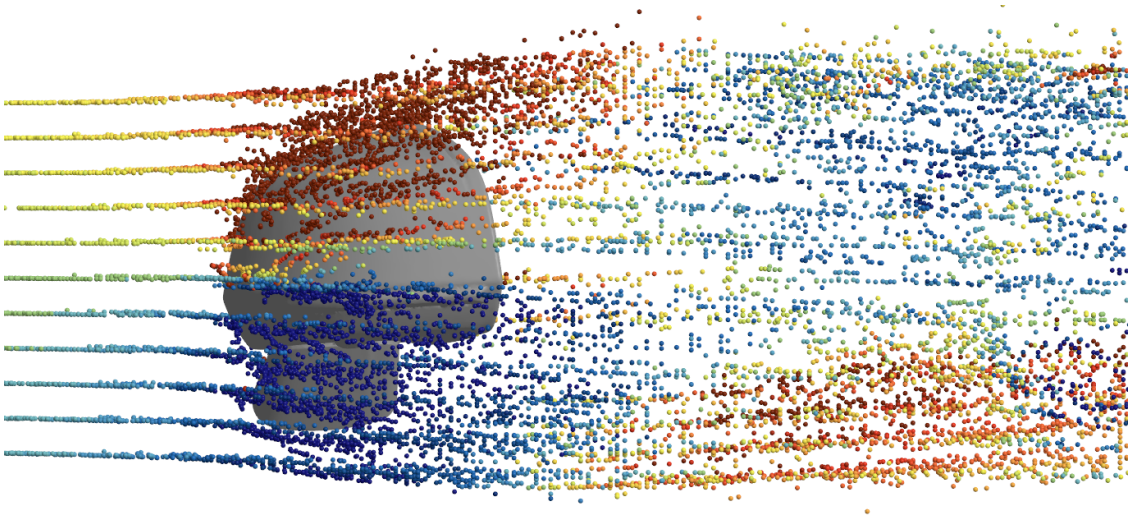
CHALMERS
UNIVERSITY OF TECHNOLOGY



Rapid Predictions of Particle Deposition Using rCFD

Implementation and investigation of recurrence computational fluid dynamics in an industrially relevant workflow for vehicle contamination

Master's thesis in Applied Mechanics

KATARINA ARVIDSSON
AINA SIGBJÖRNSSON

DEPARTMENT OF MECHANICS AND MARITIME SCIENCES
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2026
www.chalmers.se

Master's thesis 2026

Rapid Predictions of Particle Deposition Using rCFD

Implementation and investigation of recurrence computational fluid dynamics in an industrially relevant workflow for vehicle contamination

KATARINA ARVIDSSON
AINA SIGBJÖRNSSON

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Mechanics and Maritime Sciences
Division of Fluid Dynamics
Chalmers University of Technology
Gothenburg, Sweden 2026

Rapid Predictions of Particle Deposition Using rCFD
Implementation and investigation of recurrence computational fluid dynamics in an industrially relevant workflow for vehicle contamination
KATARINA ARVIDSSON
AINA SIGBJÖRNSSON

Supervisor: Torbjörn Virdung, Volvo Cars
Examiner: Henrik Ström, Department of Mechanics and Maritime Sciences

Master's Thesis 2026
Department of Mechanics and Maritime Sciences
Division of Fluid Dynamics
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Visualization of particles flowing over a Volvo EX40 mirror.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2026

Rapid Predictions of Particle Deposition Using rCFD
Implementation and investigation of recurrence computational fluid dynamics in an industrially relevant workflow for vehicle contamination
KATARINA ARVIDSSON
AINA SIGBJÖRNSSON
Department of Mechanics and Maritime Sciences
Chalmers University of Technology

Abstract

Computational Fluid Dynamics (CFD) is widely used within the automotive industry to evaluate designs during early development stages. However, simulations involving turbulent flows and particle deposition are computationally expensive. Recurrence Computational Fluid Dynamics (rCFD) is a method that aims to reduce simulation time by utilizing recurring flow behaviour. The aim of this thesis is to implement and investigate rCFD in Simcenter STAR-CCM+ within an industrially relevant workflow. The method is first applied to a cylinder and then tested on a more complex geometry; a Volvo EX40 mirror. A method for generating a recurrence path and performing particle tracking is developed using STAR-CCM+, Java and Python.

Results show that rCFD can be successfully implemented in STAR-CCM+, reducing computational time for particle simulations, while maintaining acceptable accuracy. Furthermore, the accuracy of the method is influenced by Reynolds numbers. In conclusion, rCFD shows promising potential for reducing computational cost, enabling more efficient contamination analysis in automotive applications.

Keywords: rCFD, CFD, fluid dynamics, STAR-CCM+, multiphase flow, periodicity, simulation, similarity

Acknowledgements

We would like to express our gratitude to the people who supported this project and made it possible. Our supervisor at Volvo Cars, Torbjörn Virdung, thank you for giving guidance and feedback throughout the project. We also want to thank Jacob Vikström for giving us this opportunity as well as Tobias Eidevåg for his assistance with the simulations. It has been an amazing experience to work with the contamination team at Volvo Cars. We have gained a lot of knowledge and valuable insight by working alongside the team and want to give a big thank you to everyone.

We would also like to thank our examiner, Henrik Ström at Chalmers University of Technology, for his valuable inputs and support throughout this thesis. Another thank you to Johannes Hansson for introducing us to this topic and helping us whenever questions arose. A last thank you goes to Caroline Olofsson and Emil Ljungskog at Siemens for giving us support for STAR-CCM+.

Katarina Arvidsson & Aina Sigbjörnsson, Gothenburg, June 2026

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

CFD	Computational Fluid Dynamics
CSV	Comma-Separated Values
DDES	Delayed Detached Eddy Simulation
DES	Detached Eddy Simulation
DNS	Direct Numerical Simulation
HPC	High Performance Computing
LES	Large Eddy Simulation
LPT	Lagrangian Particle Tracking
PISO	Pressure-Implicit with Splitting of Operators
RANS	Reynolds Averaged Navier Stokes
rCFD	Recurrence Computational Fluid Dynamics
RMS	Root Mean Square
SIM	Simulation File
simh	Solution History File
SIMPLE	Semi-Implicit Method for Pressure Linked Equations
URANS	Unsteady Reynolds Averaged Navier Stokes

Nomenclature

Below is the nomenclature of indices and variables that have been used throughout this thesis.

Indices

i,j	Indices for distribution of snapshots
x,y,z	Indices for direction
t,t'	Indices for time
P	Index for particle

Variables

t	Time
V	Volume
S	Surface of volume
u	Velocity
F	Force
p	Pressure
C_D	Drag coefficient
F_D	Drag force
u_∞	Free stream velocity
A	Frontal area
C_L	Lift coefficient
F_L	Lift force
$C_{L,RMS}$	Root mean square of lift coefficient
T	Time period
Re	Reynolds number

l	Characteristic length
μ	Dynamic viscosity
ν	Kinematic viscosity
ρ	Fluid density
S_{tr}	Strouhal number
f	Vortex shedding frequency
D	Diameter
F_S	Surface forces
F_V	Body forces
F_p	Pressure gradient force
F_g	Gravity force
u_s	Slip velocity
p_{static}	Static pressure
g	Gravity vector
$\mathcal{D}$	Distance norm measure
N	Normalization factor
$\mathbf{r}$	Position vector
x, y, z	Coordinates
$\mathcal{D}_k$	Distance norm measure over a jump
n	Number of jumps
δ	Radial size of a cell
$F_u, \mathbf{f}_u$	User-Defined Body Force
Δt	Time step

Contents

List of Figures

List of Tables

1

Introduction

Simulations in the vehicle industry are essential for visualizing how different designs perform during the early stages of development. Due to the complexity of real world flow systems and particle deposition, computational costs are generally high when using existing simulation methods. These simulation methods depend on turbulence models as well as solution algorithms to optimize the calculations for the specific problem. One of the most commonly used turbulence models is Detached Eddy Simulation (DES) [20]. DES is a model that offers a lower computational cost compared to other frequently used turbulence modelling approaches [8].

Optimization of simulation methods is extensively investigated in fluid dynamics, with researchers exploring new or complementary approaches to predict turbulence. All of these efforts share the common goal of reducing computational cost. By reducing simulation time, the design iteration process can become more efficient and thereby enable the possibility of discovering improved design solutions.

One recent approach aimed at reducing computational cost is Recurrence Computational Fluid Dynamics (rCFD). This method relies on a framework, generating long term approximations of the flow field by exploiting periodicity in time [8]. Although promising results have been reported, the method has not yet been widely investigated. Its implementation within vehicle design workflows could further demonstrate its applicability to industrial Computational Fluid Dynamics (CFD) problems and contribute to reducing computational cost. This could reduce the computational cost of both flow and particle simulations. The method shows considerable potential and several factors motivate further investigation for industrial applications.

1.1 Background

When developing new products, design iterations are essential to achieve an optimal solution. Simulations are performed on new designs before the prototyping phase in order to evaluate their performance and avoid unnecessary expenditures associated with suboptimal concepts. When setting up these simulations, it is crucial that they represent real world conditions as accurately as possible. Achieving such accuracy requires a well defined simulation strategy. This strategy may include suitable fluid properties, selection of a turbulence model and optimal solution algorithm for the case as well as generating an appropriate computational mesh. All of these aspects

interact with each other and significantly influence the reliability of the simulation results.

The importance of performing simulations varies depending on the vehicle attributes and physical phenomena under investigation. Contamination analysis is particularly relevant in automotive applications from a safety perspective and a user experience point of view. There are different types of contaminants, such as snow and dirt. Long distance driving in snowy conditions may lead to the headlight being partially or fully covered by snow due to its design. Snow accumulation can block cameras and other driver assistance systems. By conducting simulations for particle deposition, engineers can evaluate a larger variety of designs and identify more suitable solutions. Similar to snow, dirt deposition is also significant as it directly affects customer experience and the longevity of the car.

Fluid simulations require significant computational resources. The computational cost is often high due to the high resolution needed to capture flow gradients. Vehicle geometries are fundamentally complex and resolving turbulent regions under varying operating conditions may require simulations to run for extended periods of time. In collaboration with Johannes Hansson, a PhD student at Chalmers University of Technology, Volvo Cars is investigating the application of rCFD to enable faster predictions of the flow field. Since the study by Hansson et al. [8] was carried out using the CFD software OpenFOAM, the method is adapted to Simcenter STAR-CCM+ to be useful for Volvo Cars. If successfully implemented in industrial applications, computational time could be significantly reduced. Recurrence CFD may therefore contribute to a more efficient design iteration process, enabling improved solutions in early design stages.

1.2 Aim and objectives

The purpose of this thesis is to investigate the potential of reducing simulation time by implementing rCFD, based on earlier research within the field. To initiate the thesis, the rCFD method is applied to a simple geometry, before progressing to a more complex geometry. This thesis is divided into two main scopes; flow over a cylinder and flow over an EX40 mirror. Listed below are objectives to reach desired results and satisfy the aim.

- **Domain setup.** Generate appropriate mesh in the software STAR-CCM+, based on the cylinder case done by Hansson et al. [8]. The model is validated through a convergence study.
- **Implementation of rCFD in STAR-CCM+.** Create a database by extracting needed data from the simulated cylinder case. The database is used to create a distance matrix and recurrence path. When the desired path is obtained, modifications are done to suit the simulation programs framework. Particle simulations are subsequently performed and the results analysed.
- **Investigate industrially relevant Reynolds numbers.** Update the cylin-

der case for higher Reynolds numbers to investigate the usability of rCFD within the automotive industry.

- **Investigate different extraction frequencies.** Change the extraction frequency for the database to investigate possibilities and limitations regarding time, storage and accuracy.
- **Apply rCFD in STAR-CCM+ on a small complex geometry.** Create a new database for flow over an EX40 mirror and use it to get a new recurrence path. The implementation is done similarly as before.

1.3 Limitations

To remain within the time frame of 19th of January to 11th of June, 2026, certain limitations are defined. The implementation of rCFD is conducted exclusively in STAR-CCM+, with no generalization towards other fluid simulation software. Another limitation is the restriction of modifications in STAR-CCM+. The rCFD generated fluid field has limited integration into the software and no further modifications to the software itself are done.

Only drag force is considered to influence the particles in the cylinder case, hence, gravity among other forces are neglected. These limitations differ for the complex geometry, whereas forces such as drag, gravity and pressure gradient are included. Furthermore, all particles are assumed to be sufficiently small for the selected modelling approach to remain valid.

1.4 Societal, ethical and ecological aspects

Utilizing rCFD for rapid CFD predictions may result in less computational power being needed for future simulations. A lower computational cost has an impact on both societal and environmental aspects. By reducing the computational cost, less electrical power is needed to run the simulations, which is beneficial from an environmental perspective. A reduced computational cost also results in more efficient workflows, making it possible to find more sustainable and safer design solutions earlier. A consequence of lowered computational power may be a reduction in the workforce, since fewer people might be needed when the computational cost is reduced. On the other hand, rCFD may also create more job opportunities, as it creates a new field for development.

2

Theory

This chapter introduces the fundamental principles of fluid dynamics relevant to this thesis, displaying an overview of turbulence modelling and an introduction to multiphase flows. Particular attention is paid to Lagrangian particle tracking. Furthermore, Recurrence CFD is explained with its different methodologies; fluid based and transport based.

2.1 Fluid dynamics

Fluid dynamics is the study of fluids in motion and is a topic within the subject of fluid mechanics. Flow can either be laminar or turbulent. Several fluid properties are used to describe the physical behaviour of the fluid, including density, viscosity, temperature and pressure. Engineering problems that are related to fluid dynamics contain a fluid of varying fluid properties. Such fluid is called continuum [28]. Fluid dynamics is often described through several equations, such as Navier-Stokes. Knowledge of these equations is needed to understand the fundamentals and behaviour of fluids. A fluid can develop into turbulent flow, characterized by fluctuations in velocity and pressure, resulting in a chaotic nature and high Reynolds number, still following the Navier-Stokes equations [29].

The irregularity of turbulent flows is rather complex to solve directly and the governing equations must therefore be modelled [10]. In practice, turbulent flows are predicted using Computational Fluid Dynamics. A CFD software can be utilized for computation, combining turbulence models with solution algorithms. When calculating gradients of a fluid, different coordinate systems can be used depending on the problem. These coordinate systems are defined as Cartesian, Cylindrical and Spherical [9].

2.1.1 Governing equations

The Navier-Stokes equations are based on two of the three physical laws of conservation; mass and momentum [10].

When working with fluids, the most common approach is to deal with a region as a control volume. This allows the observer to acknowledge changes within a specific boundary of the fluid rather than following parcels. The conservation of mass is

obtained directly from the control volume as following

$$\frac{\partial}{\partial t}\int_V \rho dV + \int_S \rho \mathbf{u}\cdot\mathbf{n}dS = 0 \implies \frac{\partial \rho}{\partial t} + \nabla\cdot(\rho\mathbf{u}) = 0. \tag{2.1}$$

Where it can be simplified by using Gauss' divergence theorem [7]. In Eq. (2.1), the velocity vector of the fluid is defined as $\mathbf{u}$, the fluid density ρ and the time t. For an incompressible fluid (i.e. constant density), the following simplification can be used [26],

$$\nabla\cdot\mathbf{u} = 0. \tag{2.2}$$

Further, the conservation of momentum can be written in numerous of ways. By using the control volume method as for the mass conservation, the conservation of momentum is obtained as

$$\frac{\partial}{\partial t}\int_V \rho\mathbf{u}dV + \int_S \rho\mathbf{u}\mathbf{u}\cdot\mathbf{n}dS = \sum \mathbf{F}, \tag{2.3}$$

where $\sum \mathbf{F}$ represent the summation of forces acting on the control volume; surface- and body forces [7]. The body force is acting on the volume, denoted V and the surface force is working on the surface of the volume, denoted S [27]. This can be further simplified, leading to the Navier-Stokes equation. The Navier-Stokes equation for incompressible flow can be defined as

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u} = -\frac{\nabla p}{\rho} + \nu\nabla^2\mathbf{u}, \tag{2.4}$$

where ν is the kinematic viscosity and p the pressure [10] [7]. External forces can be appended by adding the term $\sum \mathbf{F}$ on the right hand side [5].

2.1.2 Force coefficients

A body surrounded by a fluid is subjected to aerodynamic forces developed from the pressure and shear stress distributions over the surface. For bluff bodies that are characterized by significant flow separations, the aerodynamic forces are prominent. These forces acting on the body are drag and lift. The magnitude depends on parameters such as velocity, fluid properties, and the shape of the body. Passenger vehicles are examples of bluff bodies.

To determine how aerodynamically efficient a certain shape is, the drag coefficient C_D is used. The drag coefficient is defined as

$$C_D = \frac{2F_D}{\rho u_\infty^2 A}, \tag{2.5}$$

where F_D is the drag force acting on the body, u_∞ the free stream velocity and A the frontal area [14]. Drag on a streamlined body includes both pressure drag and skin friction drag, where the latter is dominant. For a bluff body, most of the drag comes from pressure drag [9].

The lift coefficient C_L quantifies a body's ability to produce lift and is calculated as

$$C_L = \frac{2F_L}{\rho u_\infty^2 A}, \tag{2.6}$$

where F_L is the lift force [14].

