^{k+1}$

$$(\mathbf{u}^{k+1}, \mathbf{v}) = (\mathbf{u}^t, \mathbf{v}) - \mathrm{d}t(\nabla(p^{k+1} - p^k), \mathbf{v}) \quad \forall \mathbf{v} \in \mathbb{H}^1_0(\Omega). \tag{2.4}$$

The boundary conditions for these equation are problematic and discussed topic [Gresho and Sani, 1987],[Vreman, 2014]. Let us state boundary conditions

used for the computations in this thesis and for more information we refer the reader to the literature for example [Guermond et al., 2006],[Gresho and Sani, 1987],[Vreman, 2014]. Let us start with the equation (2.3). The equation is formulated with implicit boundary condition $\nabla p^{k+1}\mathbf{n} = \nabla p^k \mathbf{n}$ which is non-physical and leads to creation of a numerical boundary layer which limits the accuracy of the scheme [Guermond et al., 2006]. Since pressure is determined up to a constant an additional boundary condition is needed to solve the equation. There are couple of solutions to this problem, such as imposing Dirichlet BC for pressure on part of the boundary, Lagrange multipliers to enforce zero mean pressure or compute the minimal norm solution to the singular problem.

While it is quite straight forward to implement the Dirichlet boundary conditions for equation (2.2) it is complicated to come up with proper outflow conditions. The traction free condition

$$(-p\mathbb{I} + 2\nu\mathbb{D})\mathbf{n} = 0 \tag{2.5}$$

couples velocity and pressure together. However, notice that pressure is not an unknown in the first equation. The outflow boundary condition for the projection scheme are

$$(-p^k\mathbb{I} + 2\nu\mathbb{D}^t)\mathbf{n} = 0 \tag{2.6}$$

as stated in [Guermond et al., 2006]. The third correction step requires no boundary conditions. It should be mentioned that this scheme is known to require small time steps to give accurate results. By using the Finite Element Method one obtains a set of 3 linear problems to be solved for each time step.

## 2.2 Preconditiong of the Navier-Stokes Equations

Consider the Navier-Stokes equations, temporally discretized using the $\theta$ time stepping scheme and spatially discretized by the FEM and let the finite elements satisfy (1.8). Consider the decomposition of $\mathbf{u}_h^k, p_h^k$ into the basis of the form (1.10). Let us take a closer look at the system of the nonlinear equations which arises for the coefficients. From the balance of linear momentum we get (consider implicit euler for simplicity)

$$\{R_u(\hat{\mathbf{u}}^{k+1}, \hat{\mathbf{p}}^{k+1})\}_j = \sum_{i=1}^{N_u} \frac{1}{dt}(\hat{u}_i^{k+1}\boldsymbol{\phi}_i, \boldsymbol{\phi_j}) + \sum_{i=1}^{N_u}\sum_{k=1}^{N_u} \hat{u}_i^{k+1}\hat{u}_k^{k+1}(\nabla(\boldsymbol{\phi}_i)\boldsymbol{\phi}_k, \boldsymbol{\phi_j}) +$$
$$\nu\sum_{i=1}^{N_u}\hat{u}_i^{k+1}(\boldsymbol{\phi}_i, \boldsymbol{\phi_j}) - \sum_{i=1}^{N_p}\hat{p}_i^{k+1}(\varphi_i, \nabla\cdot\boldsymbol{\phi_j}) + G(\hat{\mathbf{u}}^k, \hat{\mathbf{p}}^k, \boldsymbol{\phi_j}) = 0 \quad \forall j = 1, .., N_u \tag{2.7}$$

where $G(\hat{\mathbf{u}}^k, \hat{\mathbf{p}}^k, \boldsymbol{\phi_j}) = 0$ is a function of tests function and discrete solution $\hat{\mathbf{u}}^k, \hat{\mathbf{p}}^k$ from previous step. From the incompressibility one obtains

$$\{R_m(\hat{\mathbf{u}}^{k+1})\}_j = \sum_{i=1}^{N_u}\hat{u}_i^{k+1}(\nabla\cdot\boldsymbol{\phi_i}, \varphi_j) = 0 \quad \forall j = 1, .., N_p. \tag{2.8}$$

Let us take a look at one iteration of Newton's method for this system of nonlinear equations that is

$$\begin{aligned} \hat{\mathbf{u}}_{new}^{k+1} &= \hat{\mathbf{u}}_{old}^{k+1} + \delta\hat{\mathbf{u}} \\ \hat{p}_{new}^{k+1} &= \hat{p}_{old}^{k+1} + \delta\hat{p} \end{aligned} \tag{2.9}$$

where $\delta\hat{\mathbf{u}}, \delta\hat{p}$ satisfies

$$J \begin{pmatrix} \delta\hat{\mathbf{u}} \\ \delta\hat{p} \end{pmatrix} = - \begin{pmatrix} \mathbf{R}_u(\hat{\mathbf{u}}_{old}^{k+1}, \hat{\mathbf{p}}_{old}^{k+1}) \\ \mathbf{R}_m(\hat{\mathbf{u}}_{old}^{k+1}) \end{pmatrix} \tag{2.10}$$

and J is the Jacobi matrix of the system. Let us take a closer look at the Jacobi matrix, denoting

$$\begin{aligned} \mathbf{M} &= \{(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j)\}_{i,j=1}^{i,j=N_u} \\ \mathbf{M}_p &= \{(\varphi_i, \varphi_j)\}_{i,j=1}^{i,j=N_p} \\ \mathbf{K} &= \{(\nabla\boldsymbol{\phi}_i \mathbf{u}_{old}^{k+1}, \boldsymbol{\phi}_j)\}_{i,j=1}^{i,j=N_u} \\ \mathbf{L} &= \{(\nabla\mathbf{u}_{old}^{k+1}\boldsymbol{\phi}_i, \boldsymbol{\phi}_j)\}_{i,j=1}^{i,j=N_u} \\ \mathbf{A} &= \{\nu((\nabla\boldsymbol{\phi}_i + (\nabla\boldsymbol{\phi}_i)^T, \nabla\boldsymbol{\phi}_j)\}_{i,j=1}^{i,j=N_u} \\ \mathbf{B} &= -\{(\nabla\cdot\boldsymbol{\phi}_j, \varphi_i)\}_{i,j=1}^{i=N_u,j=N_p} \end{aligned} \tag{2.11}$$

the system of equations (2.10) can be rewritten as:

$$\mathbf{J} \begin{pmatrix} \delta\mathbf{u} \\ \delta p \end{pmatrix} = \begin{pmatrix} \frac{1}{dt}\mathbf{M} + \mathbf{K} + \mathbf{L} + \mathbf{A} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \delta\mathbf{u} \\ \delta p \end{pmatrix} = - \begin{pmatrix} \mathbf{R}_u(\hat{\mathbf{u}}_{old}^{k+1}, \hat{\mathbf{p}}_{old}^{k+1}) \\ \mathbf{R}_m(\hat{\mathbf{u}}_{old}^{k+1}) \end{pmatrix} \tag{2.12}$$

If the $\mathbf{L}$ term is dropped, the resulting system corresponds to the Picard iteration [Elman et al., 2014]. We see that the problem has a saddle point structure and an important block structure. Consider a general problem with the matrix

$$\mathbf{C} = \begin{pmatrix} \mathbf{F} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{0} \end{pmatrix}. \tag{2.13}$$

solved by the right preconditioned GMRES with the preconditioning matrix

$$\mathbf{P} = \begin{pmatrix} \mathbf{F} & \mathbf{B}^T \\ \mathbf{0} & -\mathbf{S} \end{pmatrix}. \tag{2.14}$$

where $\mathbf{S} = \mathbf{B}\mathbf{F}^{-1}\mathbf{B}^T$ denotes the Schur complement of the matrix $\mathbf{C}$. Then the GMRES [Liesen and Strakoš, 2012] would converge in two iterations [Murphy et al., 1999], however the matrix $\mathbf{S}$ is dense and the action of the $\mathbf{F}^{-1}$ is expensive as well. That said, the action of the $\mathbf{P}^{-1}$ on a vector would be too expensive to compute. While the approximation of the $\mathbf{F}^{-1}$ is easy to deal with (see the end of the chapter), the action of $\mathbf{S}^{-1}$ is difficult to approximate. Following sections describe two ways of approximating the $\mathbf{S}^{-1}$ for the incompressible Navier Stokes problem. The options are the Pressure Convection-Diffusion-Reaction (PCDR) preconditioning and the Least Squares Commutator (LSC) preconditioning.

### 2.2.1 Pressure Convection Diffusion Reaction Preconditioning

This section is generally based on information found in [Elman et al., 2014] and the PhD. thesis of Martin Řehoř [Řehoř, 2018] and Jan Blechta [Blechta, 2019], the developers of the FENaPack [Blechta and Řehoř, 2019]. Let us make some simplifications to the problem. Consider a steady problem and $\mathbf{F}$ corresponding to the Picard iteration. The operators in the Jacobian can be formally interpreted as discretized continuous operators. The operator $\mathbf{M}^{-1}\mathbf{F}$ (acting on $\hat{\mathbf{u}}$) corresponds to the continuous operator $\mathcal{F}$ (acting on $\mathbf{u}$)

$$\mathcal{F}\mathbf{u} = \mathbf{u}_{old}^{k+1} \cdot \nabla\mathbf{u} - \mathrm{div}(2\nu\mathbb{D}) \tag{2.15}$$

where $\mathbf{u}_{old}^{k+1}$ is the known velocity obtained in last iteration of the Picard (or Newton's) method. The operator $\mathbf{M}_p^{-1}\mathbf{B}$ (acting on $\hat{\mathbf{u}}$) corresponds to $-\mathrm{div}()$ and the operator $\mathbf{M}^{-1}\mathbf{B}^T$ corresponds to $\nabla$ (acting on $\hat{p}$) (see [Elman et al., 2014]). That said the operator $\mathbf{S}$ formally corresponds to the operator

$$S = -\mathrm{div} \circ F^{-1} \circ \nabla \tag{2.16}$$

as can be observed from using the discrete operators

$$\mathbf{M}_p^{-1}\mathbf{S} = (\mathbf{M}_p^{-1}\mathbf{B})(\mathbf{M}^{-1}\mathbf{F})^{-1}(\mathbf{M}^{-1}\mathbf{B}^T) = \mathbf{M}_p^{-1}\mathbf{B}\mathbf{F}^{-1}\mathbf{B}^T. \tag{2.17}$$

The main idea in the PCDR preconditioning is to swap the order of operators and at the same time replace the operator $\mathbf{F}$ by its counterpart acting on the pressure space $\mathbf{F}_p$ (the discrete form of the operator $\mathbf{F}_p$ is specified later). That leaves us with two possible versions of the PCDR preconditioning

$$\begin{aligned} \mathbf{S}^{-1} &\approx \mathbf{M}_p\mathbf{F}_p^{-1}\mathbf{B}\mathbf{M}^{-1}\mathbf{B}^T \\ \mathbf{S}^{-1} &\approx \mathbf{B}\mathbf{M}^{-1}\mathbf{B}^T\mathbf{F}_p^{-1}\mathbf{M}_p. \end{aligned} \tag{2.18}$$

Since matrix $\mathbf{B}\mathbf{M}^{-1}\mathbf{B}^T$ is generally dense, the sparse discrete laplacian matrix $\hat{\mathbf{A}}_p$ (defined below) can be under some assumptions (see [Elman et al., 2014]) (matrix is singular, artificial boundary conditions are needed for nonenclosed flows) used instead. Another approach of approximating the matrix $\mathbf{B}\mathbf{M}^{-1}\mathbf{B}^T$ is to use $\mathbf{B}\mathbf{D}_{\mathbf{M}}^{-1}\mathbf{B}^T$ instead, where $\{\mathbf{D}_{\mathbf{M}}\}_{i,j} = \{\mathbf{M}\}_{i,j}\delta_{ij}$ is the diagonal matrix of $\mathbf{M}$. Defining $\hat{\mathbf{A}}_p, \mathbf{K}_p, \mathbf{A}_p$ analogically as in the (2.11)

$$\begin{aligned} \hat{\mathbf{A}}_p &= \{(\nabla\varphi_i, \nabla\varphi_j)\}_{i,j=1}^{i,j=N_p} \\ \mathbf{K}_p &= \{(\nabla\varphi_i, \mathbf{u}_{old}^{k+1}\varphi_j)\}_{i,j=1}^{i,j=N_p} \\ \mathbf{A}_p &= \nu\hat{\mathbf{A}}_p \end{aligned} \tag{2.19}$$

the 2 options for PCDR preconditiong can be summarized as

$$\begin{aligned} \mathbf{S}_{BRM1} &= \mathbf{A}_p\mathbf{F}_p^{-1}\mathbf{M}_p \\ \mathbf{S}_{BRM2} &= \mathbf{M}_p\mathbf{F}_p^{-1}\mathbf{A}_p \end{aligned} \tag{2.20}$$

and their inverse (and corresponding continuous counterparts)

$$\begin{aligned} \mathbf{S}_{BRM1}^{-1} &= \mathbf{M}_p^{-1}\mathbf{F}_p\mathbf{A}_p^{-1} \approx \mathcal{F}_p \circ (-\Delta)^{-1} \\ \mathbf{S}_{BRM1}^{-1} &= \mathbf{A}_p^{-1}\mathbf{F}_p\mathbf{M}_p^{-1} \approx (-\Delta)^{-1} \circ \mathcal{F}_p. \end{aligned} \tag{2.21}$$

Let us return to the original problem and specify the operators $\mathcal{F}$ and $\mathbf{F}$. The problem is more complicated for the Newton's iteration, so let us consider the operator $\mathbf{F}$ coming from the Picard iteration of the unsteady problem, that is

$$\mathbf{F} = \frac{1}{dt}\mathbf{M} + \mathbf{K} + \mathbf{A} \tag{2.22}$$

which approximately corresponds to

$$\mathcal{F}\mathbf{u} = \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}_{old}^{k+1} \cdot \nabla \mathbf{u} - \mathrm{div}(2\nu\mathbb{D}). \tag{2.23}$$

The corresponding pressure counterpart can be chosen as follows

$$\mathcal{F}_p p = \frac{\partial p}{\partial t} + \mathbf{u}_{old}^{k+1} \cdot \nabla p - \mathrm{div}(\nu \nabla p) \tag{2.24}$$

and the resulting discrete operator

$$\mathbf{F}_p = \frac{1}{\mathrm{d}t}\mathbf{M}_p + \mathbf{K}_p + \mathbf{A}_p. \tag{2.25}$$

Since (2.21) contains at least one Laplace solve an additional artificial boundary conditions are often needed to construct the preconditioner. This topic is widely discussed (mostly for stationary case) for example in [Blechta, 2019],[Elman et al., 2014],[Howle et al., 2006],[Olshanskii and Vassilevski, 2007]. The BRM1 version prescribes homogeneous Dirichlet boundary conditions on the inlet, while the BRM2 prescribes homogeneous Dirichlet boundary conditions on the outlet. More about the boundary conditions is explained in [Blechta, 2019] for both variants of preconditioning. Additional boundary modifications of the operator $\mathbf{F}_p$ are discussed in [Elman et al., 2014].

The PCDR preconditioner is implented in FENaPack in FEniCS[Alnæs et al., 2015] [Blechta and Řehoř, 2019]. By substituting (2.25) into (2.21) we obtain (BRM1 version only for simplicity)

$$\mathbf{S}_{BRM1}^{-1} = \mathbf{M}_p^{-1}(\frac{1}{\mathrm{d}t}\mathbf{M}_p + \mathbf{K}_p + \hat{\mathbf{A}}_p)\mathbf{A}_p^{-1} = \frac{1}{\mathrm{d}t}\hat{\mathbf{A}}_p^{-1} + \nu\mathbf{M}_p^{-1}(\frac{1}{\nu}\mathbf{K}_p\hat{\mathbf{A}}_p^{-1} + \mathbf{I}) \tag{2.26}$$

where $\mathbf{I}$ stands for the identity matrix. It was already mentioned that matrices $\hat{\mathbf{A}}_p$ and $\mathbf{B}\mathbf{D}_{\mathbf{M}}^{-1}\mathbf{B}^T$ have similar properties to $\mathbf{B}\mathbf{M}^{-1}\mathbf{B}^T$. The first term in (2.26) is instead approximated as

$$\mathbf{S}_{BRM1}^{-1} = \frac{1}{\mathrm{d}t}(\mathbf{B}\mathbf{D}_{\mathbf{M}}^{-1}\mathbf{B}^T)^{-1} + \nu\mathbf{M}_p^{-1}(\frac{1}{\nu}\mathbf{K}_p\hat{\mathbf{A}}_p^{-1} + \mathbf{I}) \tag{2.27}$$

This can be justified for $t \to 0_+$ as $\mathbf{S}^{-1} = (\mathbf{B}\mathbf{F}^{-1}\mathbf{B}^T)^{-1} \approx \frac{1}{\mathrm{d}t}(\mathbf{B}\mathbf{M}^{-1}\mathbf{B}^T)^{-1}$ and $\mathbf{S}_{BRM1}^{-1} \approx \frac{1}{\mathrm{d}t}(\mathbf{B}\mathbf{D}_{\mathbf{M}}^{-1}\mathbf{B}^T)^{-1}$. Approximation of the $\mathbf{S}_{BRM2}^{-1}$ is done in the same manner.