Fluctuations may appear for the lift coefficient. These fluctuations of the lift coefficient can be averaged by using the Root Mean Square (RMS) equation. It can be calculated as

$$C_{L,RMS} = \sqrt{\frac{1}{T}\int_0^T C_L(t)^2 dt}, \tag{2.7}$$

where T is the time period [4] [26].

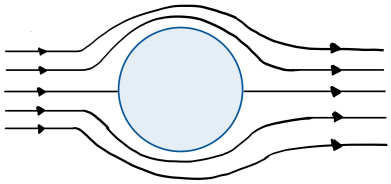
2.1.3 Reynolds number

To determine if a flow is laminar or turbulent, a dimensionless quantity known as the Reynolds number (Re) is used. The Reynolds number represents the ratio of inertial forces to viscous forces [11] and is defined as

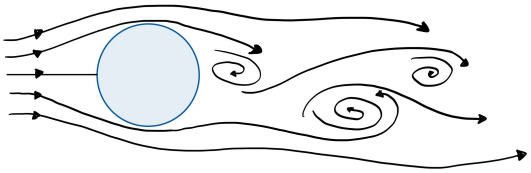
$$\mathrm{Re} = \frac{\rho u_\infty l}{\mu}. \tag{2.8}$$

Here, l denotes the characteristic length of the object. The dynamic viscosity μ and the density are defined for the operating fluid. For a passenger vehicle in motion, the Reynolds number is typically in the order of 10^6-10^7, based on a vehicle length of approximately 4-5 m and velocities in the range of 30-100 km/h, using Eq. (2.8).

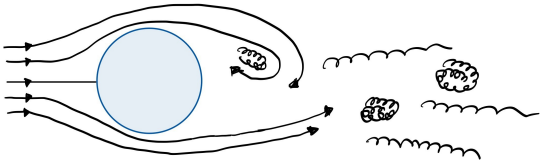
Flow with low Reynolds numbers are typically laminar, whereas high Reynolds numbers indicate turbulent flow. Transition between laminar and turbulent regimes depends on the fluid configuration. Flow around a cylinder has different characteristics depending on the Reynolds number. Higher Reynolds numbers result in increased oscillations of eddies behind the cylinder [17]. Eddies are spatial structures that appears in turbulent flows, where irregularities are present [17]. The development of the wake behind a cylinder with increasing Reynolds number can be illustrated in Figure 2.1.



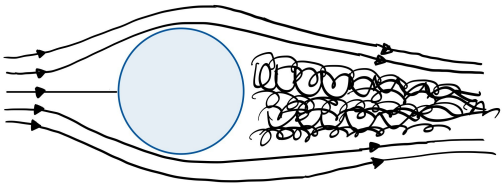
(a) Re < 1.



(b) Re $\sim 10^2$.



(c) Re $\sim 10^4$.



(d) Re $\sim 10^6$.

Figure 2.1: Illustration of theoretical flow around a cylinder for four different Reynolds numbers [6].

2.1.4 Strouhal number

The periodic vortex shedding behind a cylinder can be expressed using the dimensionless Strouhal number, S_{tr}. It describes the relationship between flow oscillations and the characteristic flow velocity and is defined as

$$S_{tr} = \frac{fD}{u_\infty}, \tag{2.9}$$

where f is the vortex shedding frequency and D the diameter of the cylinder [10] [7]. For a cylinder, the Strouhal number is approximately 0.2 over a wide range of Reynolds numbers [10].

2.1.5 Turbulence models

Several modelling approaches exist to approximate turbulent flows. Approximation of turbulence is rather intricate due to the appearance of eddies with different length and time scales that interacts with high complexity [26]. These modelling approaches are divided into three categories of turbulence methods; Models for Reynolds Averaged Navier-Stokes (RANS), Large Eddy Simulation (LES) and Direct Numerical Simulation (DNS) [26]. RANS is an early approach of turbulence modelling, with focus on the mean flow. By time averaging the incompressible Navier-Stokes equations, Eq. (2.4), the RANS equations are obtained [3]. To capture unsteady flow behaviour, Unsteady RANS (URANS) is evolved from the RANS model.

LES, apart from RANS, tracks the behaviour of the larger eddies in the flow [26].

The model captures small turbulent structures that cannot be resolved by a numerical scheme. One disadvantage of the model is the additional flow equations that must be solved, resulting in a higher computational cost [3]. DES is developed to reduce this cost, a method combining RANS and LES. Where RANS is used to resolve the boundary layers, while LES is applied outside the near wall region to capture detached eddies [3]. Delayed Detached Eddy Simulation (DDES) is an extended version of DES, in which the transition between RANS and LES is delayed, improving model behaviour and reducing computational cost [15].

DNS, is a method similar to URANS since it solves the mean flow. It has the most accurate approach to solve the Navier-Stokes without any approximations [7]. The model is not industrially used since DNS has an extremely high computational cost. This is due to the model solving all turbulent velocity fluctuations in the flow [26].

2.1.6 Solution algorithms

There are different solution algorithms available. Solution algorithms are used as a complement to solve the governing equations for incompressible flows, which lack independent equations for pressure [25] and velocity. There are different pressure-velocity coupling schemes that can be used to solve these equations. Two commonly used coupling algorithms are Semi-Implicit Method for Pressure Linked Equations (SIMPLE) [26] and Pressure-Implicit with Splitting of Operators (PISO) [25]. The main difference between them is the computational cost. SIMPLE is an iterative method to solve the equations by guessing values of the pressure [25]. PISO is known as an extension of SIMPLE but with an extra corrector step, increasing the accuracy of the results. Small time steps are recommended when using PISO since it relies on higher order temporal accuracy [26].

2.1.7 Parallel processing

When computing complex simulations, the importance of computer performance increases. High Performance Computing (HPC) is a major part within the topic. The definition of HPC is referred to the cooperation of technology, methodology and application to achieve higher computing capacity [24]. To attain higher computing capacity, the performance can be increased by computational parallelism [24]. Computational parallelism enables the possibility to perform multiple tasks at the same time, which in CFD translates to using multiple cores.

Functions and algorithms may not run efficiently in a software when applying parallelization. If they are designed for serial processing, the parallelization may not increase the speed of the process [7]. Additionally, since the processors exchange data between each other, it is not always beneficial to select as many cores as possible. Amdahl's law is commonly used to describe the correlation between the segment of the algorithm running serialized code versus the number of cores used for the parallelized code. The law proves that adding more cores than needed results in computational overheads. Where overheads refer to the system being over defined. Number of cores may therefore be selected carefully depending on the model that is

studied [16].

One solution to increase the efficiency of parallel processing is to divide the domain into subdomains and assigning one subdomain per core [7]. This can in a CFD software be done through a Partitioning Solver. A partitioning solver controls the domain decomposition resulting in individual subdomains. The partitioning solver ensures that the decomposition is updated throughout the simulation [21].

2.2 Multiphase flows

Multiphase flow is the presence of two or three phases in a domain. The definition of a phase is when a fluid exists as a solid, liquid or gas [30]. To simulate multiphase flows, there are several mathematical models. Due to the complexity related to turbulence, the modelling of multiphase flows remains an advanced task.

When modelling multiphase flows, there are two main modelling approaches; Eulerian-Lagrangian and Eulerian-Eulerian. The Eulerian-Eulerian approach considers the particles to be a continuum passing through a fixed volume. The Lagrangian, on other hand, studies the dynamics of individual particles [31].

2.2.1 Equations of motion for particles

The equations of motion for particles can be defined based on Newton's second law, $F = m \cdot a$ [13]. Where it can be combined with the Lagrangian approach, resulting in the particle motion being calculated as

$$m_P \frac{\mathrm{d}u_P}{\mathrm{d}t} = \sum \mathbf{F}, \tag{2.10}$$

$$\frac{\mathrm{d}x_P}{\mathrm{d}t} = u_P. \tag{2.11}$$

The mass of the particle is defined as m_P and $\sum \mathbf{F}$ the summation of forces acting on the particle [23] [12]. A Lagrangian approach enables the possibility to calculate the particle velocity u_P. Further, the particle position, x_P, can be updated through the calculated velocity.

2.2.2 Particle forces

There are two types of forces that acts on fluid particles; surface forces F_S and body forces F_V [26]. Particle tracking calculations varies within different commercial software. For STAR-CCM+ the surface forces consist of pressure gradient-, drag- and virtual mass forces. In addition, the body forces contain gravity, user-defined body force and Coulomb force [22].

The relevant forces, based on the limitations in Chapter 1.3, are the drag force $F_{D,P}$, pressure gradient force $F_{p,P}$ and the gravity force $F_{g,P}$. The total summation of forces acting on the particle resulting in

$$\sum \mathbf{F} = F_S + F_V = F_{D,P} + F_{p,P} + F_{g,P}. \tag{2.12}$$

Drag is the largest force affecting particles in a fluid domain [30]. The particle drag force, $F_{D,P}$ for spherical particles are defined as

$$F_{D,P} = \frac{1}{2} C_{D,P} \rho A_P |\mathbf{u}_s| \mathbf{u}_s, \tag{2.13}$$

where $C_{D,P}$ is the particle drag coefficient, ρ the density of the fluid and A_P the particle cross-section area. A_P is for spherical particles defined as

$$A_P = \frac{D_P^2 \pi}{4}, \tag{2.14}$$

with D_P being the particles diameter [2]. In Eq. (2.13), the slip velocity $\mathbf{u}_s$ is defined as the difference between the particle velocity $\mathbf{u}_P$ and the fluid velocity $\mathbf{u}$, as $\mathbf{u}_s = \mathbf{u} - \mathbf{u}_P$. To calculate the drag coefficient of the particle, the Schiller Naumann correlation is used [22] and defined as

$$C_{D,P} = \begin{cases} \frac{24}{\mathrm{Re}_P}(1 + 0.15\mathrm{Re}_P^{0.687}) & \mathrm{Re}_P \leq 10^3 \\ 0.44 & \mathrm{Re}_P > 10^3, \end{cases} \tag{2.15}$$

with Re_P being the particle Reynolds number. The particle Reynolds number is approximately calculated as

$$\mathrm{Re}_P = \frac{\rho |\mathbf{u}_s| D_P}{\mu}. \tag{2.16}$$

The dynamic viscosity is denoted as μ for the fluid. Another force acting on the particle is the pressure gradient force $F_{p,P}$ and is defined for spherical particles as

$$F_{p,P} = -\frac{\pi D_P^3}{6} \nabla p_{static}, \tag{2.17}$$

where ∇p_{static} is the static pressure gradient in the fluid [31]. In fluid domains where gravity is taken into account, a particles gravity force $F_{g,P}$ can be calculated as

$$F_{g,P} = m_P \mathbf{g}, \tag{2.18}$$

with $\mathbf{g}$ as the gravity vector [22].

2.3 Recurrence CFD

Recurrence CFD is a method dependent on recurrence statistics, which is composed by pairwise similarity measurements in a database. Once pseudo-periodic conditions are established, the database is created by saving the solved flow fields at converged time steps. Pseudo-periodic flow is characterized by its irregular fluctuations around a steady-state mean [19]. For the pairwise similarity measure for a flow based method, the distance norm measure, $\mathcal{D}$, is used on the database. It is defined by Eq. (2.19), where the normalization factor N is defined by Eq. (2.20) [1] and $\mathbf{r}$ is the position vector.

$$\mathcal{D}(t,t') = \left(\frac{1}{N}\int \left(\mathbf{u}(\mathbf{r},t) - \mathbf{u}(\mathbf{r},t')\right)^2 dr^3\right)^{1/2} \tag{2.19}$$

$$N = \max_{t,t'}\left(\int \left(\mathbf{u}(\mathbf{r},t) - \mathbf{u}(\mathbf{r},t')\right)^2 dr^3\right) \tag{2.20}$$

Solving Eqns. (2.19) and (2.20) over the whole database results in a distance matrix $\mathcal{D}_{ji}$. The distance matrix is composed of snapshots, which here are defined as the solved velocity fields dependent of the times t and t'. In the matrix the values varies from 0 to 1. A value of $\mathcal{D}_{ji} = 0$ symbolizes that the snapshots j and i are identical. Hence, if $i = j$ it implies that $\mathcal{D}_{ji}$ always equals zero. This will in practice always result in a main diagonal line with zeroes. Furthermore, the distance matrix is symmetrical over the main diagonal since $\mathcal{D}_{ji} = \mathcal{D}_{ij}$. The light blue off diagonal bands that are illustrated in Figure 2.2 represents a recurring pattern, in this case indicating that two whole oscillations are occurring in the database.

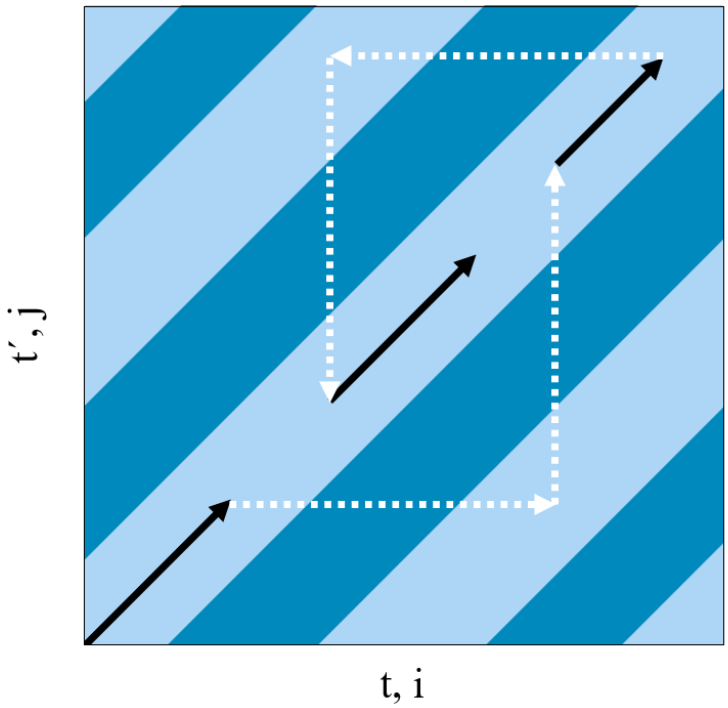


Figure 2.2: Creating the recurrence path from $\mathcal{D}_{ji}$.

From the computed distance matrix, a recurrence path is constructed and applied to extend the simulation. Firstly, an empty recurrence path is created where the starting point is the first element of the CFD database. Snapshots that are within a randomized segment length are then added to the recurrence path, where a maximum segment is the total length of the database. To increase the length of the path and thereby the simulation time, a snapshot similar to the one ending the current segment is selected from the opposite side of the database. This desired snapshot is found by moving horizontally in the distance matrix and minimizing $\mathcal{D}_{ji}$, then moving longitudinal back to the centreline, see Figure 2.2. The snapshot acts as the new starting point for the new segment from where the process continues until the path is of a desired length. In Figure 2.2, the black lines represent the snapshots added to the recurrence path from the $\mathcal{D}_{ji}$ and the white dashed lines the jumps.

2.3.1 Methods of rCFD

Two different types of rCFD methods exist; flow based and transport based. The flow based version is based on an Eulerian flow field approach, meaning that it uses snapshots of the flow field for the rCFD. Snapshots are added to a recurrence path and operate in a new order [19]. Appropriate fields to study are velocities in x, y, z directions. By reusing the already calculated flow field, the simulation is extended without performing further iterations. Hence, predicting the flow based on pseudo-periodicity. The predicted flow is used to evaluate particles positions using Lagrangian particle tracking, as if calculated alongside the flow field [8].

Transport based rCFD employs a Lagrangian framework by tracking inertialess fluid tracers. Instead of predicting the flow, this version focuses on finding recurring transportation of particles between the cells and storing them [19]. This means that a particle is observed as moving with information between cells without having to distinguish the path it takes within the time step. The particles are then mapped onto an Eulerian grid with their respective fields [18].

3

Methods

To implement the rCFD method in STAR-CCM+, a Java macro was developed. Before the macro was executed, a domain was defined and validated for convergence. The procedure of the macro is illustrated in Figure 3.1 as a flowchart, where red boxes denote processes executed in Java and blue boxes denote processes executed in Python. Initially, the process begun with creating the *Database Solution History*, by exporting data to a *Solution History* file, referred to as a simh file. The macro launched a Python code that imported the database. From the database a *Distance Matrix* was calculated to further create the *Recurrence Path*. Subsequently, the *Recurrence Path* was transferred to an *rCFD Solution History* file and imported during the *Import rCFD* step. The process continued with defining the *Field Functions* and freezing the solvers, *Freeze Solvers*. Then the *Mapping* step proceeded until the end of the macro.