### 2.2.2 Least Squares Commutator

The second option of approximating the Schur complement is called the Least Squares Commutator (LSC) and is implemented in the PETSc library [Balay et al., 2015] and described well in [Elman et al., 2014]. Let us consider steady

case and linearization of the convection term via the Picard iteration and corresponding operators

$$\begin{aligned} \mathcal{F}\mathbf{u} &= \mathbf{u}_{old}^{k+1} \cdot \nabla \mathbf{u} - \mathrm{div}(2\nu\mathbb{D}) \\ \mathcal{F}_p p &= \mathbf{u}_{old}^{k+1} \cdot \nabla p - \mathrm{div}(\nu \nabla p). \end{aligned} \tag{2.28}$$

Consider commutator of these operators with the gradient operator

$$\mathcal{E} = \nabla \cdot \mathcal{F} - \mathcal{F}_p \nabla \cdot \tag{2.29}$$

The key idea is to construct operator $\mathbf{F}_p$ such that the commutator will be small in some sense. Following relation for the discretized operator $\mathbf{E}$ is obtained by using the relations between discretized operators $\mathbf{F}, \mathbf{F}_p$ and their continuous counterparts $\mathcal{F}, \mathcal{F}_p$, mentioned in previous section.

$$\mathbf{E} = (\mathbf{M}^{-1}\mathbf{B})(\mathbf{M}^{-1}\mathbf{F}) - (\mathbf{M}_p^{-1}\mathbf{F}_p)(\mathbf{M}^{-1}\mathbf{B}) \tag{2.30}$$

If we assume that this commutator is small, i.e. $\mathbf{E} \approx \mathbf{0}$ we obtain

$$\mathbf{S}_{LSC} = \mathbf{B}\mathbf{F}^{-1}\mathbf{B}^T \approx \mathbf{M}\mathbf{F}_p^{-1}\mathbf{B}\mathbf{M}^{-1}\mathbf{B}^T \tag{2.31}$$

The construction of $\mathbf{F}_p$ can be thought as of minimization of the commutator in the least square sense. Instead of commutator $\mathcal{E}$ consider adjoint operator $\mathcal{E}^*$ defined by $(\mathcal{E}\mathbf{u}, p) = (\mathbf{u}, \mathcal{E}^* p)$. For smooth functions from pressure space we obtain for the norm of the adjoint operator (using similar relations between discretized operators and their continuous counterparts as above)

$$\begin{aligned} \sup_{p_h \neq 0} \frac{\|(-\nu\Delta + \mathbf{u}_{old}^{k+1} \cdot \nabla)\nabla - \nabla(-\nu\Delta + \mathbf{u}_{old}^{k+1} \cdot \nabla)\|}{\|p_h\|} = \\ = \sup_{\hat{\mathbf{p}} \neq 0} \frac{\|((\mathbf{M}^{-1}\mathbf{F}^T)(\mathbf{M}^{-1}\mathbf{B}^T) - (\mathbf{M}^{-1}\mathbf{B}^T)(\mathbf{M}^{-1}\mathbf{F}_p^T))\hat{\mathbf{p}}\|_{\mathbf{M}}}{\|\hat{\mathbf{p}}\|_{\mathbf{M}}} \end{aligned} \tag{2.32}$$

where $\|\hat{\mathbf{u}}\|_{\mathbf{M}} = \hat{\mathbf{u}}^T\mathbf{M}\hat{\mathbf{u}}, \|\hat{\mathbf{p}}\|_{\mathbf{M}} = \hat{\mathbf{p}}^T\mathbf{M}_p\hat{\mathbf{p}}$. The main idea is to construct $\mathbf{F}_p$ such that the norm (2.32) is as small as possible. One option is to minimize the norms of the individual vectors of the commutator. This results in a weighted least-squares problem.

$$\min \|\{\mathbf{M}^{-1}\mathbf{F}^T\mathbf{M}^{-1}\mathbf{B}^T\}_j - (\mathbf{M}^{-1}\mathbf{B}^T\mathbf{M}^{-1})\{\mathbf{F}_p^T\}_j\|_{\mathbf{M}} \quad \text{for} \quad j = 1, ..., N_p \tag{2.33}$$

The solution to the previous problem gives the solution [Elman et al., 2014]

$$\mathbf{F}_p = (\mathbf{B}\mathbf{M}^{-1}\mathbf{F}\mathbf{M}^{-1}\mathbf{B}^T)(\mathbf{B}\mathbf{M}^{-1}\mathbf{B}^T)^{-1}\mathbf{M} \tag{2.34}$$

which results in schur complement of the form

$$\mathbf{S}_{LSC} \approx (\mathbf{B}\mathbf{M}^{-1}\mathbf{B}^T)(\mathbf{B}\mathbf{M}^{-1}\mathbf{F}\mathbf{M}^{-1}\mathbf{B}^T)^{-1}(\mathbf{B}\mathbf{M}^{-1}\mathbf{B}^T) \tag{2.35}$$

Since $\mathbf{M}^{-1}$ is dense [Elman et al., 2014], it is practical to replace it with $\mathbf{D}_{\mathbf{M}}^{-1}$ (as mentioned before $\mathbf{B}\mathbf{D}_{\mathbf{M}}^{-1}\mathbf{B}^T$ is basically scaled sparse Laplacian). The final approximation to the LSC preconditioning reads

$$\mathbf{S}_{LSC} = (\mathbf{B}\mathbf{D}_{\mathbf{M}}^{-1}\mathbf{B}^T)(\mathbf{B}\mathbf{D}_{\mathbf{M}}^{-1}\mathbf{F}\mathbf{D}_{\mathbf{M}}^{-1}\mathbf{B}^T)^{-1}(\mathbf{B}\mathbf{D}_{\mathbf{M}}^{-1}\mathbf{B}^T) \tag{2.36}$$

Information regarding boundary modification of the discrete preconditioner $\mathbf{S}_{LSC}$ can be found in [Elman et al., 2014].

One can simplify the preconditioning even more by omitting the diagonal scaling obtaining

$$\mathbf{S}_{LSC} = (\mathbf{B}\mathbf{B}^T)(\mathbf{B}\mathbf{F}\mathbf{B}^T)^{-1}(\mathbf{B}\mathbf{B}^T) \tag{2.37}$$

The advantage over the PCDR preconditioning is not having to explicitly construct operator $\mathbf{F}_p$, but on the other hand, it involves two Laplace type solves instead of one compared to the PCDR.

### 2.2.3 Remarks - Newton's Method

Both of the mentioned preconditioners were derived for the Picard iteration, however it is still possible to use them together with the Newton's method with slight modification. The $\mathbf{F}$ in the (2.14) comes from the Newton's method for both types of approximation to the Schur complement.

The LSC approximation of the Schur complement $\mathbf{S}$ (2.37) is constructed from the exact $\mathbf{F}$ coming from the Newton's method withoud any modifications.

When using the PCDR preconditioner with the Newton's method, the approximation of the Schur complement is constructed using only part of $\mathbf{F}$ corresponding to the Picard iteration, cf. (2.22).

### 2.2.4 Approximation of F

There are many ways to approximate the action of the $\mathbf{F}^{-1}$ but only the best two (performance-wise) are mentioned in the thesis. One Richardson iteration together with the (hypre) AMG (algebraic multigrid) [Henson and Yang, 2002] preconditioner is used to approximate action of $\mathbf{F}^{-1}$ for the PCDR preconditioner as described in [Blechta, 2019]. For the LSC preconditioning we heuristically found two possible choices which performed well. One option for the LSC was to use BiCG$stab(l)$ [Fokkema, 1996] preconditioned by Successive-Over-Relaxation (SOR) [DeLong and Ortega, 1995] to roughly approximate the action of the inverse. Second option was to switch the SOR preconditioning by the AMG, however the first option turned out to be more efficient for the vortex ring experiment and is used in all of the computations in this chapter.

## 2.3 Numerical Experiments

This section contains the simulations of a vortex rings experiment and numerical tests and comparison of the preconditioners and the IPCS projection method.

The setup of the simulation mimics the experiment [Švančara et al., 2020]. Both 2D and 3D simulation of the experiment are considered. Let us start with the 2D model. Consider domain with boundary conditions illustrated in the Figure 2.1. The width of the domain is $W = 50mm$, the length of the domain is $L = 110mm$. The height of the jet is chosen to be $10mm$ and the width is $w = 5mm$. The 2D model is not consistent with the experiment because it does not preserve the axial (axis of the jet) symmetry of the problem, however the computed velocity and vorticity fields can still be used for the comparison

with pseudovorticity nevertheless. The inflow function on the top of the jet is a parabolic velocity profile modified in time that is

$$\mathbf{u}_{\partial\Omega_{in}}(\mathbf{x},t) = (0,u_0)((\frac{2x}{w})^2+1)f(t) \tag{2.38}$$

where $u_0$ is the maximum velocity of the inflow and

$$f(t) = \begin{cases} \frac{t}{0.1} & x \leq 0.1 \\ 1 & 0.1 \leq x \leq 0.9 \\ \frac{1-t}{0.1} & 0.9 \leq x \leq 1 \\ 0 & 1 \neq x. \end{cases} \tag{2.39}$$

The homogeneous Dirichlet boundary conditions are applied everywhere except for the topmost horizontal wall where traction free boundary condition is used. The viscosity of the fluid used is $\nu = 0.01mm^2/s$.

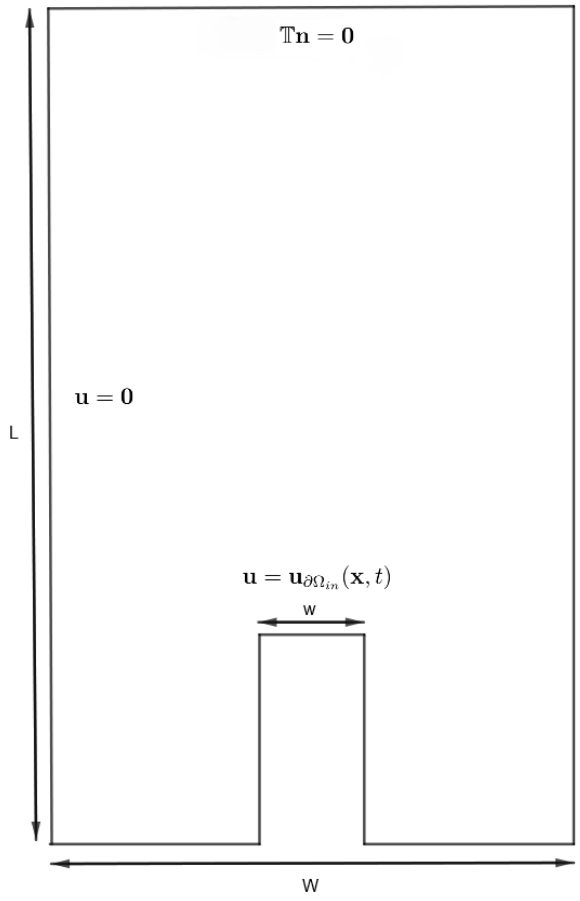


Figure 2.1: A sketch of the problem with boundary conditions.

The projection method requires use of an artificial boundary condition for the pressure. It has been verified by numerical tests that prescribed zero pressure on the topmost horizontal wall of the domain works well if the domain is long enough even-though it is not the correct condition. Better condition for pressure would be to enforce zero mean pressure via Lagrange multipliers, however since it makes

| #elements | v-dof | p-dof | total dof |
|---|---|---|---|
| 45k | 183k | 23k | 206k |
| 133k | 536k | 67k | 602k |
| 285k | 1143k | 143k | 1287k |

Figure 2.2: Number of velocity and pressure dof for the Taylor-Hood elements for the 2D meshes.

the pressure update problem much harder to solve and makes little difference in the computations. The Taylor-Hood elements were used as they can capture the vorticity more accurately than the mini elements but still retain computational efficiency when proper preconditioning is used. Three meshes of various resolution (see 2.2) in 2D were used for the computations. The computations on the most detailed mesh were carried out to study the properties of the vortex rings captured by theta-pseudovorticity [Švančara et al., 2020] and vorticity which is mentioned later on. The Figure 2.3 shows the directions of flows in the vicinity of a vortex ring pair. The vortex ring pair moves upwards in time and eventually stops.

The $\theta$ in the time stepping scheme is chosen to be 0.5, i.e. the Crank-Nicholson scheme is used. The IPCS is used in the Crank-Nicholson form as well. The computations were carried out using the time step $dt = 0.01s$ ($0.001s$ for the 1.3M dof mesh) which naturally comes up from the requirements of the experimental physicists. Smaller time steps make no significant difference in the computations. This is quite interesting, because one would expect that the projection method will require smaller time-step to give the same results, however it does not seem to be the case in this experiment. This behavior is observed if the time step is increased to $dt = 0.05s$, however such time step is too large to properly start up the simulation even with mixed formulation and the simulations then cannot be used for computations of the pseudovorticity. The length of the simulation is $15s$. Since the differences in the flows computed by the IPCS and $\theta$ scheme are almost negligible and hardly visible all the figures show results computed by the IPCS scheme.

The impact of the mesh resolution on the computation is illustrated in Figure (2.4) and (2.5). Even the coarsest mesh captures the vortex ring pair quite well and the main features of the flow remain to be the same regardless of the resolution.

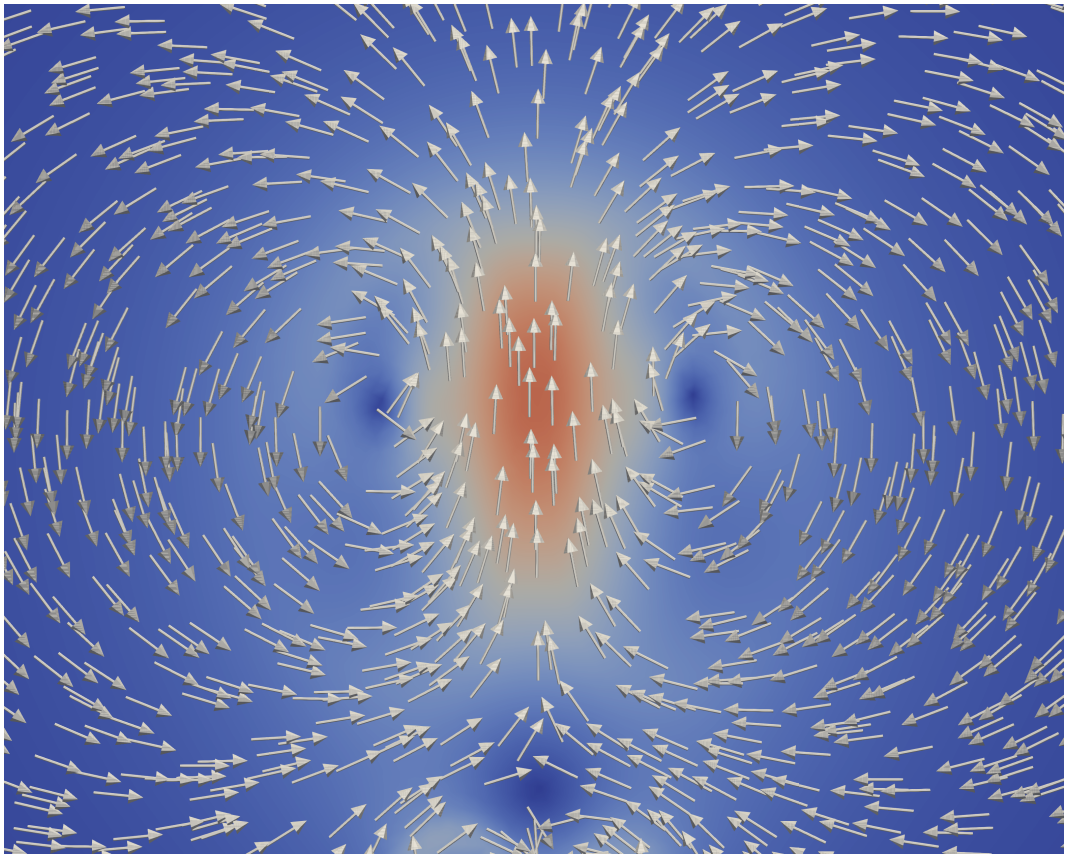

Figure 2.3: The figure illustrates direction of flows around a vortex-ring pair at time $t = 3s$ on the most detailed mesh (1.3M dof). The flow is symmetric with respect to the vertical axis of the nozzle of the jet.

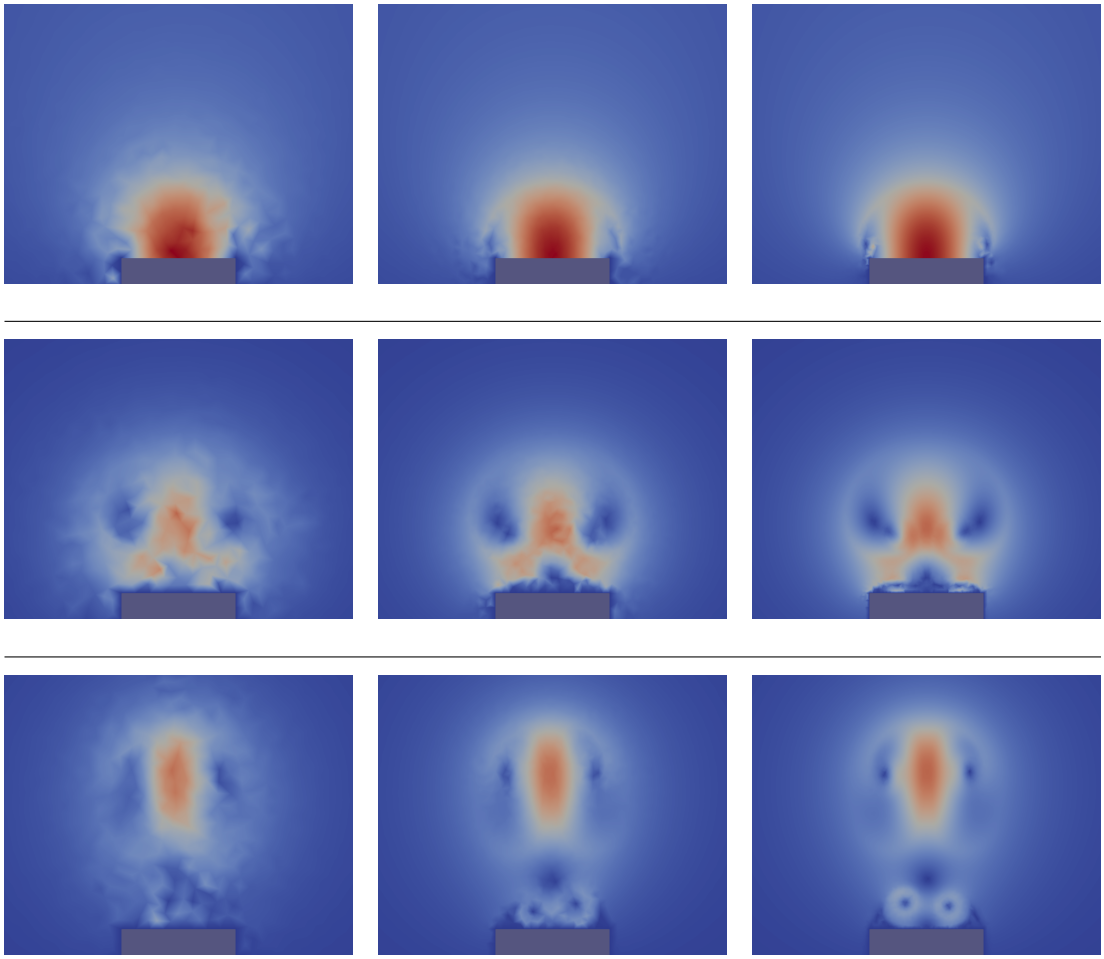

Figure 2.4: Computed velocities($u_0 = 10mm/s$) for different resolutions of the 2D mesh. On the left 200k, middle 600k and on the right 1.3M dof. Top row represents time $t = 0.5s$, middle $t = 1.5s$ and bottom $t = 3s$.

Figure 2.5: Computed pressures($u_0 = 10mm/s$) at time t=3s for different resolution of the 2D mesh (again from left 200k,600k,1.3M dof).

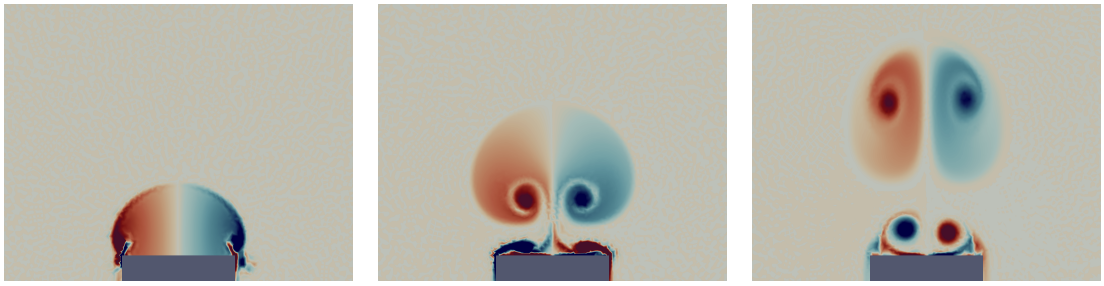

Figure 2.6: The vorticity at times (from left) $t = 0.5s, t = 1.5s, t = 3.0s$ for the most detailed 2D mesh. The red represents positive vorticity, while the blue represents areas with negative vorticity.

The interesting part of this numerical experiment is to compare performance of PCDR and LSC preconditioning against IPCS scheme and the direct solver MUMPS, their parallel scaling and the robustness with respect to the Reynolds number of the methods. The robustness is tested by changing the $u_0$ to a higher (lower) value. The only difference within the simulation is faster movement of the vortex ring. Both BRM1 and BRM2 versions of PCDR preconditioning were tested. The most simple version of the LSC preconditioning without the diagonal scaling (2.37) was used. The reason for using the simpler version of LSC is that it is implemented in PETSc. Even-though both preconditioning strategies were derived mainly for linearized approaches like Picard iteration, it was found that the preconditioners combined with Newton's method give much better performance for this problem, so only the results computed with Newton's method are summarized in the thesis. Note that the fixed time step requirement means that the projection method should be the fastest as the amount of (Krylov method) iterations needed to solve for one time step is usually lower than for fully coupled method for small time steps. If the fixed time step would not be a requirement an implementation of an adaptive time stepping scheme would yield the best comparison of the methods as the fully coupled approaches could make use of their better accuracy compared to the projection method.

The choice of proper solvers for the subproblems arising from the projection scheme or preconditioning requires fine tuning to get the best results. Let us briefly describe the setup for the projection scheme which has not been discussed before. The computation of the tentative velocity and the velocity update is done using the GMRES preconditioned by the SOR [DeLong and Ortega, 1995] mainly for its good scalability. We found that the (hypre) AMG works very well too for the flows with lower Reynolds number however it is a bit less robust with respect to Reynolds number. The pressure step (Laplace solve) is computed using the Conjugate Gradients (CG) [Liesen and Strakoš, 2012] preconditioned by (hypre) AMG.

The solution time is measured for each time step of the simulation. The sum of the solution times over all time steps is referred to as the computational time. The relative error of the computational times is estimated to be roughly 5%. The tables 2.7, 2.8 and Figure 2.9 capture computational times and strong scaling of all the methods on 200k dof and 600k dof meshes. First thing to notice is that the IPCS is by far the fastest method (as expected). It is about 5x faster than the fully coupled approaches, while giving the same results. The

fully coupled iterative approaches yield similar computational times regardless of the preconditioning type. It can be also seen that for higher inflow velocity, the direct solver tends to be faster especially for low number of cores. Overall it can be seen that the computational times increase with the increase of inflow velocity. The least affected is the direct solver MUMPS while the most affected preconditioner is the LSC. The robustness of the LSC preconditioner could be improved by using the scaled version of the preconditioning. Both versions of the PCDR preconditioning yield very similar result, however the BRM2 is a bit faster for more detailed meshes and small number of cores.

| #cores | BRM1 | BRM2 | LSC | MUMPS | IPCS |
|---|---|---|---|---|---|
| 2 | 8053 | 8010 | 5714 | 5623 | 1504 |
| 6 | 3139 | 3157 | 2372 | 2760 | 595 |
| 12 | 1595 | 1588 | 1322 | 1425 | 345 |
| 6 | 9799 | 9701 | 6606 | 8791 | 1525 |
| 12 | 4642 | 4312 | 3960 | 5006 | 912 |
| 18 | 3160 | 2754 | 2819 | 3447 | 588 |
| 24 | 2208 | 2455 | 2114 | 2591 | 515 |

Figure 2.7: Relation between computational time (in seconds) and number of cores used for $u_0 = 10mm/s$. Upper part of the table describes results for 200k dof mesh. Lower part captures the computational times for the 600k dof 2D mesh.

| #cores | BRM1 | BRM2 | LSC | MUMPS | IPCS |
|---|---|---|---|---|---|
| 2 | 8168 | 8396 | 7613 | 5737 | 1525 |
| 6 | 3122 | 3148 | 2530 | 2803 | 606 |
| 12 | 1617 | 1608 | 1545 | 1453 | 350 |
| 6 | 10949 | 9422 | 10526 | 9140 | 1848 |
| 12 | 6532 | 6173 | 6382 | 5331 | 1084 |
| 18 | 4372 | 4497 | 4403 | 3506 | 799 |
| 24 | 3061 | 3069 | 2871 | 2877 | 614 |

Figure 2.8: Relation between computational time (in seconds) and the number of cores used for $u_0 = 20mm/s$. Upper part of the table describes results for the 200k dof mesh. Lower part captures computational times for 600k dof 2D mesh.

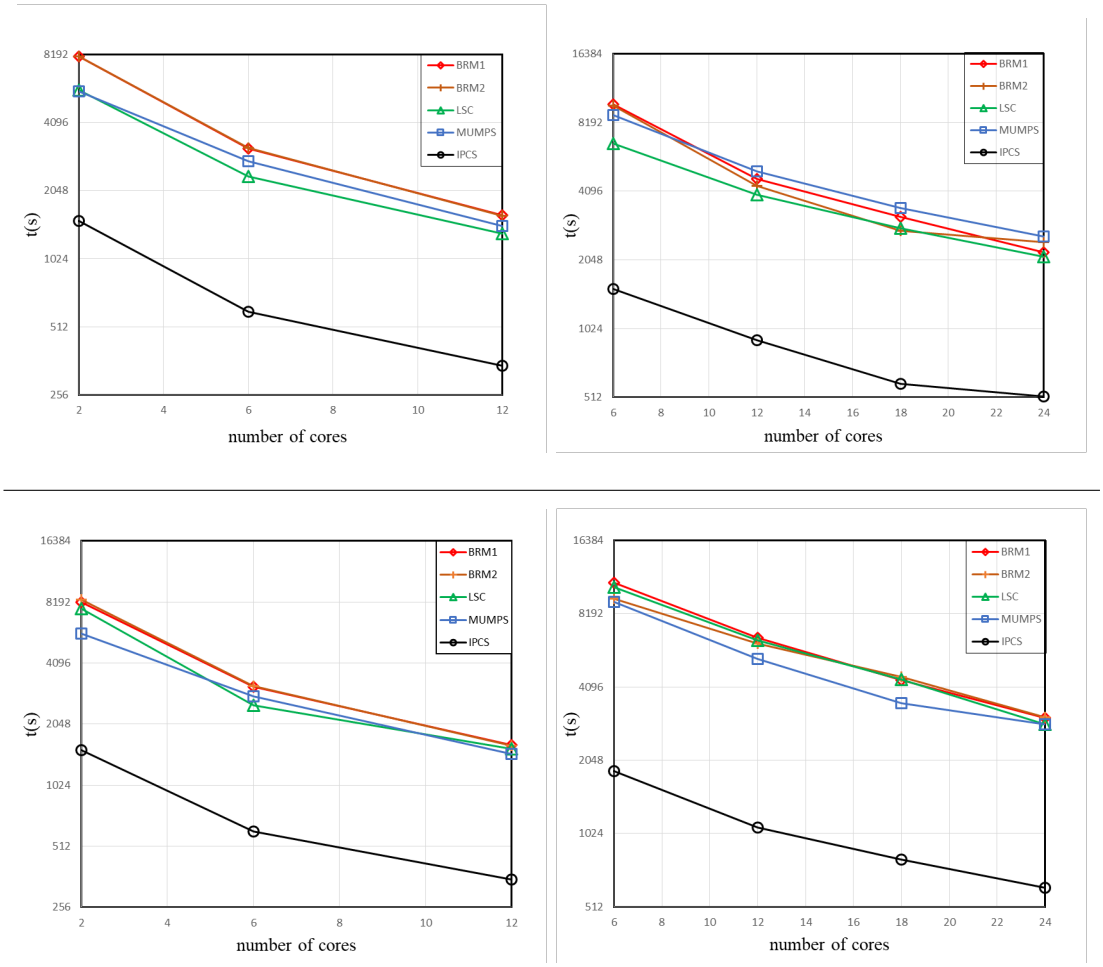


Figure 2.9: The computational times for $u_0 = 10mm/s$ are depicted on the top and $u_0 = 20mm/s$ on the bottom. Left pictures illustrate results for 200k dof 2D mesh, right results for 600k 2D dof mesh. The IPCS is by far the fastest method.

The advantage of the iterative solvers is their scalability. While the MUMPS does not scale much beyond 24 cores, the iterative solvers scale very well which makes them more suitable for larger computations which is described in the following 3D analogue of previous simulation.

In the 3D simulation, the height of the box is reduced to $70mm$, the width is reduced to $40mm$ to reduce the number of degrees of freedom. The radius and height of the nozzle represented by a cylinder remains the same. The simulation is consequently shorter (8s) to reduce the effects of the top wall on the vortex ring. The 3D mesh captures the geometry of the experiment quite accurately. Note that the 2D simulation is not consistent with the 3D simulation of the experiment because the 3D computational domain is (roughly) created by rotating the 2D computational domain around the axis of the nozzle.

The mesh is refined around the axis of the jet as depicted in Figure 2.10. One mesh yields together with Taylor-Hood elements approximately 1.2M dof and serves well for testing purposes and comparing the performance of the solvers. A finer meshes yielding about 4.5M, 8M and 13M dof were used to get more detailed results of the flow. The 3D simulation was conducted with the inflow velocity $u_0 = 10mm/s$ as more detailed mesh or additional stabilization is needed

to increase the inflow velocity (to $u_0 = 20mm/s$).

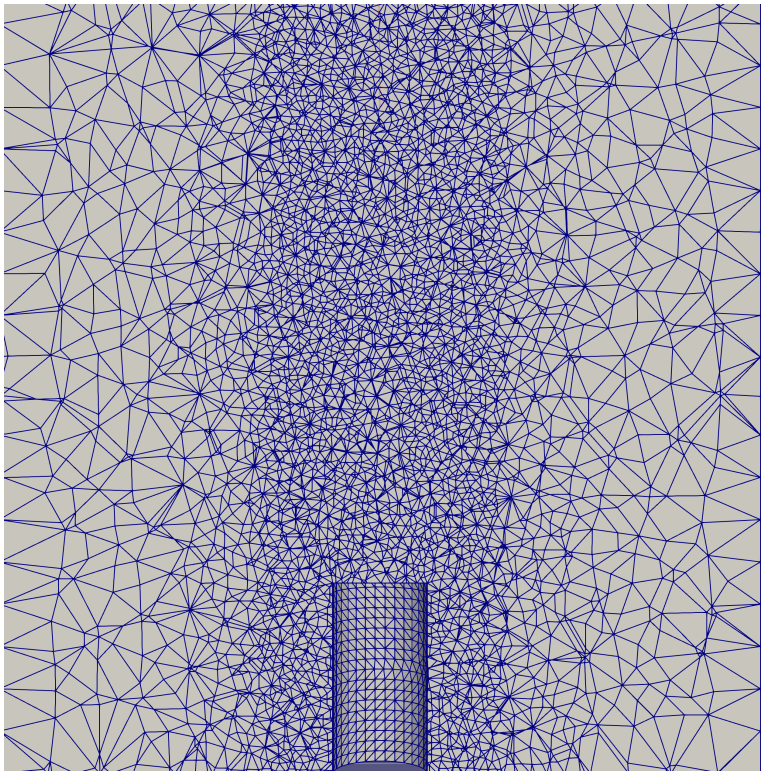

Figure 2.10: Lower part of the mesh (1.2M dofs) used for 3D simulation. The mesh is refined around the axis of the nozzle to get more detailed flow of the vortexring while keeping the number of degrees of freedoms as low as possible.

Both the projection method and fully coupled approaches yield practically the same results barely distinguishable by a simple comparison. The slice trough a vortex ring pair is illustrated in the 2.11 and verifies that the flow patterns are similar as in 2D eventhough the models are not consistent. The evolution of the vortex ring in 3D illustrated in the Figure 2.12.The simulation suggest that the vortices are bigger in the 2D simulation (Fig. 2.3) when compared to 3D results (Fig. 2.11). While it has been observed that the velocity of the vortex ring pair is roughly the same for both 3D and 2D simulation after time $t = 3s$, the previous figures suggest that the vortex ring in 3D is faster for the initial $3s$.

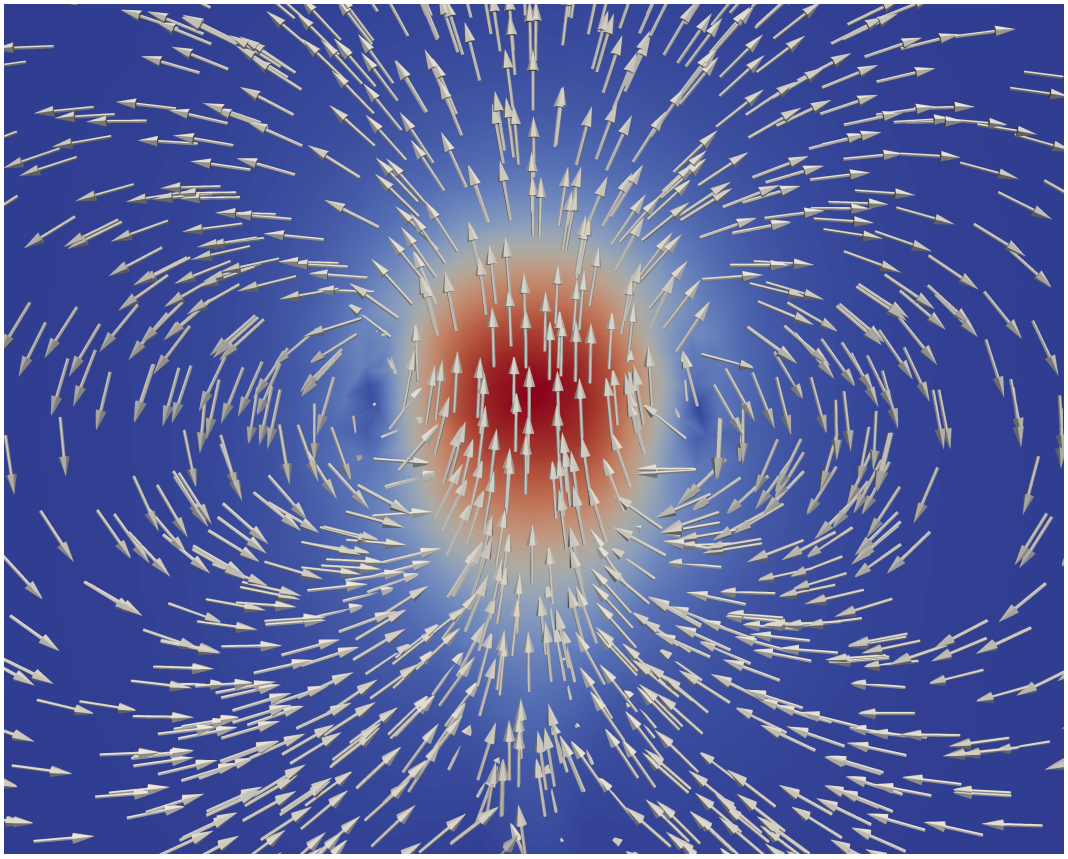

Figure 2.11: Flow patterns in the plane $x = 0$ of a vortex ring at time $t = 3s$ in 3D on the detailed mesh (13M dofs). The main features of the flow in the plane $x = 0$ are similar as the flows obtained in 2D simulations.

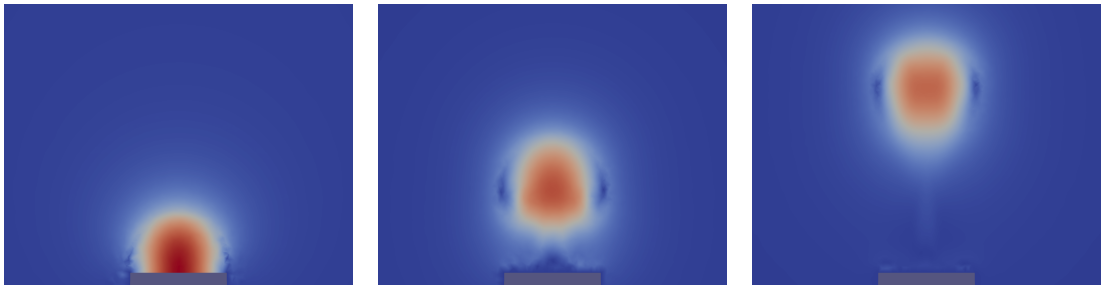

Figure 2.12: Computed velocities in the plane $x = 0$ ($u_0 = 10mm/s$) at times (from left) $0.5s$,$1.5s$ and $3s$ for the fine 3D mesh (13M dofs).

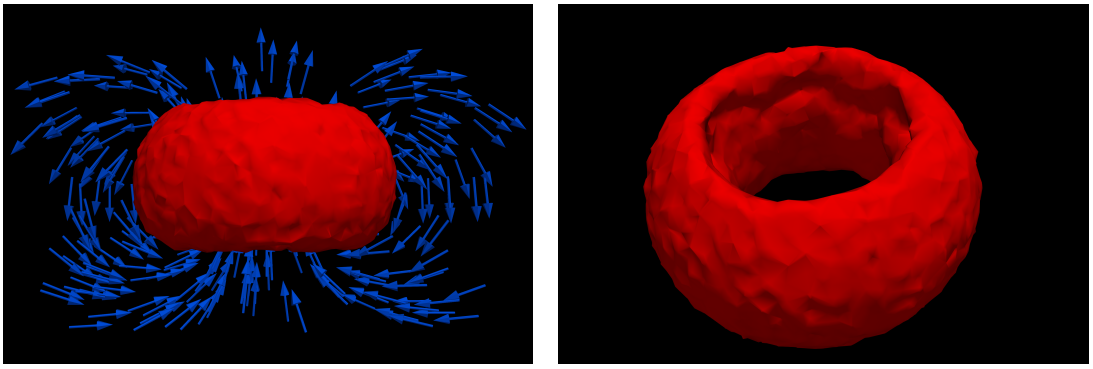

Figure 2.13: The isosurface of the magnitude of the vorticity in the shape of a torus (ring) at time $t = 4s$ (13M dofs). The arrows on the left represent the direction of the flow.

The performance of the direct solver MUMPS is so low in this 3D setup that it fails to give any results within reasonable time. The performance of the iterative solvers is visible in the table 2.14. The projection method is the fastest method and is roughly 6x faster compared to the fully coupled approaches. From these results, the projection method IPCS was chosen as the solver for the detailed 4.5M, 8M and 13M dofs simulation.

Both versions of the PCDR preconditioner yield very similar performance, but it seems that for this example, the BRM2 is slightly faster. One can see that while the LSC is faster for number of cores 36 and 72 the PCDR BRM2 preconditioner scales better with more cores and is as efficient as the LSC when run on 108 cores.

| #cores | BRM1 | BRM2 | LSC | IPCS |
|---|---|---|---|---|
| 36 | 13095 | 12567 | 10090 | 2128 |
| 72 | 7326 | 7332 | 6820 | 1091 |
| 108 | 5552 | 5162 | 5114 | 876 |

Figure 2.14: Computational times (in seconds) for the 3D problem on a less detailed mesh (1.2M dofs).

The weak scaling of the method is illustrated in the figure 2.15. The test was carried out on 4 3D meshes with 1.2M dof, 4.5M dof, 8M dof and 13M dof. The large initial increase is partially caused by slower internodal communication (if the 1.2M dof computation is run on 2 nodes 18 core each instead 1 node 36 cores then the computational time increases to $2370s$). Further weak scalability shows positive results as the computational time remains (almost) constant.