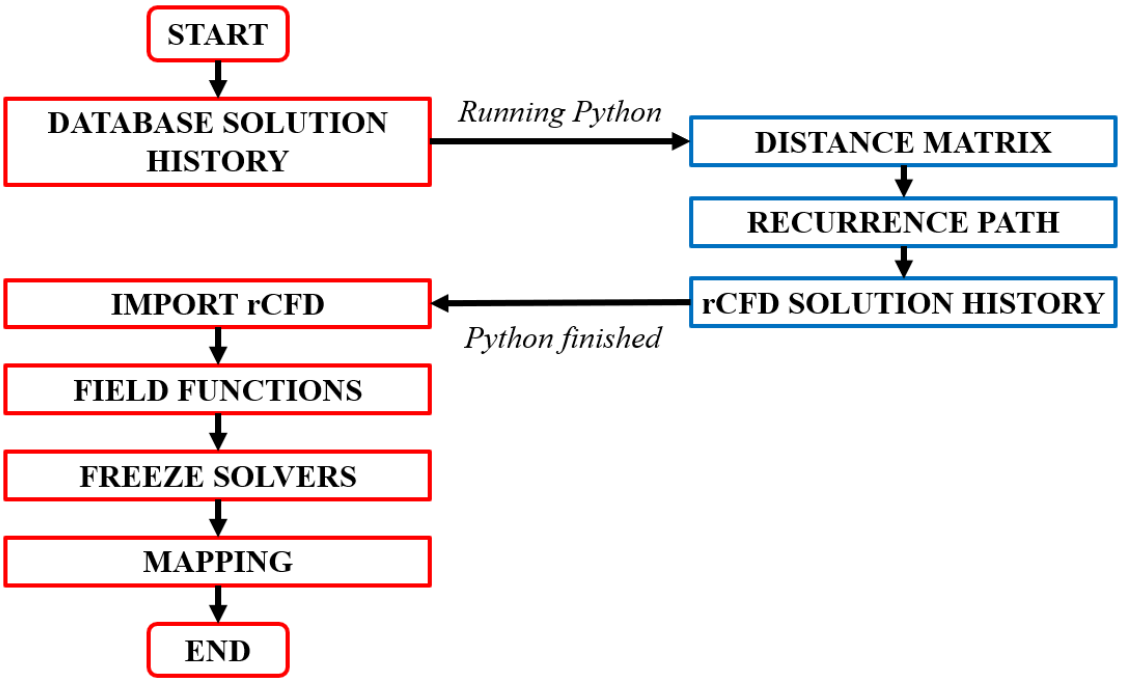


Figure 3.1: Flowchart of the codes used to implement the rCFD method.

The macro ran for the different geometries, including a cylinder and a mirror of an

EX40. Where the cases denoted as baseline, were simulated using conventional CFD for both of the geometries.

3.1 Create database

Creating a database containing fluid fields from a simulation was necessary to initialize the rCFD method. When the database was obtained, the distance norm measure, Eq. (2.19), could be applied. From this a recurrence path was determined and transformed into a software compatible file format. Furthermore, validations and results were obtained.

3.1.1 Define the domain

To acquire the database, a domain was set up. The first case contained a cylinder clearly exhibiting vortex shedding when affected by an inlet velocity. The fluid domain was defined according to the coordinates in Table 3.1 [8].

Table 3.1: Domain coordinates for the cylinder case.

	x	**y**	**z**
Start coordinate [m]	-20	-20	0
End coordinate [m]	40	20	12

The boundaries of the domain were defined as seen in Figure 3.2. At the coordinates [0,0,6] m a cylinder with a length of 12 m and diameter of 2 m was positioned inside the fluid. Hence, the cylinder was spanning through the entire z-direction of the domain.

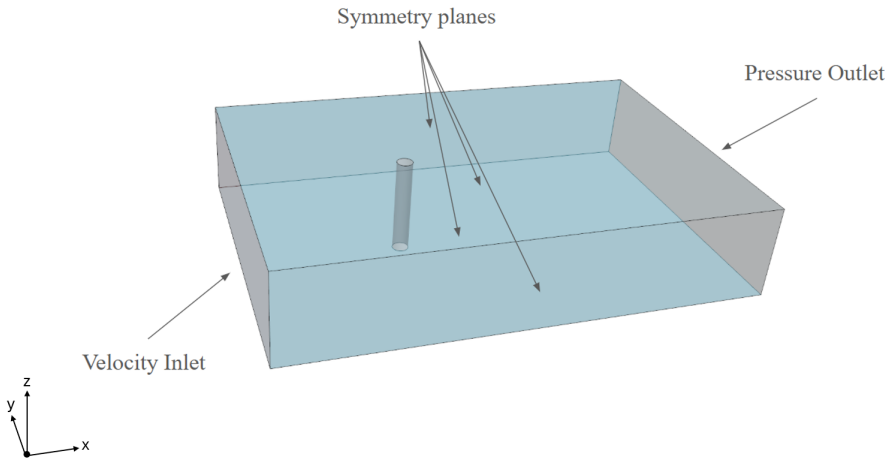


Figure 3.2: Fluid domain with boundary conditions.

To simplify the procedure selecting the Reynolds number, the density for the working fluid was set to 1 kg/m^3.

Databases were created for various Reynolds numbers such as 10^2, 10^4, 10^5 and 10^6. This was done to observe how different Reynolds numbers would affect the results, including industrially relevant magnitudes. The inlet velocity was set to 1 m/s for all of the cases. Further, the fluid viscosity was set according to Eq. (2.8), based on the selected Reynolds number and the chosen cylinder diameter.

For the mesh of the domain, a polyhedral mesh was used along with prism layers close to the cylinder surface. The prism layers were fine enough to ensure that y^+ did not exceed approximately a value of one. This verified that the viscous sublayer was sufficiently resolved in the simulation. Each prism layer cell received a stretching of 1.1.

To capture the gradients of the flow, two refinement regions were defined in the fluid domain. They were placed in regions where strong gradients were expected, such as around the cylinder and in the wake, see Figure 3.3.

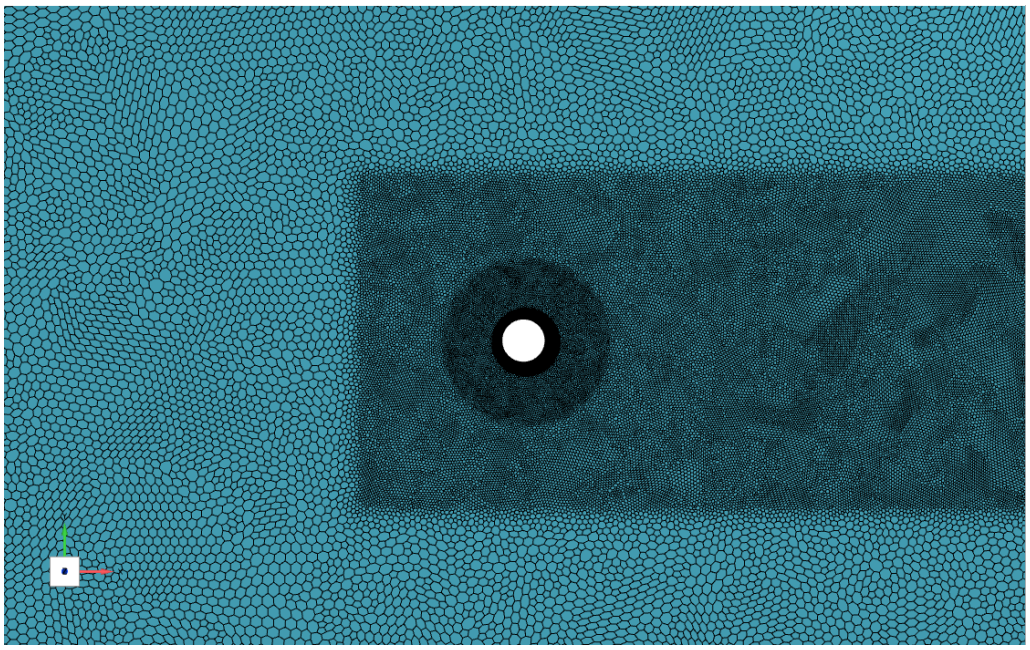


Figure 3.3: Meshed domain for a Reynolds number of 10^4 with a rectangular and circular refinement capturing the wake and turbulent region close to the surface.

One refinement region surrounded the cylinder, with a diameter of 5 m, capturing the gradients close to its surface. Furthermore, a rectangular refinement was used to thoroughly resolve the whole wake including the stagnation area in the front of the cylinder. The final mesh of the domain was approximately $2.3 \cdot 10^6$ cells for a Reynolds number of 10^4. When changing the Reynolds numbers, the meshing settings were modified. Their corresponding mesh refinement, number of cells among other parameters are presented in Appendix B.

Another simulation parameter was the time step, which was set to 0.00625 s. The DES turbulence model was employed together with the PISO solver due to its stability for time dependent cases.

3.1.2 Validation of convergence

To investigate the validity of the simulation, ensuring that the mesh is fine enough to properly solve the flow, several parameters were compared to existing studies. The parameters consisted of C_D mean (Eq. 2.5), $C_{L,RMS}$ (Eq. 2.7) and the Strouhal number (Eq. 2.9). For the validation, a length to diameter ratio of six was used, corresponding to the ratio employed in the referenced studies. This enabled a direct comparison with the values presented in Table 3.2 [14]. The physical time interval considered for the validation ranged from 200 s to 600 s.

Table 3.2: Values for fluid convergence comparability at a Reynolds number of 10^4, for a length to diameter ratio of six.

	C_D **mean**	$C_{L,RMS}$	S_{tr}
Nguyen and Nguyen (2016) [14]	1.1329	0.3629	0.1961
This thesis	1.1474	0.3431	0.2002

3.1.3 Data extraction

The database was created by extracting data from the verified simulation, containing snapshots of the flow fields. To save time and storage, a restriction was made to only extract the velocity from the flow. It should be emphasised that this was not a limitation of the rCFD but rather a deprioritization when conducting this thesis. Five field functions were exported from the software including a velocity vector, see Table 3.3. The velocity scalars in the directions x, y and z were used for the distance norm calculations, Eq. (2.19), while the velocity magnitude was used for post processing. Furthermore, the velocity vector was applied later for data mapping, see Chapter 3.3.2.

Table 3.3: Chosen input functions for the database. It should be noted that the directions are denoted as [x, y, z] in this thesis and [i, j, k] in STAR-CCM+.

Input functions	**Type**
Velocity	Vector
Velocity: magnitude	Scalar
Velocity x	Scalar
Velocity y	Scalar
Velocity z	Scalar

Extraction of the velocity fields was initiated once the simulation had reached pseudo-periodicity. Every 16th time step within a limited physical time span were

exported into a *Solution History* file. The chosen time span ensured that the database covered three vortex shedding periods, to capture the main flow features, see Appendix B.

3.2 Recurrence CFD

Once the database file was finalised, the *Solution History* file was used to create the recurrence path. Firstly, the database was imported from STAR-CCM+ to Python, to simplify the handling of the *Solution History* file. In Python, an empty *Solution History* file was created, with its original structure intact, matching the structure of the database file. The velocities in x, y and z directions were used to calculate Eq. (2.19) and Eq. (2.20), which composed the distance matrix $\mathcal{D}_{ji}$. The matrix was validated by visually locating off diagonal lines, which implies that there exist periodicity. Once the distance matrix was computed, a recurrence path could be constructed from snapshots of the velocity fields. For each snapshot within the recurrence path the corresponding snapshot from the database was fully copied and added into the rCFD *Solution History* file. This was performed until desired physical time for the rCFD simulation was achieved.

The rCFD file was automatically loaded into the STAR-CCM+ simulation. Within the software, the variables stored in the rCFD file, such as velocity magnitude, could be visualized and treated as if they had been directly computed by STAR-CCM+. Additionally, an output text file was saved during the Python scripts execution, containing valuable information such as the recurrence path and accuracy of the rCFD. The calculated distance norm values were saved in a Comma-Separated Values (CSV) file. As a result, the matrix only needed to be calculated once for a given database. Further iterations of rCFD were then executed with the benefit of reduced calculation time.

The accuracy of the rCFD was calculated based on the difference between the snapshot at the end of a segment length and the start of the next segment. The value was collected directly from the corresponding position in the distance matrix $\mathcal{D}_{ji}$, defined as $\mathcal{D}_k$. Then the total rCFD accuracy was calculated as

$$\mathcal{D}_{ji,accuracy} = 1 - \frac{\sum_{k=1}^{n} \mathcal{D}_k}{n}, \tag{3.1}$$

where n represents the number of jumps in the recurrence path.

3.3 Particles

To simulate the particles using rCFD, their properties and injection positions were selected. Furthermore, the particles movement was calculated by combining particle forces and the rCFD velocity field.

3.3.1 Defining particles

For the cylinder case, the particles were defined as solid spheres of H_2O, with a constant density of 1000 kg/m^3 [8]. These were injected through an injection plane in the domain, see Table 3.4. To optimize the computational resources required for the simulation, parcel depletion was implemented using a field function that defined the particle removal plane, see Table 3.5. Visualised in Figure 3.4 are the injection and removal planes. The field function for the parcel depletion can be found in Appendix A.

Table 3.4: Injection plane.

	x	**y**	**z**
Start coordinate [m]	-10	-1	0
End coordinate [m]	-10	1	12

Table 3.5: Removal plane.

	x	**y**	**z**
Start coordinate [m]	8	-20	0
End coordinate [m]	8	20	12

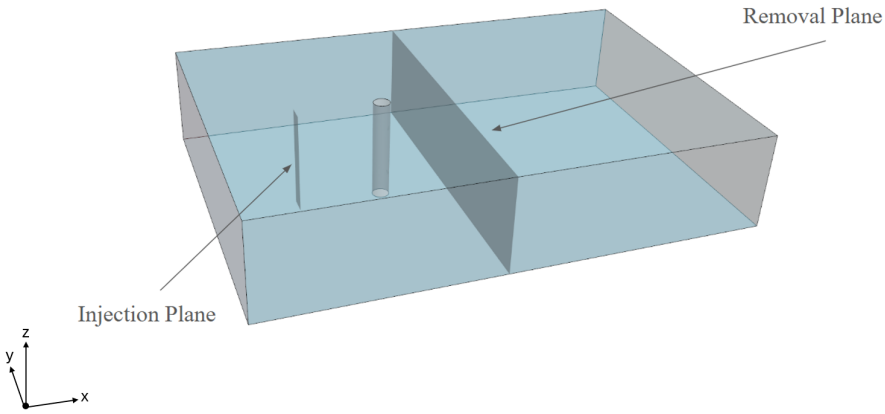


Figure 3.4: Fluid domain with particle injection and removal plane.

The particles were injected over a definite time period. To visualise the final deposition of the particles, the simulation continued running after the injection period for a specified time. The specified time was set to twice the physical time of the respective database. Particle size was determined by using Eq. (3.2) and was updated for every Reynolds number since the smallest cell size differ.

$$\delta \approx \frac{D_P}{2} \tag{3.2}$$

The radial size of the first cell was defined as δ, thereby determining the smallest particle diameter that could be resolved within the boundary layer. In Appendix B the particles diameter for the different Reynolds numbers can be seen.

3.3.2 Simulate particles with rCFD data

To simulate the particles with the recurrence CFD data, a Lagrangian Particle Tracking (LPT) solver was implemented. The built in particle tracking in the software was not applicable for the rCFD data. To calculate the particle velocity, the velocity vector was used from every snapshot in the rCFD *Solution History*. For the LPT implementation, *User-Defined Body Force* was enabled and its method was set as a field function. By using Eqns. (2.13) - (2.16) the field functions needed to calculate the drag force were computed. The *User-Defined Body Force* $\mathbf{F}_u$ was defined as following

$$\mathbf{F}_u = V_P \mathbf{f}_u \tag{3.3}$$

and thereby the drag force had to be divided by the particle volume V_p when being used in STAR-CCM+ [22], see Appendix A. The external force acting on the particle was limited to drag force, therefore being the only force calculated for the cylinder case.

To update the particle velocity based on the created rCFD velocity field, data mapping was used. Mapping allows the user to insert previous extracted fields into the simulation. Since the mapping only could be done for one snapshot at a time to get the corresponding velocity field, the process was automated. This process included selecting a specific snapshot and mapping its velocity field. By running the simulation for one rCFD time step,

$$\Delta t_{rCFD} = \Delta t_{database} \cdot frequency, \tag{3.4}$$

the particle velocity was updated based on the calculated drag force. Since there was a software limitation when recreating the particle distribution, additional grid layers were created for the injector, based on the rCFD time step. By calculating the approximate position that the particles moved in the x-direction for one time step of 0.00625 s, new grid layers were created. Therefore, enhancing the visualization for the simulated particles when implementing the rCFD method. Figure 3.5a shows the injector for the baseline case with only one grid layer, while Figure 3.5b shows the same grid layer but with 15 identical layers added to the injector. These were added to compensate for the remaining starting positions when using the rCFD time step. The distance between the grid layers was set to 0.0057 m. When running for

other frequencies, the layers had to be compensated to suit the rCFD time step, leading to 32 and 64 layers for frequencies 32 and 64, respectively.

(a) One grid layer. **(b)** 16 grid layers.

Figure 3.5: Frequency dependent particle injection for the cylinder with a frequency of 16.

3.4 Comparison

Time and file size comparisons were carried out to investigate the differences between the rCFD method and the currently used CFD method. Here they are called the rCFD case and baseline case, respectively. To get as accurate comparison as possible the simulations were based on the same case, hence, the same starting conditions. To accommodate for the simulation time needed for the rCFD database, the *Pre injection* was run for both cases for the same amount of time, but without exporting to a *Solution History* file for the baseline. The *Distance matrix* $\mathcal{D}_{ji}$ segment was only relevant for the rCFD case and it added no extra physical time to the simulation. During the *Injection* segment particles were injected at the same mass flow rate of 1 kg/s within the same time frame. The particles moved freely for the *Post injection* without any new particles added until the simulation end time was reached. The different segments were timed according to Table 3.6, noting the start and end time for each new occurring segment. In total, the simulation time corresponded to the end of *Post injection* subtracted by the start of *Pre injection*.

Table 3.6: Time segments for time comparison between the baseline and rCFD case, including physical simulation time for each segment.