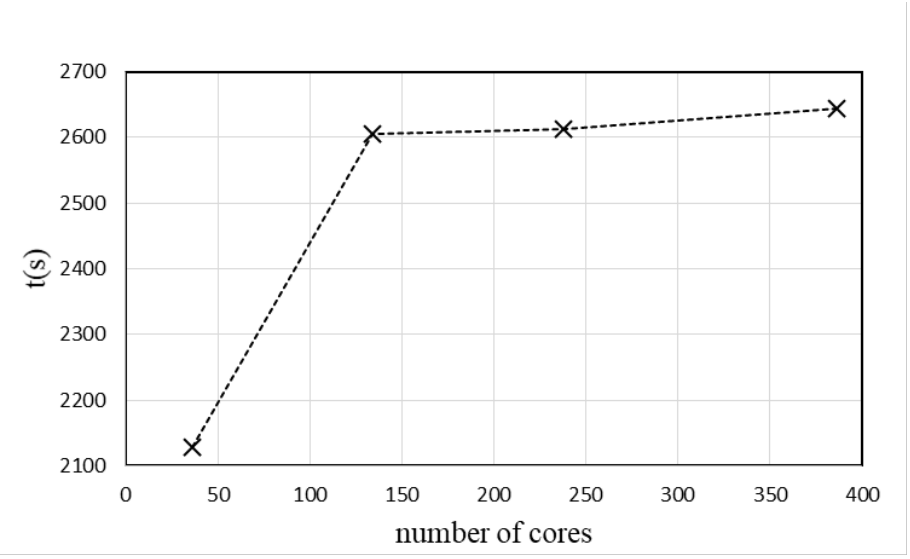


Figure 2.15: Weak scaling of the IPCS (cca 34k dof per core, 1.2M dof, 4.5M dof, 8M dof and 13M dof). The initial increase is caused by running computations on more than one node which decreases performance. The method scales very well with more degrees of freedom.

## 2.4 Pseudovorticity

This section will shortly introduce the pseudovorticity, its relation to vorticity and how it can be used to identify the vortex rings (vortices) in the simulations and the experiments. While the complete velocity and vorticity field can be obtained from the simulations, it is not obtainable from the experiments where typically only Lagrangian data (positions of particles) can be tracked. The pseudovorticity is a quantity which allows to estimate strength of the (macroscopic) vortices from the positions of tracked particles. More details about the relationship of vorticity and pseudovorticity and their ability to track various properties of the vortex rings will be hopefully published later this year [Outrata et al.].

The pseudovorticity $\mathbf{T}$ is defined as

$$\mathbf{T}(\mathbf{x}, t) = \frac{1}{N} \sum_{i=1}^{N} \frac{(\mathbf{x_i} - \mathbf{x}) \times \mathbf{u_i}}{(\mathbf{x} - \mathbf{x_i})^2} \tag{2.40}$$

where $N$ is the number of particles with position $\mathbf{x}_i$ and velocity $\mathbf{u}_i$ found inside of an annulus with inner diameter $R_{in}$ and out diameter $R_{out}$ and within $10ms$ time window centered in $t$ and placed side by side. It can be shown that under certain assumptions (such as massless particles, smooth velocity field, see [Švančara et al., 2020]) it holds that

$$\mathbf{T} = \frac{1}{2} \nabla \times \mathbf{u} \tag{2.41}$$

in some limit sense.

For our purposes $R_{out} = 3mm$ is used which roughly relates to size of the single vortex. $R_{in} = 10^{-4}mm$ was used to prevent computations with particles located exactly at the mesh grid points. The particles are in our case massless, and their

velocity at a given point corresponds to velocity of the flow at the given point. These particles are put in the simulation at time $t = 1s$ near the nozzle such that there are as many particles inside of the vortices as possible. Initial position of the particles in velocity field is illustrated in the Figure 2.16. Software Paraview and their function ParticleTracer was used for this purpose. The pseudovorticity was evaluated on a square mesh. The vorticity was interpolated on the same mesh. The values of the pseudovorticity depend on the number of particles if there is not enough particles. It has been found, that if the number of particles is 10k or more, the pseudovorticity does not change with more particles. In this simulation, 22.5k particles placed as in the Figure 2.16 were used for the computation of pseudovorticity. The 1.3M dof 2D simulation was used. The comparison of the pseudovorticity and vorticity can be done in 3D as well but it is very computationally intensive.

An additional simulation where 5.5k particles are distributed evenly troughout the computational domain is also considered and referred to as the pseudovorticity - experiment. This distribution of particles is motivated by experiment and is similar to what would be observed in the experiment. The particles are put into the fluid a time $t = 0s$ on a rectangle of dimension $20mm \times 79mm$ centered above the nozzle. For this distribution of particles the parameters $R_{out} = 5mm$ and $R_{in} = 1mm$ were used (as for the experiment [Švančara et al., 2020]).

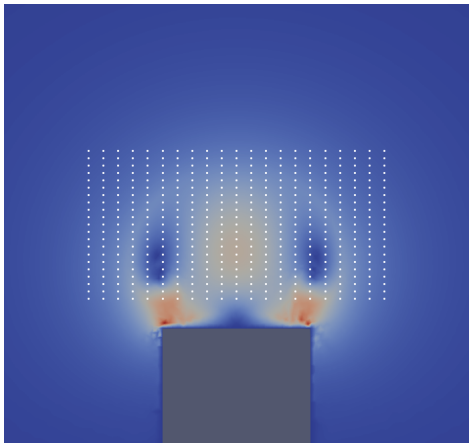

Figure 2.16: Initial position of particles at time $t = 1s$ for computation of pseudovorticity. The dimension of the particle array is $10mm \times 5mm$.

### 2.4.1 Results-pseudovorticity vs vorticity

Following section describes how the pseudovorticity can be used to track certain properties (position and velocity) of the vortex ring and how it compares to tracking the vortex ring by the vorticity.

We want to focus only on fully developed vortex rings so times lower than $t = 4s$ are neglected. The Figure 2.17 shows comparison of pseudovorticity and vorticity for the vortex rings simulation. Vortex ring is obtained by applying the threshold of approximately 10% of a local maximum of the pseudovorticity at time $t = 10s$ denoted by $\mathbf{T}_{max}$ (because the simulation is 2D, the pseudovorticity has only one nonzero component so the maximum makes sense). This means,

all mesh points that meet $T > 0.1\mathbf{T}_{max}$ are assigned to the positive (clockwise) vortex, while the points that meet $\mathbf{T} < 0.1\mathbf{T}_{max}$ are linked with the negative vortex. The centre of the positive and negative vortices are then $\mathbf{T}$ weighted centres of these points. The position of a vortex ring is defined as $\mathbf{T}$ weighted average of the position of the 2 vortices. Vortex ring measured by the vorticity are obtained in the same way. The threshold applied is again approximately 10% of the maximum of vorticity at time $t = 10s$. The vorticity in the close viscinity of the nozzle is not taken in account ( as can be seen in the Figures 2.4 and 2.6 , there are small vortices near the nozzle for later times, however these are of no interest to us)

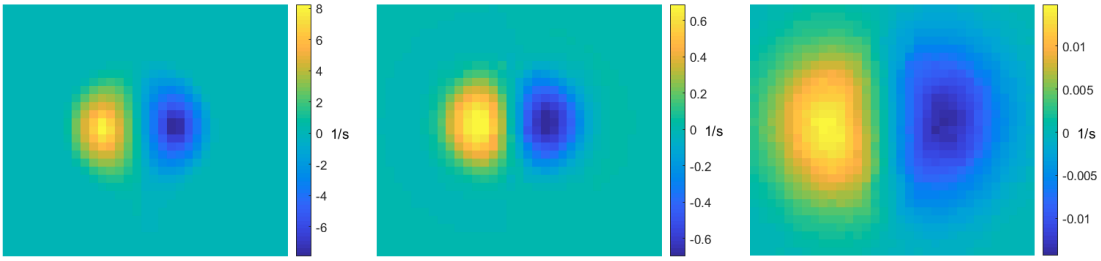


Figure 2.17: The vorticity (left), pseudovorticity (middle) and pseudovorticity - experiment (right) of the vortex ring at time $t = 11s$.

### 2.4.2 Position

Vertical position of a vortex ring is tracked accurately by the pseudovorticity if compared to the vorticity as shown in the Figure 2.18. The difference between the two is almost negligible. Pseudovorticity - experiment captures the position of the vortex accurately as well.

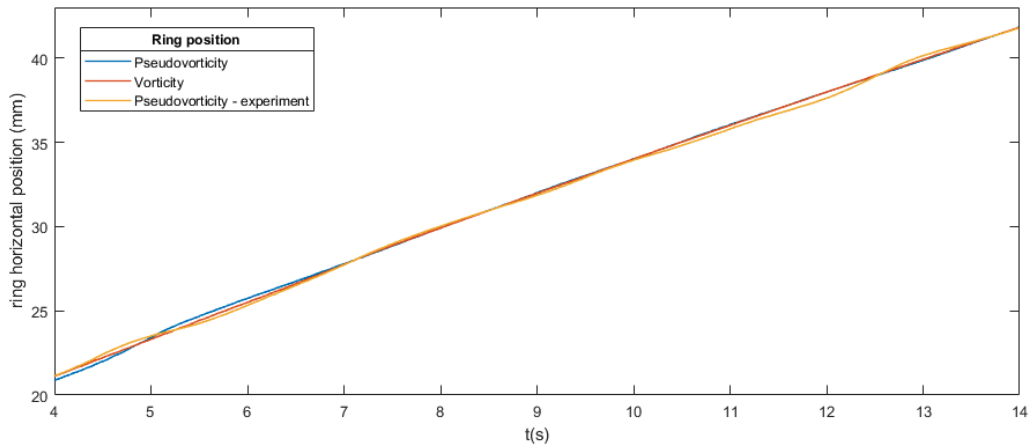


Figure 2.18: Vortex ring position vs time.

### 2.4.3 Velocity

One can track the velocity of a vortex ring by both pseudovorticity and vorticity by a convolution of the position of the vortex ring with a suitable Gaussian kernel (as in [Švančara et al., 2020]). As demonstrated in the Figure 2.19, the decreasing trend is similar for both vorticity and pseudovorticity, especially for $t > 7s$ the pseudovorticity can capture the vortex ring velocity quite well if compared to vorticity, however, there are slightly bigger oscillations in the velocity of vortex ring tracked by pseudovorticity. The oscilations are much bigger for the pseudovorticity - experiment simulation.

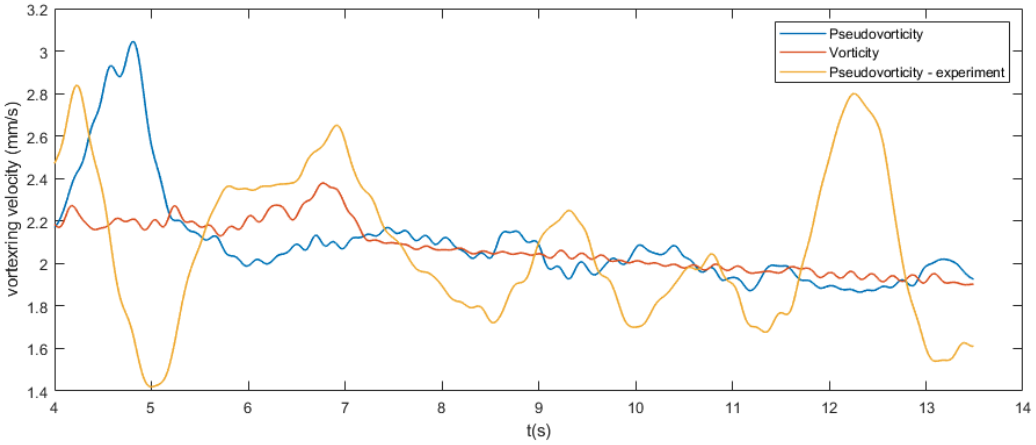


Figure 2.19: Vortex ring velocity vs time.

# 3. Fictitious Boundary Method and Arbitrary Lagrangian Eulerian Formulation

Two ways of dealing with moving boundaries are presented in this chapter. The first is called the Fictitious Boundary method (FIB) and the second is called the Arbitrary Lagrangian Eulerian (ALE) formulation of the Navier-Stokes equations.

Generally, there are two main approaches to solving problems on a time-dependent domain, either transform it into some fixed reference domain and solve the transformed problem and transform the solution back or use some solution method in the physical time-dependent domain. For example consider the first problem described in the next chapter, that is an moving cylinder in a rectangular domain. The Figure 4.1 illustrates the physical time-dependent domain where the cylinder moves trough the domain. The reference domain can be for example the domain with the cylinder located exactly in the middle.

The first approach usually includes some ALE-type methods such as [Lozovskiy et al., 2018]. The class of the solution methods in the physical domain includes methods such as the FIB [Turek et al., 2003], CutFEM [Burman et al., 2015], penalty methods [Shirokoff and Nave, 2015], full space-time FEM [Mittal et al., 1991], Lagrange Multipliers [Glowinski et al., 2001].

In the following section, the FIB will be presented as one of the simplest solution techniques in the physical domain.

## 3.1 Fictitious Boundary Method

The main idea of the FIB method is to generate one mesh which already contains most of the geometrical details and then apply the classic solution techniques and enforce the boundary conditions on the algebraic level. The idea is well summed up in [Turek et al., 2003]:

> *"Employ a (rough) boundary parametrization which sufficiently describes all large-scale structures with regard to the boundary conditions. Treat all fine-scale features as interior objects such that the corresponding components in all matrices and vectors are unknown degrees of freedom which are implicitly incorporated into all iterative solution steps. Hence, standard tools for grid refinement in interior regions are easily applicable and highly accurate approximations can be obtained."*

The treatment of the Dirichlet boundary conditions is as follows. Use the expansion (1.10) and assemble the nonlinear system of equations of the form

$$\mathbf{A}_h(\mathbf{U}_h) = \mathbf{B}_h \tag{3.1}$$

for unknown degrees of freedom $\mathbf{U}_h$ with the operator $\mathbf{A}_h$ and right hand side $\mathbf{B}_h$. Mark the degrees of freedom corresponding to the boundary (for example

if you have a moving object occupying part of the domain $\Omega_c$, then the dofs corresponding to the boundary of the object are the ones located inside or on the boundary of $\Omega_c$). In each time step replace the rows corresponding to the boundary of $\mathbf{A}_h$ by identity. Prescribe the given Dirichlet values for all the components of the vectors $\mathbf{U}_h, \mathbf{B}_h$ which belong to the boundary. This approach is called the *semi-implicit boundary treatment* and most of the software PDE solvers use it to implement Dirichlet boundary conditions. More about this method, convergence and error estimates can be found in [Turek et al., 2003].

## 3.2 The Arbitrary Lagrangian Eulerian Formulation

The second option of dealing with the time-dependent domain described in this thesis is the Arbitrary Lagrangian Eulerian formulation. The formulation chosen in this thesis comes up naturally from the weak formulation and its linearized version is documented in [Lozovskiy et al., 2018] (they refer to the method as quasi-Lagrangian, however we still prefer to call it an ALE method). While there exist other ALE formulations such as [Zhu and Yan, 2019] or [Duarte et al., 2004], their implementation is more complicated for the problems considered in the thesis.

Consider the Navier-Stokes equations in the time-dependent domain (homogeneous Dirichlet BC for simplicity) $\Omega(t)$, that is

$$\begin{aligned}(\frac{\partial \mathbf{u}}{\partial t}, \mathbf{v})_{\Omega(t)} + (\mathbf{u}\cdot\nabla\mathbf{u}, \mathbf{v})_{\Omega(t)} + \nu(2\mathbb{D}, \nabla\mathbf{v})_{\Omega(t)} - (p, \nabla\cdot\mathbf{v})_{\Omega(t)} = 0 \quad \forall \mathbf{v}\in\mathbb{X} \\ (\nabla\cdot\mathbf{u}, q)_{\Omega(t)} = 0 \quad \forall q \in Q\end{aligned} \tag{3.2}$$

where $\mathbb{X}$ and $Q$ are corresponding function spaces. Consider domain $\Omega_0$ and mapping $\chi : \Omega_0 \times [0,T] \to \Omega(t)$, such that the mapping is a bijection and is smooth enough. The main idea of this method is to transform the previous equations into the fixed domain $\Omega_0$ and compute the solution there. The coordinates in the physical domain $\Omega$ are denoted $\mathbf{x}$ and coordinates in $\Omega_0$ are denoted $\mathbf{X}$. Let us define

$$\begin{aligned}\mathbf{F}(\mathbf{X},t) &= \nabla\chi(\mathbf{X},t) \\ J(\mathbf{X},t) &= \det(\mathbf{F}(\mathbf{X},t)) \\ \tilde{\mathbf{u}}(\mathbf{X},t) &= \mathbf{u}(\chi(\mathbf{X},t),t) \\ \tilde{p}(\mathbf{X},t) &= p(\chi(\mathbf{X},t),t).\end{aligned} \tag{3.3}$$

Then it follows that (the gradients are always taken with respect to the variable of the function, i.e. $\nabla\tilde{\mathbf{u}}(\mathbf{X},t) = \frac{\partial\tilde{\mathbf{u}}}{\partial\mathbf{X}}, \nabla\mathbf{u}(\mathbf{x},t) = \frac{\partial\mathbf{u}}{\partial\mathbf{x}}$)

$$\begin{aligned}\nabla\tilde{\mathbf{u}}(\mathbf{X},t) &= \nabla\mathbf{u}(\mathbf{x},t)|_{\mathbf{x}=\chi(\mathbf{X},t)}\mathbf{F}(\mathbf{X},t) \\ \frac{\partial\tilde{\mathbf{u}}(\mathbf{X},t)}{\partial t} &= \frac{\partial\mathbf{u}(\mathbf{x},t)}{\partial t}|_{\mathbf{x}=\chi(\mathbf{X},t)} + \nabla\mathbf{u}(\mathbf{x},t)|_{\mathbf{x}=\chi(\mathbf{X},t)}\tilde{\mathbf{u}}_m\end{aligned} \tag{3.4}$$

where $\tilde{\mathbf{u}}_m$ denotes $\frac{\partial\chi(\mathbf{X},t)}{\partial t}$ and represents the velocity of the transformation of the

domain. This yields the following set of equation in the reference domain

$$\begin{aligned}(\frac{\partial\tilde{\mathbf{u}}}{\partial t},\tilde{\mathbf{v}}J)_{\Omega_0}+((\tilde{\mathbf{u}}-\tilde{\mathbf{u}}_m)\cdot\nabla\tilde{\mathbf{u}}\mathbf{F}^{-1},\tilde{\mathbf{v}}J)_{\Omega_0}\\ +\nu(\nabla\tilde{\mathbf{u}}\mathbf{F}^{-1}+(\nabla\tilde{\mathbf{u}}\mathbf{F}^{-1})^T,\nabla\tilde{\mathbf{v}}\mathbf{F}^{-1}J)_{\Omega_0}-(\tilde{p},\mathrm{Tr}(\nabla\tilde{\mathbf{v}}\mathbf{F}^{-1})J)\Omega_0=0\quad\forall\mathbf{v}\in\mathbb{X}\\ (\mathrm{Tr}(\nabla\tilde{\mathbf{u}}\mathbf{F}^{-1}),qJ)\Omega_0=0\quad\forall q\in Q.\end{aligned}\tag{3.5}$$

If the transformation $\chi$ is smooth enough, one can interchange the spaces $\mathbb{X}$ and $Q$ for their counterparts in the reference domain $\tilde{\mathbb{X}}$ and $\tilde{Q}$. That yields the following set of equation on the reference domain to be solved.

$$\begin{aligned}(\frac{\partial\tilde{\mathbf{u}}}{\partial t},\mathbf{v}J)_{\Omega_0}+((\tilde{\mathbf{u}}-\tilde{\mathbf{u}}_m)\cdot\nabla\tilde{\mathbf{u}}\mathbf{F}^{-1},\mathbf{v}J)_{\Omega_0}+\\ \nu(\nabla\tilde{\mathbf{u}}\mathbf{F}^{-1}+(\nabla\tilde{\mathbf{u}}\mathbf{F}^{-1})^T,\nabla\mathbf{v}\mathbf{F}^{-1}J)_{\Omega_0}-(\tilde{p},\mathrm{Tr}(\nabla\mathbf{v}\mathbf{F}^{-1})J)_{\Omega_0}=0\quad\forall\mathbf{v}\in\tilde{\mathbb{X}}\\ (\mathrm{Tr}(\nabla\mathbf{v}\mathbf{F}^{-1}),qJ)_{\Omega_0}=0\quad\forall q\in\tilde{Q}\end{aligned}\tag{3.6}$$

Note that the mesh velocity $\tilde{\mathbf{u}}_m$ is independent of the velocity $\tilde{\mathbf{u}}$. The choice of the mapping $\chi$ and the reference domain is completely arbitrary as long as mapping $\chi$ is a bijection and smooth.