Baseline case	**rCFD case**	**Physical time [s]**
Pre injection	Pre injection	31
–	Distance matrix $\mathcal{D}_{ji}$	–
Injection	Injection	10
Post injection	Post injection	60

The total storage was calculated based on the size of the Simulation File (SIM) along

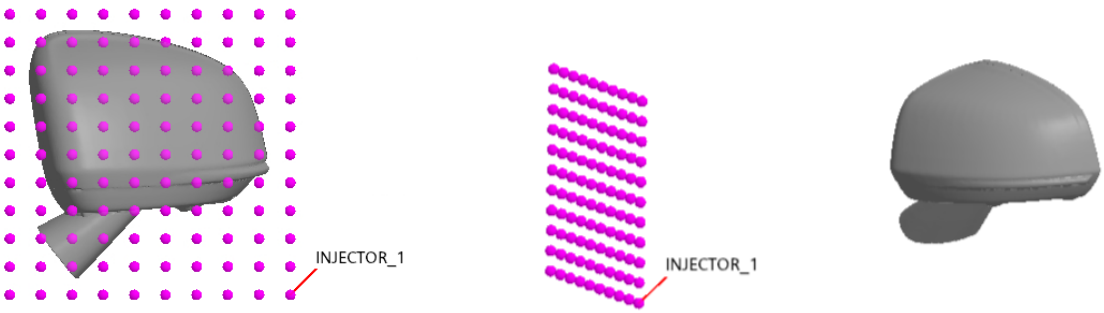
with the size of the *Solution Histories* for the rCFD case. Time and storage tests were executed for a Reynolds number of 10^4 combined with frequencies of 16, 32 and 64 for the rCFD case. An accuracy test was conducted, based on the calculated accuracy from Eq. (3.1), including visually exerting the velocity field over an rCFD jump.

A particle deposition comparison test was also executed by evaluating the rear side of the cylinder. Hence, analysing the particles final positions for the two cases. The same amount of particles were dispersed into the domain with the same properties. Changes were done for the particle deposition comparison. The particle mass flow rate was doubled as well as the resolution of the grid of the injection plane. The mesh was also updated to enable the use of smaller particles. It resulted in approximately 10 million cells and a particle diameter of 0.8 mm, see Appendix B for more details. The changes ensured that the amount of particles recirculating back onto the cylinders rear side increased, compared to previous setup. The simulation ran for the same physical times as in Table 3.6 with the exception of *Post injection*, which ran for 125.7 seconds.

3.5 EX40 mirror

The rCFD method was implemented on a mirror of a Volvo EX40 to further challenge the method. To compare the rCFD result of the mirror, two cases were executed; baseline and rCFD. The baseline was solved using conventional CFD, meanwhile the other case utilized rCFD to extend simulation time. A suitable size of the domain was chosen to capture the flow around the mirror. The mesh was generated using previously established parameters and consequently no further mesh validation was performed. The generated mesh of the domain contained 6.4 million cells and the mirror was exposed to a velocity of 70 km/h.

The mirror was exposed to snow particles, injected through 110 injector points per grid layer. These were placed in front of the mirror with a particle flow rate of 2000 particles per second, see Figure 3.6.



(a) Front view of the injector. **(b)** ISO view of the injector.

Figure 3.6: EX40 mirror and injector points for different views.

The distance between the layers was calculated to 0.08 mm, corresponding to the largest particle diameter in the domain. Creating additional grid layers was important to enable comparison of the results from the baseline case with the rCFD case.

The database was generated in the same way as in Chapter 3.1, performed with the implicit solver instead of PISO. Furthermore, the recurrence path was obtained using the same procedure as in Chapter 3.2, with an extraction frequency of 16. To accommodate for the rCFD time step, 15 grid layers were added. Additional forces were included in the *User-Defined Body Force* for the mirror simulation. Specifically, the gravitational force, Eq. (2.18), and the pressure gradient force, Eq. (2.17), were incorporated to further improve the LPT solver. All of the applied forces such as drag, gravity and pressure force were divided by the volume, to fit into the units of STAR-CCM+ (Eq. 3.3).

At the end of the simulations a time, storage and accuracy comparison was conducted to compare the simulation results of the mirror with and without the rCFD method. These tests were done in the same way as in Chapter 3.4. One difference between the cylinder and EX40 mirror simulations was that the *Post Injection* procedure was not included for the mirror case. To optimize the calculating performance, the number of cores used for the simulation was determined by using one core per 10 thousand cells. This led to a total number of 640 cores for the simulation. To obtain comparable results, the number of used cores were the same for the rCFD and baseline case.

4

Results cylinder

This chapter presents the results from the cylinder case. Distance matrices for the different Reynold numbers are displayed including respective accuracy from the recurrence path. The results also contain a comparison between velocity fields, as well as particle deposition. Moreover, the outcome from the storage and time tests will be displayed with histograms and plots.

4.1 Distance matrices

The obtained distance matrices from the cylinder case are shown for four different Reynolds numbers; 10^2, 10^4, 10^5 and 10^6, see Figures 4.1 - 4.4. For the following Reynolds numbers, the same frequency of 16 has been used when exporting the database. In Figure 4.1, the Reynolds number was set to 10^2. It can be seen that the periodicity is clear due to the distinct off diagonal lines appearing in the plot. A similar pattern to the main diagonal can be observed for the off diagonal lines.

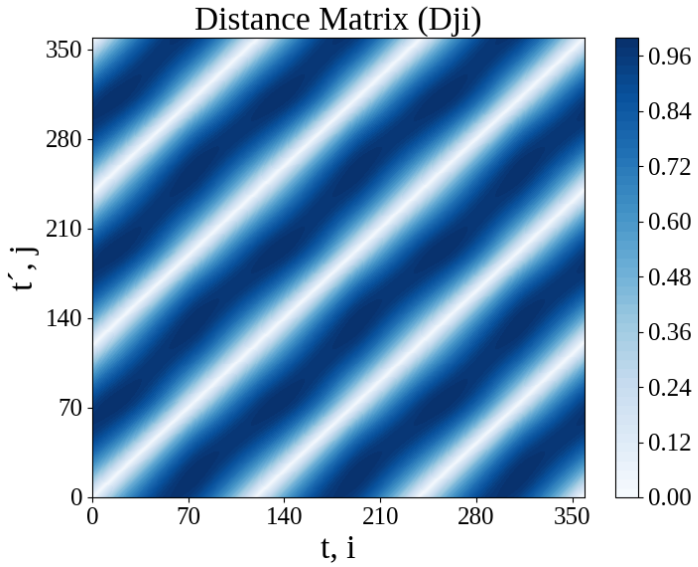


Figure 4.1: Distance matrix for a Reynolds number of 10^2.

By observing the distance matrix in Figure 4.2 for Reynolds number 10^4, the main

diagonal line can be seen to appear clearly. The off diagonal lines are less defined compared to the distance matrix of Reynolds number 10^2. They are still visible, but not appearing as distinct as for the lower Reynolds number.

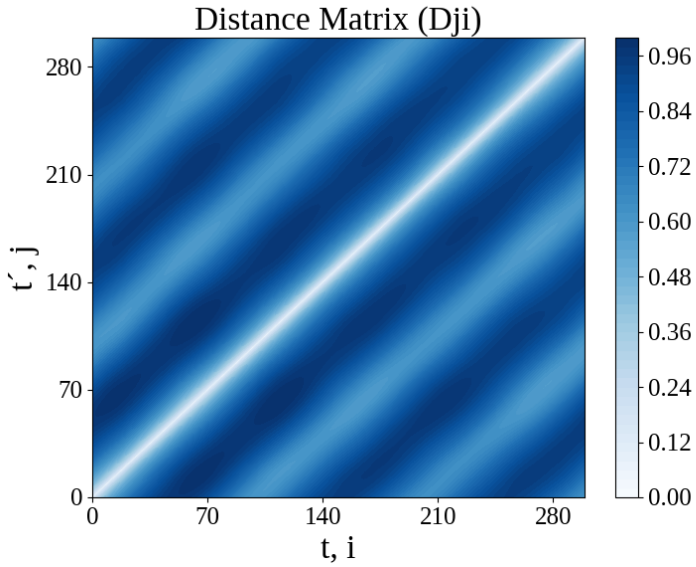


Figure 4.2: Distance matrix for a Reynolds number of 10^4.

Figure 4.3 shows the distance matrix for Reynolds number 10^5 and has very similar appearance as for 10^4. One remarkable thing is that the distance matrix for Reynolds number 10^5 has less defined off diagonal lines after approximately snapshot 160.

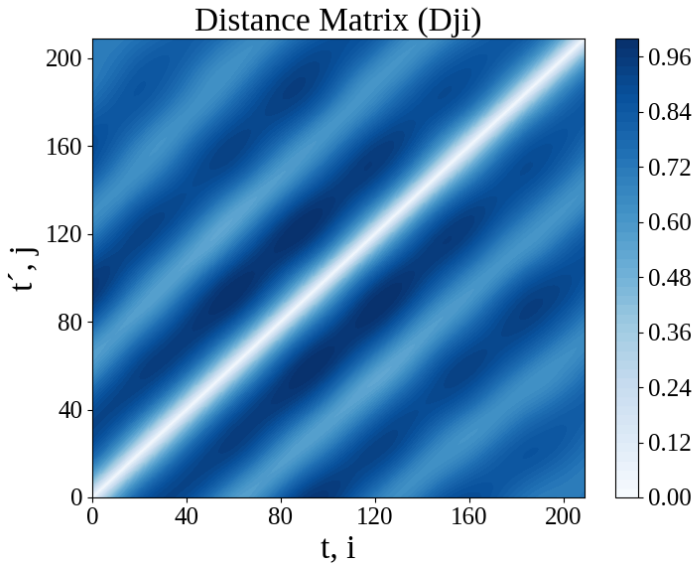


Figure 4.3: Distance matrix for a Reynolds number of 10^5.

Figure 4.4 displays the distance matrix for Reynolds number 10^6. The off diagonal

lines are still visible but not as distinct as for the other Reynolds numbers. It can also be noted that the outer edges of the off diagonal lines are jagged.

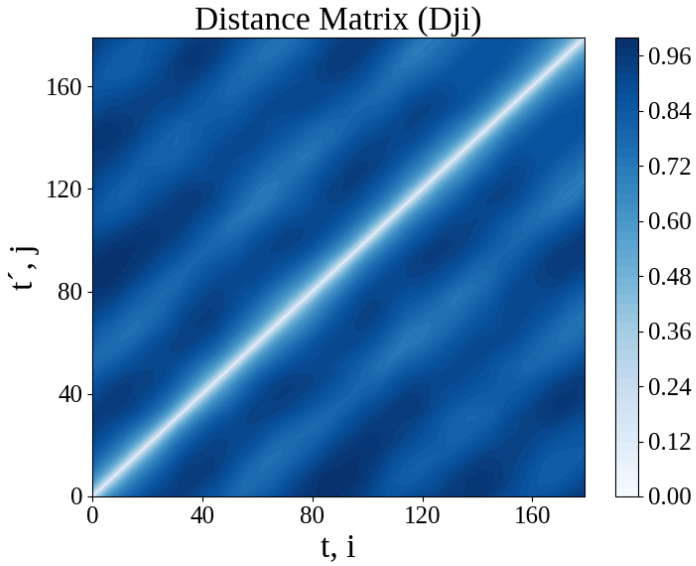
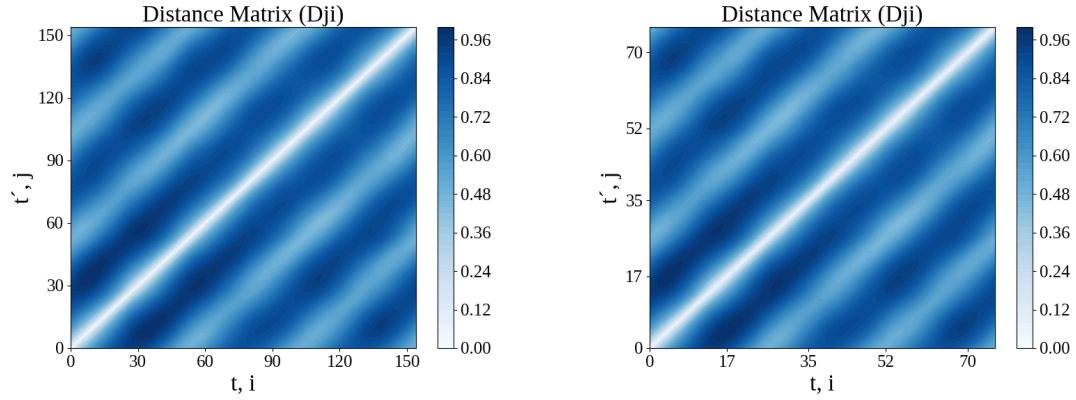


Figure 4.4: Distance matrix for a Reynolds number of 10^6.

The distance matrices for extraction frequencies 32 and 64 are shown in Figure 4.5 for Reynolds number 10^4. Both matrices show similar patterns to the matrix corresponding to a frequency of 16. The main difference is that the off diagonal lines are less pronounced and more irregular. It can be observed that they have a similar appearance to each other.



(a) Frequency 32. **(b)** Frequency 64.

Figure 4.5: Distance matrices for a Reynolds number of 10^4 with frequencies 32 and 64.

4.2 Velocity field and accuracy

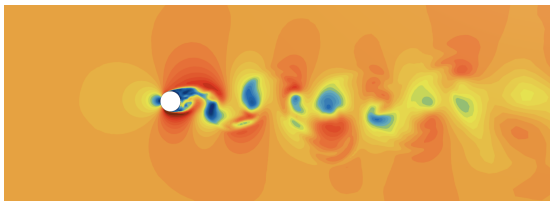
The overall rCFD accuracy for the different Reynolds numbers for the cylinder case are displayed in Table 4.1. It can be noticed that the accuracy decreased as the Reynolds number increased. With an accuracy for Reynolds number 10^2 of 99.66 % and decreasing to 29.32 % for Reynolds number 10^6. The largest difference is seen between Reynolds number 10^2 and 10^4 with a deviation of approximately 48 percentage points. It can also be noticed that the accuracy slightly increases with larger extraction frequency values for Reynolds number 10^4 from 51.58 % to 53.88 %.

Table 4.1: Accuracy of rCFD for the cylinder case.

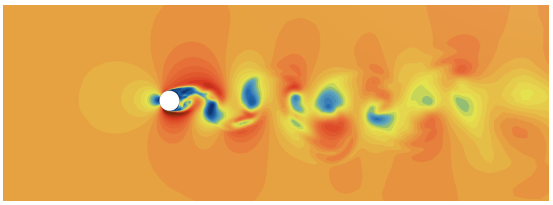
Reynolds number	**Frequency**	**Accuracy** [%]
10^2	16	99.66
10^4	16	51.58
10^4	32	52.67
10^4	64	53.88
10^5	16	48.61
10^6	16	29.32

To visually evaluate the rCFD methods accuracy the velocity fields were compared side by side. By showing one snapshot together with its upcoming snapshot and thereby catching the changes over a rCFD jump. The velocity field was taken from the middle of the domain where z=6 m and the normal was [0,0,1].

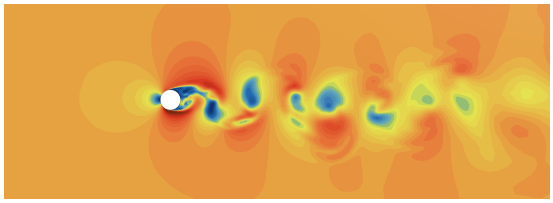
In Figure 4.6 the velocity magnitude fields between two snapshots can be seen. Where Figures 4.6a and 4.6b represent the baseline case for Reynolds number 10^4 with a frequency of 16. Figure 4.6c and 4.6d represent the first rCFD jump for the velocity field, showing the difference between two snapshots. Note that Figure 4.6a and 4.6c show the same velocity field, i.e., the same snapshot. It can be seen that there is no clear visual difference in the baseline case between the snapshots. For the rCFD case, however, differences can be observed in the wake. The flow field appears more similar close to the cylinder surface, while larger deviations are observed further downstream compared to the baseline case. The accuracy for the first jump was 46.28 %.



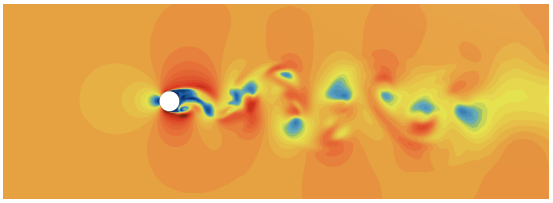
(a) Snapshot 57, baseline case.



(b) Snapshot 58, baseline case.



(c) Snapshot 57, rCFD case.



(d) Snapshot 58, rCFD case.

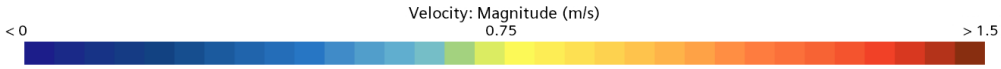
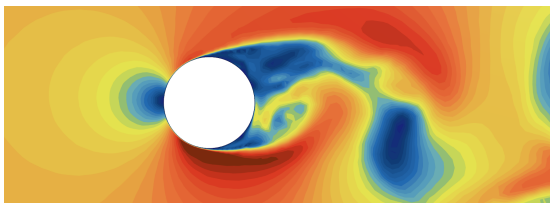


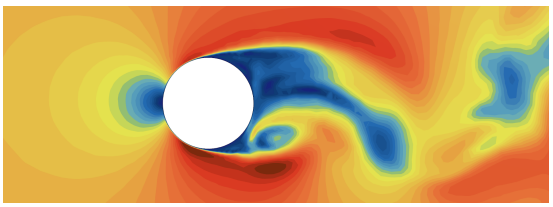
Figure 4.6: Snapshot comparison for the baseline and rCFD cases with frequency of 16 and Reynolds number 10^4.