These equations can be again discretized using the Finite Element Method and $\theta$ time stepping scheme. For simplicity only the implicit Euler version is described. Since the Jacobian depends on time as well, one more term arises in the formulation. The fully discretized problem reads (by $\mathbf{F}$ it is always meant $\mathbf{F}^{k+1}$ to simplify the notation)

$$\begin{aligned}(\frac{\tilde{\mathbf{u}}_h^{k+1}J^{k+1}-\tilde{\mathbf{u}}_h^kJ^k}{dt}-\frac{\partial J}{\partial t}^{k+1}\tilde{\mathbf{u}}_h^{k+1},\mathbf{v})_{\Omega_0}\\ +((\tilde{\mathbf{u}}_h^{k+1}-\tilde{\mathbf{u}}_{m,h}^{k+1})\cdot\nabla\tilde{\mathbf{u}}_h^{k+1}\mathbf{F}^{-1},\mathbf{v}J^{k+1})_{\Omega_0}+\\ \nu(\nabla\tilde{\mathbf{u}}_h^{k+1}\mathbf{F}^{-1}+(\nabla\tilde{\mathbf{u}}_h^{k+1}\mathbf{F}^{-1})^T,\nabla\mathbf{v}\mathbf{F}^{-1}J^{k+1})_{\Omega_0}\\ -(\tilde{p},\mathrm{Tr}(\nabla\mathbf{v}\mathbf{F}^{-1})J^{k+1})_{\Omega_0}=0\quad\forall\mathbf{v}\in\tilde{\mathbb{X}}_h\\ (\mathrm{Tr}(\nabla\tilde{\mathbf{u}}_h^{k+1}\mathbf{F}^{-1}),qJ^{k+1})_{\Omega_0}=0\quad\forall q\in\tilde{Q}_h.\end{aligned}\tag{3.7}$$

Generally, the time derivative of the Jacobian is not available, however for the examples considered in this thesis the time derivative is explicitely known so no approximations to the derivative have to be made. The corresponding Dirichlet boundary conditions just simply transform to the reference mesh. Stability of the linearized version of scheme the scheme can be found in [Lozovskiy et al., 2018].

The projection IPCS can be also modified to the IPCS-ALE ( fully coupled formulation is reffered to only as ALE). The formulation is analogical to classic IPCS pulled back to the reference domain (for simplicity only implicit Euler version). Initially, the tentative velocity is computed from the equation

$$\begin{aligned}(\frac{\tilde{\mathbf{u}}_h^tJ^{k+1}-\tilde{\mathbf{u}}_h^kJ^k}{dt},\mathbf{v})_{\Omega_0}-(\frac{\partial J}{\partial t}^{k+1}\mathbf{u}_h^{k+1},\mathbf{v})_{\Omega_0}+((\tilde{\mathbf{u}}_h^t-\tilde{\mathbf{u}}_{m,h}^{k+1})\cdot\nabla\tilde{\mathbf{u}}_h^k\mathbf{F}^{-1},\mathbf{v}J^{k+1})_{\Omega_0}\\ +2\nu(\nabla\tilde{\mathbf{u}}_h^{k+1}\mathbf{F}^{-1}+(\nabla\tilde{\mathbf{u}}_h^{k+1}\mathbf{F}^{-1})^T,\nabla\mathbf{v}\mathbf{F}^{-1}J^{k+1})_{\Omega_0}\\ =(\tilde{p}_h^k,\mathrm{Tr}(\nabla\mathbf{v}\mathbf{F}^{-1})J^{k+1})_{\Omega_0}\quad\forall\mathbf{v}\in\mathbb{H}_0^1(\Omega_0).\end{aligned}\tag{3.8}$$

After obtaining the tentative velocity, the new pressure $\tilde{p}^{k+1}$ is computed using

$$(\nabla \mathbf{F}^{-1}\tilde{p}_h^{k+1}, \nabla \mathbf{F}^{-1} q J^{k+1})_{\Omega_0} = (\frac{Tr(\nabla \tilde{\mathbf{u}}_h^t \mathbf{F}^{-1})}{\mathrm{d}t}, q J^{k+1})_{\Omega_0} + (\nabla \mathbf{F}^{-1}\tilde{p}_h^k, \nabla \mathbf{F}^{-1} q J^{k+1})_{\Omega_0} \quad \forall q \in L^2(\Omega_0). \quad (3.9)$$

The final velocity $\tilde{\mathbf{u}}_h^{k+1}$ is obtained by correcting the tentative velocity using the new pressure.

$$(\tilde{\mathbf{u}}_h^{k+1}, \mathbf{v} J^{k+1})_{\Omega_0} = (\tilde{\mathbf{u}}_h^t, \mathbf{v} J^{k+1})_{\Omega_0} - \mathrm{d}t(\nabla \mathbf{F}^{-1}(p_h^{k+1} - p_h^k), \mathbf{v} J^{k+1})_{\Omega_0} \quad \forall \mathbf{v} \in \mathbb{H}_0^1(\Omega_0) \quad (3.10)$$

The boundary equations for these equations are analogical with the standart IPCS method.

## 3.3 Special Boundary Conditions and Stabilization

### 3.3.1 Boundary Conditions

The simulation of the experiments requires special outflow boundary conditions to allow the vortices to leave the domain without ruining the computation. The boundary condition used in this thesis [Arndt et al., 2016] are called Directional Do-Nothing (DDN) boundary condition and mimic the classical traction free boundary condition but lead to an enhanced stability.

Consider part of the domain $\partial\Omega^N$ where a traction free boundary condition is employed, that is

$$((-p\mathbb{I} + 2\nu\mathbb{D})\mathbf{n}, \mathbf{v})_{\partial\Omega^N} = 0 \quad (3.11)$$

The Directional Do-Nothing boundary condition can be written as

$$(-p\mathbb{I} + \nu\nabla\mathbf{u})\mathbf{n} = \frac{1}{2}(\mathbf{u}\cdot\mathbf{n})_-\mathbf{u} \text{ on } \partial\Omega^N, \quad (3.12)$$

where $(\mathbf{u}\cdot\mathbf{n})_- = 0$ if $(\mathbf{u}\cdot\mathbf{n}) \geq 0$ and $(\mathbf{u}\cdot\mathbf{n})_- = \mathbf{u}\cdot\mathbf{n}$ if $(\mathbf{u}\cdot\mathbf{n}) \leq 0$. The boundary conditions for the pure outflow turn into classic traction free conditions. Since the divergence form of Navier-Stokes equations is used, then

$$((-p\mathbb{I} + 2\nu\mathbb{D})\mathbf{n}, \mathbf{v})_{\partial\Omega^N} = (\frac{1}{2}(\mathbf{u}\cdot\mathbf{n})_-\mathbf{u} + \nu(\nabla\mathbf{u})^T\mathbf{n}, \mathbf{v})_{\partial\Omega^N}. \quad (3.13)$$

When the ALE formulation is used, the boundary conditions still have to be mapped into the reference domain (suppose that the boundary $\partial\Omega^N$ is mapped to $\partial\Omega_0^N$), that is

$$((-\tilde{p}\mathbb{I} + \nu(\nabla\tilde{\mathbf{u}}\mathbf{F}^{-1} + (\nabla\tilde{\mathbf{u}}\mathbf{F}^{-1})^T))\mathbf{F}^{-T}\mathbf{n}_0, \mathbf{v}J)_{\partial\Omega_0^N} = (\frac{1}{2}(\tilde{\mathbf{u}}\cdot\mathbf{F}^{-T}\mathbf{n}_0)_-\tilde{\mathbf{u}} + \nu(\nabla\tilde{\mathbf{u}}\mathbf{F}^{-1})^T\mathbf{F}^{-T}\mathbf{n}_0, \mathbf{v}J)_{\partial\Omega_0^N}. \quad (3.14)$$

where $\mathbf{n}_0$ is the normal vector in the reference configuration to the boundary $\partial\Omega_0^N$.

These boundary conditions are not suitable for the projection scheme as they are not linear in velocity. We did not manage to find better outflow boundary conditions for the projection scheme, thus the projection method will not be tested in the simulation of the experiment.

### 3.3.2 Stabilization

When dealing with flows reaching high Reynolds numbers an additional stabilization often has to be introduced to add "regularity" into the equations. One possible option is to use interior penalty type of stabilization ([Burman et al., 2004],[Schott, 2017]) or some older and well known approaches like Streamline Upwind Petrov-Galerkin (SUPG) [Brooks and Hughes, 1982]. In this thesis, the SUPG stabilization is used because the transformation to the reference domain is more straight-forward than the transformation of the interior-penalty type stabilizations and gave overall better results. For the FIB method, both approaches yielded similar results. The classical SUPG stabilization adds to the LHS of the weak formulation (implicit Euler is used to simplify the notation) (1.9) term

$$\sum_{K\in\Omega_h} \tau_K((\nabla \mathbf{v})\mathbf{u}_h^{k+1}, \mathbf{R})_K \tag{3.15}$$

where $\mathbf{R}$ is the strong residual

$$\mathbf{R} = \frac{\mathbf{u_h^{k+1}} - \mathbf{u}_h^k}{\mathrm{d}t} + \nabla p_h^{k+1} + ((\nabla \mathbf{u}_h^{k+1})\mathbf{u}_h^{k+1} - 2\nu \mathrm{div}(\mathbb{D}_h^{k+1}))) \tag{3.16}$$

and $\tau_K$ is the stabilization parameter projected on element $K$ defined later.

The version of the SUPG stabilization for similar ALE formulation can be found for example in [Zhu and Yan, 2019]. It corresponds to adding the following term to the LHS of the weak ALE formualation (3.7)

$$\sum_{K\in\Omega_h} \tau_K((\nabla \mathbf{v}\mathbf{F}^{-1})(\tilde{\mathbf{u}}_h^{k+1} - \tilde{\mathbf{u}}_{m,h}^{k+1}), \mathbf{R}_{ALE})_K \tag{3.17}$$

where $\mathbf{R}_{ALE}$ is the strong residual of the transformed equation, that is

$$\begin{aligned}\mathbf{R}_{ALE}^{k+1} = &\frac{\tilde{\mathbf{u}}_h^{k+1}J^{k+1} - \tilde{\mathbf{u}}_h^k J^k}{\mathrm{d}t} - \frac{\partial J^{k+1}}{\partial t}\tilde{\mathbf{u}}_h^{k+1} \\ &(\tilde{\mathbf{u}}_h^{k+1} - \tilde{\mathbf{u}}_{m,h}^{k+1})\cdot\nabla\tilde{\mathbf{u}}_h^{k+1}\mathbf{F}^{-1}J^{k+1} + \nu\mathrm{div}((\nabla\tilde{\mathbf{u}}_h^{k+1}\mathbf{F}^{-1} + (\nabla\tilde{\mathbf{u}}_h^{k+1}\mathbf{F}^{-1})^T)\mathbf{F}^{-T}J^{k+1}) \\ &- \nabla\tilde{p}\mathbf{F}^{-1}J^{k+1}.\end{aligned} \tag{3.18}$$

The choice of the stabilization parameter is difficult and widely discussed topic [Tezduyar and Sathe, 2003], [John and Knobloch, 2008], [Tezduyar, 1991],[Tezduyar and Osawa, 2000]. We decided to use a form ($h$ denotes diameter of a cell)

$$\tau = \alpha((\frac{2}{\mathrm{d}t})^2 + (\frac{2|\mathbf{u}_h^k|}{h})^2 + 9(\frac{4\nu}{h^2})^2)^{-\frac{1}{2}} \tag{3.19}$$

where $\alpha > 0$ is an additional parameter which can be tuned to give the best results (typically $\alpha \leq 1$ ). It should be noted that that this coefficient has to be chosen carefully, big enough to stabilize the system but small enough to reduce artificial viscosity added to the system by the stabilization. The current magnitude of the velocity $\mathbf{u}_h^{k+1}$ is usually used in the coefficient instead of $\mathbf{u}_h^k$ but this choice makes a significant difference when using a Newton's method and ruins the quadratic convergence. The LSIC (Least Squares Incompressibility Constraint) (see [Tezduyar, 2002],[Zhu and Yan, 2019]) type of stabilization is often added together with SUPG as well. The stabilization reads, add

$$\sum_{K\in\Omega_h} \tau_K^{lsic}(\mathrm{div}(\mathbf{u}_h^{k+1}), \mathrm{div}(\mathbf{v}))_K \tag{3.20}$$

to the LHS of the weak formulation (3.7) where $\tau_K^{lsic}$ is the stabilization coefficient projected on the element $K$. The corresponding stabilization of the ALE formulation is

$$\sum_{K\in\Omega_h} \tau_K^{lsic}(\mathrm{Tr}((\nabla\tilde{\mathbf{u}}_h^{k+1})\mathbf{F}^{-1}), \mathrm{Tr}((\nabla\mathbf{v})\mathbf{F}^{-1})J^{k+1})_K. \tag{3.21}$$

The coefficient is chosen in the similar fashion as in [Zhu and Yan, 2019]

$$\tau_K^{lsic} = \frac{\alpha^2 h^2}{12\tau} \tag{3.22}$$

If the stabilization coefficient is too high, it adds too much dissipation into the system and leads to incorrect results.

# 4. Numerical Results - time-dependent Domains

This chapter of the thesis is focused on a simulation of the experiment conducted by [Duda et al., 2015]. The experimented has already been attempted to simulate in [Outrata, 2018] by the FIB method [Turek et al., 2003] with unsatisfactory results. The behavior reached by the FIB did not reach our expectations. The aim of this chapter is to get the desired behavior by the Arbitrary Lagrangian Eulerian (ALE) formulation of Navier-Stokes equations and compare them with the FIB. First of all, a benchmark test [Schott, 2017] is conducted to verify the methods. The second part of this chapter is focused at the simulation of the experiment.

## 4.1 Benchmark Test - Moving Cyllinder

The benchmark simulation of the moving cylinder problem is described and compared to the results obtained in [Schott, 2017] in this section. The benchmark test simulates flow over a moving cylinder immersed in an incompressible viscous fluid. The original benchmark test and results can be seen in [Codina et al., 2009]. The benchmark simulated in this thesis and in [Schott, 2017] differs from the original benchmark by a slightly asymmetric position of the cylinder with respect to vertical coordinate.

Consider rectangular domain $[0, 2.2] \times [0, 0.44]$ with cylinder of diameter $d = 0.2$ and the coordinates of its center $(x_c, y_c)$. The movement of the cylinder is prescribed to be

$$\begin{aligned} x_c(t) &= 1.1 - 0.8 \sin(\frac{2\pi}{3}(t - 0.75)) \\ y_c(t) &= 0.23. \end{aligned} \tag{4.1}$$

It can be seen, that at time $t = 3$ the cylinder is returned to its initial position. The viscosity of the fluid is $\nu = 0.001$. The no-slip boundary conditions are imposed on the cylinder, top, left and bottom wall. The traction free boundary condition is imposed at the right wall of the domain. The problem and boundary conditions are illustrated in the Figure 4.1. The function $\chi$ is defined as follows:

$$\chi(\mathbf{X}, t) = \mathbf{X} + \begin{pmatrix} k(t) \\ 0 \end{pmatrix} * f(X_1) \tag{4.2}$$

where $k(t) = -0.8 \sin(\frac{2\pi}{3}(t - 0.75))$ and $f(X_1)$ ($X_1$ denotes x coordinate in reference domain) is a smooth ($C^1$) function illustrated in the Figure 4.2. This function is constructed using piece-wise linear functions combined with cubic polynomials to make $f$ smooth. Note that the domain is deformed only in the $x$ direction while there is no shift in the $y$ direction and that the reference configuration corresponds to the location of the cylinder in the middle of the domain.

Measured quantities are the density of the force $(f_1, f_2)$ acting on the cylinder at two points , the leftmost point of the cylinder and the topmost point of the cylinder. The forces at the point $a$ are computed using the formula

$$(f_1, f_2)(a) = (-p\mathbb{I} + 2\nu\mathbb{D})(a) * \mathbf{n}(a) \tag{4.3}$$

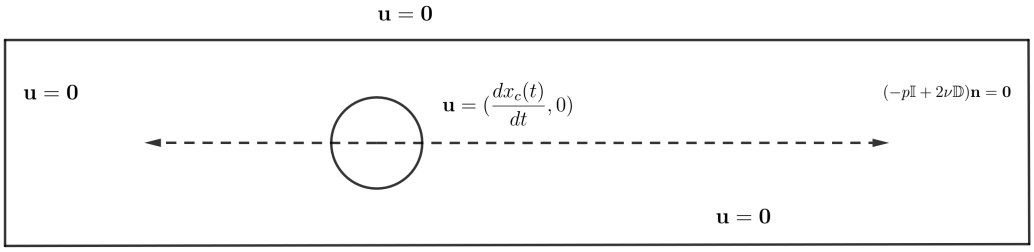


Figure 4.1: A sketch of the problem with boundary conditions. The dotted line illustrates the motion of the cylinder.

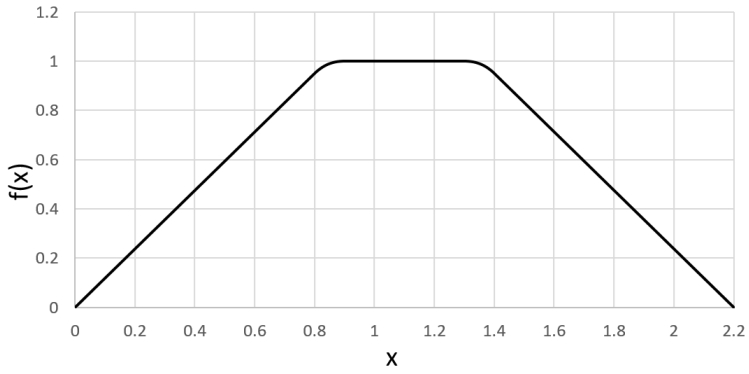


Figure 4.2: Part of the displacement function. The displacement in the vicinity of the cylinder is only a translation.

where $\mathbf{n}(a)$ is the normal at point $a$. These are compared to the results in [Schott, 2017], however all of the values in [Schott, 2017] are scaled by a constant which is not obtainable in the current setup of the benchmark. Since the measurement in the Fictitious Boundary method is not precise (quantities are measured at different cells over time) and there are oscillations of pressure, the values are averaged over 5 time steps to dampen the oscillations. It should be noted that since the force is computed from point-wise values, the results can be sensitive to the choice of the mesh.

The benchmark is conducted on 2 meshes roughly the same resolution as the mesh used in [Schott, 2017]. One mesh has the cylinder cut out of the mesh and is unstructured, which leads to better representation of the cylinder, while the other is a structured mesh with crossed diagonals where the cylinder is only marked as a subdomain in mesh leading to less accurate representation of the cylinder as illustrated in the Figure 4.3. The ALE formulations are computed on both meshes. The FIB method is only computed on the structured mesh. Recall from second chapter that when using the FIB, the mesh has to be fine enough to resolve the moving object, a cylinder in this case which it is as can be seen in figures 4.4 and 4.5.

All the schemes are tested in the implicit Euler versions (results in [Schott, 2017] are computed using implicit Euler as well). The time step chosen for the

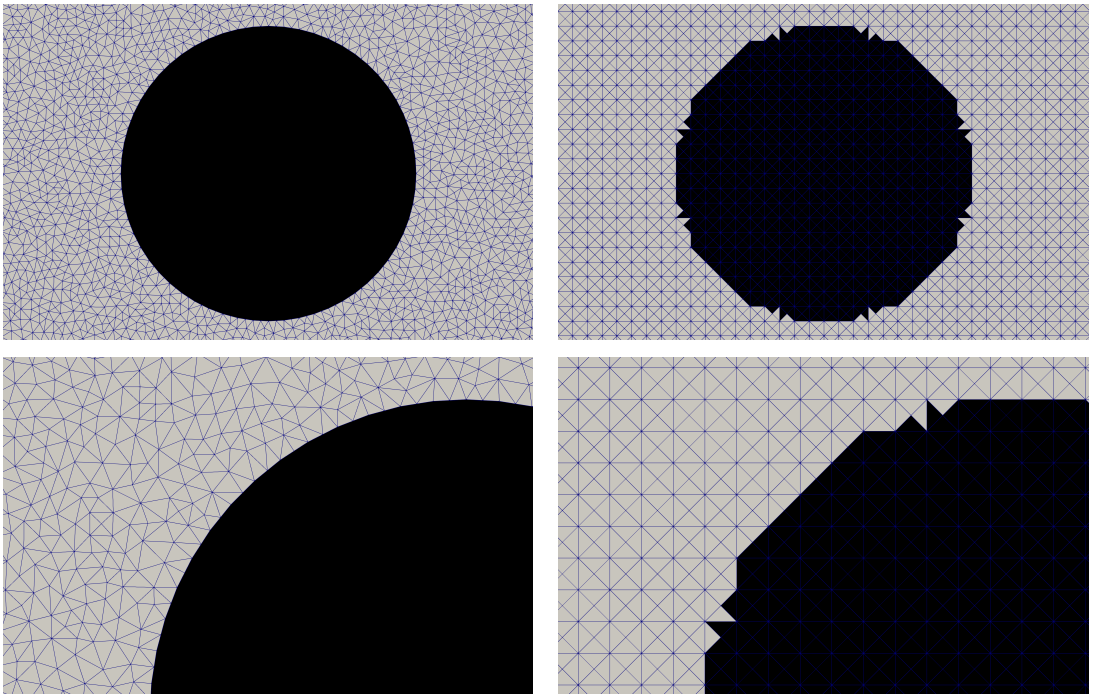

Figure 4.3: The structured and unstructured meshes used for the computation. The close up picture shows the representation of the cylinder for the two meshes. The representation of the cylinder on the structured mesh is inaccurate regardless of the method. While the representation of the cylinder for the ALE is fixed (it is inaccurate but it does not change time) the representation of the cylinder in the FIB changes for every time step.

fully coupled methods is the same as in [Schott, 2017], that is $dt = 0.001$ or $dt = 0.002$, however the result are very similar, so only the results for $dt = 0.001$ are presented. The IPCS-ALE requires about 10x smaller time step to give comparable results so time step $dt = 0.0001$ was chosen for the projection method which makes the total computational time roughly 5x bigger compared to the fully coupled methods.

The zero mean pressure is chosen as the artificial (boundary) condition for the pressure computation in the IPCS scheme. The condition is enforced via Lagrange multipliers and resulting algebraic system is a saddle point system. A direct solver MUMPS is used to solve for the pressure. The solver for the tentative velocity a velocity update remains to be the same as in the previous chapter.

The computations were carried out using the FEniCS platform [Alnæs et al., 2015] and postproccesed in Paraview. The Taylor-Hood $\mathcal{P}_2/\mathcal{P}_1$ elements were used for both methods.

The computed velocities and pressures are illustrated in the Figures 4.4 and 4.5. Both the FIB and ALE yield very good results compared to [Schott, 2017] regardless of the used mesh. There are minor differences, however the main features of the flow are the same for all the methods, more specifically the FIB and ALE yield practically the same results on the same mesh (only FIB is shown). The results on the structured meshes are almost identical to the results illustrated in [Schott, 2017], while on the unstructured mesh there are minor differences at

later times (t=2.29,t=2.46). The IPCS-ALE yields similar flow patterns as the fully coupled ALE on both structured and unstructured mesh.

The densities of forces are illustrated in the Figure 4.6. Both the FIB and the ALE capture the main trend very well regardless of the mesh and are in good comparison with [Schott, 2017]. The IPCS-ALE cannot capture the magnitude of the force as well as the ALE or the FIB. This can be explained by the creation of an artificial boundary layer created by the use of the implicit boundary condition for pressure as mentioned before.

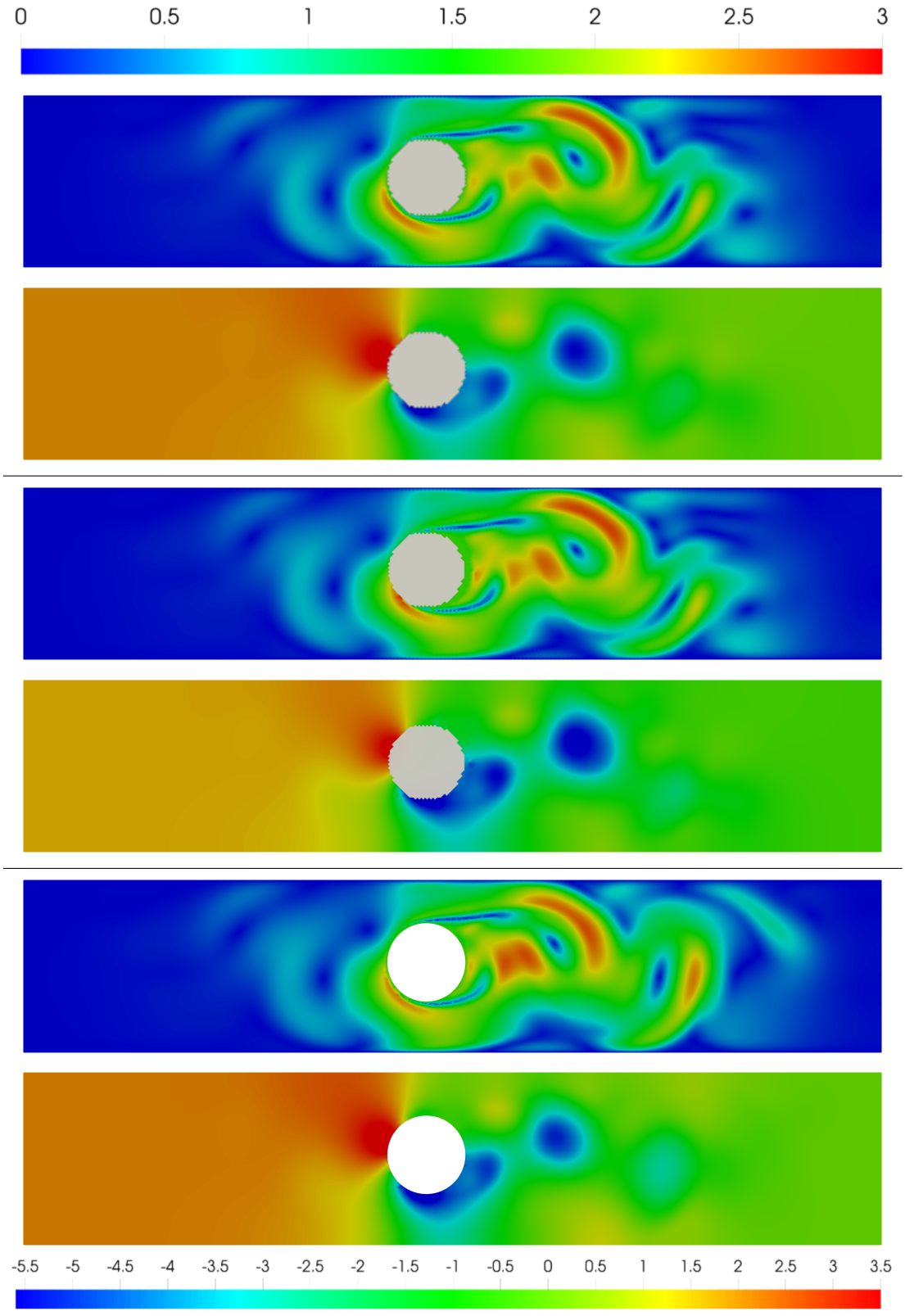


Figure 4.4: Velocity (scale at the top) and pressure (scale at the bottom) results at time t=2.29. The FIB method on the structured mesh is at the top, the IPCS-ALE on the structured mesh (note that the cylinder is only a fixed marked subdomain in the mesh which leads to its bad representation) is in the middle and the ALE on the unstructured mesh is at the bottom.

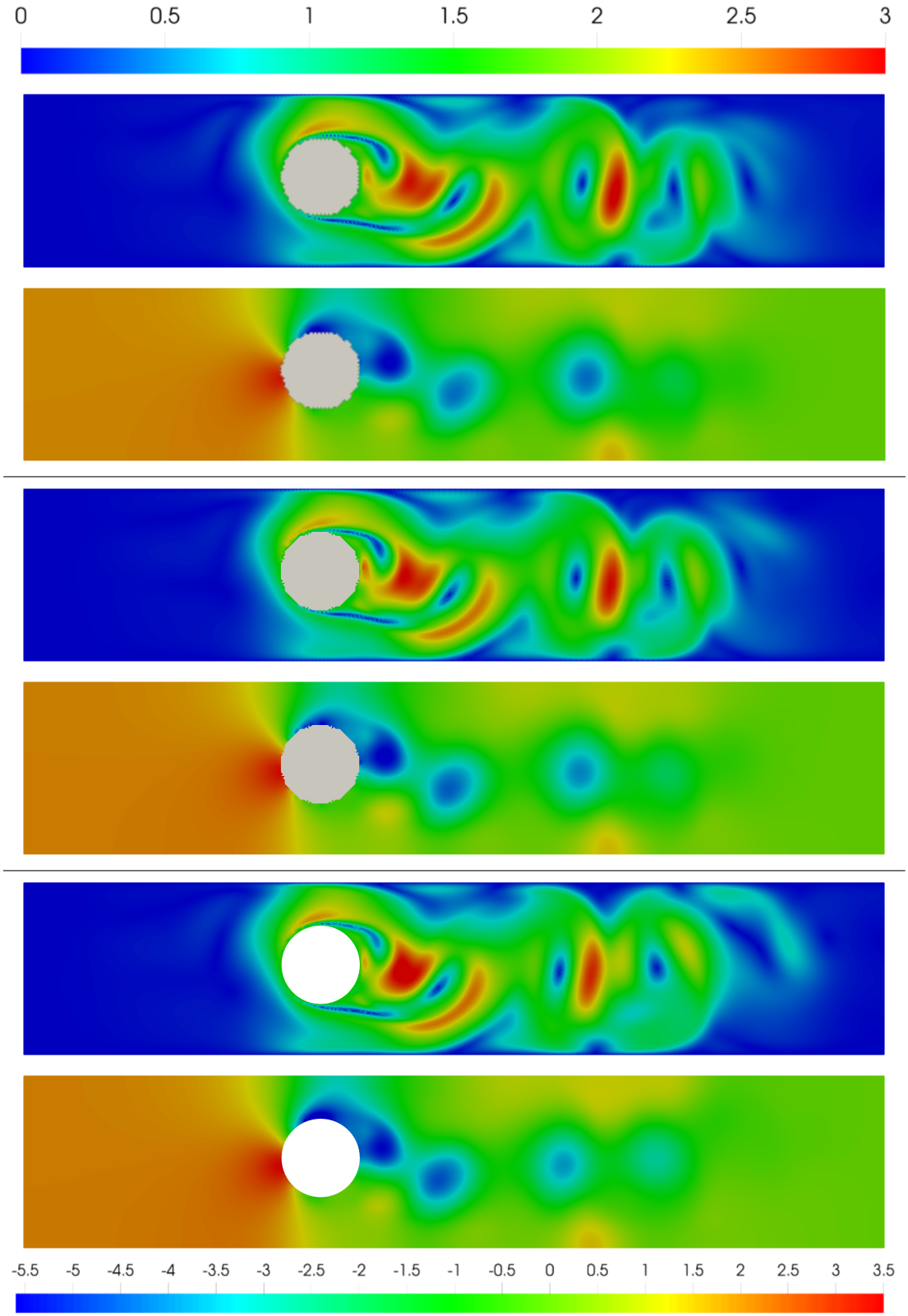


Figure 4.5: Velocity and pressure results at time t=2.46. The order and scaling is the same as in the previous figure.

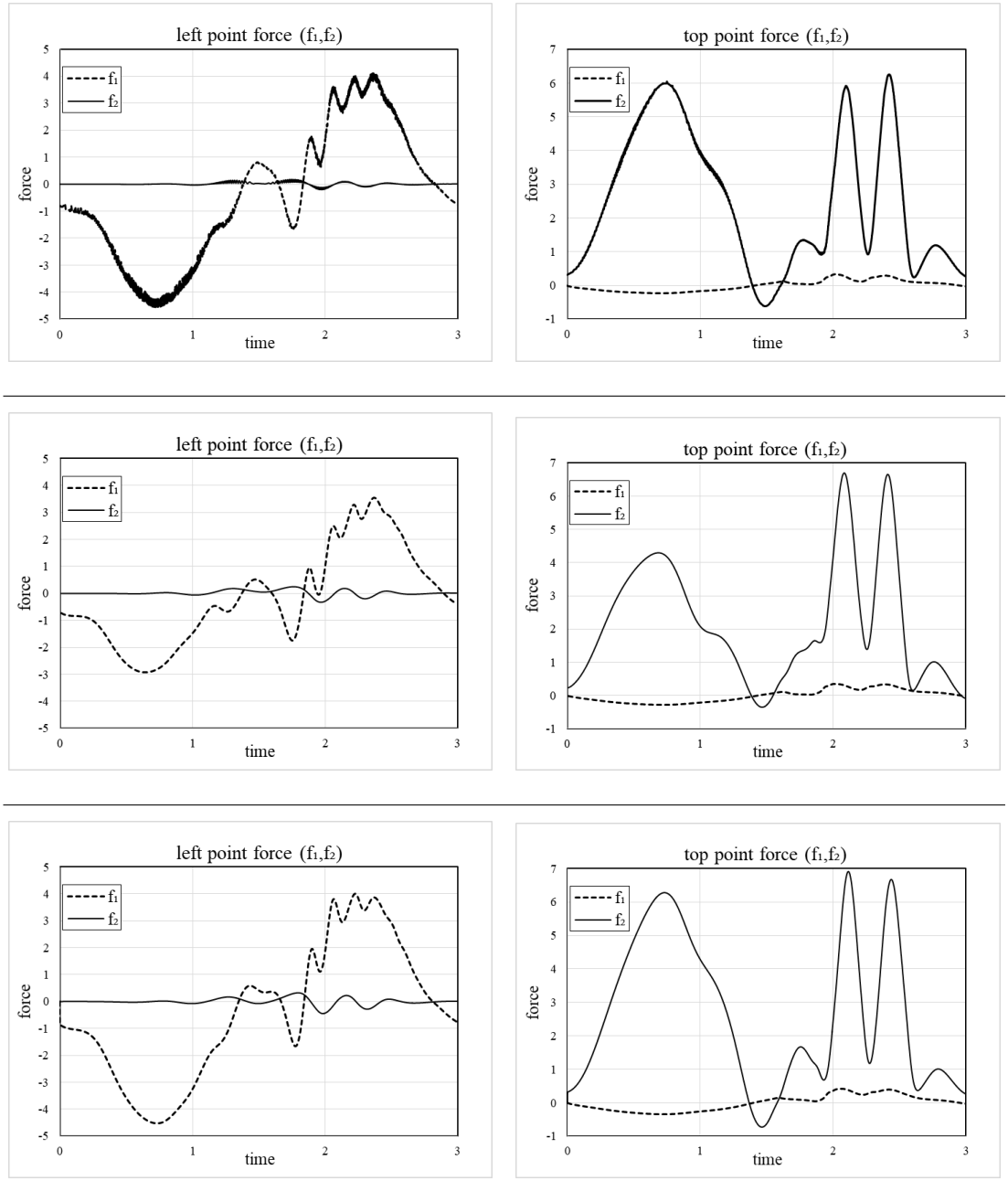


Figure 4.6: Density of force at the leftmost point and the topmost point of the cylinder. The FIB is at the top, IPCS-ALE on structured mesh is in the middle and ALE on the unstructured mesh is at the bottom (the ALE on structured mesh yields same results as FIB without the oscillations).

## 4.2 Simulation of the Experiment

This section is focused on the simulation of an oscillating cyllinder immersed in incompressible fluid governed by the Navier-Stokes equations. The motivation behind this simulation is the experiment [Duda et al., 2015] conducted by the superfluidity research group at Charles university. The experiment involved an oscillating cylinder of rectangular cross section immersed in liquid $He$ II at low temperatures (1,25K). The $He$ II can be described as if it displayed two fluid behavior, one component would be viscous, second would be superfluid. [Barenghi et al., 2014],[Skrbek and Sreenivasan, 2012] suggests, that on large scales $He$ II can be considered Newtonian if the temperature is constant. The aim of this section is to provide numerical simulation of this experiment on large scales. The author has already tried to simulate the model in his bachelor thesis [Outrata, 2018], however the results were not satisfying.

The model of this experiment will be 2D for two reasons. Firstly, it would be extremely computationally challenging to provide a reliable 3D model (as a rough estimate, at least 5 million dof would be needed). Secondly, there should not exist any major flows in the plane perpendicular to this axis of oscillation according to [Duda et al., 2015].

The setup is illustrated in the Figure 4.7. The computational domain is a rectangle of width $W = 50mm$ and height $H = 80mm$. In the center of the domain a cyllinder of rectangular cross section (rectangle) oscillates along the vertical axis. The motion of the center of the cylinder is prescribed by

$$\begin{aligned} x_c &= 0 \\ y_c(t) &= -5\sin(\pi t). \end{aligned} \tag{4.4}$$

The height of the cylinder is $l = 3mm$ and width is $D = 10mm$. The frequency of the oscillations is $f = 0.5Hz$. The no-slip boundary conditions are applied on the cylinder and the vertical walls of the domain. The DDN boundary conditions are applied on the horizontal walls at the top and bottom of the domain.

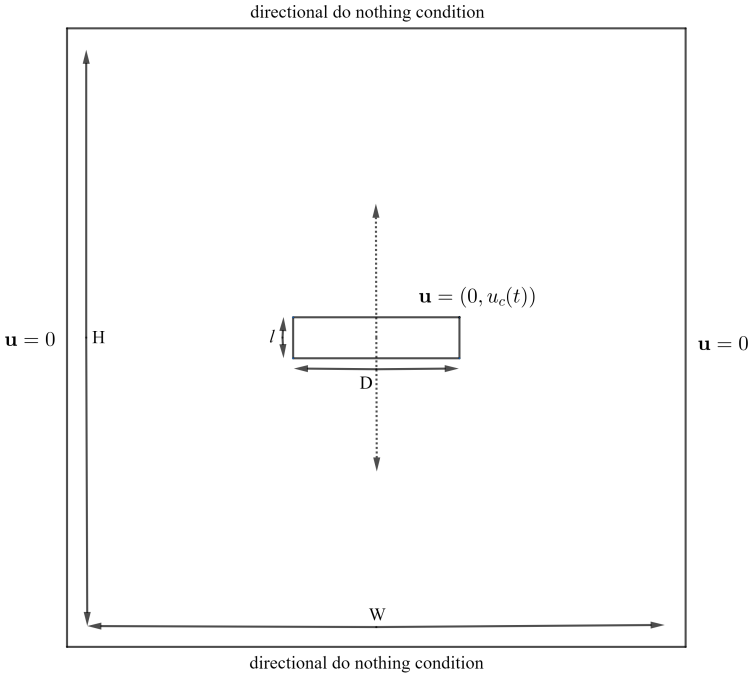


Figure 4.7: A sketch of the problem. The dotted line illustrates the motion of the cylinder.