A close up on the velocity magnitude field near the cylinder is seen in Figure 4.7. The figures show snapshot 58 after the jump for rCFD and the baseline, representing the same physical time in the simulation. Overall, the wake structure appears similar in both shape and direction, capturing the general flow behaviour

The velocity field magnitude comparison for Reynolds number 10^2, 10^4, 10^5 and 10^6 is found in Appendix C, including the extraction frequencies of 32 and 64 for 10^4.



(a) Baseline case.



(b) rCFD case.

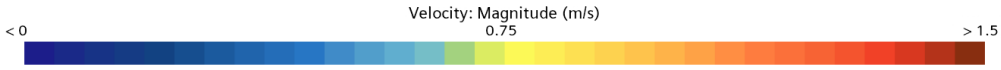


Figure 4.7: Zoomed in snapshot comparison of snapshot 58 for the baseline and rCFD cases with a frequency of 16 and Reynolds number 10^4.

4.3 Particles

In Figure 4.8 the particle deposition over the cylinders rear side can be seen for the baseline case, displaying the particles stuck to the surface. It is evident that there are accumulation zones at the ends of the cylinder. Overall, the particles are centred horizontally along the middle of the cylinder.

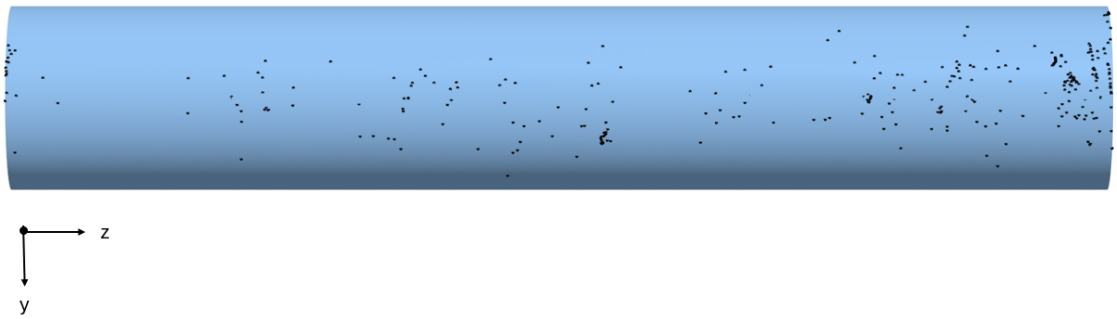


Figure 4.8: Particle deposition on the rear side of the cylinder for the baseline case with Reynolds number 10^4.

Figure 4.9 visualises the particle deposition when the rCFD method was used. An accumulation zone can clearly be seen at the right end of the cylinder. There are generally more particles located towards the right side of the cylinder, similarly as for the baseline case.

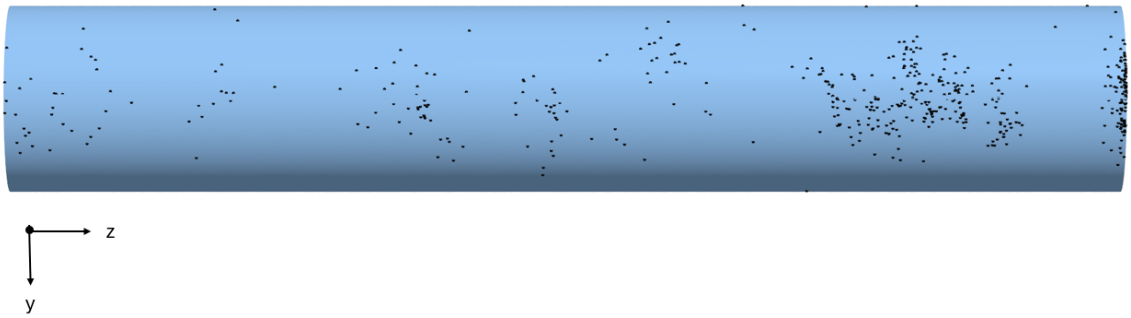


Figure 4.9: Particle deposition on the rear side of the cylinder for the rCFD case, with a frequency of 16 and Reynolds number of 10^4.

4.4 Storage and time comparison

In Figure 4.10, a histogram over the time duration for each segment can be seen. The time tests over the cylinder were simulated with Reynolds number 10^4 with frequencies of 16, 32 and 64 for the rCFD case. These are defined as rCFD_{16}, rCFD_{32} and rCFD_{64}, respectively.

The baseline case took a total of 61.59 minutes, where the main time was spent on the *Post injection* when tracking the particles. When comparing the rCFD_{16}

case with the baseline, it is evident that the baseline case was approximately two minutes faster. The other frequencies were faster with a large margin, 39.76 minutes for $rCFD_{32}$ and 34.8 minutes for $rCFD_{64}$. It can be noticed that the time did not decrease linearly with larger frequency values. For all the rCFD cases the main time can be seen spent on the *Pre injection* segment. Moreover, the *Post injection* for all of the rCFD frequencies were faster than for the baseline case.

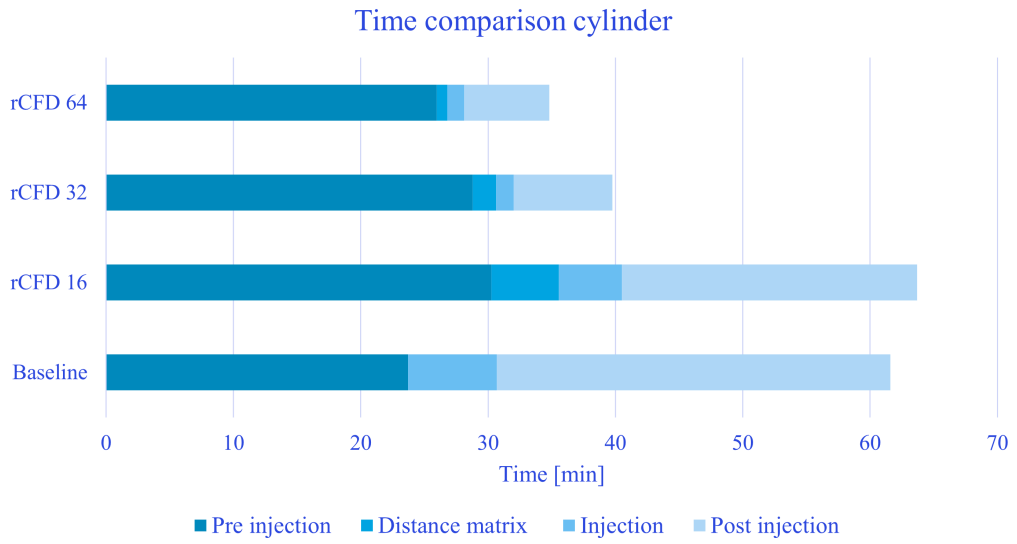


Figure 4.10: Duration for each segment for the baseline case and rCFD cases, units in minutes.

In Figure 4.11 the real time duration is visualised against the physical time in the simulation. The *Pre injection* is seen from physical time zero to 31 seconds and *Injection* is between 31 to 41 seconds. Within the *Injection* segment it can be seen that $rCFD_{64}$ overtook the baseline, thereby being the faster method. Between 41 and 101 seconds is the *Post injection* segment. $rCFD_{32}$ overtook the baseline case almost instantly after the *Injection* was finished.

It can clearly be noticed that the last segment, *Post Injection*, has the steepest slope for the baseline case. Indicating a less efficient particle tracking. Additionally, the baseline is more consistent in speed meanwhile the rCFD becomes faster for the injection and particle tracking.

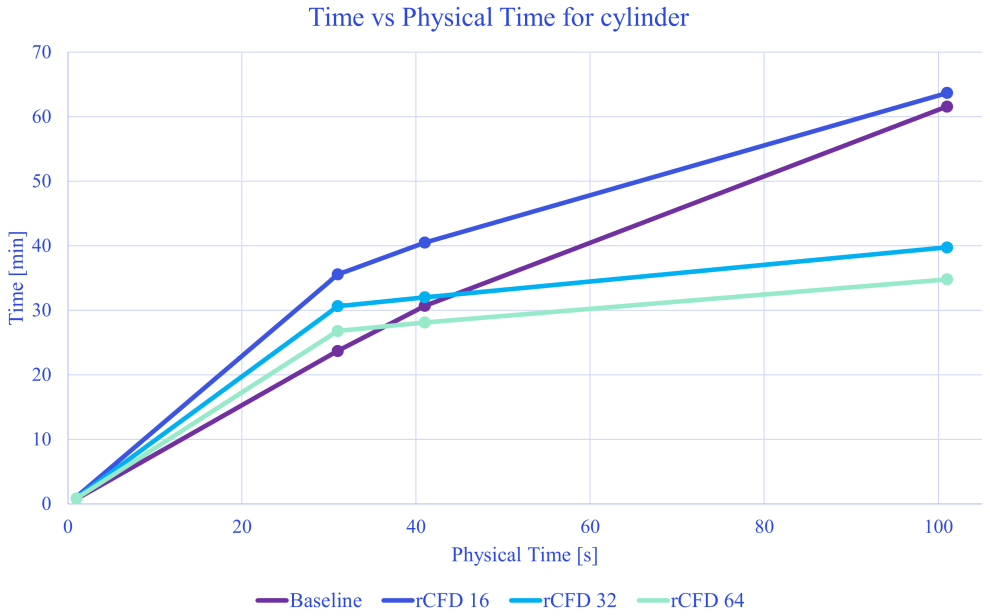


Figure 4.11: Duration for the baseline case and rCFD case versus Physical time in the simulation.

The total storage can be seen in Figure 4.12 containing the SIM, the database *Solution History* and the rCFD *Solution History*. For the baseline case the SIM file was slightly smaller then for the rCFD cases. The Solution Histories got approximately halved in size between frequency 16 and 32. Thereby, the needed storage decreased with increased frequency values.

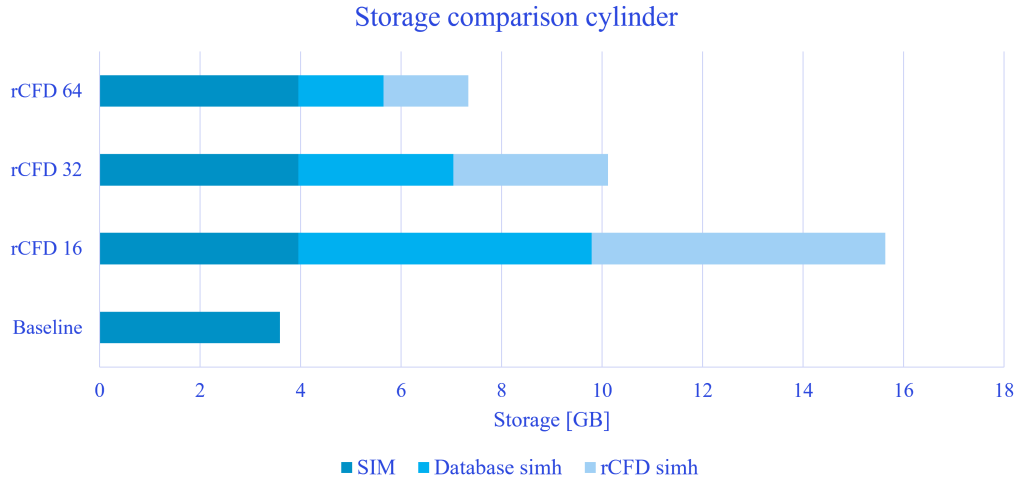


Figure 4.12: Histogram over the storage for the baseline case, rCFD_{16}, rCFD_{32} and rCFD_{64} cases, units in GB.

5

Results EX40 mirror

This chapter presents the results obtained for the EX40 mirror. First, the distance matrix and the accuracy of the recurrence paths are presented. Subsequently, the velocity fields and particle clouds generated by the simulations are compared. Finally, the results from the time tests and the total storage are discussed.

5.1 Distance matrix

Figure 5.1 reveals the results of the distance matrix $\mathcal{D}_{ji}$ for the EX40 mirror. The database used to obtain the distance matrix had 125 snapshots. In comparison to the earlier distance matrices, the distance matrix for the EX40 mirror has no distinct off diagonal lines. Additionally, it can be noted that there are some darker vertical and horizontal lines appearing around snapshot 40 and 100.

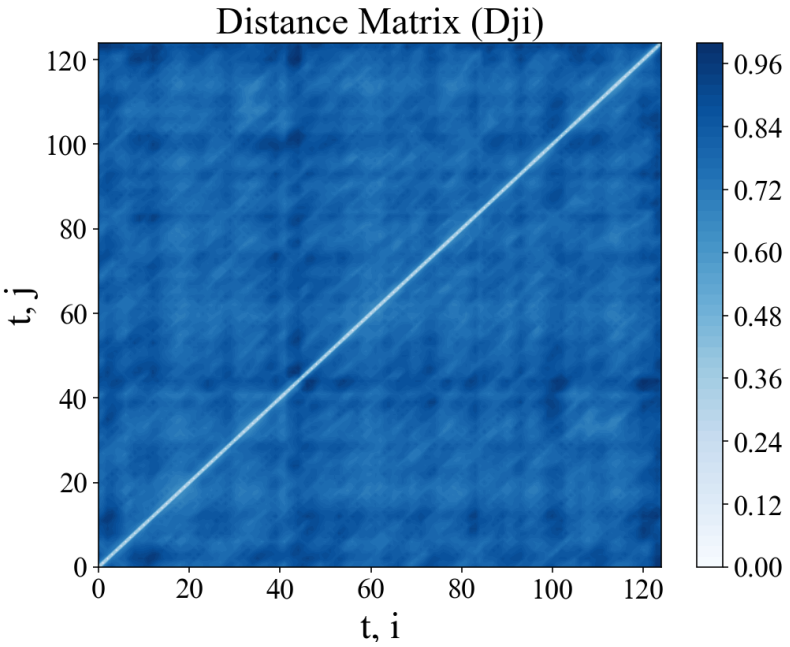
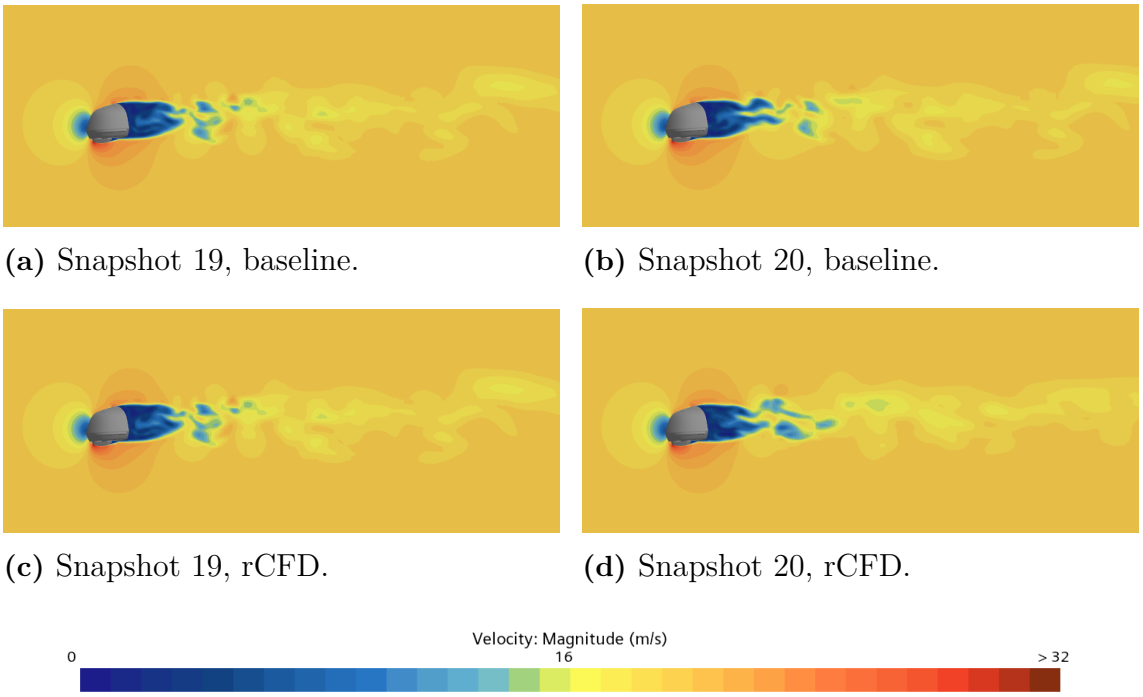


Figure 5.1: Distance matrix for the EX40 mirror.

5.2 Velocity field and accuracy

The rCFD *Solution History* file has a total of 357 snapshots, with an overall accuracy of 28.65 %. Figure 5.2 displays the differences between the jump from snapshot 19 to 20 for the baseline case and the rCFD case. Figure 5.2a and 5.2b shows the differences for the baseline case between the two snapshots, for a frequency of 16. Further, Figure 5.2c and 5.2d, displays the difference for the rCFD case between the corresponding snapshots, with an accuracy of 31.48 %. By investigating these snapshots, it is found that the difference between the snapshots for both the baseline case and rCFD case have similar tendencies. The similarity between the two cases is clearly seen in the wake that developed two tails for both of the cases. For snapshots 20, the wake further away from the mirror has less prominent appearance relative to each other.