The computations were carried out on uniform symmetric structured meshes as illustrated in the Figure 4.3. The cylinder is aligned with the elements such that its representation is exact. The mini elements were used because of their computational performance to reduce computational time. This setup yields about $230k$ dof on the coarser mesh and $600k$ dof on more detailed mesh. The viscosity of the fluid is chosen as $\nu = 0.001mm^2/s$. The Newton's method together with direct MUMPS solver was used to solve the system of nonlinear equations.

The biggest issue regarding the simulations seems to be the choice of the time stepping scheme and the stabilization. As for the choice of the stabilization, the ALE method is computed with the SUPG and LSIC stabilizations and the FIB method is computed with the SUPG stabilization. The coefficient $\alpha$ in the stabilization was chosen heuristically. The FIB method is extremely sensitive to the choice of the stabilization parameter and gives chaotic results otherwise.

The implicit Euler time stepping was used mainly for its dissipative properties. Other time stepping schemes have been tried as well (Crank-Nicholson, BDF2) (see the Figure 4.8) however the smaller (or none) dissipation of the schemes lead to chaotic behavior of the system for both methods. The results become more chaotic with decreasing time step even with implicit Euler time stepping scheme. This behaviour might reflect some properties of the $He$ II which incompressible Navier-Stokes model is not able to capture, however the implementation of different fluid model (two fluid model [Landau, 1941],[Barenghi et al., 2014]) would be very difficult to implement due to the computationally intensive nature of the problem (the simulation is $50s - 100s$ long and the chosen time step is $dt = 0.01s$). A more detailed mesh yielding roughly $700k$ dof was employed to confirm these results and it appears that the dissipation coming from the implicit Euler time stepping scheme is essential for stability of the computations.

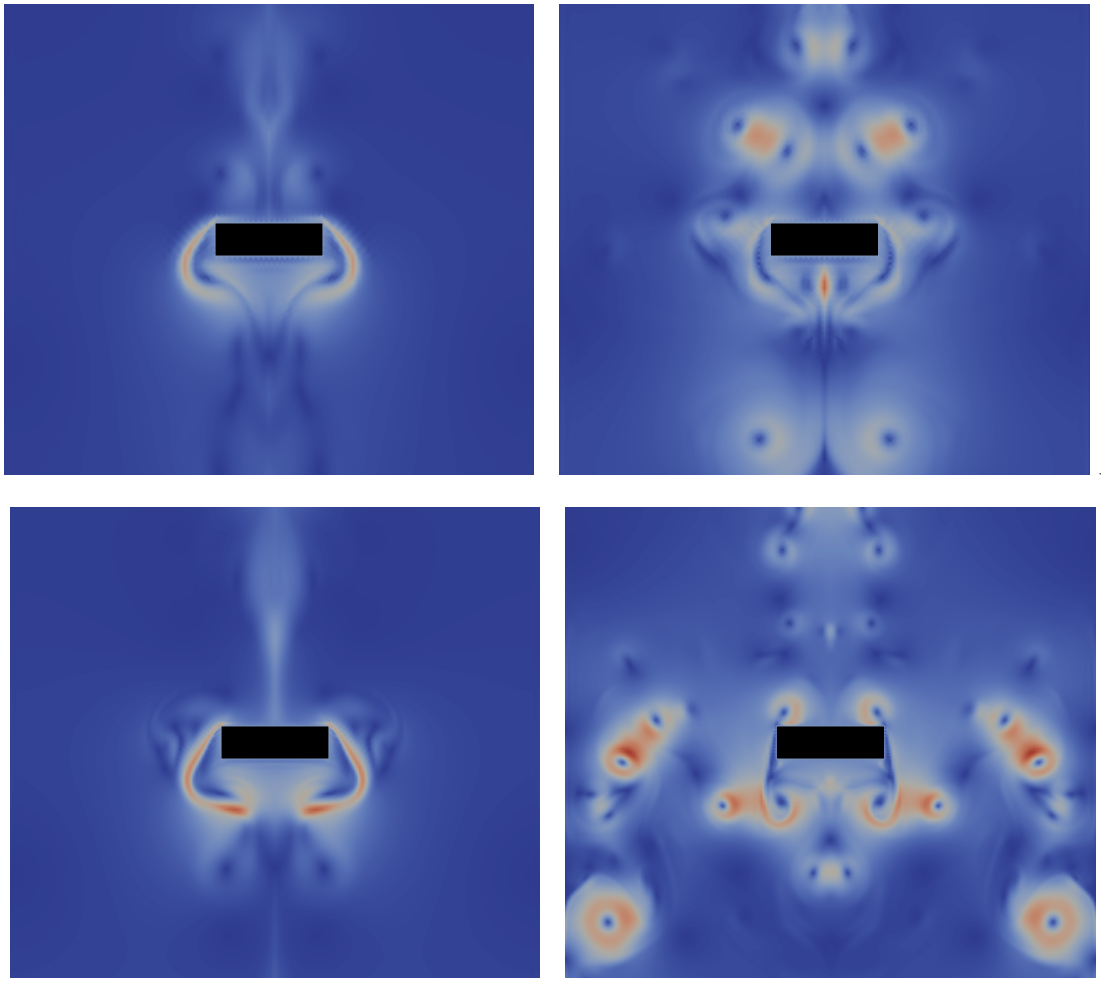

Figure 4.8: Implicit euler (left) vs the Crank Nicholson (right) time stepping schemes in time $t = 18.5s$ (ALE). Bottom pictures are computed on more detailed mesh. It can be seen that the dissipation caused by the Euler scheme makes a significant difference in the long term behavior.

The results of the simulation are illustrated in the Figure 4.9 and 4.10. It can be seen that the results computed by ALE and FIB method are similar and both qualitatively agree with the experiment. Both methods yield the vortex shedding typical for the experiments and keep the symmetry for the time of the simulation (50s). There are differences in the shape of the vortices, but that could be explained by inaccurate representation of the cylinder in the FIB method (the used mesh was quite coarse for the problem). The results of the ALE do not change much with more detailed mesh. Additional smaller vortices are formed with the use of the FIB method on more detailed mesh which cannot be seen for the ALE computations.

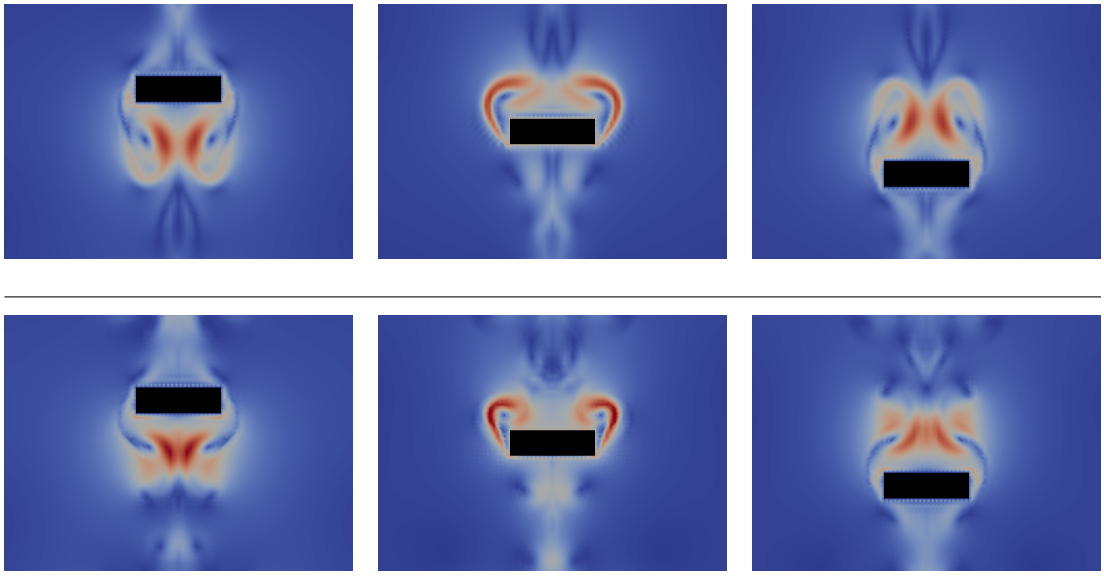

Figure 4.9: ALE method on the top, FIB method on the bottom at times $t = 15s$ on the left, $t = 15.5s$ in the middle and $t = 16s$ on the right. Results for the coarser mesh. Both methods give similar results.

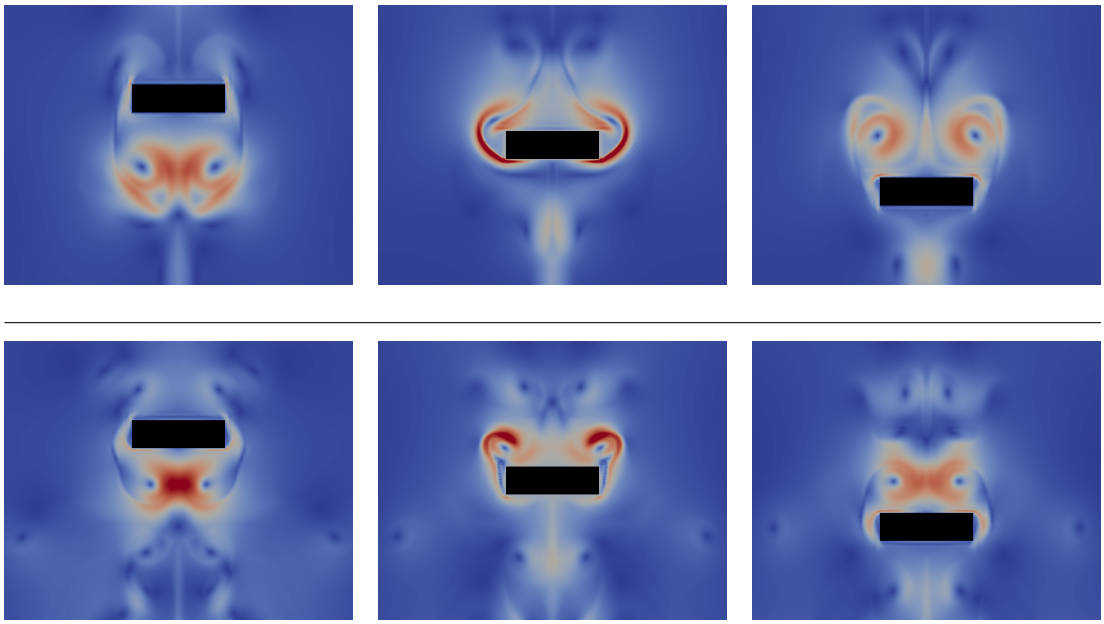

Figure 4.10: Results on the more detailed $600k$ dof mesh. The ordering and times are the same as in the previous figure. The ALE results do not change much with detailed mesh unlike the FIB results.

The Figure 4.11 illustrates the sensitivity of the problem on the used mesh. If one uses mesh with diagonals in one preferred direction, left instead of crossed, one gets completely different results in the long run. This has shown to be an important factor which was not considered in the previous work. Computations yield the same results for viscosity high enough ($\nu \approx 1mm^2s^{-1}$) regardless of the

mesh, however with increasing effects of the convection and decreasing viscosity the system becomes chaotic.

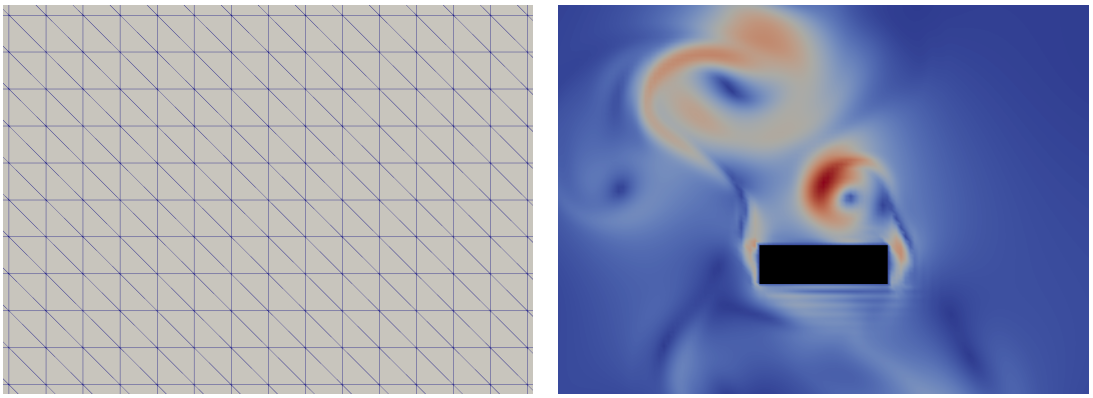

Figure 4.11: Image of structured mesh with left diagonals on the right and behavior of the velocity at time $t = 16s$ (ALE). If compared with the Figure 4.9 one can see that the structure and symmetry of the mesh plays an important role.

The force (or more precisely linear density of force because of the 2D simulation) acting on the cylinder is illustrated in figure 4.12. The force was computed utilizing the formula

$$\mathbf{f} = \rho(-p + 2\nu\mathbb{D}, \mathbf{n})_{\partial\Omega_{cylinder}} \tag{4.5}$$

where $\mathbf{n}$ is the outer normal to the cylinder and $\rho = 0.125g/cm^3$ is roughly the density of the liquid helium. Because of the symmetry of the problem only the vertical force is illustrated and computed. Both methods yield the same trend of the force regardless of the mesh used. The force is the highest when the vortices are being shed, that is about $0.1s$ (a little earlier for FIB) before the cylinder reaches the maximum velocity. The oscillations in the FIB method are caused by oscillations of pressure and inexact representation of the cylinder, which can have quite a big impact on the fluid structure interface. The ALE method is in this sense much more reliable because of the exact representation of the cylinder and no oscillatory behavior of the pressure. The forces will be compared to the experiment in the future.

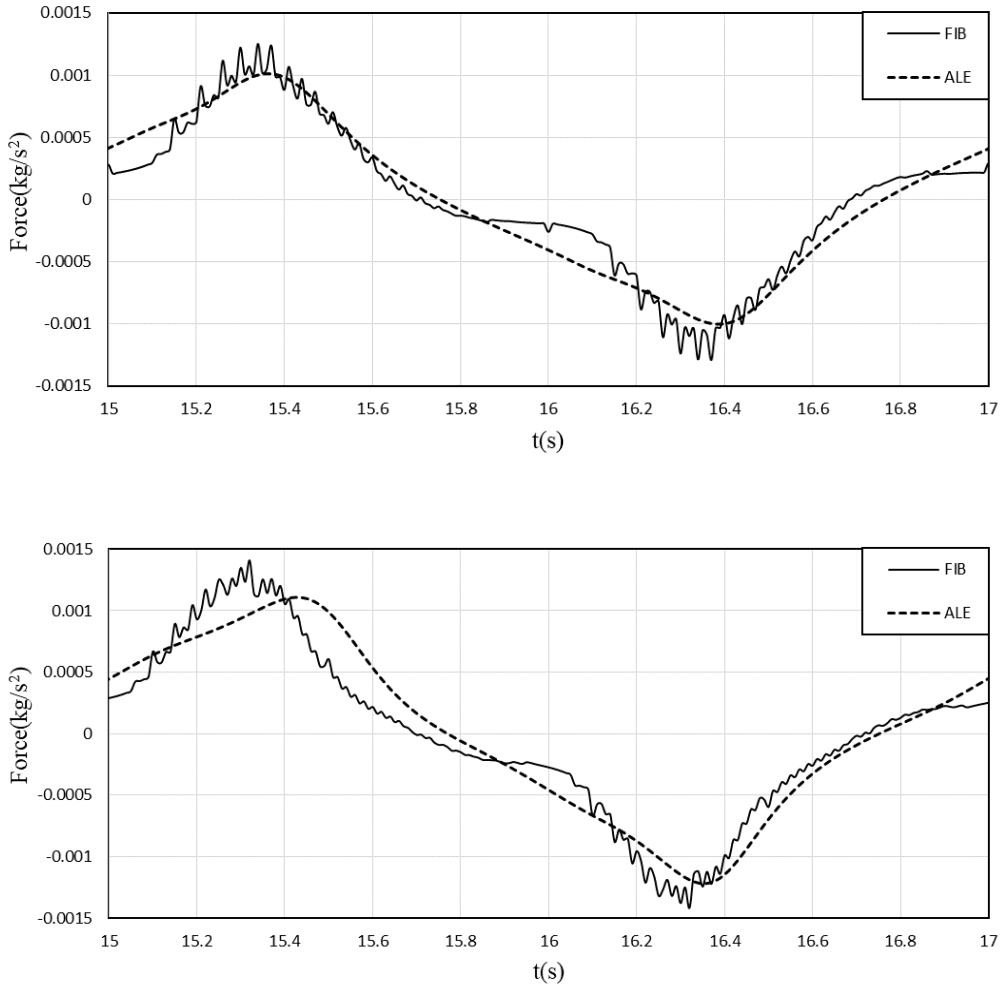


Figure 4.12: Force (linear density of force)(y-component) acting on the cylinder for both methods during one period for the coarse mesh (top) and the more detailed mesh (bottom). Compare with the Figure 4.9 to see the position of the cylinder (at time $t = 15s$ cylinder is at the top, at time $t = 16s$ at the bottom). Both methods give similar trend, however the oscillations in the FIB method are quite significant regardless of the mesh.

The pseuovorticity can be computed for these simulations to get direct comparison with the experiment, however there are open boundary which make it difficult to stabilize the amount of particles in the field of view. This problem is yet to be addressed as well as the fact that the particles in the experiment have mass.

# 5. Discussion

The thesis describes two topics connected to the numerical solution of the incompressible Navier-Stokes equations. The preconditioning and solution methods of algebraic saddle point systems arising from the equations are described and tested in the second chapter. The third chapter describes two distinct methods of dealing with time-dependent domains which are then used for the benchmark test and simulation of the experiment in the fourth chapter.

The simulation of the vortex rings was successful as all tested methods gave the same results regardless of the mesh (in 2D and 3D as well). The computed results qualitatively agree with the experiment [Švančara et al., 2020] on a sufficiently fine mesh in 3D. The most suitable method for this particular simulation proved to be the projection IPCS method, which yielded the best computational time as well as good scaling. One drawback of the method (and projection methods in general) is the unclear boundary conditions which can be harder to implement for more complicated problems such as the simulation of the experiment and generally fluid-structure interface problems. The modified projection scheme IPCS-ALE was slower in the benchmark test than the fully coupled approaches and gave the least accurate results.

The preconditioners LSC and both versions of PCDR appeared to have similar performance in the the vortex ring simulation. The LSC does not seem to be as robust with respect to the Reynolds number as the PCDR. The robustness and performance of the LSC could be rapidly improved by using the scaled version of the LSC (2.36) according to the [Elman et al., 2006]. As the scaled version of the LSC is not available in the PETSc nor FEniCS its implementation might present an interesting opportunity for further research. An alternative to the preconditioning strategies would be the SIMPLE method [Patankar and Spalding, 1972] or SPAC [Elman et al., 2006]. The performance of the preconditioning varies with elements. It has been observed that the performance of the preconditioners (and projection method) was worse while using the mini elements compared to the Taylor-Hood elements. On the other hand, the structure of the matrix arising from mini elements is more easily invertible by the MUMPS than the matrix arising from the Taylor-Hood elements from which we can draw the conclusion that the comparison may vary with different elements.