(a) Snapshot 19, baseline.

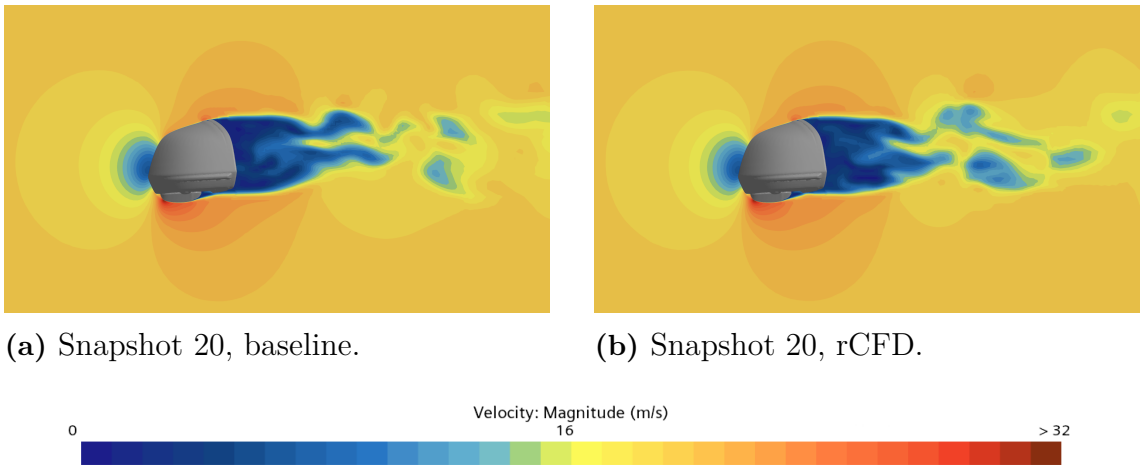
(b) Snapshot 20, baseline.

(c) Snapshot 19, rCFD.

(d) Snapshot 20, rCFD.

Figure 5.2: Snapshot comparisons for the velocity magnitude between the baseline case and rCFD case, for the EX40 mirror.

Figure 5.3 shows a close up of the wake for the EX40 mirror. The two tails are more visible. It can be seen that the first separation of the wake appears closer to the surface for the rCFD case than for the baseline case.



(a) Snapshot 20, baseline.

(b) Snapshot 20, rCFD.

Figure 5.3: Snapshot 20 comparison between baseline case and rCFD case, for the EX40 mirror. Close up of the wake.

5.3 Particles

Figure 5.4a displays the results of the particle distribution for the baseline case. It can be seen that the wake behind the mirror has an irregular shape. Furthermore, Figure 5.4b shows the particle injection for the rCFD case. Both of the wakes exhibit irregularities, but with different shapes. Closer to the mirror it can be seen that the particles appear as more compact for the rCFD case.

(a) Particle cloud for baseline.

(b) Particle cloud for rCFD.

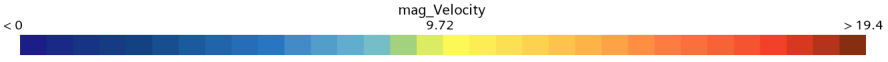
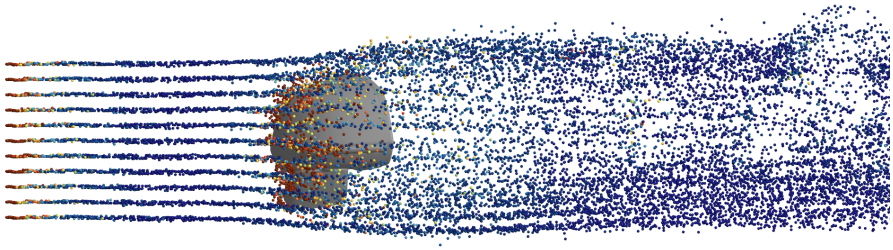


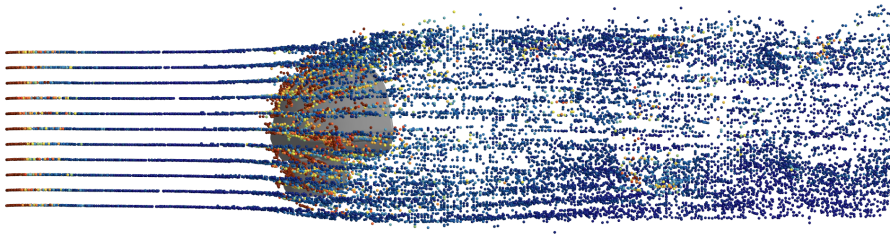
Figure 5.4: Comparison of the Particle clouds of velocity magnitude for baseline case and rCFD case, for EX40 mirror.

The particle drag force for both cases are shown in Figure 5.5. Figure 5.5a repre-

sents the drag force on the particles for the baseline mirror case. There are large fluctuations in front of the mirror, compared to the rCFD case in Figure 5.5b. The particle drag force has similar tendencies for both of the cases. Higher drag force is seen at the particle injection plane as well as when interacting with the mirror. Further,the particles were more dispersed for the baseline case than for the rCFD case.



(a) Particle drag force baseline mirror.



(b) Particle drag force rCFD mirror.

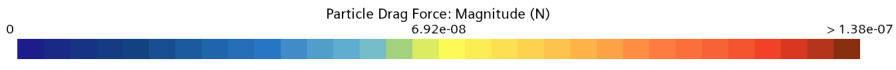
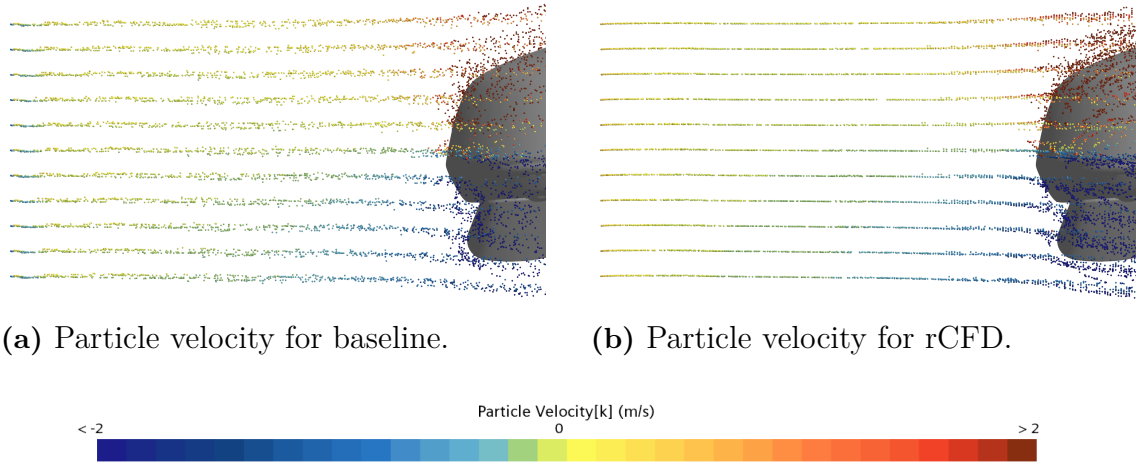


Figure 5.5: Comparison of the particle drag force of the baseline case and rCFD case.

In Figure 5.6, the fluctuation can clearly be seen in front of the mirror. Figure 5.6a and 5.6b shows the particle velocity in the z-directions, which is the velocity in the vertical direction of the figures. There are some differences that can be

spotted when comparing the two methods. For the CFD method, the particles close to the injection plane has higher negative velocities than for the rCFD method. The velocity differences between the cases appears smaller further away from the injection plane. When comparing the particles closer to the mirror, there are not any notable differences between the particle velocities.



(a) Particle velocity for baseline.

(b) Particle velocity for rCFD.

Figure 5.6: Comparison of the particle velocity in z-direction for the baseline case and rCFD case, for EX40 mirror.

5.4 Storage and time comparison

To be able to evaluate how time efficient the rCFD method is, a time comparison between the baseline and rCFD cases can be seen in Figure 5.7. The *Pre injection* phase exhibited longer execution time for the rCFD case compared to the baseline. However, it compensated with a significantly faster *Injection* segment. The total time difference between the cases was 55.93 minutes.

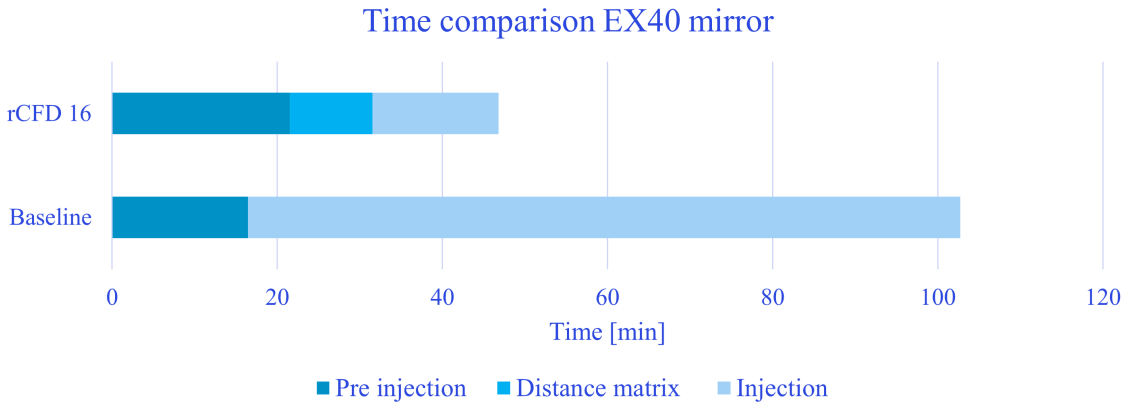


Figure 5.7: Duration for each segment for the baseline case and rCFD case for the EX40 mirror. Units in minutes.

In Figure 5.8 the real time duration can be seen plotted against the physical simulation time. The first second is displaying the *Pre injection* time. Closely after the start of the *Injection*, around 1.6 seconds, the rCFD case overtakes the baseline case.

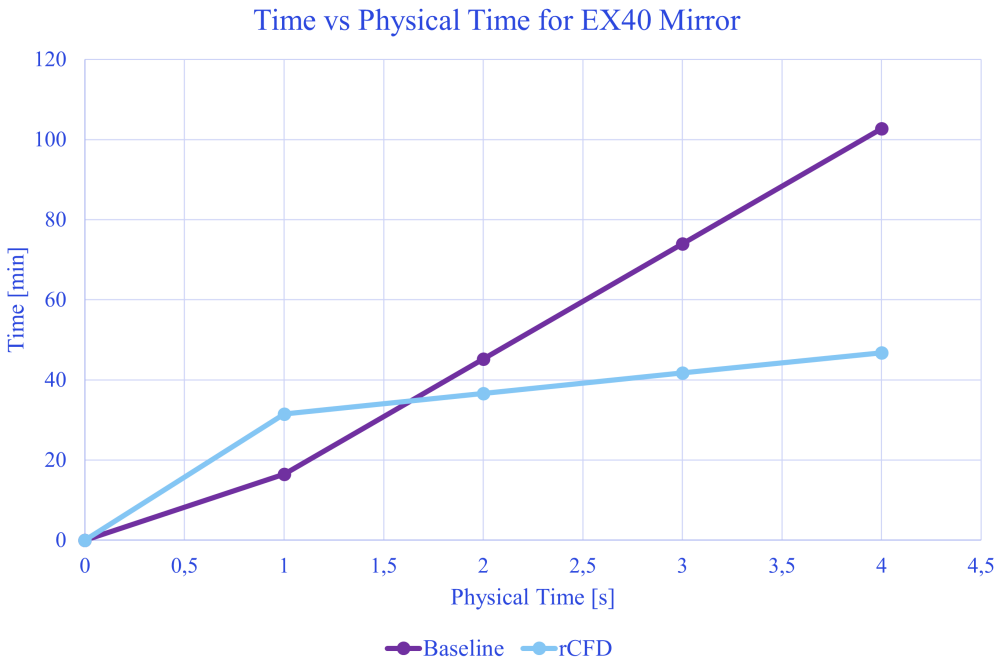


Figure 5.8: Duration for the baseline case and rCFD case versus Physical time in the simulation for the EX40 mirror.

The total storage comparison consisted of a SIM file and *Solution History* files for the database and rCFD, see Figure 5.9. When comparing the SIM files, it is observed that the rCFD method requires more storage. Further, the rCFD method needed additional storage for the database as well as the rCFD *Solution History* file. The total file size difference between the cases was 6.66 GB.

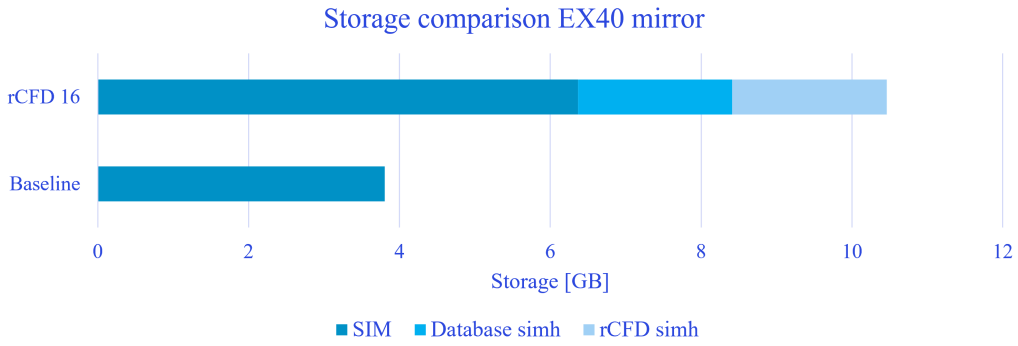


Figure 5.9: Histogram over the storage for the baseline case and rCFD case of the EX40 mirror, units in GB.

6

Discussion

This chapter presents a discussion and evaluation of the obtained results with the aim of investigating the rCFD methods applicability. The discussion includes analysis of the recurrence patterns, flow field, accuracy and particle deposition for the different cases. In addition, computational aspects such as simulation time, storage and potential error sources are considered to provide a broader investigation of the methods efficiency.

6.1 Distance matrices

Every distance matrix for the cylinder have the same general appearance characterised by off diagonal lines representing periodicity. The database captures three vortex shedding periods which is visible in the distance matrices, as expected. Theoretically, a Reynolds number of 10^2 should result in clear periodicity, see Figure 2.1b. For rCFD, this should result in an accuracy close to 100 %, as confirmed by the results, indicating an ideal recurrence path and efficient execution of the rCFD method. It can be noted that the periodicity becomes less evident as more turbulence is introduced through higher Reynolds numbers in the domain. In the distance matrices this is displayed as less distinct off diagonal lines. By comparing the distance matrices to the accuracy results it can be seen that despite the similar appearances for Reynolds numbers 10^5 and 10^4, the accuracy is higher for 10^4. Only a small difference in accuracy is observed and the results may vary due to the randomized segment lengths.

The distance matrix for the EX40 mirror has a dissimilar appearance compared to the cylinders matrices. If examining Figure 5.1 closely, lighter off diagonal lines can be seen. These lines can be seen on each side of the main diagonal, neither fully straight nor clearly visible. The darker vertical and horizontal lines in the matrix indicates that a certain event has occurred for a snapshot, causing it to differ from the other snapshots. These events could have appeared because of the flow interacting with the boundaries of the domain, for example the ground. The events should not affect the final results since the most similar snapshot was chosen.

6.2 Velocity field and accuracy

When examining the velocity fields for the cylinder it can be seen that the flow has the same tendencies for the rCFD as for the baseline. Closer to the cylinder surface, the snapshots have a similar appearance, sharing the same overall shape in the near wake and moving in the same direction. This is applicable for all the Reynolds number cases. Further downstream of the flow it can be seen that the baseline and rCFD cases differ, except for Reynolds number 10^2. They exhibit similar flow behaviour with oscillations clearly visible. However, the vortex shedding are not placed in the same positions in the fluid domain. This is possibly due to increased differences between the snapshots when increasing the Reynolds number since turbulence is of a chaotic nature.

The different Reynolds numbers provide varying accuracy, decreasing with increased Reynolds number. For Reynolds number 10^2 the accuracy is high, leading to no significant differences for the rCFD jump. This is expected since a Reynolds number of 10^2 should have similar snapshots for each vortex shedding. The remaining Reynolds numbers are exhibiting turbulent flow structures and the vortex shedding will not be identical. When applying the rCFD method to turbulent cases the accuracy decreases. Therefore, the rCFD method might not be suitable for all applications, especially where high accuracy is required. Nevertheless, it may be beneficial for simulations that need to be run over long periods of time. Recurrence CFD combined with particle tracking could be beneficial to find accumulation zones.

To save additional simulation time, larger extraction frequency values can be used when exporting data to *Solution Histories*, reducing the time needed for each segment. This also saves storage since the *Solution History* size decreases when using large extraction frequency values. When changing the frequency, the off diagonal lines appear more irregular in shape in the distance matrices, see Figure 4.5. Despite these irregularities the accuracies for the different frequencies are still relatively high. No distinct difference between the distance matrices for the varying frequency can be seen. However, the accuracy can be seen increasing despite fewer snapshots in the database. This accuracy pattern could be a coincidence based on the randomization factor in the recurrence path and further investigation would be needed to fully draw a conclusion. When comparing the velocity fields the flow depict similar tendencies regardless of frequency.

A possible error source is the differences between the jumps. The first rCFD jump is shown in the results. However, it should be noted that subsequent jumps will differ from the corresponding snapshots in the baseline case. This is due to the randomized segments which will alter the flow path. The rCFD snapshot before the jump might not match the corresponding snapshot for the baseline.

6.3 Particle analysis

The differences of the particle deposition for the cylinder case could have emerged from several different factors. First and foremost, when using the regular CFD method, the particle deposition will differ between each run of the simulation. This is because the flow field will appear differently between the runs, even if the model and settings remains the same. Due to this factor, the results may be complicated to compare.

Another factor of the particle distribution that may cause a difference is the forces used to compute the Lagrangian particle tracking. Only particle drag force was included in the calculations for the cylinder case, resulting in several forces affecting the particles being neglected. However, the neglected forces should not have a considerable effect on the LPT due to their relatively small size compared to the drag force. To increase accuracy of the LPT model all forces should be included to work similarly to the one used in the software.

For the EX40 mirror, the particle clouds have some similarities as well as evident differences. In this case, the two additional forces, gravity force and pressure gradient force, were added to the LPT model. These forces are included because of their increased relevance due to the updated flow specifications and behaviour. However, the same problem regarding the accuracy still remain since not all forces are added.

Additional grid layers were introduced for the frequencies to maintain a particle distribution visually comparable to that of the baseline case. Without this adjustment, particles were injected in discrete batches due to the larger mapping interval, producing differences in the visualized particle clouds. The adjustment was therefore primarily introduced for visualization purposes rather than to improve the physical accuracy of the simulations.

6.4 Storage and time comparison

For the time comparison, it can be noted that the database extraction time increases as the extraction frequency value decreases. For example, a frequency of 16 requires more time than a frequency of 64. This is reasonable since more data needs to be exported from the simulation, additionally leading to a larger *Solution History* file. A larger *Solution History* file will increase the number of snapshots added into the distance matrix and hence, increasing computation time for that segment. To accommodate for this extra computation time needed for the rCFD, the *Injection* and *Post injection* segments have to be more efficient compared to the baseline case. For Reynolds number 10^4 it can be seen that the *Post injection* phase is more rapid for the rCFD cases. Although the simulation with an extraction frequency of 16 requires more computational time than the baseline, the trend suggests that it may eventually outperform the baseline for longer simulation times.

For the EX40 mirror it can be seen that the rCFD method is more time efficient than the original CFD. The *Pre injection*, including the *Distance matrix* $\mathcal{D}_{ji}$ calculations,

are more time consuming but the rCFD case outperforms the baseline once the *Injection* is started. The mapping and LPT is approximately six times faster than the inbuilt LPT solver. One possible explanation is that only the particle motion is solved in rCFD, whereas the baseline simulation must solve both the particle motion and the flow field.

Another factor affecting the time is the number of cores used. As mentioned earlier in Chapter 2.1.7, a larger number of cores does not necessarily result in increased simulation efficiency. Since this thesis did not include any study of the optimal number of cores, there may be a more suitable amount. Additionally, the different sequences of the total simulations can either be serial or parallel. As discussed in Chapter 2.1.7, the ratio of serial to parallel code influences the achievable reduction in simulation time. In addition, larger simulations with finer meshes may be limited by the increased storage. For the geometries considered in this thesis, storage limitations were not encountered. Further investigations of larger meshes should be considered.

7

Conclusion

It is evident that rCFD is implementable for STAR-CCM+ for different geometries. The accuracy varies depending on the active Reynolds number. However, it still gives relevant results and is deemed applicable. It is suitable for certain industrial application, such as vehicle contamination. A study of the methods usefulness is needed to evaluate the trade off between time and accuracy, since reduced time from rCFD corresponds to lower accuracy. Results of the rCFD are showing promising possibilities of reduced time, thereby also reducing computational cost. By using the rCFD method, contamination on vehicles have further potential to be examined and studied, enabling a broad foundation for decision making. There are several possibilities of further increasing the time efficiency of rCFD. In this thesis, it is proven that the extraction frequency of the database affects the simulation time. In summary, the method enables rapid predictions of particle deposition and is applicable for STAR-CCM+. The method demonstrates potential for further optimization of vehicle contamination, providing a basis for continued investigation.

7.1 Future work

Several aspects have been investigated throughout this thesis, with more to be studied. Reduction of the storage for the simh files, including database and rCFD, is worth analysing. Furthermore, the particle tracking can be optimized by integrating the remaining forces in the LPT model. To increase the speed depending on accuracy, an extended frequency study could be executed. Hence, making a correlation between frequency, accuracy and time efficiency. The time can further be decreased by investigating the optimal number of cores used for the simulations. Another area of interest is the investigation of geometries with more cells in STAR-CCM+ to evaluate limits of the rCFD method.

Declaration about use of artificial intelligence

Parts of the Java macro and Python script have been generated and edited by Claude Sonnet 4.6. Claude Sonnet 4.6 has been used for troubleshooting the codes to make the process more time efficient. To decrease human error margins, Claude Sonnet 4.6 has together with us, verified the syntax for the equations used in the code.

In terms of the report, the LMM GPT 5.5 has been used for editing and to propose a first draft of the abstract. The editing includes spell checking, synonyms and suggestions for paraphrasing certain sentences of this thesis.

Bibliography

[1] S. Abbasi, S. Pirker, and T. Lichtenegger. Application of recurrence cfd (rcfd) to species transport in turbulent vortex shedding. *Computers Fluids*, 196:104348, 2020.

[2] Jelena Andrić. Cfd with openfoam software: Lagrangian particle tracking, 2009.

[3] Jiri Blazek. *Computational Fluid Dynamics: Principles and Applications*. Butterworth-Heinemann, third edition edition, 2015.

[4] Marina Bulat, Stefan Mirković, Nemanja Gazivoda, Dragan Pejić, Marjan Urekar, and Boris Antić. An improved algorithm for the estimation of the root mean square value as an optimal solution for commercial measurement equipment. *Microprocessors and Microsystems*, 106:105042, 2024.

[5] Lars Davidson. *Fluid mechanics, turbulent flow and turbulence modeling*. 2025.

[6] P. A. Davidson. *Turbulence - An Introduction for Scientists and Engineers*. Oxford University Press, second edition edition, 2015.

[7] Joel H. Ferziger, Milovan Peric, and Robert L. Street. *ComputationalMethods for FluidDynamics*. Springer, forth edition edition, 2020.

[8] Johannes Hansson, Thomas Lichtenegger, Stefan Pirker, Srdjan Sasic, and Henrik Ström. Recurrence computational fluid dynamics for efficient predictions of long-term particle deposition on a cylinder. *Physics of Fluids*, 37(9):093320, 09 2025.

[9] JR. John D. Andersson. *Fundamentials of Aerodynamics*. McGraw-Hill, fifth edition edition, 2011.

[10] Pijush K. Kundu, Ira M. Cohen, and David R. Dowling. *Fluid Mechanics*. Academic Press, fifth edition edition, 2012.

[11] Paul J. LaNasa and E. Loy Upp. *Fluid Flow Measurement*, chapter 2 - Basic Flow Measurement Laws, pages 19–29. Butterworth-Heinemann, third edition edition, 2014.

[12] Efstathios E. Michaelides, Martin Sommerfeld, and Berend van Wachem. *Multiphase Flows with Droplets and Particles*. CRC Press, third edition edition,

2023.

[13] NASA Glenn Research Center. Newton's laws of motion, 2024.

[14] Vinh-Tan Nguyen and Hoang Huy Nguyen. Detached eddy simulations of flow induced vibrations of circular cylinders at high reynolds numbers. *Journal of Fluids and Structures*, 63:103–119, 2016.

[15] Davide Oberto, Davide Fransos, and Stefano Berrone. Using delayed detached eddy simulation to create datasets for data-driven turbulence modeling: A periodic hills with parameterized geometry case. *Computers Fluids*, 2025.

[16] Rob Oshana. *Multicore Software Development Techniques*. Elsevier Inc, 2016.

[17] Ronald L. Panton. *Incompressible Flow*. John Wiley Sons, Inc., forth edition edition, 2013.

[18] S. Pirker, M. Atzori, S. Heinrich, and T. Lichtenegger. Particle size segregation in bi and penta-disperse gas–solid fluidized beds: Cfd-dem and recurrence cfd simulations. *Chemical Engineering Science*, 309:121469, 2025.

[19] S. Pirker and T. Lichtenegger. Efficient time-extrapolation of single- and multiphase simulations by transport based recurrence cfd (rcfd). *Chemical Engineering Science*, 188:65–83, 2018.

[20] Tomas Schuetz. *Aerodynamics of Road Vehicles*. SAE International, 5 edition, 2016.

[21] Siemens. Paritioning solver. `https://docs.sw.siemens.com/documentation/external/PL20250714501835942/en-US/userManual/starccmp_userguide_html/STARCCMP/GUID-1859AAF4-F407-4DC2-B2D3-B1D55BF93AAD.html?hl=partitioning%2Csolver`. Accessed: 2026-05-07.

[22] Siemens. Particle equations of motion. `https://docs.sw.siemens.com/documentation/external/PL20250714501835942/en-US/userManual/starccmp_userguide_html/STARCCMP/GUID-1CAF3A65-9203-4AFF-8316-70CBEA981BDB.html`. Accessed: 2026-03-04.

[23] Martin Sommerfeld. *Numerical Methods for Dispersed Multiphase Flows*, pages 327–396. Springer International Publishing AG, 07 2017.

[24] Thomas Sterling, Matthew Anderson, and Maciej Brodowicz. *High Performance Computing*. Elsevier Inc, 2018.

[25] Jiyuan Tu, Guan-Heng Yeoh, and Chaoqun Liu. *Computational Fluid Dynamics*, chapter Chapter 5 - CFD Techniques: The Basics, pages 155–210. Butterworth-Heinemann, third edition edition, 2018.

[26] H K Versteeg and W Malalasekera. *An Introduction To Computational Fluid Dynamics*. Pearson Education Limited, second edition edition, 2007.

[27] Pieter Wesseling. *Principles of Computational Fluid Dynamics*. Springer-Verlag, 2001.

[28] Frank M. White. *Fluid Mechanics*. McGraw-Hill Education, eight edition edition, 2016.

[29] David C. Wilcox. *Turbulence Modeling for CFD*. DCW Industries, Inc, second edition edition, 2004.

[30] Guan Heng Yeoh and Jiyuan Tu. *Computational Techniques for Multiphase Flows*. Butterworth-Heinemann, 2010.

[31] Chao Zhu, Liang-Shih Fan, and Zhao Yu. *Dynamics of Multiphase Flows*. Cambridge University Press, 2021.

A

Field functions

Filter condition | Scalar | Cylinder case

```
$${Position}[0]>8?1:0
```

Particle RE rCFD | Scalar | Cylinder case and EX40 mirror case

```
${ParticleDiameter}*mag($${MappedVelocity}-$${ParticleVelocity})*${Fluid
Density}/${Fluid Viscosity}
```

Drag Coefficient rCFD | Scalar | Cylinder case and EX40 mirror case

```
max(24/max(1e-10,${Particle Re rCFD})*(1+0.15*pow(${Particle Re rCFD},0.687))
,0.44)
```

Drag Force rCFD | Vector | Cylinder case and EX40 mirror case

```
0.5 * ${ParticleSurfaceArea}/4 * ${Rho_ref} * ${Drag Coefficient rCFD} *
mag($${MappedVelocity}-$${ParticleVelocity})*
($${MappedVelocity}-$${ParticleVelocity})/${ParticleVolume}
```

Gravity Force rCFD | Vector | EX40 mirror case

```
${ParticleMass}*$${GravityVector}/${ParticleVolume}
```

PG Force rCFD (Pressure gradient force) | Vector | EX40 mirror case

```
-grad(${StaticPressure})
```

Forces rCFD (User-Defined Body Force) | Vector | EX40 mirror case

```
$${Drag Force rCFD}+$${Gravity Force rCFD}+$${PG Force rCFD}
```

Forces rCFD (User-Defined Body Force) | Vector | Cylinder case

`$${Drag Force rCFD}`

B

Table of mesh and rCFD parameters

Table B.1: Mesh data.

	Meshing data				
Re	Mesh Size	Prism layers	Refine. cyl.	Refine. rect.	y+ max
10^2	1249908	10	18%	25%	0.176
10^4	2274388	26	15%	20%	0.896
10^5	2563356	38	13%	20%	1.58
10^6	13605620	47	5%	13%	1.93
10^4 (particle deposition test)	9958276	42	7%	13%	0.17

Table B.2: rCFD parameters.

	rCFD data					
Re	No. vortices	Time simulating database [s]	Freq.	Timestep	No. States database	Accuracy
10^2	3	37	16	0.00625	360	0.9966
10^4	3	31	16	0.00625	300	0.4507
10^5	3	22	16	0.00625	210	0.4991
10^6	3	19	16	0.00625	180	0.2932
10^4 (particle deposition test)	3	31	16	0.00625	310	-

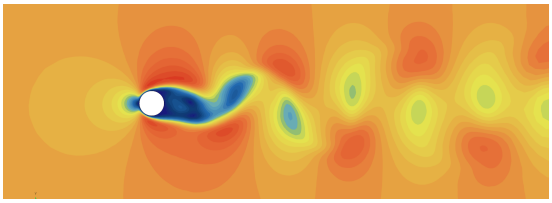
Table B.3: Particle data.

	Particle data
Re	Particle diameter [m]
10^2	0.03
10^4	0.004
10^5	0.001
10^6	0.0005
10^4 (particle deposition test)	0.0008

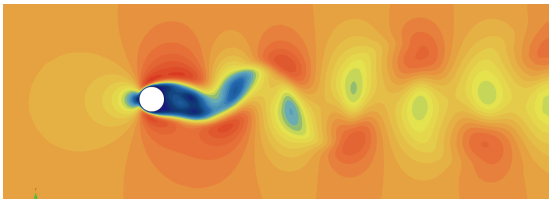
C

Additional cylinder results

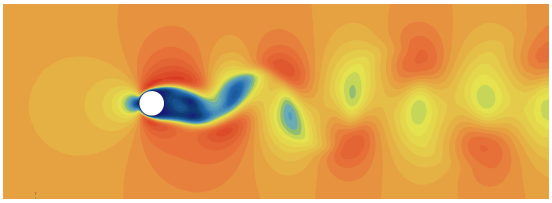
C.1 Velocity field for Re $= 10^2$ with frequency 16



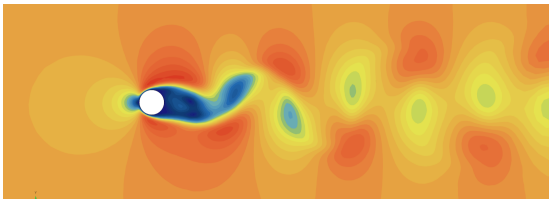
(a) Snapshot 54 baseline.



(b) Snapshot 55 baseline.



(c) Snapshot 54 rCFD.



(d) Snapshot 55 rCFD.

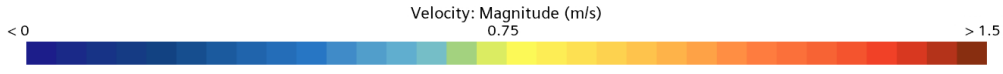
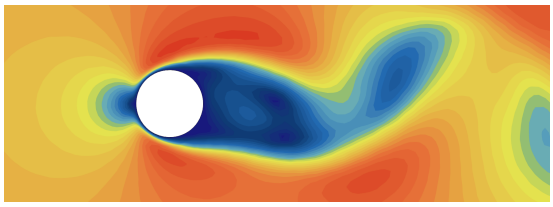
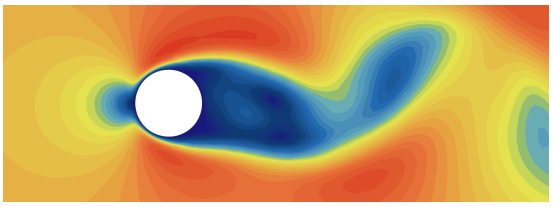


Figure C.1: Snapshot comparison for baseline and rCFD with frequency 16 and Re $= 10^2$.



(a) Snapshot 55 baseline zoomed in.



(b) Snapshot 55 rCFD zoomed in.

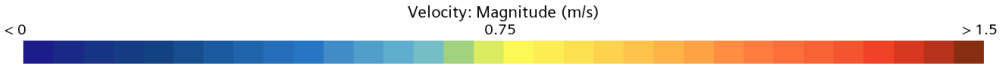
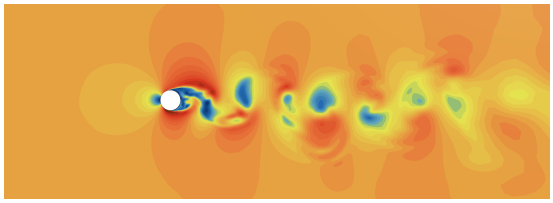
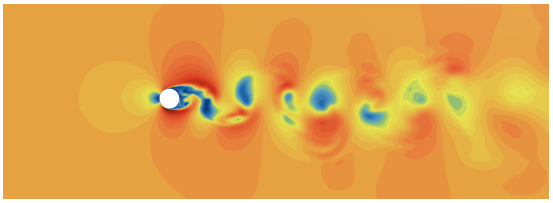


Figure C.2: Zoomed in snapshot comparison for baseline and rCFD with frequency 16 and $\mathrm{Re} = 10^2$.