Two different approaches of simulating the incompressible viscous Navier-Stokes fluid in time-dependent domains are presented in the thesis. Both methods have proved to be reliable at lower Reynolds numbers in the benchmark test, unfortunately, both methods struggled to simulate the full scale experimental setup.

The results of the benchmark tests were reproduced at a very good accuracy for both methods regardless of the mesh and elements used. The fully coupled approaches gave more accurate results and were faster than the projection IPCS-ALE scheme.

The simulation of the experiment has shown to be a more complex problem than the benchmark test. The results computed by the implicit Euler with given time step on the symmetric mesh were useful and gave meaningful results. Unfortunately, with the choice of more accurate time stepping scheme (such as

Crank-Nicholson or BDF2) the system became rather chaotic and the computed results were completely different than those observed in the experiment. A similar behavior was observed when the mesh was not completely symmetric. The computations quickly tended to lose symmetry and became nonphysical. Since this happened for both ALE and FIB, it might be the incompressible Navier-Stokes model of the fluid that is actually not capable of capturing the observed phenomena. The $He$ II could be described as if it consists of two components, viscous and non-viscous which are coupled together and sum of their densities is constant, however, their ratio changes with temperature, which means that if the temperature were really constant, the mixture should resemble the classical incompressible viscous fluid as suggested by [Landau, 1941],[Barenghi et al., 2014]. The authors of the experiment claim to have conducted the experiment with a constant temperature and were interested in whether the experiment could be captured by the incompressible Navier-Stokes. What could happen is that there actually exists a temperature gradient in the areas with higher velocity (near the cylinder) which changes the dynamics of the fluid and adds some dissipation into the system (the ratio of viscous part increases with temperature). If one would want to capture these effects within a solid framework, there are two approaches: the two fluid model [Landau, 1941] or the single component model described in [Mongiovì et al., 2018]. It is possible that the dissipative properties of the implicit Euler time stepping schemes capture similar effect which is crucial for the stability of the computations and is difficult to impose by other means (just simply add viscosity for example).

A second option might be that the viscosity is in fact so low, that the flows are turbulent. A solution to this would be either to use extremely fine mesh and time step, or to employ a turbulence model. Both of the approaches would be very difficult to realize as the turbulence is currently not a fully understood problem. The validity of the 2D model can be questioned as well, however, due to the computational resources, it is not possible to implement a 3D model in the time period given.

This conclusion has been drawn after years of struggling to get valid results after testing many approaches (some of which are not documented in this thesis) to deal with time-dependent nature of the problem e.g., transforming into the reference domain of the cylinder, Lagrange multiplier method and the most basic penalty term method [Shirokoff and Nave, 2015]. The reason why the Lagrange multipliers are less suitable than other methods is that the resulting algebraic system requires additional special solution approach or it becomes too costly to solve. The basic penalty term method struggles to converge as the oscillation of the cylinder becomes $0.5Hz$. Another method which could still be tried but would require more time to implement, would be CutFEM [Burman et al., 2015](the CutFEM is yet to be implemented for nonlinear problems), or better penalty term method [Shirokoff and Nave, 2015] (which works for $C^1$ domains which is not our case, so some smoothing would be required). The problem of the stabilization (regardless of the type) is that it adds unwanted artificial viscosity to the system. This is especially important at high Reynolds numbers, such as in the experiment. As for the choice of the stabilization, both the interior penalty type stabilizations and SUPG work well for the Fictitious Boundary Method. It is difficult to get the exact value of the coefficients to not over-

stabilize the system. The behavior of the interior penalty method with FIB was previously tested in the author's bachelor thesis. It was rather challenging to find a proper way to transform the interior penalty stabilization into the reference domain, that is why the SUPG was chosen as the transformation of the method was fairly straight-forward. The LSIC stabilization improved the stability of the computations when using the ALE formulation (however, it adds a bit more dissipation). This combination has proven to work very well since it could stabilize the computations with the increased frequency of the oscillations $f = 1Hz$ with the same stabilization parameter and time step. The preconditioners discussed in the second chapter can be directly used with the FIB, however, they require further modification so they can be used together with the ALE.

The ALE method has the ability to properly capture the effects on the fluid-cylinder interface such as the force acting on the cylinder. The FIB gives the same trend of the forces for the same choice of the mesh after suitable spatial and temporal averaging; however, the results should be interpreted cautiously as the method uses very rough (low order) approximation to the fluid-structure interface.

# Conclusion

Two aspects of the numerical solution to the incompressible Navier-Stokes equations are explained and tested on real world experiments in the first part of the thesis. The PCDR and LSC preconditioners were compared with an IPCS projection method on the vortex rings experiment. While giving the same results, the IPCS method was much more efficient than the fully coupled approaches for the vortex ring experiment. The IPCS method showed great parallel scaling up to 13M dof making it the best of all tested methods for the vortex ring experiment. It has been verified that the pseudovorticity can closely approximate the position of a vortex ring tracked by vorticity.

Second part of the thesis contains description and comparison of the Arbitrary Lagrangian Eulerian formulation of Navier-Stokes equations with the Fictitious Boundary method as two distinct approaches of dealing with time-dependent domains. Both approaches yield the same results on the benchmark test and similar results for the simulation of the oscillating cylinder experiment. The results of the benchmark test obtained by [Schott, 2017] have been accurately reproduced by both methods using the fully coupled approach. It is probable that the incompressible Navier-Stokes model is not a suitable model for the oscillating cylinder experiment and $He$ II specific models should be employed in order to achieve more accurate results, however the FIB and the ALE still provide results which qualitatively agree with the experiment (under special circumstances).

# Bibliography


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

P.R. Amestoy, I. S. Duff, J. Koster, and J.-Y. L'Excellent. A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM Journal on Matrix Analysis and Applications*, 23(1):15–41, 2001.

D. Arndt, M. Braack, and G. Lube. Finite elements for the navier-stokes problem with outflow condition. In Bülent Karasözen, Murat Manguoğlu, Münevver Tezer-Sezgin, Serdar Göktepe, and Ömür Uğur, editors, *Numerical Mathematics and Advanced Applications ENUMATH 2015*, pages 95–103, Cham, 2016. Springer International Publishing. ISBN 978-3-319-39929-4.

S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, K. Rupp, B. F. Smith, S. Zampini, and H. Zhang. PETSc Web page. `http://www.mcs.anl.gov/petsc`, 2015. URL `http://www.mcs.anl.gov/petsc`.

C. F. Barenghi, L. Skrbek, and K. R. Sreenivasan. Introduction to quantum turbulence. *Proceedings of the National Academy of Sciences*, 111(Supplement 1):4647–4652, 2014. ISSN 0027-8424. doi: 10.1073/pnas.1400033111. URL `https://www.pnas.org/content/111/Supplement_1/4647`.

J. Blechta. *Towards efficient numerical computation of flows of non-Newtonian fluids*. Dissertation, Charles University, Prague, 2019.

J. Blechta and M. Řehoř. Fenapack 2019.1.0 (navier-stokes preconditioning package), 2019. URL `http://fenapack.readthedocs.io`.

F. Brezzi and M. Fortin. *Mixed and hybrid finite element methods*, volume 15. Springer Science & Business Media, 2012.

A. N. Brooks and T. J.R. Hughes. Streamline upwind/petrov-galerkin formulations for convection dominated flows with particular emphasis on the incompressible navier-stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 32(1):199 – 259, 1982. ISSN 0045-7825. doi: https://doi.org/10.1016/0045-7825(82)90071-8. URL `http://www.sciencedirect.com/science/article/pii/0045782582900718`.

R Bruce, J Reynaud, S Pascali, and B Baudouy. 3d CFD transient numerical simulation of superfluid helium. *IOP Conference Series: Materials Science and Engineering*, 278:012057, dec 2017. doi: 10.1088/1757-899x/278/1/012057. URL `https://doi.org/10.1088%2F1757-899x%2F278%2F1%2F012057`.

E. Burman, M. A. Fernández, and P. Hansbo. Edge Stabilization for the Incompressible Navier-Stokes Equations: a Continuous Interior Penalty Finite

Element Method. Research Report RR-5349, INRIA, 2004. URL https://hal.inria.fr/inria-00070653.

E. Burman, S. Claus, P. Hansbo, M. G. Larson, and A. Massing. Cutfem: Discretizing geometry and partial differential equations. *International Journal for Numerical Methods in Engineering*, 104(7):472–501, 2015. doi: 10.1002/nme.4823. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.4823.

R. Codina, G. Houzeaux, H. Coppola-Owen, and J. Baiges. The fixed-mesh ale approach for the numerical approximation of flows in moving domains. *J. Comput. Phys.*, 228(5):1591–1611, March 2009. ISSN 0021-9991. doi: 10.1016/j.jcp.2008.11.004. URL https://doi.org/10.1016/j.jcp.2008.11.004.

M.A. DeLong and J.M. Ortega. Sor as a preconditioner. *Applied Numerical Mathematics*, 18(4):431 – 440, 1995. ISSN 0168-9274. doi: https://doi.org/10.1016/0168-9274(95)00080-E. URL http://www.sciencedirect.com/science/article/pii/016892749500080E.

Fabián Duarte, Raúl Gormaz, and Srinivasan Natesan. Arbitrary lagrangian–eulerian method for navier–stokes equations with moving boundaries. *Computer Methods in Applied Mechanics and Engineering*, 193(45):4819 – 4836, 2004. ISSN 0045-7825. doi: https://doi.org/10.1016/j.cma.2004.05.003. URL http://www.sciencedirect.com/science/article/pii/S0045782504002476.

D. Duda, P. Švančara, M. La Mantia, M. Rotter, and L. Skrbek. Visualization of viscous and quantum flows of liquid $^4$He due to an oscillating cylinder of rectangular cross section. *Phys. Rev. B*, 92:064519, 2015. doi: 10.1103/PhysRevB.92.064519. URL https://link.aps.org/doi/10.1103/PhysRevB.92.064519.

M. Řehoř. *Diffuse interface models in theory of interacting continua*. Dissertation, Charles University, Prague, 2018.

H. Elman, V. Howle, J. Shadid, R. Shuttleworth, and R. Tuminaro. Block preconditioners based on approximate commutators. *SIAM Journal on Scientific Computing*, 27(5):1651–1668, January 2006. doi: 10.1137/040608817. URL https://doi.org/10.1137/040608817.

H. Elman, D. Silvester, and A. Wathen. *Finite Elements and Fast Iterative Solvers*. Oxford University Press, June 2014. doi: 10.1093/acprof:oso/9780199678792.001.0001. URL https://doi.org/10.1093/acprof:oso/9780199678792.001.0001.

L.C. Evans and American Mathematical Society. *Partial Differential Equations*. Graduate studies in mathematics. American Mathematical Society, 1998. ISBN 9780821807729. URL https://books.google.cz/books?id=5Pv4LVB_m8AC.

M. Feistauer. Theory and numerics for problems of fluid dynamics. *Charles University Prague, Faculty of Mathematics and Physics*, 2006.

D.R. Fokkema. *Enhanced Implementation of Bicgstab(l) for Solving Linear Systems of Equations*. Universiteit Utrecht. Mathematisch Instituut, 1996. URL https://books.google.cz/books?id=j7ACrgEACAAJ.

R. Glowinski. Finite element methods for incompressible viscous flow. *Handbook of numerical analysis*, 9:3–1176, 2003.

R. Glowinski, T.W. Pan, T.I. Hesla, D.D. Joseph, and J. Périaux. A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: Application to particulate flow. *Journal of Computational Physics*, 169(2):363 – 426, 2001. ISSN 0021-9991. doi: https://doi.org/10.1006/jcph.2000.6542. URL http://www.sciencedirect.com/science/article/pii/S0021999100965422.

P. M. Gresho and R. L. Sani. On pressure boundary conditions for the incompressible navier-stokes equations. *International Journal for Numerical Methods in Fluids*, 7(10):1111–1145, 1987. doi: 10.1002/fld.1650071008. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.1650071008.

J.L. Guermond, P. Minev, and J. Shen. An overview of projection methods for incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 195(44):6011 – 6045, 2006. ISSN 0045-7825. doi: https://doi.org/10.1016/j.cma.2005.10.010. URL http://www.sciencedirect.com/science/article/pii/S0045782505004640.

V. E. Henson and U. M. Yang. Boomeramg: A parallel algebraic multigrid solver and preconditioner. *Applied Numerical Mathematics*, 41(1):155 – 177, 2002. ISSN 0168-9274. doi: https://doi.org/10.1016/S0168-9274(01)00115-5. URL http://www.sciencedirect.com/science/article/pii/S0168927401001155. Developments and Trends in Iterative Methods for Large Systems of Equations - in memorium Rudiger Weiss.

V. E. Howle, J. Schroder, and R. Tuminaro. The effect of boundary conditions within pressure convection–diffusion preconditioners. Technical report, Technical Report SAND2006-4466, Sandia National Laboratories, Livermore, CA, 2006.

V. John and P. Knobloch. On the choice of parameters in stabilization methods for convection–diffusion equations. In Karl Kunisch, G. Of, and O. Steinbach, editors, *Numerical Mathematics and Advanced Applications*, pages 297–304, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg. ISBN 978-3-540-69777-0.

L. Landau. Theory of the superfluidity of helium ii. *Phys. Rev.*, 60:356–358, Aug 1941. doi: 10.1103/PhysRev.60.356. URL https://link.aps.org/doi/10.1103/PhysRev.60.356.

J. Liesen and Z. Strakoš. Krylov subspace methods: Principles and analysis. *Krylov Subspace Methods: Principles and Analysis*, 01 2012. doi: 10.1093/acprof:oso/9780199655410.001.0001.

A. Lozovskiy, M. Olshanskii, and Y. Vassilevski. A quasi-lagrangian finite element method for the navier-stokes equations in a time-dependent domain. *Computer Methods in Applied Mechanics and Engineering*, 333:55–73, 01 2018. doi: 10.1016/j.cma.2018.01.024.

S. Mittal, A. Ratner, D. Hastreiter, and T. Tezduyar. Space-time finite element computation of incompressible flows with emphasis on flows involving oscillating cylinders. *International Video Journal of Engineering Research*, 1:83–96, 12 1991.

M. S. Mongiovì, D. Jou, and M. Sciacca. Non-equilibrium thermodynamics, heat transport and thermal waves in laminar and turbulent superfluid helium. *Physics Reports*, 726:1 – 71, 2018. ISSN 0370-1573. doi: https://doi.org/10.1016/j.physrep.2017.10.004. URL `http://www.sciencedirect.com/science/article/pii/S0370157317303472`. Non-equilibrium thermodynamics, heat transport and thermal waves in laminar and turbulent superfluid helium.

M. F. Murphy, G. H. Golub, and A. J. Wathen. A note on preconditioning for indefinite linear systems. *SIAM J. Scientific Computing*, 21:1969–1972, 1999.

M. Olshanskii and Y. Vassilevski. Pressure schur complement preconditioners for the discrete oseen problem. *SIAM Journal on Scientific Computing*, 29: 2686–2704, 01 2007. doi: 10.1137/070679776.

O. Outrata. *Computation of viscous flows due to an oscillating cylinder of rectangular cross section*. Bachelor thesis, Charles University, Prague, 2018.

O. Outrata, M. La Mantia, P. Švančara, J. Hron, M. Pavelka, J. I. Polanco, and G. Krstulovic. Lagrangian estimate of the strength of the macroscopic vortex rings and its relation with the underlaying eulerian vorticity. In preparation, 2020.

S.V Patankar and D.B Spalding. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *International Journal of Heat and Mass Transfer*, 15(10):1787 – 1806, 1972. ISSN 0017-9310. doi: https://doi.org/10.1016/0017-9310(72)90054-3. URL `http://www.sciencedirect.com/science/article/pii/0017931072900543`.

B. Schott. *Stabilized Cut Finite Element Methods for Complex Interface Coupled Flow Problems*. Dissertation, Technische Universität München, München, 2017.

D. Shirokoff and J.-C. Nave. A sharp-interface active penalty method for the incompressible navier–stokes equations. *Journal of Scientific Computing*, 62 (1):53–77, Jan 2015. ISSN 1573-7691. doi: 10.1007/s10915-014-9849-6. URL `https://doi.org/10.1007/s10915-014-9849-6`.

L. Skrbek and K. Sreenivasan. Developed quantum turbulence and its decay. *Physics of Fluids*, 24, 01 2012. doi: 10.1063/1.3678335.

T. Tezduyar and S. Sathe. Stabilization parameters in supg and pspg formulations. *Journal of Computational and Applied Mechanics*, 4:71–88, 01 2003.

T. E Tezduyar and Y. Osawa. Finite element stabilization parameters computed from element matrices and vectors. *Computer Methods in Applied Mechanics and Engineering*, 190(3):411 – 430, 2000. ISSN 0045-7825. doi: https://doi.org/10.1016/S0045-7825(00)00211-5. URL http://www.sciencedirect.com/science/article/pii/S0045782500002115.

Tayfun Tezduyar. Stabilization parameters and local length scales in supg and pspg formulations. In *Proceedings of the Fifth World Congress on Computational Mechanics, On-line publication: http://wccm. tuwien. ac. at/, Paper-ID*, volume 81508, 2002.

T.E. Tezduyar. Stabilized finite element formulations for incompressible flow computations††this research was sponsored by nasa-johnson space center (under grant nag 9-449), nsf (under grant msm-8796352), u.s. army (under contract daal03-89-c-0038), and the university of paris vi. volume 28 of *Advances in Applied Mechanics*, pages 1 – 44. Elsevier, 1991. doi: https://doi.org/10.1016/S0065-2156(08)70153-4. URL http://www.sciencedirect.com/science/article/pii/S0065215608701534.

S. Turek, D. Wan, and L. S. Rivkind. The fictitious boundary method for the implicit treatment of dirichlet boundary conditions with applications to incompressible flow simulations. In *Lecture Notes in Computational Science and Engineering*, pages 37–68. Springer Berlin Heidelberg, 2003. doi: 10.1007/978-3-642-19014-8_3. URL https://doi.org/10.1007/978-3-642-19014-8_3.

S. Turek, L. Rivkind, J. Hron, and R. Glowinski. Numerical study of a modified time-stepping $\theta$-scheme for incompressible flow simulations. *Journal of Scientific Computing*, 28(2):533–547, 2006. ISSN 1573-7691. doi: 10.1007/s10915-006-9083-y. URL https://doi.org/10.1007/s10915-006-9083-y.

P. Švančara, M. Pavelka, and M. La Mantia. An experimental study of turbulent vortex rings in superfluid 4he. *Journal of Fluid Mechanics*, 889:A24, 2020. doi: 10.1017/jfm.2020.96.

A.W. Vreman. The projection method for the incompressible navier–stokes equations: The pressure near a no-slip wall. *Journal of Computational Physics*, J Comp Phys:353–374, 04 2014. doi: 10.1016/j.jcp.2014.01.035.

Q. Zhu and J. Yan. A moving-domain cfd solver in fenics with applications to tidal turbine simulations in turbulent flows. *Computers and Mathematics with Applications*, 2019. ISSN 0898-1221. doi: https://doi.org/10.1016/j.camwa.2019.07.034. URL http://www.sciencedirect.com/science/article/pii/S0898122119303906.