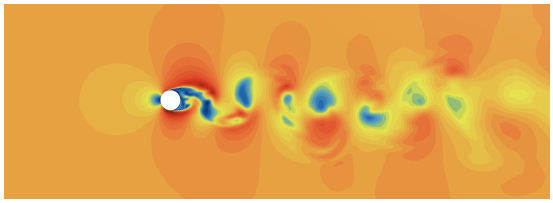
C.2 Velocity field for $\mathrm{Re} = 10^4$ with frequency 32



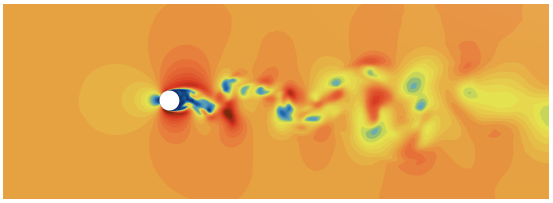
(a) Snapshot 24 baseline.



(b) Snapshot 25 baseline.



(c) Snapshot 24 rCFD.



(d) Snapshot 25 rCFD.

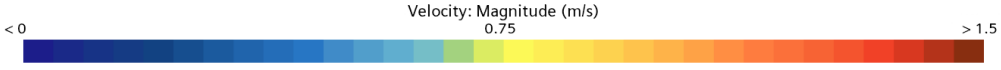
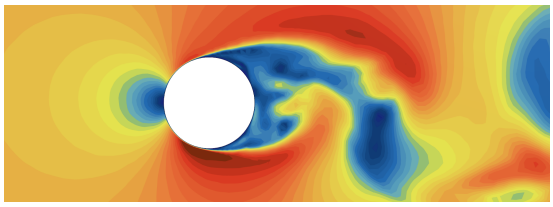
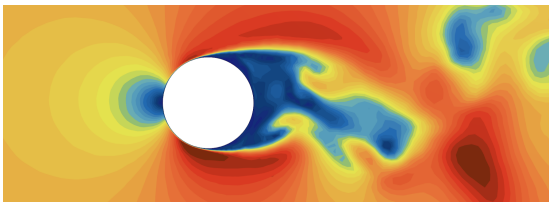


Figure C.3: Snapshot comparison for baseline and rCFD with frequency 32 and $\mathrm{Re} = 10^4$.



(a) Snapshot 25 baseline zoomed in.



(b) Snapshot 25 rCFD zoomed in.

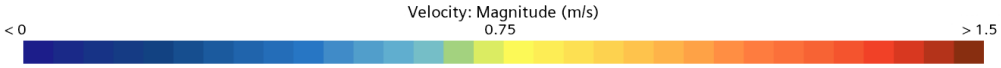
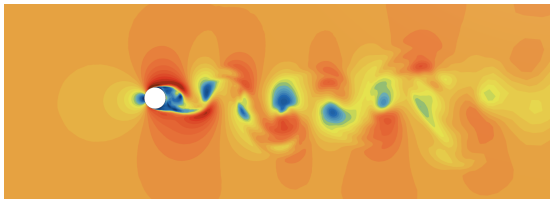
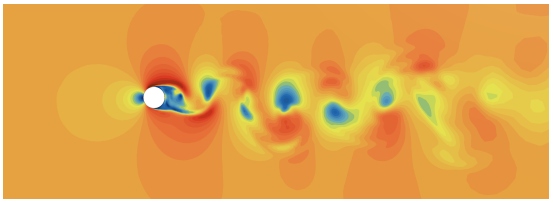


Figure C.4: Zoomed in snapshot comparison for baseline and rCFD with frequency 32 and Re = 10^4.

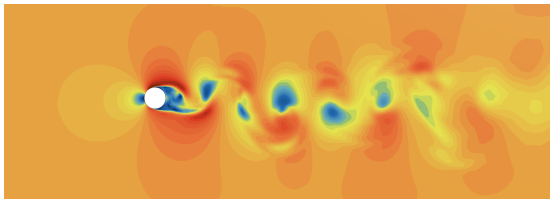
C.3 Velocity field for Re = 10^4 with frequency 64



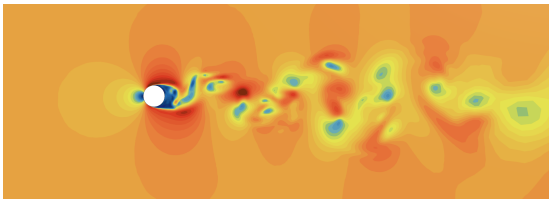
(a) Snapshot 3 baseline.



(b) Snapshot 4 baseline.



(c) Snapshot 3 rCFD.



(d) Snapshot 4 rCFD.

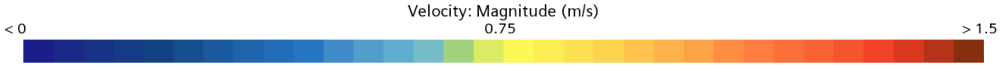
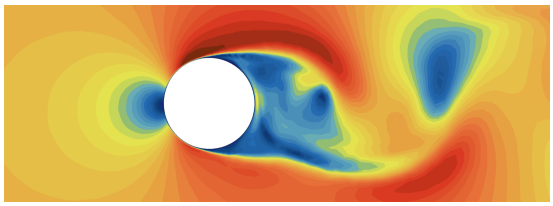
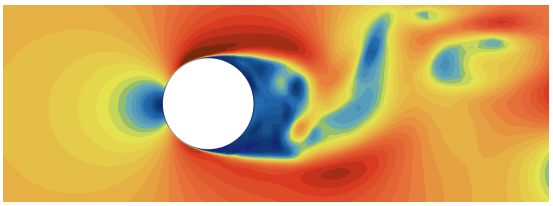


Figure C.5: Snapshot comparison for baseline and rCFD with frequency 64 and Re = 10^4.



(a) Snapshot 4 baseline zoomed in.



(b) Snapshot 4 rCFD zoomed in.

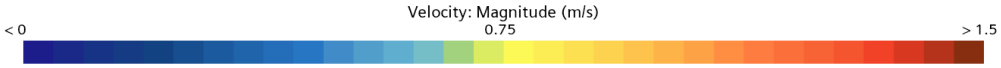
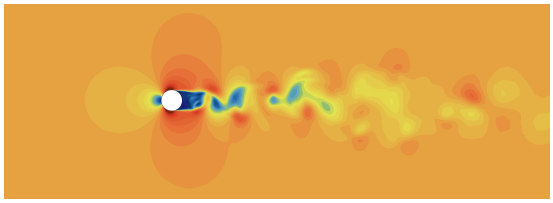
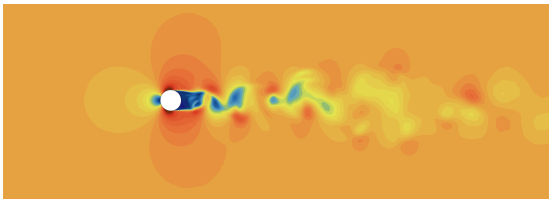


Figure C.6: Zoomed in snapshot comparison for baseline and rCFD with frequency 64 and $\mathrm{Re} = 10^4$.

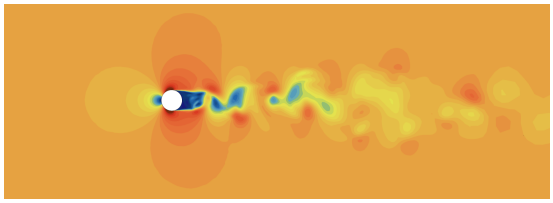
C.4 Velocity field for $\mathrm{Re} = 10^5$ with frequency 16



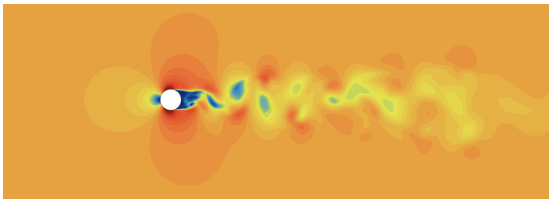
(a) Snapshot 40 baseline.



(b) Snapshot 41 baseline.



(c) Snapshot 40 rCFD.



(d) Snapshot 41 rCFD.

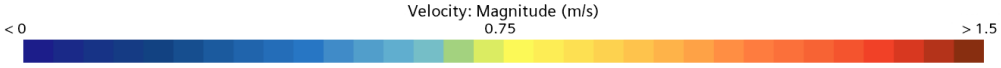
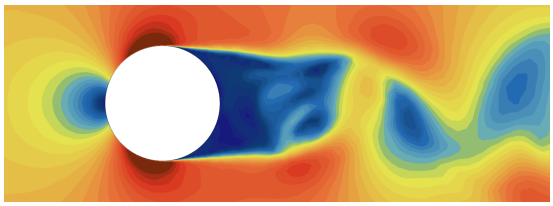
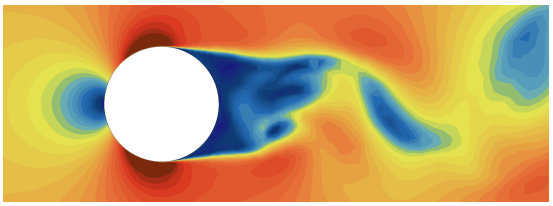


Figure C.7: Snapshot comparison for baseline and rCFD with frequency 16 and $\mathrm{Re} = 10^5$.



(a) Snapshot 41 baseline zoomed in.



(b) Snapshot 41 rCFD zoomed in.

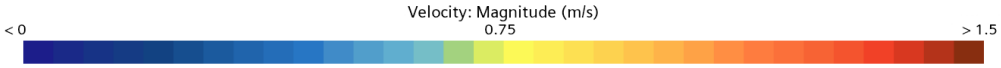
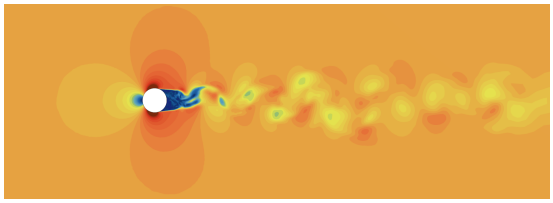
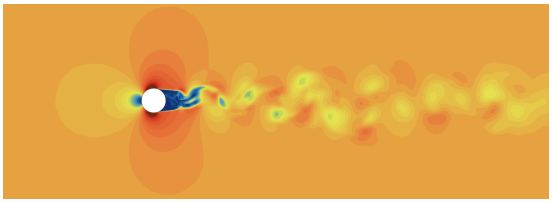


Figure C.8: Zoomed in snapshot comparison for CFD and rCFD with frequency 16 and $\mathrm{Re} = 10^5$.

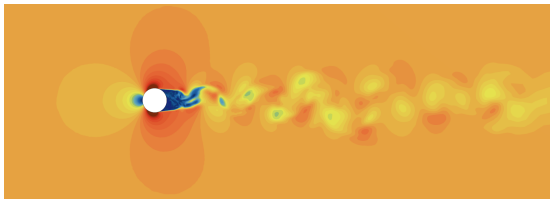
C.5 Velocity field for $\mathrm{Re} = 10^6$ with frequency 16



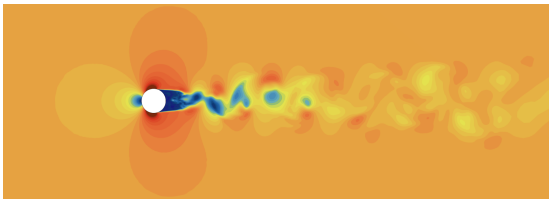
(a) Snapshot 18 baseline.



(b) Snapshot 19 baseline.



(c) Snapshot 18 rCFD.



(d) Snapshot 19 rCFD.

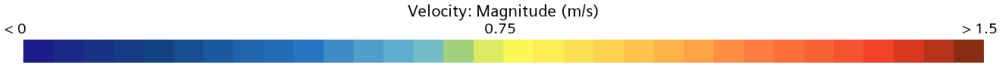
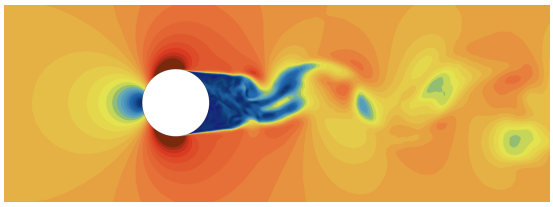
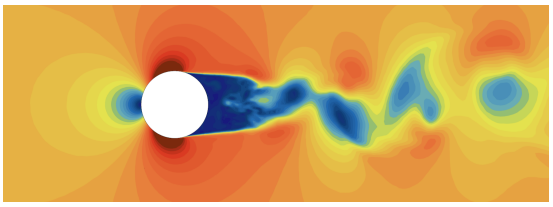


Figure C.9: Snapshot comparison for baseline and rCFD with frequency 16 and $\mathrm{Re} = 10^6$.



(a) Snapshot 19 baseline zoomed in.



(b) Snapshot 19 rCFD zoomed in.

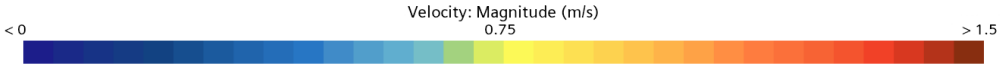
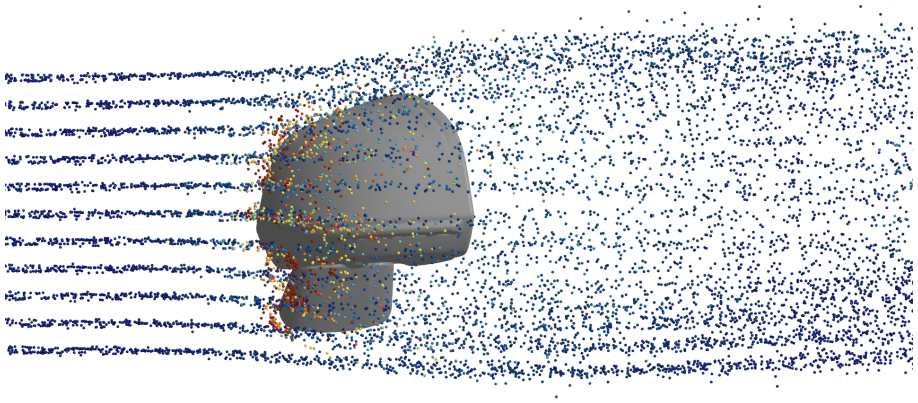


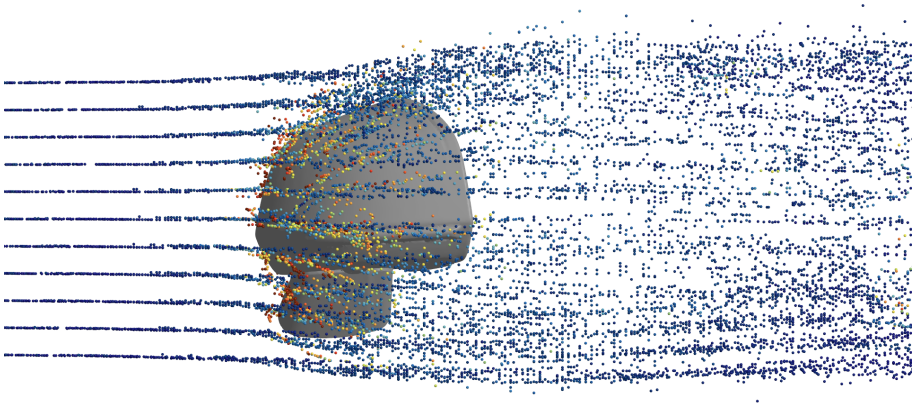
Figure C.10: Zoomed in snapshot comparison for baseline and rCFD with frequency 16 and $\mathrm{Re} = 10^6$.

D

Additional EX40 mirror results



(a) Baseline.



(b) rCFD.

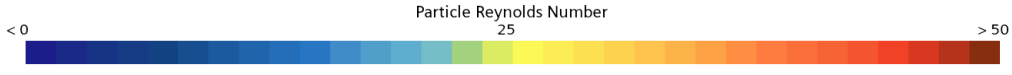
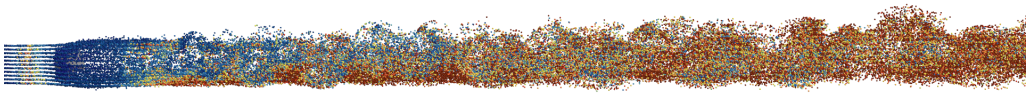
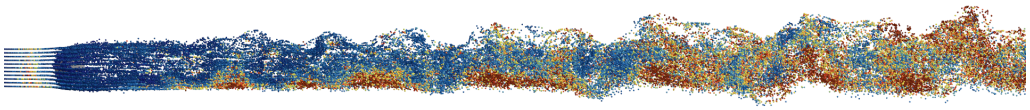


Figure D.1: Particle cloud visualizing the particle Reynolds number for the EX40 mirror.



(a) Baseline.



(b) rCFD.

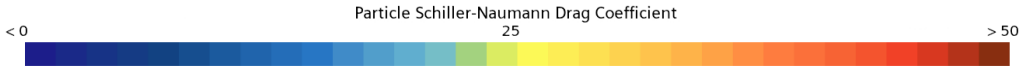


Figure D.2: Particle cloud drag coefficient for the EX40 mirror.

DEPARTMENT OF MECHANICS AND MARITIME SCIENCES
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY