ISBN 978-82-326-3404-0 (printed version)
ISBN 978-82-326-3405-7 (electronic version)
ISSN 1503-8181

NTNU
Norwegian University of
Science and Technology

Jørgen Røysland Aarnes

Doctoral theses at NTNU, 2018:307

NTNU

Doctoral Thesis

Doctoral theses at NTNU, 2018:307

Jørgen Røysland Aarnes

Simple Geometry – Complex Flow

A computational study of turbulent particle-laden flows impinging on a cylinder

NTNU
Norwegian University of
Science and Technology
Faculty of Engineering
Department of Energy and Process Engineering

NTNU
Norwegian University of
Science and Technology

Jørgen Røysland Aarnes

Simple Geometry – Complex Flow

A computational study of turbulent particle-laden flows impinging on a cylinder

Thesis for the degree of Philosophiae Doctor

Trondheim, September 2018

Norwegian University of Science and Technology
Faculty of Engineering
Department of Energy and Process Engineering

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Thesis for the degree of Philosophiae Doctor

Faculty of Engineering
Department of Energy and Process Engineering

ISBN 978-82-326-3404-0 (printed version)
ISBN 978-82-326-3405-7 (electronic version)
ISSN 1503-8181

Doctoral theses at NTNU, 2018:307

MILJØMERKET 241 TRYKKSAK 731

Printed by Skipnes Kommunikasjon as

Preface

This thesis is submitted to the Norwegian University of Science and Technology (NTNU) for partial fulfillment of the requirements for the degree of philosophiae doctor. The doctoral work has been performed at the Department of Energy and Process Engineering, NTNU, Trondheim. The supervisors of the work are Professor Helge I. Andersson and Associate Professor Nils E. L. Haugen.

The research has been funded in full by the Research Council of Norway (Norges Forskingsråd) under the FRINATEK Grant.

Trondheim, September 2018
Jørgen Røysland Aarnes

Abstract

Direct numerical simulations were performed to study the effects of free-stream turbulence on different flow regimes past a circular cylinder. The focus was on the transition-in-wake state of the flow and particle impact on the cylinder. To increase accuracy of the simulations, a considerable effort was made to improve the numerical methods used within the simulations. The focus of these improvements was on an existing immersed boundary method and a newly developed overset grid method for the representation of the solid cylinder in the open-source Pencil Code.

Transition-in-wake is the first stage of transition to turbulent flow around a cylinder, and can be characterized by three-dimensional effects developing in the wake, along with a decrease in the vortex shedding frequency (at $Re \approx 190$). For medium to high-intensity turbulent free-stream flow, the Reynolds number bandwidth for transition is increased. That is, the transition is initiated at a lower Reynolds number and endures to a higher Reynolds number than when the free-stream is laminar. Relevant mechanisms for this bandwidth increase include intermittent vortex dislocations and quasi-stable states at lower Reynolds numbers within the transition, and stimulations of larger modes of instabilities for higher Reynolds numbers.

For inertial particle impaction on a cylinder at moderate Reynolds number ($Re = 100$ and $Re = 400$), high-intensity free-stream turbulence significantly amplifies the number of particles that impact the cylinder (for certain particle Stokes numbers, St). The peak amplification of impaction is observed at $St = 0.3$. This peak is related to a change in impaction mechanism, from boundary stopping to boundary interception, and it will therefore dependent on the size of the particles as well as the Stokes number. The amplification decreases rapidly when the difference between a particle's Stokes number and Stokes number at peak impaction increases. These observations were reproduced from laminar impaction data using an expression for expectation value to estimate the particle impaction within the turbulent free-stream.

Improving how ghost- and mirror-points reconstruct the boundary conditions in an existing immersed boundary method yielded a more efficient way to perform simulations of a cylinder in a cross-flow using the Pencil Code. Yet impaction of small particles proved computational costly, limiting $St \geq 0.2$. Overset grids were developed as a means to overcome this limitation. The overset grid implementation uses sixth-order finite-differences for the governing equations, with summation-by-part operators at the boundary, Padé filtering and local time-step restrictions. The result is high-order accuracy at the cylinder boundary reproducing particle impaction with a 90% reduction in the necessary grid points. For 2D flow simulations, the number of grid points could be reduced by factors of 14–18 for flows with $Re = 100$–400 when using overset grids rather than the immersed boundary method.

Acknowledgments

No man is an island, but can a Ph.D. student perhaps be one? At times it appears so, in particular after hours, days or weeks of battling an unlimited number of bugs, impenetrable equations, server crashes and other general frustrations. Fortunately, for one who has family, friends and colleagues, the occasional vacation from the island of isolation, back to the continent of social community prevents one from drifting away during the years where more and more effort is invested into research on topics interesting perhaps to only oneself. I am exaggerating, of course. Yet, the help from people around me, not necessarily with research, but with withstanding the pressure of a four-year long journey of too much to do, and too little time to do it all, cannot be overestimated.

There are many whom I would like to acknowledge in this regard. First of all, I would like to extend my gratitude to my supervisors, Helge Andersson and Nils Erland Haugen. You have not only been a source of continuous support and encouragement during your time as my supervisors, you have also given me the freedom to grow during this time and to make the research project my own. A special thanks goes to Nils, for the many hours we have spent together on making code improvements and discussing results, and for introducing me to the Pencil Code and its community of researchers.

I would also like to thank my colleges in the Fluid Dynamics group at the Department of Energy and Process Engineering. I am particularly grateful to Bernhard Müller, for several years of collaboration on teaching computational fluid dynamics to Master's students and for much appreciated advice and expertise on numerical problems that were outside of my grasp. I must also thank those I have struggled together with: my fellow PhD-students and postdocs. There are too many to mention, but you definitely know who you are. Thank you for all the movie nights, dinners, rounds of beers, and outrageous lunch discussions.

There certainly is a lot more to Trondheim than a university, and a large part of my time here has been spent at another crucial institution in this student city: the student society, *Studentersamfundet*. I would like to give a very special thanks to my many close friends in Kulturutvalget ved Studentersamfundet, for all the early morning walks home, film festivals, Rocky Horror Picture Shows and genuine (yet sometimes misguided) efforts to make this city a better/more fun place.

Unfortunately, some of the people closest to me live the farthest away. I would like to extend my gratitude to the support and involvement from my family and friends in Stavanger, Bergen, Oslo, Tromsø and Copenhagen In particular, I would like to acknowledge the support from my closest family, Peter, Andreas, Tone and Per. Thank you for calling to check up on me every week. You know I am far too forgetful to do so myself.

Last, but not least, I wish to thank the person who has chosen to spend (almost) every day in my company. Your great patience and empathy, your north-Norwegian temperament, your sincere humor, sharp intellect, and your lovely smile, gets me out of the bed every morning (and back in every night!). Thank you Maria, for sharing your time with me.

Contents

Part I

Everything flows.

Heraclitus[1]

1 Introduction

In this thesis, computational efforts to investigate flows past a cylinder are presented. Flows of varying complexity are studied, ranging from simple steady flows to particle-laden flows in the unsteady vortex-shedding regime, with high-intensity turbulence present in the free-stream. To study such flows at reasonable computational costs, novel numerical methods have been developed and implemented in the open-source code known as the Pencil Code [15, 16, 100]. The aim of this introductory chapter is to provide the motivation for this work and to give an overview of relevant physical and numerical research.

1.1. Motivation

It can be quite relaxing to lie down next to a river and daydream, while watching twigs floating by and eddies breaking apart on rocks. Contemplating the physics of the phenomena one is observing, however, can be enough to make the most carefree daydreamer awaken to a nightmare of non-linear interactions and unsolved problems. Yet, if one takes a special interest in the physics of flows, it is perhaps the knowledge that what one is observing is yet to be fully understood that causes the relaxation itself.

[1] Variants of this aphorism appear in Plato's *Cratylus* [73] and Simplicius' *Commentary on Aristotle's Physics* [99].

Although it is now four centuries since the great mathematician and physicist Isaac Newton lived, there are many problems of a mechanical nature that are far from fully understood by natural scientists. One such problem is the complex interplay of fluids and objects in a turbulent flow. One needs not be a daydreamer or a romantic to find such a problem interesting, as turbulent flows are found not only in nature. Such flows exist in abundance within industrial applications, and increased knowledge about turbulence facilitates improvements in such application. Turbulent flow past an object may be complex enough, but it can be further complicated by considering a multi-phase flow, such as a two-phase flow where one phase is a fluid and the other consists of particles suspended in the flow. The flow problem considered in this thesis is a turbulent, particle-laden flow, where an obstruction is blocking the path of the suspended particles.

A river flowing around a rock and the fly-ash of a bio-mass fired power plant heating tubes of water vapor are examples of turbulent particle-laden flows navigating an obstruction. Although these examples of water flowing and gas rising may appear quite different, the flow fields can be very similar. Further similarities can be found in the transport and deposition mechanisms of suspended particles in a river, sediments that wear rocks over time, and particles from combusted bio-mass transported by the fly-ash in an incinerator and deposited on heating tubes. In both examples, one should expect intermittent, chaotic fluctuations of velocity in the flow. These fluctuations, the turbulence in the flow, may be due to the fluid interacting with the object or other factors upstream or surrounding the object. Examples include rapids in the river and an incinerating flame in the bio-mass power plant.

A clear description and understanding of the fundamental physical processes of such flows can be motivation enough to take on the research required to write a doctoral thesis. Any applications of such research in industry can be considered a beneficial side-effect, or the main motivation for such research. Some possible applications are consideration below.

Particle motion, accumulation and deposition in turbulent flows are of great importance for many critical applications in the modern world. Two relevant examples are filter applications and industrial boilers. Particles impacting on a solid object in the flow can lead to the build-up of a deposition layer on the solid-fluid interface, or it can lead to erosion of the solid object. For a bio-mass power plant, a deposition layer will lead to reduced efficiency (power output), while in a water turbine, particles in the water will contribute to erosion and corrosion of the blades (also lowering efficiency). In both cases efforts are therefore made to avoid particle deposition. In the diesel filter of a car, particles passing the filter contribute to air pollution. In this case (and in other filter applications) the goal is to maximize particle deposition. Regardless of the particular application, increased knowledge of the flow problem can improve design and, hence, efficiency.

Computational methods (simulations) were used in all new research presented within this thesis. With modern supercomputers available, idealizing complex flow problems and solving them numerically is a common approach yielding knowledge previously unattainable by purely theoretical or experimental studies. The computations are based on the well-founded claim that all (non-relativistic continuum) flows can be described by the same fundamental laws: the Navier-Stokes equations. Similarities between water in a river and gaseous flow in a power plant were noted above. In effect, only the boundary conditions and certain physical parameters (viscosity, density, etc.) separate these flows when described by the Navier-Stokes equations. This computational approach does, however, have drawbacks. Not only must a considerable effort be made to prove the validity of the numerical results, certain methods may turn out less effective than initially expected. This necessitates the development and implementation of novel numerical methods. Quite a large part of the research presented in this thesis is related to such development and the implementation of a numerical framework capable of describing complex flow problems.

1.2. Flow problems

The domain of all flows considered are variations of a cylinder obstructing the flow. Although there is little variation in the geometry, the flows themselves are quite different. This leads to different flow problems, which are described in the following section.

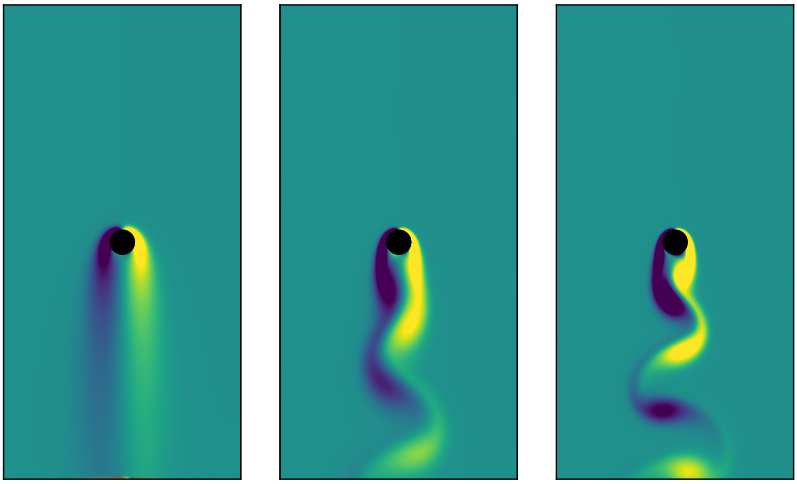
Transition-in-wake

The flow past a circular cylinder is usually described by a single non-dimensional number, the (cylinder) Reynolds number[2]. The Reynolds number (Re) is defined as:

$$Re = U_0 D / \nu \, , \tag{1.1}$$

where U_0, D and ν are the mean flow velocity, cylinder diameter and kinematic viscosity, respectfully. The Reynolds number is the primary parameter describing the viscous behavior of all Newtonian fluids [114]. At low Reynolds numbers, viscous forces dominate over advective inertial forces. This flow regime is termed laminar flow (or creeping flow, for very low Reynolds numbers). For flow past a

[2] Named after Osborne Reynolds, for his pioneering research on a broad range of topics in fluid mechanics [82].



(a) $Re = 20$ (b) $Re = 60$ (c) $Re = 100$

Figure 1.1.: Visualization of instantaneous vorticity normal to the view plane, plotted for three different Reynolds numbers. Transition from steady laminar flow (a) to unsteady laminar flow without (b) and with (c) vortex shedding in the wake. Inflow at the top plane.

cylinder, laminar flow does not necessarily indicate steady flow, i.e., flow constant in time. In the very low Reynolds numbers range of laminar flow, the flow is steady, but when the Reynolds number is increased past $Re \simeq 47$ the steady flow becomes unsteady through a Hopf bifurcation [79]. At this point, periodic oscillations develop in the wake. The wake eventually breaks up and form eddies downstream of the cylinder as the Reynolds number is increased ($Re > 90$). This pattern is known as the Kármán vortex street[3]. This development, from steady laminar flow to the Kármán vortex street, is seen in Figure 1.1.

Up to this point, the flow could be described as purely two-dimensional, as there was no variation of flow variables in the spanwise direction of the cylinder. With a further increase of the Reynolds number, however, three-dimensional fluctuations develop in the cylinder's wake. This occurs as the flow enters the transition-in-

[3] Although named after the engineer and scientist Theodor von Kármán, for his early research of this wake pattern, some controversy exists on who should be attributed as discovering this pattern. Von Kármán himself remarked that he made peace with the French scientist Henri Bénard on this matter, by jokingly suggesting that what in Berlin and London is called "Kármán street" should be called "Avenue de Henri Bénard" in Paris [109].

wake state. This state of the flow spans $(180-200) < Re < (350-400)$ (see, e.g., Zdravkovich [126]). For $Re > 400$ the flow progresses through other transitional stages as the Reynolds number is increased. These are denoted transition-in-shear-layers and transition-in-boundary-layers. Following the last transition, the flow can be described as fully turbulent; a state that is reached when all disturbed flow regions around the cylinder are turbulent [126]. All flows investigated in this thesis have $Re \leq 400$.

A flow simulation operating in the Reynolds number range of the transition-of-wake state should be able to reproduce the distinctive three-dimensional flow behavior observed in experiments focusing on this transition. The transition-in-wake was selected as the test case for three-dimensional simulations performed in the present work, and motivated an in-depth look at this transitional state of the flow. The particular focus was on the effect of a turbulent free-stream (TFS), i.e. the flow approaching the cylinder was turbulent. Key contribution to the understanding of the transition-in-wake state include work by Roshko [85], Williamson [115, 117], Williamson and Roshko [119], and Barkley and Henderson [8]. A brief background on this material is presented below, including specific work on the effect of free-stream turbulence.

It was noted above that this transition starts at $180 < Re < 200$. The Reynolds number of onset is typically identified by a sharp decrease in the vortex shedding frequency of the Kármán vortex street. The onset is identified over a range of Reynolds numbers, and this reflects two things. Firstly, there is a hysteresis pattern in the vortex shedding frequency when the Reynolds number is varied in the region were the transition begins. Secondly, there is quite a large scatter in the observed onset of the transition. This is especially prominent within experimental results, which can be seen by comparing data from the literature.

The onset of transition was observed at Reynolds numbers (in chronological order): 150 by Roshko [85] and Tritton [102], 140 by Gerrard [35], 170–180 by Williamson [115], 168 by Norberg [67], and 180–194 by Williamson [117]. Using Floquet stability analysis, Barkley and Henderson [8] identified the onset of transition at $Re = 189$ and Henderson and Barkley [43] further proved that the transition was sub-critical, explaining the hysteresis effect observed experimentally by Williamson [115] and others. The Reynolds number computed by Barkley and Henderson [8] was not only confirmed by the more recent results by Williamson [117], but also by high-accuracy numerical simulations [77].

Instability modes with a spanwise length of approximately four cylinder diameters develop in the flow at the onset of three-dimensionality in the wake. These are called mode A instabilities [115, 117] and are the dominant flow features in the spanwise direction during the first part of the transition-in-wake regime, along with spot-like *vortex dislocations*. Vortex dislocations are large-scale intermittent structures that grow downstream of the cylinder. These dislocations were discovered experimentally by Williamson [116], and similar features have been reproduced in numerical simulations (see, e.g., [127, 42]). Zhang et al. [127] labeled these structures "vortex adhesion", due to vortices evidently adhering to the cylinder over many shedding periods. The structures were also found to be self-sustained in the range $160 < Re < 230$. Henderson [42] pointed out that the spot-like dislocations must be generated by the mode A instability, and concluded that a nonlinear interaction between self-excited modes in the A-band is responsible for the appearance of large-scale structures in the wake. Here, the A-band refers to the different wavelengths of the mode A instability[4].

The mode B instability develops in the wake with a further increase of the Reynolds number. Mode B instabilities are streamwise structures with a spanwise wavelength of approximately one cylinder diameter, dominating the flow at $Re \gtrsim 260$ (see Williamson [118] and references therein). At Reynolds numbers between 210 and 220, the mode A and mode B instabilities start to co-exist in the wake. The transition from a flow dominated by one mode to the other is gradual, with energy in the flow shifting continuously from the larger (mode A) to the smaller (mode B) instabilities over a range of Reynolds numbers (see Barkley et al. [9]). Note that unlike the transition where mode A instabilities first occur, the second transition during the transition-in-wake state of the flow is supercritical [9].

The work on transitional flow presented in the present thesis does not focus on the general nature of the instabilities that develop in the flow. Rather, the transitional flow and flow with a Reynolds number near the transition-in-wake regime was studied under the disturbance of a turbulent free-stream. Experimental studies exist with this as a partial focus, conducted by Bloor [14], Hussain and Ramjee [45] and Norberg [66]. No effect was observed by Hussain and Ramjee [45], leading Zdravkovich [125] to draw the conclusion that the transition-in-wake state of the flow was insensitive to the free-stream turbulence. This finding is not valid, as Hussain and Ramjee performed experiments with $Re \lesssim 160$, that is, for Reynolds numbers outside the transition-in-wake state of the flow. Bloor [14] and Nordberg [66] observed an effect of the turbulence on the transition, in the bandwidth of Reynolds numbers spanning the transitional regime, but an analysis of the process was not performed.

[4] Note that the observed spanwise wavelength ($\lambda_A = 4D$) of the three-dimensional pattern developing in the wake corresponds to the most unstable wavelength of the A-band.

Particle deposition

Investigating particles in a flow convected towards and (possibly) past a circular cylinder is quite different from the study of transitional flow, and can be studied completely independently. Nevertheless, knowledge of the effect of free-stream turbulence on the transition-in-wake can be useful when particle-laden flow simulations under turbulent flow conditions are performed. For example in understanding unexpected behavior of the particle-laden flow at certain Reynolds numbers (perhaps due to an early onset of the transition-in-wake state), or in setting up flow cases with Reynolds numbers outside the most complex region of the transition. The latter was done when particle simulations within this work were set up to conduct an investigation of free-stream turbulence effects on particle deposition without unnecessary disturbances.

Particle-laden flow is a huge field of research itself, and has been a focus in recent years. For a general overview, with emphasis on numerical simulations of such flows, the reader is referred to reviews by Kuerten [55] and Marchioli [60]. For the purpose of the research presented in the present thesis, only particle-laden flows with a cylinder obstructing the path of the particles will be considered, and the focus will be on the impaction of particles on the surface of the cylinder.

When considering a large number of particles impacting on an obstruction in the flow, a practical way of quantifying the probability of impaction is by computing impaction efficiencies, η. The impaction efficiencies are the ratios of particles impacting on the cylinder to the total number of particles with a centre of mass that would hit the cylinder if not disturbed by the flow[5]. The impaction efficiencies are typically split into front-side and back-side impaction, η_f and η_b, respectively. These figures are represented as functions of the particle size and density, described by the Stokes number. The Stokes number (St) is the ratio of a particle's Stokes time to the timescale of the fluid flow the particle is suspended in:

$$St = \tau_p / \tau_f \,. \tag{1.2}$$

In the context of particle impaction on a cylinder, the fluid time scale is the ratio between the cylinder diameter (or radius) and the mean flow velocity.

There are several different methods for determining the impaction efficiencies of particles in flow past a cylinder. The simplest way to compute η is to use a potential flow approximation derived by Israel and Rosner [48]. The potential flow method for computing impaction efficiencies is well accepted for Stokes numbers larger

[5] In practice, this means that $\eta > 1$ is possible, since particles with finite radii and center of mass outside the projected cylinder area can contact the cylinder even without the flow affecting their trajectory.

than unity. For small Stokes numbers, the particles follow the flow to a large extent, hence, rotational and viscous effects in the vicinity of the cylinder will be important for the transport of these particles. These effects are not resolved with a potential flow approximation, thus, other methods must be used to calculate accurate particle impaction efficiencies for small Stokes numbers[6]. Other studies on impaction and deposition found in the literature used experimental methods (Schweers et al. [89] and Kasper et al. [50]), numerical simulations (Yilmaz and Cliffe [122], Li et al. [57], Haugen and Kragset [39], Haugen et al. [40], and Wacławiak and Kalisz [110]) and phenomenological modeling (Huang et al. [44]). These studies regarded smooth laminar flow past one or more cylinders.

Inertial particle impaction on a cylinder in a cross-flow can be split into three different impaction modes, based on what drove the trajectories of the convected particles during impact. The modes are classical impaction (particle inertia driven trajectory), boundary stopping (boundary layer driven trajectory) and boundary interception (mass center of particles do not come in contact with the cylinder) (cf. Haugen and Kragset [39] and Weber et al. [111]). The mode is determined by the Stokes number (with some dependence on Reynolds number). Generally, classical impaction occurs for Stokes numbers $St \gtrsim 0.9$, boundary stopping[7] for $0.2 < St < 0.9$ and boundary interception for $St < 0.2$. Potential flow expressions for impaction efficiencies are applicable for the classical impaction mode, while they are only somewhat useful for the boundary stopping mode and are not at all applicable for boundary interception.

It is reasonable to expect that turbulence in the flow influences the rate of particle impaction in many applications. The turbulence can be due to transitional eddies in the free shear layers of the cylinder at high Reynolds numbers, combustion upstream of the cylinder or wall turbulence for a cylinder in a confined space, etc. The velocity fluctuations will affect the inertial particle impactions on the cylinder surface, as particle trajectories deviate from the mean flow streamlines when turbulence is present. This is particularly important for particles with small Stokes numbers, as such particles primarily follow the flow.

An exception to the experimental studies on particles in smooth laminar flow, is the measurements by Douglas and Ilias [29] on the effect of turbulence on particle impaction. Douglas and Ilias [29] considered a cylinder situated within a channel

[6] A consequence of using the potential flow approximation for computing particle efficiencies, is that $\eta = 0$ (no impaction) for particles with $St < 1/8$ (see Ingham et al. [46]).

[7] The boundary stopping mode is partly overtaken by the classical impaction mode for high Reynolds numbers. Haugen and Kragset [39] identified boundary stopping for $0.2 < St < 0.3$ and classical impaction for $St > 0.3$ when $Re = 1685$. With $Re = 20$ the boundary stopping mode existed for $0.3 < St < 0.7$.

with turbulence generated by the channel walls. The results showed increased impaction efficiencies when turbulence was present in the flow, and eddy diffusion was a contributing mechanism. The scatter in the data was, however, quite large for small Stokes numbers.

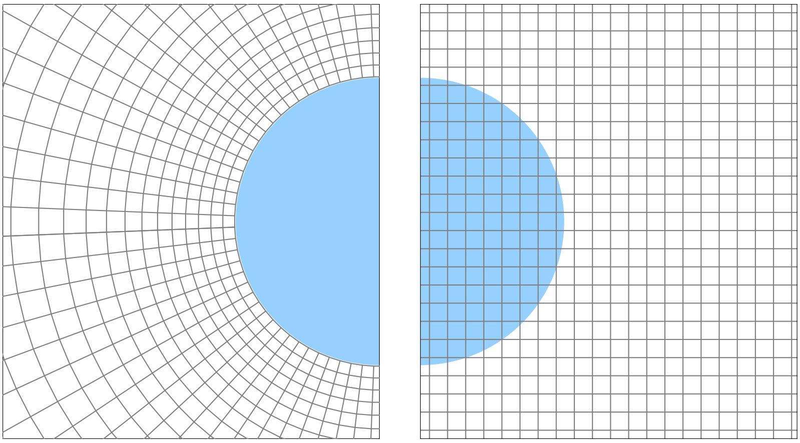
Several recent computational studies on particle impaction include turbulence in the flow (see Lee and Lockwood [56], Weber et al. [111], Beckmann et al. [10], and Pérez et al. [70], and a mini-review by Weber et al. [112]), but these were limited by the use of Reynolds Average Navier-Stokes modeling[8]. In one such study, performed by Weber et al. [111], it was found that the turbulence played a minor role for particles with Stokes number larger than a critical value. The turbulence increasing particle impactions for particles with Stokes numbers below the critical value. However, as pointed out by the authors of said study, the accuracy of the CFD simulation was limited by the lack of rigorous testing of the particle tracking procedure. Further limitations were introduced by modeling and time-averaging the flow, and by not accounting for boundary interception.

The work on particle-impaction presented in this thesis is twofold. Firstly, the effect of free-stream turbulence on impaction efficiencies of inertial particles for Stokes numbers from 0.2 to 10 was considered by Direct Numerical Simulations (DNS). Secondly, the particle impaction for a large range of Stokes numbers was used as a test case for a new numerical method developed as part of this doctoral work. This numerical method was motivated by the very high accuracy requirement for impaction of small particles ($St \lesssim 0.2$). The crux of the computations was to resolve the flow without needing a very fine grid, and at the same time have a sufficiently fine grid to resolve the particle trajectories in the vicinity of the cylinder. The interpolation of fluid velocities used to update the forces acting upon particles was particularly sensitive to the grid spacing for very small, tracer-like particles.

1.3. Numerical methods

Fluid flow in a domain that contains an immersed solid object is a common test case for computational fluid dynamics. Obstructions in the flow include (but are not limited to) cylinders, spheres, flat plates, rectangular or elliptical cylinders and spheroids, triangles and complex geometries made out of a combination of these shapes. Finding a method to represent such objects optimally in simulations is not a trivial task, and the numerical method used is often chosen specifically for the

[8] This limitation is, perhaps, not of much concern to the authors of said papers, as the focus of these articles is not on effects of the turbulence on the impaction. An exception to this is the article by Weber et al. [111] where this is a partial focus.


(a) Body-conformal

(b) Non-body conformal

Figure 1.2.: Representation of a solid object in the flow domain with different types of structured meshes. Domain boundaries not shown.

problem at hand.

For many generic shapes, such as cylinders, spheres, plates, etc., body-fitted structured meshes are commonly used to represent the object(s) in the flow. These meshes conform to the object(s) in the flow domain, as seen in Figure 1.2(a), and typically to the other physical boundaries of the domain (inlet, outlet, walls, etc.) as well. Depending on the shape of the flow domain and the object in the flow, this may require a deformation of the grid to conform to domain boundaries, in addition to the procedures to map the grid in the flow domain to a simple computational domain. This deformation may result in a grid with unnecessary local variations of the grid (e.g., a grid that is denser than necessary in certain areas of the domain). This can also contribute to time consuming grid generation (see, e.g., Versteeg and Malalasekera [105]). A popular alternative to such meshes, particularly when the shape of the flow domain or the objects within is more complex, is an unstructured mesh. Unstructured meshes provide the highest flexibility in grid adaptation to a particular flow geometry, and are a good alternative for complex geometries when finite-volume or finite-element formulations of the governing equations are used (see Mavriplis [62], Owen [68], and Tannehill et al. [98]). Disadvantages of such grids include larger storage requirements, the need for intricate mesh generation techniques and difficulties in achieving high-order accuracy.

Immersed boundary method

The object(s) in the flow and the flow domain can alternatively be represented without the grid conforming to the object(s), as seen in Figure 1.2(b). Typically, this is done by using a Cartesian grid, with a modification in either the flow equations or the grid cells in the immediate vicinity of the solid object(s). Popular techniques include immersed boundary methods (IBMs) (Peskin [71], Peskin [72], and Mittal and Iaccarino [64]) and cut-cell methods (Quirk [81], Causon et al. [20], Ingram et al. [47], and Schneiders et al. [88]). These two methods differ in that the IBM uses a Cartesian grid in the entire flow domain, while the cut-cell method grids are "cut" near objects and/or domain boundaries that do not conform to the grids. Due to this cell cutting, care must be taken such that the cut cells do not become too small, since this may introduce numerical instabilities.

For the IBM, rather than modifying the grid cells near the solid object, the boundary conditions of the solid are imposed directly in the flow equations. This is done either by a continuous or discrete forcing technique. In both cases a body-force, present due to non-conforming boundaries in the flow, is introduced in the Navier-Stokes equations. This is done either before discretization (continuous forcing) or after (discrete forcing) [64]. The latter is the preferred method for IBM used to represent rigid boundaries.

A further development of the discrete forcing method is to treat the immersed boundary as a sharp interface, and to impose the boundary conditions directly by using a combination of ghost-points inside the solid and mirror/image-points in the flow domain (set by interpolation) to reconstruct the solid (Tseng and Ferziger [103] and Berthelsen and Faltinsen [12]). An advantage to this approach is that the boundary conditions are represented without any added force in the flow equations, hence, the method can easily be implemented in an existing flow solver.

Disadvantages of IBMs are the lack of mass conservation[9] and reduced accuracy in the vicinity of the surface. Note that recent developments show that some of the challenges related to high-order accurate reconstructions of velocities near the surface can be overcome (Linnick and Fasel [58], Seo and Mittal [90], and Xia et al. [121]).

Overset grids

Roughly ten years after the emergence of the IBM, a method of multiple grids overset on one another was proposed to represent solids in a flow (see Steger

[9] Finite-volume approaches with cut-cell methodology are appropriate if mass and momentum conservation must be guaranteed [64].

et al. [93], Steger and Benek [94], and Benek et al. [11]). Such overset grids, or Chimera methods, employ body-conformal grids on solid objects in the flow domain, but the grids do not extend to the domain boundaries. Instead, a non-conforming background grid (typically uniform Cartesian) is used, and updated flow information of overlapping grid regions is communicated between grids at every time step. In this way, a flow simulation is split into multiple sub-simulations, one for each grid, and the information exchanged in overlapping regions enter as boundary conditions into the flow equations of the sub-simulations. The background grid is used to compute the flow outside the smaller body-fitted grids, and the communication between the different grids is done by interpolation. The flow domain may contain a single grid overlapping another, or several grids overlapping. The latter case necessitates a priority of communication and computation for solutions on different grids (see, e.g., Chesshire and Henshaw [21] and Meakin [63]). For complex configurations, this may require extensive preprocessing for fixed objects (Rogers et al. [84]) or intricate grid handling during the simulation for moving bodies (Noack [65]).

Overset grid methods have the advantages of being highly accurate at the solid-fluid interface. This is due to the use of body-fitted grids in these regions, and the flexibility in the grid stretching when several grids are used. Additionally, no grid deformation is necessary to conform to domain boundaries, due to the use of an appropriate non body-conformal background grid. If the flow domain is circular, a cylindrical grid can be used as the background grid, if rectangular, a Cartesian grid, etc.

The communication between the grids is the limiting factor in terms of the accuracy of overset grid methods. Interpolation of flow variables between grids in the overlapping regions is detrimental to mass conservation. However, conservative, mass-correcting overset grid methods do exist for finite-volume codes (see e.g. Pärt-Enander and Sjögreen [69] and Zang and Street [124]). Using high-order interpolation between grids has proven beneficial in regard to the overall accuracy and stability of the overset grid method for both finite-difference and finite-volume implementations (Sherer and Scott [91], Chicheportiche and Gloerfelt [22], and Völkner et al. [108]).

While advantageous in terms of accuracy, high-order interpolation techniques have the disadvantages of an increase in complexity and larger interpolation stencils, leading to more inter-processor communication and floating-point operations. Furthermore, straightforward extension to high-order interpolation (e.g. from second-order to fourth-order Lagrangian interpolation) does not guarantee a better solution. Possible overshoots in the interpolation polynomials can have a devastating impact on the accuracy and stability of the numerical simulation.

1.4. Goals and thesis outline

The main goal of this thesis is to document research on the transport and deposition of small particles on a cylindrical geometry in moderate Reynolds number flows, as well as the necessary numerical methods developed to perform the research. The achievements of the research as a whole can be split into four parts:

- An existing immersed boundary method was improved, such that accurate Direct Numerical Simulations (DNS) could be performed at reasonable computational costs for flows with moderate Reynolds numbers.
- The improved IBM was used to investigate the complex flow patterns in the transition-in-wake state by DNS, with the focus on how the transition is affected by high-intensity free-stream turbulence.
- DNS of a particle laden-flow interacting with a circular cylinder under conditions of laminar and turbulent free-streams was performed. Observation and analysis of how the impaction efficiencies were affected by high-intensity free-stream turbulence was done. Further, a predictive expression of the turbulence effect was suggested and the quality of this expression was considered.
- The numerical handling of the cylinder was improved by the development of an overset grid method, in order to (1) make possible very accurate simulations of small particles impacting on the cylinder and to (2) reduce the computational cost such that flows with larger Reynolds numbers can be considered in future research.

Background information related to the milestones has been given in this introductory chapter. In the next chapter the focus shifts to the theory of fluid flow, and a thorough description of the relevant numerical methods for discretization of the flow and particle equations and representation of solid objects in the flow. Following this, the contributions to the present thesis are described in Chapter 3. Conclusions are drawn in the last chapter, where suggestions for further work are also given. The scientific articles making up the bulk of the documented work can be found in the second part of the thesis.

What we observe is not nature itself, but nature exposed to our method of questioning.

Werner Heisenberg[10]

2 Theory and methodology

In this chapter the theoretical framework used to describe fluid flow and particle transport is described, and details of the discretization and numerical methods in use are presented.

2.1. Compressible Navier-Stokes equations

Fluid flow can be described by the Navier-Stokes equations, a set of equations derived from Newton's second law of motion, by considering viscous and pressure forces acting on a fluid element. For compressible Newtonian fluids, the Navier-Stokes equations for continuity and momentum are:

$$\frac{\mathrm{D}\rho}{\mathrm{D}t} + \rho \boldsymbol{\nabla} \cdot \boldsymbol{u} = 0\,, \tag{2.1}$$

and

$$\rho \frac{\mathrm{D}\boldsymbol{u}}{\mathrm{D}t} = -\boldsymbol{\nabla} p + \boldsymbol{\nabla} \cdot (2\mu \boldsymbol{S}) + \boldsymbol{F}\,, \tag{2.2}$$

respectively. Here ρ, t, $\boldsymbol{u}$, p and μ are the density, time, velocity vector, pressure, and dynamic viscosity ($\mu = \rho\nu$, with kinematic viscosity ν), respectively, and

$$\frac{\mathrm{D}}{\mathrm{D}t} = \frac{\partial}{\partial t} + \boldsymbol{u} \cdot \boldsymbol{\nabla} \tag{2.3}$$

[10] In *Physics and Philosophy: The Revolution in Modern Science* [41]

is the substantial derivative operator. The compressible rate of strain tensor $\boldsymbol{S}$ is given by:

$$\boldsymbol{S} = \frac{1}{2}\left(\boldsymbol{\nabla u} + (\boldsymbol{\nabla u})^T\right) - \boldsymbol{I}\left(\frac{1}{3}\boldsymbol{\nabla}\cdot\boldsymbol{u}\right), \tag{2.4}$$

where $\boldsymbol{I}$ is the identity matrix. The vector $\boldsymbol{F}$ is an external body force, e.g. gravity, drag from particles, etc. To close this set of equations, an equation of state is required. Here, the ideal gas law is applied:

$$p = c_s^2\rho\,, \tag{2.5}$$

where c_s is the speed of sound.

The Navier-Stokes equations can be re-written into a form with only the differentiation of variables with respect to time on the left hand side of the equations. This is the format of the equations in the Pencil Code. With a constant speed of sound (for the case of an isothermal fluid), constant kinematic viscosity and zero body-force, the flow equations are:

$$\frac{\partial\rho}{\partial t} = -\left(\boldsymbol{u}\cdot\boldsymbol{\nabla}\right)\rho - \rho\boldsymbol{\nabla}\cdot\boldsymbol{u}, \tag{2.6}$$

and

$$\begin{aligned}\frac{\partial\boldsymbol{u}}{\partial t} = &-\left(\boldsymbol{u}\cdot\boldsymbol{\nabla}\right)\boldsymbol{u} - c_s^2\boldsymbol{\nabla}\left(\ln\rho\right)\\ &+\nu\left(\boldsymbol{\nabla}^2\boldsymbol{u} + \frac{1}{3}\boldsymbol{\nabla}\left(\boldsymbol{\nabla}\cdot\boldsymbol{u}\right) + 2\boldsymbol{S}\cdot\boldsymbol{\nabla}\left(\ln\rho\right)\right).\end{aligned} \tag{2.7}$$

To develop a computational Navier-Stokes solver, fully expanded versions of the equations above are needed. Despite the utility, the fully expanded Navier-Stokes equations are rarely stated for compressible flow, particularly for coordinates systems other than Cartesian. For this reason the expressions are included in Appendix A.

2.2. Particle equations

Describing the fluid flow by the Navier-Stokes equations entails describing the fluid as a continuum. This is known as a Eulerian description of the flow. Particle descriptions in fluids are categorized as either Lagrangian tracking or Eulerian modeling approaches (Crowe et al. [24]). Using a Lagrangian formalism means that an individual particle (or parcel of particles) is tracked as it moves through the flow field. An Eulerian approach treats the particles and the fluid as two inter-penetrating continua. A Eulerian-Eulerian description for the fluid-particle flow is useful for when the volume fraction of the solid matter is high (see, e.g. van der Hoef et al. [104]), but this is not the case for the flows considered in this thesis. Instead Eulerian-Lagrangian tracking is used for particle-laden flows, such that each particle is tracked individually as it interacts with the surrounding fluid.

The considered particles are idealized as point-particles, that is, the particle volume is neglected (although a finite diameter is used in drag calculations and collisions, more on the latter point in Section 2.6). Although simulations are performed with several million particles in the flow[11], the small particle sizes makes the flow dilute, justifying the use of a point-particle approach. This also justifies the use of one-way coupling for fluid and particle phases. The one-way coupling indicates that the flow of one phase (the fluid) affects the other (the particles) while there is no reverse effect [24]. Alternatively, two-way coupling implies a mutual effect between the flows of both phases.

For a one-way coupled, Lagrangian, point-particle, the particle's velocity and position is described by:

$$\frac{\mathrm{d}\boldsymbol{v}_p}{\mathrm{d}t} = \frac{\boldsymbol{F}_{D,p}}{m_p}, \tag{2.8}$$

$$\frac{\mathrm{d}\boldsymbol{x}_p}{\mathrm{d}t} = \boldsymbol{v}_p, \tag{2.9}$$

where $\boldsymbol{v}_p$, $\boldsymbol{x}_p$ and m_p are the velocity, centre of mass position and mass of the particle, respectively. The force acting upon the particle is the drag force, $\boldsymbol{F}_{D,p}$, due to the particle moving through the fluid. The particle drag force is given by:

$$\boldsymbol{F}_{D,p} = \frac{1}{2C_c}\rho C_{D,p} A_p \left|\boldsymbol{u} - \boldsymbol{v}_p\right| \left(\boldsymbol{u} - \boldsymbol{v}_p\right), \tag{2.10}$$

where $A_p = \pi d_p^2/4$ is the cross sectional area of the particle and

$$C_c = 1 + \frac{2\lambda}{d_p}\left(1.257 + 0.4e^{(-1.1d_p/2\lambda)}\right), \tag{2.11}$$

is the Stokes-Cunningham factor with parameters set for air (Cunningham [25] and Davies [27]), and a particle diameter d_p. The mean free path $\lambda = 67\,\mathrm{nm}$ accounts

[11] Note that although several million particles are inserted into the flow during a simulation, there are no more than a few hundred thousand particles present in the flow in the simulations at any given time. This is due to the removal of particles from the flow when impacting on the cylinder or reaching the outlet boundary.

for the fact that for very small particles, the surrounding medium cannot be regarded as a continuum but, rather as distinct particles. The particle drag coefficient is given by (see Schiller and Neumann [87] and Clift et al. [23]):

$$C_{D,p} = \begin{cases} 0.44\,, & \text{if } Re_p > 1000\,, \\ \dfrac{24}{Re_p}\left(1+0.15Re_p^{0.687}\right), & \text{if } Re_p \lesssim 1000\,, \end{cases} \tag{2.12}$$

where Re_p is the particle Reynolds number, $Re_p = d_p\left|\boldsymbol{v_p}-\boldsymbol{u}\right|/\nu$. When $Re_p <$ 1000, which is the case for all the particles considered, the particle drag force of Eq. (2.10) can be written as:

$$\boldsymbol{F}_{D,p} = \frac{m_p}{\tau_p}\left(\boldsymbol{u}-\boldsymbol{v}_p\right)\,, \tag{2.13}$$

where

$$\tau_p = \frac{Sd_p^2C_c}{18\nu(1+f_c)}\,, \tag{2.14}$$

is the particle response time, with $f_c = 0.15Re_p^{0.687}$ and density ratio $S = \rho_p/\rho$. Note that the drag force on the particle is reduced to the Stokes drag (Stokes [95]), $\boldsymbol{F}_{D,p} = 6\pi\nu\rho r_p\left(\boldsymbol{u}-\boldsymbol{v_p}\right)$, in the limit $C_c = 1$ and $Re_p \ll 1$.

With the particle time scale defined, the Stokes number can be used as a non-dimensional number to describe a particle in the flow. As mentioned in the introduction, the Stokes number is the ratio between the particle and fluid time scales (Eq. (1.2)). The particle time scale is given by Eq. (2.14) and the fluid time scale is:

$$\tau_f = \frac{D}{2U_0}\,, \tag{2.15}$$

where the factor two is included by convention. The Stokes number can be regarded as a measure of particle inertia. Particles with small Stokes numbers follow the flow to a large extent, while particles with large Stokes numbers are negligibly affected by flow conditions.

2.3. Numerical methods for the Navier-Stokes equations

All simulations performed during the Ph.D. studies used the high-order finite-difference code known as the Pencil Code [100]. The code is a modular, open-source, partial differential equation solver programmed in Fortran95[12]. In addition to using this code for flow simulations, extensions were made and a new module was added. The main code extensions were on the representation of solid surfaces in the fluid flow. Details of the methods developed will be given in Section 2.5. First, the discretization of the Navier-Stokes equations is considered.

Governing equations

The spatial derivatives in the governing equations (right hand side of Eqs. (2.6) and (2.7)) are discretized with sixth-order central differences. Finite-difference approximations are applied to first and second order derivatives and to mixed-derivatives (Eqs. (A.6)–(A.10) and (A.16)–(A.20) in Appendix A). The finite-difference stencils for first and second order derivatives of a variable f, at grid point i, in direction h are:

$$\delta h \frac{\partial f_i}{\partial h} \approx \frac{1}{60} f_{i+3} - \frac{3}{20} f_{i+2} + \frac{3}{4} f_{i+1} - \frac{3}{4} f_{i-1} + \frac{3}{20} f_{i-2} - \frac{1}{60} f_{i-3}\,, \qquad (2.16)$$

$$(\delta h)^2 \frac{\partial^2 f_i}{\partial h^2} \approx \frac{1}{90} f_{i+3} - \frac{3}{20} f_{i+2} + \frac{3}{2} f_{i+1} - \frac{49}{18} f_i + \frac{3}{2} f_{i-1} - \frac{3}{20} f_{i-2} + \frac{1}{90} f_{i-3}\,, \qquad (2.17)$$

on a standard grid with grid spacing δh and grid points denoted by $i-1, i, i+1$, etc. (Fornberg [33] and Gustafsson [37]). For mixed-derivatives of $f_{i,j}$ along directions h and k, the finite-difference operators can be found by applying the first order derivative in an orthogonal direction to each term in Eq. (2.16), yielding:

$$\begin{aligned} \delta k \frac{\partial^2 f_{i,j}}{\partial h \partial k} \approx & \frac{1}{60} \frac{\partial f_{i,j+3}}{\partial h} - \frac{3}{20} \frac{\partial f_{i,j+2}}{\partial h} + \frac{3}{4} \frac{\partial f_{i,j+1}}{\partial h} \\ & - \frac{3}{4} \frac{\partial f_{i,j-1}}{\partial h} + \frac{3}{20} \frac{\partial f_{i,j-2}}{\partial h} - \frac{1}{60} \frac{\partial f_{i,j-3}}{\partial h}\,, \end{aligned} \qquad (2.18)$$

where the derivatives with respect to h are approximated by Eq. (2.16) and δk is the grid spacing in the k-direction (along the j index). Alternatively, using bidiagonal operators:

$$\delta h \delta k \frac{\partial^2 f_{i,j}}{\partial h \partial k} \approx \frac{1}{360} \left(f_{i+3,j+3} - f_{i-3,j+3} \right) - \frac{3}{80} \left(f_{i+2,j+2} - f_{i-2,j+2} \right)$$

[12] Some modules make use of C and Fortran 2003 functionality. Post-processing routines are implemented in IDL and Python. Makefiles, configuration files for compilation, and other auxiliaries use Perl. Massively parallel by MPI, with recent CUDA-extensions allowing for parallel simulations on graphical processing units.

$$
\begin{aligned}
&+\frac{3}{8}\left(f_{i+1,j+1}-f_{i-1,j+1}\right)-\frac{3}{8}\left(f_{i+1,j-1}-f_{i-1,j-1}\right) \\
&+\frac{3}{80}\left(f_{i+2,j-2}-f_{i-2,j-2}\right)-\frac{1}{360}\left(f_{i+3,j-3}-f_{i-3,j-3}\right).
\end{aligned} \tag{2.19}
$$

Notice that the bidiagonal operators reduce the number of grid points used by the finite-difference stencil from 36 (in Eq. (2.18)) to 12 (in Eq. (2.19)).

The integration in time of the Navier-Stokes equations and the particle equations is by a third-order memory-efficient Runge-Kutta scheme[13] [120]. For a function

$$
\frac{\partial \phi}{\partial t}=\boldsymbol{f}(\phi(t)), \tag{2.20}
$$

the update from timestep n to $n+1$, that is, from $\phi(t_n)$ to $\phi(t_{n+1})$, by this explicit scheme, is according to the following algorithm:

$$
\begin{aligned}
\delta \boldsymbol{f}_1 &= \delta t\, \boldsymbol{f}(\phi(t_n)), & \phi_1 &= \phi(t_n)+\frac{1}{3}\delta \boldsymbol{f}_1, \\
\delta \boldsymbol{f}_2 &= \delta t\, \boldsymbol{f}(\phi_1)-\frac{5}{9}\delta \boldsymbol{f}_1, & \phi_2 &= \phi_1+\frac{15}{16}\delta \boldsymbol{f}_2, \\
\delta \boldsymbol{f}_3 &= \delta t\, \boldsymbol{f}(\phi_2)-\frac{153}{128}\delta \boldsymbol{f}_2, & \phi(t_{n+1}) &= \phi_2+\frac{8}{15}\delta \boldsymbol{f}_3,
\end{aligned} \tag{2.21}
$$

where $\delta t=(t_{n+1}-t_n)$. The memory efficiency of this Runge-Kutta scheme is due to it only requiring one temporary array $\delta \boldsymbol{f}$ to store data as the time-integration of ϕ is performed. No temporary arrays of ϕ are needed because the array ϕ is overwritten when updated from one stage to the next.

Boundary closures

The finite-difference operators in Eqs. (2.16)–(2.19) are only valid for grid points that are surrounded by enough neighbouring grid points in each direction such that all values for $f_{i+3}, f_{i-3}, f_{i+2}$, etc., which are necessary to calculate the derivatives, exist. Hence, special handling is required to compute the derivatives of velocity and density in the vicinity of boundaries. The accuracy of a simulation as a whole is very sensitive to the choice of boundary closures. In the present work, three types of boundaries occur, each with its own closure for the finite-difference stencils. The different type of boundaries are: inlet/outlet, periodic and solid.

[13] Williamson [120] handles round-off errors explicitly in his Runge-Kutta algorithm. This is not done in the implementation in the Pencil Code.

Navier-Stokes characteristic boundary conditions

Navier-Stokes characteristic boundary conditions (NSCBC) are used for the inlet and outlet boundaries to minimize reflection. The NSCBC is a well-posed boundary formulation that makes use of local one-dimensional characteristic wave relations (LODI) to approximate the amplitude of incoming acoustic waves. This allows waves to pass through the boundaries, yielding partially reflecting inlet and outlet boundaries. For details on NSCBC and LODI-relations the reader is referred to the seminal paper by Poinsot and Lele [74].

To compute necessary gradients for the LODI-relations at the inlet and outlet, fourth-order, one-sided finite-difference stencils [33] are used:

$$\delta h \frac{\partial f_i}{\partial h_i} \approx \pm \left(-\frac{25}{12} f_i + 4 f_{i\pm1} - 3 f_{i\pm2} + \frac{4}{3} f_{i\pm3} - \frac{1}{4} f_{i\pm4} \right) , \tag{2.22}$$

where points $i, i+1, \ldots$ are used for left-side boundaries and $i, i-1, \ldots$ for right-side boundaries. Note that the NSCBC implementation in the Pencil Code uses modifications suggested by Yoo et al. [123] and Lodato et al. [59] to account for transversal flow effects (necessary, e.g. for turbulent flow at the inlet).

Periodic boundary

Unlike inlet/outlet and solid boundaries, periodic boundaries can be handled in a straightforward, almost trivial, manner in many fluid dynamics codes. A periodic boundary simply implies that two boundaries of the flow domain (e.g., top and bottom) are directly coupled, such that what exits through the one boundary enters the other, and vice versa (see, e.g., Versteeg and Malalasekera [105]). Hence, a periodic boundary is in principle folded onto itself. This makes the size of the physical flow domain (artificially) infinitely long in the direction of periodicity.

The derivatives of Eqs. (2.16)–(2.19) can be solved in the same manner as the interior points, provided that the *outlying* points are included appropriately. For example, if the first derivatives are computed at grid points $i = 0, 1, 2, \ldots, N_i - 1, N_i$ and the boundaries are periodic, the gradient at grid point N_i will need data at $f_{N_i+1}, f_{N_i+2}, f_{N_i+3}$ from points outside the domain. Due to the periodicity the grid points $i = 0$ and $i = N_i$ overlap (connecting the folded domain), hence, $f_{N_i+1} = f_1, f_{N_i+2} = f_2, f_{N_i+3} = f_3$, and Eq. (2.16) can be used. In the Pencil Code the periodic boundaries are handled by the use of a three-point deep ghost-zone on each side of the array of flow variables. Hence, variables are stored for grid points $i = -2, -1, 0, 1, 2, \ldots, N_i - 1, N_i, N_{i+1}, N_{i+2}, N_{i+3}$, where the outliers are updated using:

$$f_{N_i+1} = f_1 , \qquad f_{N_i+2} = f_2 , \qquad f_{N_i+3} = f_3 ,$$

$$f_{-2} = f_{N_i-2}\,, \qquad f_{-1} = f_{N_i-1}\,, \qquad f_0 = f_{N_i}\,,$$

and gradients are computed for $i = 1,2,\ldots,N_i$ by Eq. (2.16) etc.[14]

Flow arrays are padded with ghost-points on each side by default in the Pencil Code, due to the domain decomposition and communication between processors that do not share physical memory. Hence, the periodicity does not add complexity in this regard.

Solid boundary

In the present thesis, solid boundaries in the flow domain represent a solid object that obstructs the fluid flow. The solid object's boundary may or may not conform to the computational grid. As mentioned in Section 1.3, both body-conformal and non-body conformal grids are used to resolve such boundaries in the present work. Body-conformal grids are used with overset grids, while grids that are not fitted to the surface are used with immersed boundary methods.

In the case of surface representation by an immersed boundary method, the finite-difference stencils in Eqs. (2.16),(2.17) and (2.19) are used for first, second and mixed derivatives, respectively, at all grid points in the flow domain[15]. This is possible due to the use of a several layers of ghost-points inside the solid, set by corresponding mirror-points in the flow domain (details in Section 2.5). For overset grids, asymmetric stencils are necessary for grid points close to the surface, as there are no grid points inside the solid when the grid is body-conformal.

The asymmetric stencils that are used near the boundary, when represented by overset grids, are summation-by-parts (SBP) operators (see Strand [96] and Mattsson and Nordström [61]). The choice of SBP-operators near the surface is based on favorable features, such as of well-posedness and stability enhancement. For the sixth-order centred stencils used for first and second derivatives, third-order accurate SBP boundary closures exist. For an array of flow variables, $\boldsymbol{f} = (f_1, f_2, \ldots, f_{N_i})$, the first derivative stencils are given by:

$$\delta h \frac{\partial \boldsymbol{f}}{\partial h} = \boldsymbol{Q} \cdot \boldsymbol{f}^T \tag{2.23}$$

[14] Note that this means reducing the number of grid points from $N_i + 1$ to N_i. This can countered by adding one grid point to any periodic direction during initialization.

[15] There exist some exceptions to this. The exceptions are grid points handled by NSCBC, and grid points that are within a very small distance from the cylinder surface (usually set to $\sqrt{2}\delta x$). The latter exception will be described when details of the immersed boundary method are given in Section 2.5

with

$$\boldsymbol{Q} = \left(\begin{array}{cccccccccccc}
q_{1,1} & q_{1,2} & q_{1,3} & q_{1,4} & q_{1,5} & q_{1,6} & 0 & \cdots & & & & \\
q_{2,1} & 0 & q_{2,3} & q_{2,4} & q_{2,5} & q_{2,6} & 0 & \cdots & & & & \\
q_{3,1} & q_{3,2} & 0 & q_{3,4} & q_{3,5} & q_{3,6} & 0 & \cdots & & & & \\
q_{4,1} & q_{4,2} & q_{4,3} & 0 & q_{4,5} & q_{4,6} & q_{4,7} & 0 & \cdots & & & \\
q_{5,1} & q_{5,2} & q_{5,3} & q_{5,4} & 0 & q_{5,6} & q_{5,7} & q_{5,8} & 0 & \cdots & & \\
q_{6,1} & q_{6,2} & q_{6,3} & q_{6,4} & q_{6,5} & 0 & q_{6,7} & q_{6,8} & q_{6,9} & 0 & \cdots & \\
 & \cdots & 0 & -\frac{1}{60} & \frac{3}{20} & -\frac{3}{4} & 0 & \frac{3}{4} & -\frac{3}{20} & \frac{1}{60} & 0 & \cdots \\
 & & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots
\end{array}\right) \tag{2.24}$$

where coefficients $q_{j,k}$ can be found in Strand [96]. Similarly, the second derivative stencils are given by:

$$\delta h^2 \frac{\partial \boldsymbol{f}}{\partial h} = \boldsymbol{P} \cdot \boldsymbol{f}^T \tag{2.25}$$

with

$$\boldsymbol{P} = \left(\begin{array}{cccccccccccc}
p_{1,1} & p_{1,2} & p_{1,3} & p_{1,4} & p_{1,5} & p_{1,6} & 0 & \cdots & & & & \\
p_{2,1} & p_{2,2} & p_{2,3} & p_{2,4} & p_{2,5} & p_{2,6} & 0 & \cdots & & & & \\
p_{3,1} & p_{3,2} & p_{3,3} & p_{3,4} & p_{3,5} & p_{3,6} & 0 & \cdots & & & & \\
p_{4,1} & p_{4,2} & p_{4,3} & p_{4,4} & p_{4,5} & p_{4,6} & p_{4,7} & 0 & \cdots & & & \\
p_{5,1} & p_{5,2} & p_{5,3} & p_{5,4} & p_{5,5} & p_{5,6} & p_{5,7} & p_{5,8} & 0 & \cdots & & \\
p_{6,1} & p_{6,2} & p_{6,3} & p_{6,4} & p_{6,5} & p_{6,6} & p_{6,7} & p_{6,8} & p_{6,9} & 0 & \cdots & \\
 & \cdots & 0 & \frac{1}{90} & -\frac{3}{20} & \frac{3}{2} & -\frac{49}{18} & \frac{3}{2} & -\frac{3}{20} & \frac{1}{90} & 0 & \cdots \\
 & & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots
\end{array}\right) \tag{2.26}$$

where coefficients $p_{j,k}$ can be found in Mattsson and Nordström [61]. Alternative values for $q_{j,k}$ and $p_{j,k}$ can be found in Gustafsson [37]. Note that the SBP-operators for the first and second derivatives are consistent with the sixth-order centred finite-difference schemes on the interior of the domain. This can be seen from lines seven and on of the matrices $\boldsymbol{Q}$ and $\boldsymbol{P}$ being equal to matrix formulations of Eqs. (2.16) and (2.17), respectively. For the mixed derivatives, Eq. (2.18) is used, with SBP-operators from Eqs. (2.23) in the direction orthogonal to the surface.

Filtering

The centred finite-difference schemes used to discretize the governing equations are non-dissipative. This can cause problems due to the potential growth of high-frequency modes in the solutions, leading to spurious noise (wiggles) in computed fields or numerical instability. In the present work, such instabilities showed up in the density array, in particular on very fine non-uniform grids. Different methods of filtering the instabilities related to non-dissipative schemes are implemented in the Pencil Code. Alternatives include using upwinding rather than central-differences for certain computations, using a high-order implicit filter, or using hyper-diffusion. The latter option was not used. For details on hyper-diffusion in the Pencil Code, the reader is referred to Brandenburg and Sarson [18] and Haugen and Brandenburg [38].

The upwinding filter *turns on* dissipation in the computation of the density field. The procedure makes use of (dissipative) fifth-order upwind stencils for the advective operators of the density field. Because the difference between the sixth-order central and fifth-order upwind derivative is proportional to the operator of second-order accuracy, the upwind gradient can be computed by:

$$-u_h \delta h \left.\frac{\partial f}{\partial h}\right|_{\text{upwd}} = -u_h \delta h \left.\frac{\partial f}{\partial h}\right|_{\text{cent}} + \frac{1}{60}\left|u_h\right| \delta h^6 \left.\frac{\partial^6 f}{\partial h^6}\right|_{\text{cent}}, \tag{2.27}$$

for any sign of the velocity components in the h-direction, u_h, with the first term on the left-hand side computed by Eq. (2.16) and the second term (sixth derivative) computed by:

$$\delta h^6 \frac{\partial^6 f}{\partial h_i^6} \approx f_{i+3} - 6f_{i+2} + 15f_{i+1} - 20f_i + 15f_{i-1} - 6f_{i-2} + f_{i-3}\,. \tag{2.28}$$

For details see Dobler et al. [28].

Upwind filtering is an inexpensive way to add dissipation to the density field. The filter should, however, only be used for grids that are not body-conformal. This is because the necessary SBP boundary closures for the body-fitted grids are not implemented for the fifth-order upwind scheme or for the second-order central differences of the sixth derivative. The SBP boundary closures in Eqs. (2.24) and (2.26) are only appropriate for sixth-order central difference schemes of the first and second derivative, respectively[16]

In simulations performed during the development and implementation of the overset grid method in the Pencil Code, the impact of noise in the density field increased when cylindrical polar coordinates were used. This was especially true

[16] SBP operators for a fifth-order upwind scheme exist, and have been used for computation of the Euler equations [97]. The extension to the Navier-Stokes equations, if possible, requires a modification of boundary handling by simultaneous approximation terms which complicate the process considerably (see [31] and references therein).

when the grid was refined and when the flow was unsteady. As mentioned above, the upwind filtering in the Pencil Code cannot be applied to body-fitted cylindrical grids. Instead, high-order Padé filtering is used. The Padé filter is an implicit filter introduced by Visbal and Gaitonde [106] that filters high-frequency modes by:

$$\alpha_f \hat{\phi}_{i-1} + \hat{\phi}_i + \alpha_f \hat{\phi}_{i+1} = \sum_{n=0}^{N} \frac{\alpha_n}{2} \left(\phi_{i+n} + \phi_{i-n} \right), \tag{2.29}$$

where $\hat{\phi}_k$ and ϕ_k are components k of the filtered and unfiltered solution vectors, respectively. The parameter α_f is a free (filter) parameter ($\left|\alpha_f\right| \leq 0.5$) and N depends on the filter order. For the tenth-order filter used here, $N = 5$, and the fixed parameters α_n are:

$$a_0 = \frac{193+126a_f}{256}, \qquad a_1 = \frac{105+302a_f}{256}, \qquad a_2 = \frac{15(-1+2a_f)}{64},$$
$$a_3 = \frac{45(1-2a_f)}{512}, \qquad a_4 = \frac{5(-1+2a_f)}{256}, \qquad a_5 = \frac{1-2a_f}{512}.$$

Asymmetric filters are used near the solid boundary, where the one-sidedness of the filter increases as the boundary is approached. Different options have been implemented, and up to tenth-order one-sided filters can be used with overset grids. The filter used in the work done by Aarnes et al. [2] is a “0-6-8-8-8-10-" filter in the vicinity of the boundary. That is, the boundary value is not filtered, and the filter order for increasing distance from the surface is sixth-order, eighth-order (three grid points) and tenth-order (all points in the interior). The boundary stencils and parameter values can be found in Gaitonde and Visbal [34].

The high-order Padé filtering requires solving a tridiagonal linear system of equations (Eq. (2.29)) for every strip of grid points in each direction. In the radial direction a system of equations with i from 1 to N_r (where N_r is the number of grid point in the radial direction) is solved for every strip in the θ-direction. The filter is one-dimensional, making the system of equations manageable. Further, the requirement $\left|\alpha_f\right| \leq 0.5$ ensures that the system is diagonally dominant, hence it can be solved directly by the Thomas algorithm (a simplified form of Gaussian elimination without pivoting, developed by Thomas [101]). For strips along the θ-direction, the domain is periodic. This results in a cyclic tridiagonal system, that can be solved with the Thomas algorithm after application of the Sherman-Morrison formula [78].

The disadvantage of using an implicit filter (Padé filtering) as compared to explicit filters (upwinding, hyper-diffusion) is increased computational cost due to solving linear systems and increased communication between processors. The implicit Padé filter is, however, known to outperform explicit filtering schemes [106, 107] in terms of accuracy, and having less of an effect on larger flow structures.

Grid stretching

Many options for grid stretching are available in the Pencil Code. For grid stretching in the radial direction of the cylindrical grid, grid stretching by hyperbolic sine functions is appropriate. For grid points $i = 1,2,\ldots,N_\xi - 1$, the grid points in physical space[17], ξ_i, are computed by:

$$\xi_i = L_\xi \frac{\sinh\left(\dfrac{\gamma i}{N_\xi - 1}\right)}{\sinh\gamma} + \xi_0\,, \tag{2.30}$$

where

$$\gamma \equiv \alpha_g L_\xi\,, \tag{2.31}$$

L_ξ and N_ξ are the physical length and the number of grid points in the ξ-direction, respectively. The parameter α_g is a free (grid) parameter and ξ_0 is the location of the first grid point[18]. With this set-up for grid stretching, the smallest and largest grid spacings become:

$$\delta\xi_{\min} = \frac{L_\xi\gamma}{N_\xi - 1}\frac{1}{\sinh\gamma}, \tag{2.32}$$

$$\delta\xi_{\max} = \frac{L_\xi\gamma}{N_\xi - 1}\coth\gamma. \tag{2.33}$$

For communication between the background and body-fitted grids in the overset grid method, it is advantageous to have grid cells with similar grid spacing in the interpolation region. In addition, having near-quadratic cells close to the surface of the cylinder is recommended. The grid spacing in the tangential direction on the overset cylindrical grid (θ-direction) is simply $(2\pi r_i)/N_\theta$, where N_θ is the number of grid cells in the θ-direction and r_i is the radial coordinate of a strip of grid points. Hence, the ratio between the smallest and largest grid cells in the tangential direction is

$$\frac{\delta\theta_{\max}}{\delta\theta_{\min}} = \frac{r_{\text{grid}}}{r_{\text{cyl}}}, \tag{2.34}$$

[17] The grid set-up in the Pencil Code is not computed directly, as presented here, but uses coordinates in computational space and later converts to physical space. For the present purpose, however, the physical space description is adequate (and significantly easier to comprehend).

[18] Note that the expressions are simplified due to the inflection point of the stretching function being set to the boundary point ξ_0. More general expressions can be found in the grid-generation module in Pencil Code [100].

where r_{grid} is the radius of the body-fitted cylindrical grid and r_{cyl} is the radius of the cylinder that the grid is fitted to. With grid stretching in the radial direction by Eq. (2.30), the ratio between largest and smallest grid cells in the radial direction is:

$$\frac{\delta r_{\text{max}}}{\delta r_{\text{min}}} = \cosh \gamma. \tag{2.35}$$

Hence, to achieve similar grid spacing in the interpolation region on all grids and near-quadratic cells in the vicinity of the cylinder surface, the following steps should be taken:

- Set N_θ to an appropriate value for the cylinder grid size, such that $\delta \theta_{\text{max}} \approx \delta x$, where δx is the grid spacing on the Cartesian grid.
- Compute the free parameter α_g by combining Eqs.(2.31), (2.34) and (2.35) to get:

  $$\alpha_g = \frac{1}{L_r} \cosh^{-1}\left(\frac{r_{\text{grid}}}{r_{\text{cyl}}}\right), \tag{2.36}$$

  where $L_r = r_{\text{grid}} - r_{\text{cyl}}$.
- Set the number of grid points in the radial direction by inserting $\delta r_{\text{max}} \approx \delta \theta_{\text{max}}$ into Eq. (2.33), yielding

  $$N_r \approx \frac{N_\theta}{2\pi r_{\text{grid}}} \alpha_g L_r^2 \coth(\alpha_g L_r). \tag{2.37}$$

Other types of grid stretching can be used, but the stretching described here has the advantage of being a straightforward way to yield a grid that is finest in the vicinity of the cylinder and gradually becomes coarser as the distance from the cylinder is increased.

The finite-difference stencils of Eqs. (2.16)–(2.19) are expressed for uniform grids, yet with a slight modification these are also applicable to non-uniform grids. The first and second order derivatives in physical space (ξ) can be expressed as derivatives in computational space (h) by:

$$\frac{\partial f}{\partial \xi} = h'(\xi)\frac{\partial f}{\partial h}, \quad \frac{\partial^2 f}{\partial \xi^2} = \left(h'(\xi)\right)^2 \frac{\partial^2 f}{\partial h^2} - h''(\xi)h'(\xi)\frac{\partial f}{\partial h}. \tag{2.38}$$

The functions $h'(\xi)$ and $h''(\xi)$ are the first and second derivatives of h with respect to ξ. These can be set during grid generation and do not vary in time unless the grid is deformed during a simulation[19].

[19] The grid in computational space is generated such that it is equidistant with grid spacing $\delta h = 1$. Using deforming grids is not an option.

2.4. Turbulence

A central focus for much of the research within this work is effects of turbulence in the flow field on other flow phenomena. Turbulent flows feature a velocity field that fluctuates randomly in time and is highly disordered in space, with fluctuations that exhibit a wide range of length scales. Turbulence is unpredictable in the sense that a minute change to the initial conditions will produce a large change to the subsequent motion [26].

Important length scales used to characterize a turbulent flow field are (from largest to smallest) the integral scale, the Taylor microscale and the Kolmogorov scale. The integral scale and the Kolmogorov scale are characteristic length, velocity and time scales of the largest and smallest eddies in the turbulent flow field, respectively. The Taylor microscale is an intermediate scale in-between these two[20]. The energy cascade describes the relation between the scales of turbulence (as described by Richardson [83] and Kolmogorov [52, 53]). In brief, the largest eddies contain the largest proportion of the energy in the flow. Energy is continuously transferred from larger to smaller and smaller eddies, in a cascade, until the eddies become so small that they are strongly affected by the viscosity. At the scale of these isotropic (Kolmogorov sized) eddies, the kinetic energy is dissipated (into heat).

Computational simulations of turbulent flows

For simulation of turbulent flows, there are three main types of computations: Reynolds-Averaged Navier-Stokes (RANS) simulations, Large-Eddy Simulations (LES) and Direct Numerical Simulations (DNS).

In DNS, the Navier-Stokes equations are solved directly by the simulation software, with sufficiently fine resolution to resolve all scales (spatial and temporal) of the turbulent flow. The approach was popularized by Kim et al. [51] for a fully developed channel flow, and has been applied to a large range of flows since then. Because no modeling of the turbulence is used in DNS, it is regarded as the most accurate way to simulate turbulent flows. A major drawback of DNS is that such simulations are very computationally expensive. Resolving all scales of turbulence (which often span several orders of magnitude), yields the number of grid points as scaling by $Re_{\tau}^{2.25}$ (Re_{τ} is the Reynolds number defined in terms of the Taylor microscale) and the total cost as scaling by Re_{τ}^{3} in DNS [75]. All simulations in the

[20] Note that the Taylor microscale is not an arithmetic mean in-between the integral and the Kolmogorov scales, but a well-defined length scale that can be computed from two-point correlations of the flow field. A Taylor scale based Reynolds number is commonly used to quantify the turbulence of a flow field (see, e.g., Pope [75]).

present work utilize DNS.

LES is an approach where the largest scales of motion are represented explicitly while the small scales are treated by an approximate model [32]. In effect, low-pass filtered versions of the Navier-Stokes equations are solved, allowing for the use of larger grid spacing and time step sizes than in DNS. To avoid losing valuable information in the filtering process, the smaller scales are represented by sub-grid-scale (SGS) models. A popular SGS-model is the Smagorinsky model (Smagorinsky [92]). Although LES is much less expensive to perform than DNS, such simulations are in many cases too expensive for practical applications. For such applications, RANS simulation is the preferred method. RANS simulates time-averaged versions of the flow equations[21]. The time-averaging yields an additional stress term in the equations, which must be modeled by a turbulence closure model. Such closure models range from simple algebraic models (mixing length model) to second-order closure models (Reynolds Stress Models) [105]. Alternatives to the mentioned turbulence simulation strategies are unsteady RANS (URANS) and Very Large-Eddy simulations (VLES).

Turbulence generation

For turbulence to develop in the flow past a cylinder, the Reynolds number must be sufficiently high. The first instance of turbulence develops downstream of the cylinder in the transition-in-wake state of the flow. To develop turbulence in the shear layers or boundary layers around the cylinder, the Reynolds number must be much higher than in the transition-in-wake.

Another possible source of turbulence in such a flow is turbulence inserted in the free-stream. The present work is limited to moderate Reynolds numbers, and insertion of free-stream turbulence at the inlet. Before a turbulent flow field is inserted into the simulations, it is generated in a turbulence generation domain.

To generate turbulence, a rectangular box with periodic boundaries in all directions is used. Initially, the density and velocity fields are constant. Homogeneous isotropic turbulence is generated in this periodic box by forcing in random directions on a selected range of wave numbers. A forcing function is used to generate the force term in the momentum equation (Eq. (2.2)), as described by Brandenburg [17], Haugen and Brandenburg [38], and Babkovskaia et al. [7]. The forcing function is:

$$\boldsymbol{F}(\boldsymbol{x},t) = \mathscr{R}\{N \boldsymbol{f}_{\boldsymbol{k}(t)} e^{i\boldsymbol{k}(t)x + i\varphi(t)}\}, \tag{2.39}$$

where $\boldsymbol{x}$ is the position vector and $\mathscr{R}$ indicates the real part of the expression. The wave vector $\boldsymbol{k}(t)$ and random phase $-\pi \leq \varphi(t) \leq \pi$ change at every time step, so

[21] To emphasize the time-averaging, the term *steady* RANS is often used.

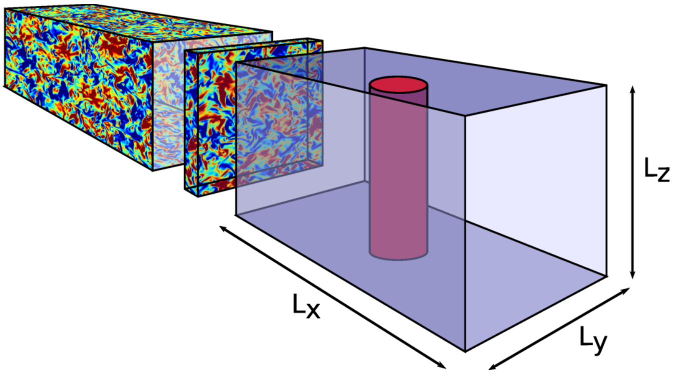


Figure 2.1.: The insertion of a turbulent flow field at the inlet of the flow domain. Thin slices of velocity data are taken from the turbulence domain (left box) and added to the velocity field at the inlet of flow domain (right box). Figure reproduced from Aarnes et al. [4].

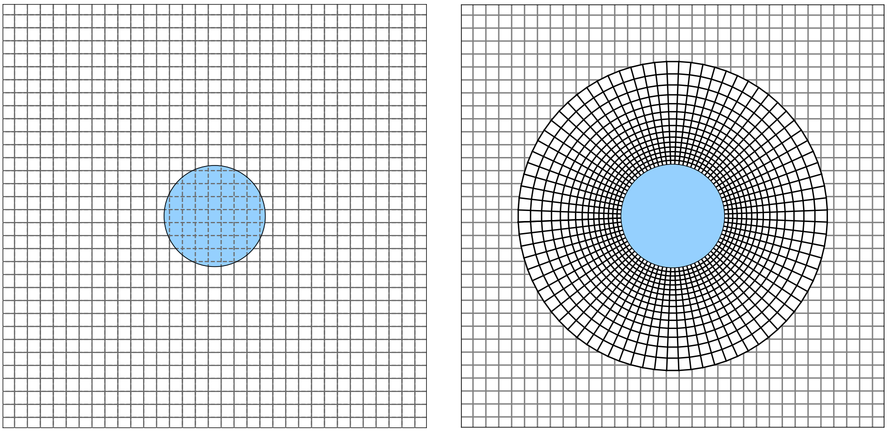
$\boldsymbol{F}(\boldsymbol{x},t)$ is delta-correlated in time. On dimensional grounds, the prefactor is chosen to be $N = f_0 c_s \sqrt{k c_s/\delta t}$, where $k = |\boldsymbol{k}|$ and f_0 are non-dimensional factors chosen to regulate the intensity of the turbulence.

At each time step, one of many possible wave vectors in a certain range around a given forcing wave number is selected. For example, with a given forcing wave number $k_f = 5$ this range may be $4.5 < |\boldsymbol{k}| < 5.5$. The system is then forced with non-helical transversal waves,

$$\boldsymbol{f}_k = \frac{\boldsymbol{k} \times \hat{\boldsymbol{e}}}{\sqrt{\boldsymbol{k}^2 - (\boldsymbol{k} \cdot \hat{\boldsymbol{e}})^2}}\,, \tag{2.40}$$

where $\hat{\boldsymbol{e}}$ is an arbitrary unit vector that is real and not aligned with $\boldsymbol{k}$. Note that $|\boldsymbol{f}_k|^2 = 1$.

The turbulence generation is run until it is statistically stationary. When this point is reached, the turbulence can be inserted at the flow domain's inlet. The insertion process is straightforward, with slices of data taken from the turbulence generation domain and added to the inlet of the flow domain at every timestep (see Fig. 2.1). The inlet of the flow domain has a fixed mean velocity. With mean flow in the x-direction, the mean velocity is $\boldsymbol{U}_0 = (U_0, 0, 0)$. Since the turbulence generation domain has a mean velocity of zero (isotropic homogeneous turbulence) the mean inlet velocity is not affected by the added turbulent flow field. The velocities at the inlet are updated as $\boldsymbol{U} = \boldsymbol{U}_0 + \boldsymbol{u}'$ at every time step, where $\boldsymbol{u}'$ are the velocity fluctuations in the slice taken from the turbulence generation domain.



(a) IBM

(b) Overset grids

Figure 2.2.: Illustration of grids used to represent solid objects in the Pencil Code, by either immersed boundary methods (a) or overset grids (b).

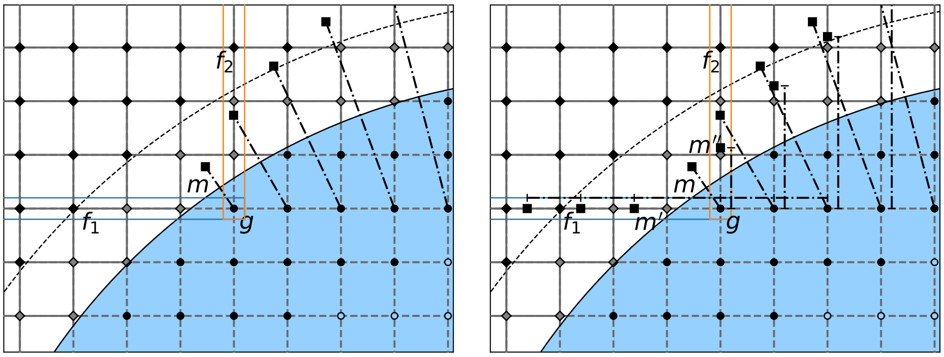
To avoid outflow at the inlet, f_0 used in the forcing function is chosen such that $\max\{|u'_x|, |u'_y|, |u'_z|\} < U_0$. Note that the size of the inlet (L_y and L_z in Fig. 2.1) must be the same as the size of the data slices taken from the turbulence generation domain. The two domains must also have the same number of grid points in these directions.

2.5. Resolving solid boundaries

Representing a solid cylinder in a flow simulation may appear trivial, but the degree of accuracy necessary for resolving impaction of very small embedded particles in the flow made this one of the major challenges in this research. The background for the two different methods used for this purpose was given in Section 1.3. In this section a description of the implementations in the Pencil Code is presented. It is based in part on material from research articles included in Chapter 5, in particular from Aarnes et al. [1] and Aarnes et al. [5].

Immersed boundary methods

As mentioned in the introductory chapter, IBMs are methods that resolve boundaries in the flow without fitting a grid to the boundaries. Typically, a non-body conformal Cartesian grid is used for the entire flow domain, as illustrated in Figure 2.2(a). In



(a) Orthogonal mirror-point IBM

(b) Cartesian mirror-point IBM

Figure 2.3.: Immersed boundary method. A zone of ghost-points is used when computing fluid-points. Ghost-points are set from corresponding mirror-points in the flow domain, after identification of mirror-points along surface normals (a) or along Cartesian grid lines and surface normals (b). Fluid-points within a fixed distance from the surface are computed by interpolation. (●) ghost-points, (◆) fluid-points, (◈) interpolated fluid-points, (○) hole-points, (■) mirror-points.

this illustration, the intersections of solid grid lines represent fluid-points, where the governing equations are solved by using the finite-difference stencils defined in Eqs. (2.16),(2.17) and (2.19). The intersection points of dashed grid lines are grid points inside the solid (solid-points) where the flow equations are not solved. The solid-points can be split into two groups: ghost-points and hole-points. The ghost-points are set by corresponding mirror-points and used in the finite-difference stencil of fluid-points in the vicinity of the solid surface, while hole-points are unused grid-points. The two IBM implementations in the Pencil Code that are considered here differ only in one aspect: how the velocity components in the ghost-points are set.

Ghost-points and mirror-points

Flow variables (velocity and density) in the ghost-points are not computed by solving the governing equations. Rather, they are set from corresponding mirror-points in the flow domain, by imposing boundary conditions at the solid surface. At the mirror-points the flow variables are computed by interpolation from surrounding fluid-points. When the ghost-points are used by other fluid-points, the boundary conditions are imposed on the fluid flow. The boundary conditions used for the simulations of a cylinder in a cross-flow are no-slip and impermeability for the velocity components, and zero gradient for the density normal to the surface[22]. The distance from a ghost-point to a mirror-point is twice the distance from the ghost-point to the surface. This yields the following relationship between a flow variables in a ghost-point (ϕ_g) and its corresponding mirror-point (ϕ_m):

$$\phi_g(\boldsymbol{u}) = -\phi_m(\boldsymbol{u}), \tag{2.41}$$

$$\phi_g(\rho) = \phi_m(\rho). \tag{2.42}$$

The two different methods used to compute ghost-points in the present work are by identifying mirror-points along surface normals through ghost-points or along grid-lines through ghost-points. The different methods are denoted orthogonal mirror-point IBM (OIBM) and Cartesian mirror-point IBM (CIBM).

In the OIBM, each ghost-point is related to a single mirror-point, as seen in Figure 2.3(a), where points g and m make up a ghost/mirror-point pair. The values set in ghost-point g from m (by Eqs.(2.41) and (2.42)) is used in the finite-difference stencils of fluid-points f_1 and f_2, when computing derivatives along the horizontal and vertical directions, respectively. Conversely in the CIBM, each ghost-point is related to several mirror-points, as seen Figure 2.3(b), where ghost-point g is connected to mirror-points m, m' and m''. The density in point g is set identically to the orthogonal mirror-point IBM, using data interpolated to point m. Velocity components are set from points m' and m''. When g enters as a grid-point in the horizontal gradients of point f_1, velocities in g as set from m' are used. When the vertical gradients in point f_2 use g, velocities in g are set from mirror-point m''.

The handling of ghost- and mirror-points in the CIBM is somewhat more intricate than in the OIBM, but it has the advantages of reducing distances between fluid-points and mirror-points, and simplifying interpolation of mirror-points. Flow variables in mirror-points are calculated by Lagrangian interpolation (Abramowitz and Stegun [6]); bi-linear for OIBM and quadratic for CIBM (velocity components only). With OIBM, flow variable ϕ in a mirror-point m at a position (x_m, y_m) is computed from the four surrounding grid points $(x_i, y_j), (x_{i+1}, y_j), (x_i, y_{j+1}), (x_{i+1}, y_{j+1})$ by

$$\phi_m = C_1 + C_2\tilde{x} + C_3\tilde{y} + C_4\tilde{x}\tilde{y}, \tag{2.43}$$

[22] The boundary condition for density can be derived by applying the boundary layer approximation for pressure, $\partial p/\partial r = 0$ (White [113]), to the ideal gas law for an isothermal flow.

where

$$
\begin{aligned}
C_1 &= \phi_{i,j}\,, \\
C_2 &= -\phi_{i,j} + \phi_{i+1,j}\,, \\
C_2 &= -\phi_{i,j} + \phi_{i,j+1}\,, \\
C_3 &= \phi_{i,j} - \phi_{i+1,j} - \phi_{i,j+1} + \phi_{i+1,j+1}\,,
\end{aligned}
\tag{2.44}
$$

and

$$
\tilde{x} = \frac{x_m - x_i}{x_{i+1} - x_i}\,, \quad \tilde{y} = \frac{y_m - y_j}{y_{j+1} - y_j}\,. \tag{2.45}
$$

With CIBM, velocity components are interpolated using one-dimensional interpolants along the grid-lines. This makes increasing the order of interpolation from linear to quadratic trivial. The quadratic interpolation stencils use three grid-points, where one of the three grid-points is always the interception point of the grid-line with the solid boundary. Because velocity components are zero at this point, the interpolation of values to a mirror-point in-between fluid-points (x_i, y_j) and (x_{i+1}, y_j), next to a boundary interception point (x_b, y_j) is:

$$
\phi_m = L_1 \phi_{i,j} + L_2 \phi_{i+1,j}\,, \tag{2.46}
$$

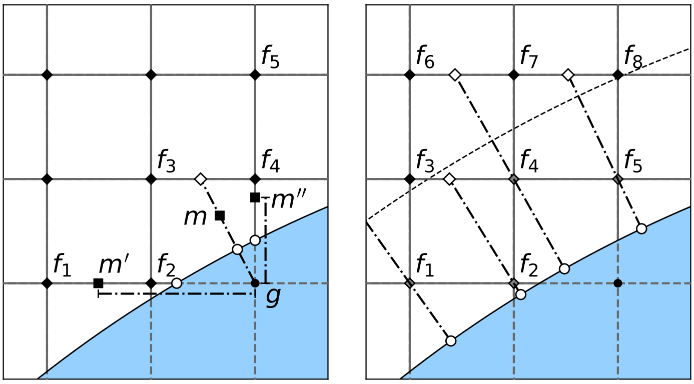
where

$$
\begin{aligned}
L_1(x_m, x_i, x_{i+1}) &= \left(\frac{x_m - x_b}{x_i - x_b}\right)\left(\frac{x_m - x_{i+1}}{x_i - x_{i+1}}\right)\,, \\
L_2(x_m, x_i, x_{i+1}) &= \left(\frac{x_m - x_b}{x_{i+1} - x_b}\right)\left(\frac{x_m - x_i}{x_{i+1} - x_i}\right)\,,
\end{aligned}
\tag{2.47}
$$

and similarly for mirror-points in-between grid points (x_i, y_j) and (x_i, y_{j+1}) using $L_1(y_m, y_j, y_{j+1})$ and $L_2(y_m, y_j, y_{j+1})$.

Exceptions

Not all mirror-points can be set by interpolation from surrounding fluid-points. Mirror-points that are very close to the surface, like point m in Figure 2.3(a), are not surrounded by enough fluid-points to use Eqs. (2.43)–(2.45) directly, without including ghost-points in the interpolation stencil. Flow velocities at such points are computed by interpolation along the surface normal, where the intersection points of the surface normal with the solid boundary and with the first grid-line outside the solid boundary are used for interpolation. Data at the intersection with the grid-line is interpolated from the two neighboring points along the grid-line.



(a) Mirror-points

(b) Fluid-points

Figure 2.4.: Special handling of mirror-points and fluid-points that are very close to the solid surface. (●) ghost-points, (◆) fluid-points, (◇) grid-line interception-point, (○) boundary interception-point, (■) mirror-points.

Since the velocity normal to the surface is expected to increase as a second order function with distance from the surface, it is beneficial to interpolate this variable by a quadratic expression. Velocities at the two interception points are first decomposed into radial and tangential velocities. The radial component of the velocity at the mirror-point is interpolated by the quadratic expression:

$$u_{r,m} = u_{r,GI} \left(\frac{\delta r_m}{\delta r_{GI}} \right)^2 , \tag{2.48}$$

where $u_{r,GI}$ is the radial velocity at the grid-line interception point, and δr_m and δr_{GI} are the distances from the mirror-point and grid-line interception point to the boundary interception point, respectively. Note that this interpolation expression is quadratic, yet guarantees that there are no overshoots in the interpolated values. The tangential velocity component is interpolated by:

$$u_{\theta,m} = u_{\theta,GI} \left(\frac{\delta r_m}{\delta r_{GI}} \right) . \tag{2.49}$$

The expressions for interpolation of $u_{r,m}$ and $u_{\theta,m}$ are valid only when $\boldsymbol{u} = 0$ at the surface.

Figure 2.4(a) depicts such a case, for mirror-point m (used by ghost-point g). Data at fluid-points f_3 and f_4 are used to interpolate values to the grid-intersection point (◇), such that the velocities at m can be computed from this and the known boundary value at the boundary interception point (○). Note that this handling is not used for density, as the density at the boundary is not known (Neumann boundary condition). Instead, regular interpolation stencils are applied, which works for density, due to very little variation in the density field and a symmetric boundary condition.

If a mirror-point is identified along a grid-line with the CIBM closer to the surface than the nearest fluid-point along the grid line, the interpolation is simply shifted one cell away from the surface. In Figure 2.4(a) the mirror-point m' is in-between fluid-points f_1 and f_2, and uses these fluid-points and the boundary interception-point next to f_2 in the interpolation. Mirror-point m'' is too close to the surface for such handling, and shifts the interpolation stencil to use f_4 and f_5. Since the boundary interception-point next to m'' is also used in the interpolation, extrapolation is avoided and the shifting of the boundary stencil is does not produce problems.

The second type of exception in the IBM implementation is the handling of fluid-points very close to the surface. These points are particularly sensitive to disturbances in the flow field, and for this reason an option is included to compute them directly by interpolation from surrounding points and boundary interception-points, rather than using finite-difference stencils[23]. If a fluid-point is within a pre-defined cut-off distance from the cylinder[24], the flow variables at the grid point are computed by interpolation along surface normals, similarly to mirror-points in the OIBM that were too close to the surface to be surrounded by enough fluid-points. Figure 2.4(b) depicts such an interpolation: The fluid points f_1, f_2, f_4 and f_5 are within the cut-off distance (dashed line). For each of these fluid-points, the boundary interception-point and the grid-line interception at the first grid-line from the surface are used in interpolation. As for mirror-point handling, data at the grid-line interception is interpolated from neighboring fluid-points, e.g. values at f_6 and f_7 are used to interpolate data to the grid-line interception point used for interpolation to fluid-point f_4. As for the mirror-points, the radial velocity component is interpolated by the quadratic expression in Eq. (2.48).

Overset grids

Representing a solid in the flow by an overset grid means applying a body-fitted grid to said object. Other considerations, like the shape of the flow domain, number

[23] Finite-difference stencils used to compute grid points very close to the surface would include ghost-points quite far into the solid cylinder, in particularly for computation of mixed derivatives. This is avoided by handling such points as exceptions.

[24] Default value is $\sqrt{2}\delta x$.

of objects in the flow[25], inlet and outlet conditions, etc., are specified by the background grid and by communication between the overset grids.

The governing flow equations are discretized on each grid, which allows for different coordinate systems to be used in the same flow simulation. For flow past a cylinder in a rectangular domain, the appropriate choice is a Cartesian background grid where Eqs. (A.1)–(A.4) are used for the flow. A cylindrical grid is fitted to the solid (see Figure 2.2(b)) and the governing equations in Eqs. (A.12)–(A.15) are discretized on this grid. Hence, a generalized flow solver (or two different flow solvers) is necessary to update the solution on the different grids.

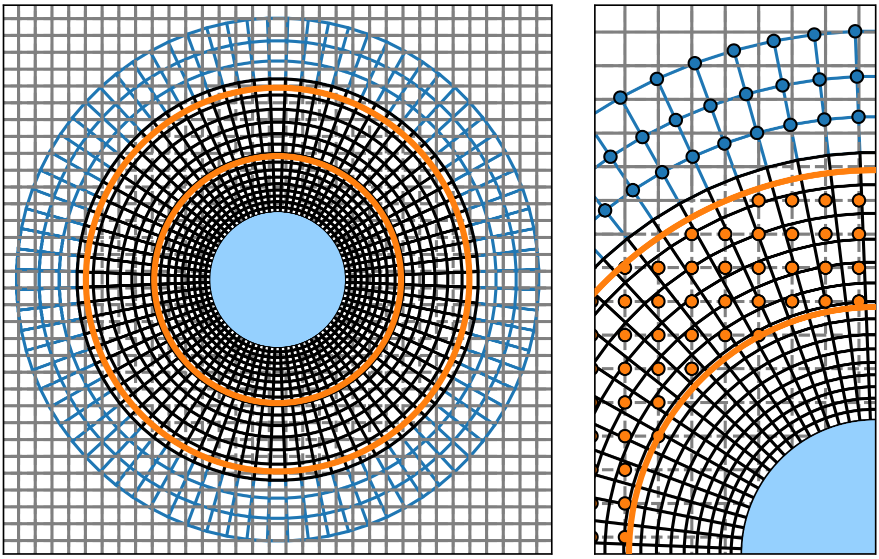
For the case of a cylinder in a cross-flow, the inlet, outlet, periodic boundaries, etc., are handled by the background grid. Using overset grids does not affect how the domain boundaries are specified. The solid object's boundary, however, is handled in a very different manner than when resolved with IBMs. Since the grid surrounding the cylinder is body-conformal, SBP boundary closures of Eqs. (2.23)–(2.26) can be used for third-order accurate, time-stable solutions at the boundary. Hence, no ghost/mirror-points or interpolation is used to enforce the boundary conditions at the cylinder surface. The physical boundary conditions (no-slip and impenetrability for velocity and zero density gradient normal to the surface) do not change.

Communication between grids

Each of the grids must take into account one additional boundary: the numerical boundary where data is communicated between grids by interpolation. This boundary consists of a region of overlapping background and body-fitted grids. In Figure 2.2(b) this region is hidden to make the illustration of the overset grid clearer. In practice, the Cartesian grid penetrates into the cylindrical grid, and the cylindrical grid is extended with grid points extending further out over the Cartesian grid points. In the overlapping region, the grid points can be split into regular fluid-points (some which are used as donor-points in interpolation), interpolated fluid-points (fringe-points) and unused grid points (hole-points). A more detailed depiction of the overset grids can be seen in Figure 2.5.

Identification of fringe-points on the two different grids is quite different. The cylindrical grid simply uses a zone of fringe-points three points thick outside the fluid-points (blue grid-lines/grid-points in Figure 2.5). For the background grid, an inner and outer radius of fringe-points is set during pre-processing, and all points within the bounds of these radii are identified as fringe-points (orange

[25] At present, the overset grid implementation in the Pencil Code only allows for a single grid overset a background grid. Multiple grids overset one another will therefore not be considered here.



(a) Identification of fringe-points

(b) Fringe-points

Figure 2.5.: Identification of fringe-points in region of overlapping grids on a background grid with $\delta x/D = 8$.

circles/grid points in Figure 2.5). The outer radius is set first, in accordance with a criterion of explicit interpolation[26]; hence, the outer radius is smaller if higher-order interpolation is used. The inner radius is computed from the outer radius, thick enough to guarantee that all fluid-points use only other fluid-points or fringe-points in the finite-difference stencils (not hole-points). All grid points on the background grid within the inner radius are identified as hole-points.

After identification of fringe-points, the grid points on the opposing grid surrounding each fringe-point are identified as donor-points. Flow variables from these donor-points are interpolated to the fringe-point. The interpolated velocity components must be transformed to the appropriate coordinate system, before used in the flow arrays. The analytic relations between Cartesian and cylindrical (polar) coordinates are:

$$\begin{aligned} x &= r\cos\theta\,, \quad & r &= \sqrt{x^2+y^2}\,, \\ y &= r\sin\theta\,, \quad & \theta &= \arctan\left(\frac{y}{x}\right)\,. \end{aligned} \tag{2.50}$$

[26] Explicit interpolation on overset grids means interpolating from donor-points to fringe-points, where all donor-points are fluid-points. Using implicit interpolation between grids means that donor-points can also be fringe-points (Chesshire and Henshaw [21])

The velocity transformations can be done in a straightforward way by:

$$\begin{aligned} u_x &= u_r \cos\theta - u_\theta \sin\theta\,, \quad & u_r &= u_x \cos\theta + u_y \sin\theta\,, \\ u_y &= u_r \sin\theta + u_\theta \cos\theta\,, \quad & u_\theta &= -u_x \sin\theta + u_y \cos\theta\,. \end{aligned} \tag{2.51}$$

The number of donor-points for each fringe-point depends on the interpolation stencil. Typically, the z-plane is shared between the Cartesian and the cylindrical systems, hence stencils with four or nine donor-points are used for bi-linear or bi-quadratic interpolation, respectively. If bi-linear interpolation is selected, Eqs. (2.43)–(2.45) can be used directly regardless of whether the interpolation is the Cartesian to the curvilinear grid or vice versa, because the grid points are all on rectangular grid-cells in computational space. For quadratic interpolation, spline interpolation stencils can be used. In the Pencil Code, the form of the quadratic splines for interpolation of a point p at position x_p from surrounding donor points $i-1, i, i+1$ is:

$$\phi_p = S_0 \phi_{i-1} + S_1 \phi_i + S_2 \phi_{i+1}\,, \tag{2.52}$$

where

$$S_0 = \frac{1}{2}\left(\frac{1}{2} - \delta x_p\right)^2, \tag{2.53}$$

$$S_1 = \frac{3}{4} - \delta x_p^2\,, \tag{2.54}$$

$$S_2 = \frac{1}{2}\left(\frac{1}{2} + \delta x_p\right)^2, \tag{2.55}$$

with

$$\delta x_p = \frac{x_p - x_i}{\delta x_i}\,, \tag{2.56}$$

and δx_i is the grid spacing at grid point i. Other, higher-order interpolants can also be used, but the implementations of these have not been properly tested at present.

Timestep on overset grids

When overset grids are used to resolve a solid boundary, the timestep is split into four parts: updating the flow array using the governing equations on the Cartesian grid, communicating the boundary data from the Cartesian to the cylindrical grid, updating the flow array with the governing equations on the cylindrical grid, and, lastly, communicating the boundary data from the cylindrical to the Cartesian grid.

In some overset grid implementations (see, e.g. Sherer and Scott [91]) such splitting is done for each sub-iteration of the timestep in the Runge-Kutta method. This is not done in the overset grid implementation in the Pencil Code. The Cartesian flow solver instead completes an entire Runge-Kutta timestep before communicating flow data to the cylindrical grid. Following this, several Runge-Kutta timesteps are performed on the cylindrical grid, before data is sent back to the Cartesian grid.

There are two main advantages of splitting the timestep as it is done in the Pencil Code. Firstly, the data communication between grids is reduced. This saves both computational time on each processor and communication time used to transfer data between processors. Communicating data once per Runge-Kutta timestep reduced these costs by a factor of three, when compared to communicating once per sub-iteration. Secondly (and more importantly), completing several timesteps on the cylindrical grid for each timestep on the background grid is a way to circumvent the very strict timestep requirement imposed by using an explicit method for the compressible Navier-Stokes equations. This timestep restriction is the minimum of the advective and diffusive timestep restrictions,

$$\delta t_\nu \leq \frac{C_\nu \delta\xi_{\min}^2}{\nu}, \quad \delta t_u \leq \frac{C_u \delta\xi_{\min}}{|\boldsymbol{u}|+c_s}, \tag{2.57}$$

respectively, where $\delta\xi_{min}$ is the smallest grid spacing in any direction (on the relevant grid), and C_ν and C_u are the diffusive and advective Courant numbers, respectively. For weakly compressible flows past a cylinder, the timestep restriction is typically much smaller than the necessary timestep to accurately resolve the flow, due to the necessity for a fine grid in the vicinity of the cylinder. By allowing several timesteps on the cylindrical grid for each time step on the background grid, the timestep restriction from the fine grid cells in the vicinity of the cylinder no longer applies as a global timestep restriction. Thus, not only is computational cost related to updating the region of fringe-points reduced, but the cost of the flow solver as a whole is dramatically reduced by allowing a much larger timestep in the major regions of the flow domain.

2.6. Simulating particle-laden flow

When the flow is particle-laden, the fluid-phase of the flow is handled with the numerical methods described in the preceding sections, in the same way as if there were no particles in the flow. For the Lagrangian particles, the third-order Runge-Kutta method used for time integration in the same way as for the flow equations. Other than this, the handling of the particle equations is fundamentally different from the flow equations. New considerations include the insertion of particles into the flow, interpolation of flow data to particle positions, and particles impaction on the cylinder.

Particle insertion

Particles are inserted at random positions within a limited volume near the inlet. The intention of inserting particles into the flow is to study particle impaction on the cylinder. For this reason, it is only necessary to insert particles over a volume where the particles have a chance of hitting the cylinder. When the flow is laminar, the front-side of the volume is the area that covers the projected cylinder area. No particles inserted outside of this area will hit the cylinder[27]. For turbulent flow, the particle trajectories do not follow the mean flow, and a larger insertion volume is necessary. The size is dependent on the turbulence intensity and distance from the inlet to the cylinder[28]. Whether particles are inserted over the entire flow domain or not, the impaction efficiency:

$$\eta = N_{\text{impact}}/N_{\text{insert}} \tag{2.58}$$

only takes into account the number of inserted particles (N_{insert}) over the projected cylinder area when computed from the number of particles that impact the cylinder (N_{impact}).

For the smallest particle sizes $St = 0.01$–0.1, the number of particles that impact on the cylinder is very small ($\eta_f \approx 10^{-5}$–10^{-4}). For confidence in the simulation statistics, when the impaction efficiencies are computed, N_{impact} should be large enough that a single impaction does not significantly impact η. With $\eta_f \approx 10^{-5}$ for $St = 0.01$, this would require N_{insert} of order 10^7 for this particular particle size. Very small particles are light enough to follow the fluid almost perfectly. Hence, it is reasonable to assume that only particles inserted very close to the center of the inlet will have a chance of impacting on the cylinder, as fluid is *pushed away* from the centerline upon approach to the cylinder. Consequently, inserting very small particles close to the centerline only, and scaling up N_{insert} correspondingly during post-processing is a reasonable way to reduce the number of particles required when very small particles in a laminar flow are considered. This technique has been used with success in Aarnes et al. [2], where particles with $St \leq 0.1$ where

[27] In theory, particles inserted outside the projected cylinder area can hit the cylinder, due to their finite radii. However, the force from the fluid upon the particles will push the particles away from the cylinder, and no particles inserted outside the area hit the cylinder in these simulations.

[28] Insertion over the entire inlet was used for simplicity, for the particle-laden turbulent flow simulations in Aarnes et al. [3]

inserted only over one-tenth of the projected cylinder area. Note that for flow with turbulence in the free-stream, this would not be an appropriate way of inserting the particles, because particles inserted farther from the centerline can be convected towards and past the centerline when approaching the cylinder.

Updating the particle equations

To update the particle equations (Eqs. (2.8) and (2.9)), the fluid velocity and density at each particle position is necessary. The particle positions do not, in general, coincide with the grid points, so these values must be set using surrounding grid points.

The simplest way to compute the fluid velocity and density at a particle position is by using the nearest grid point-approximation, which consists of setting $\boldsymbol{u}$ and ρ equal to the values at the grid points closest to $\boldsymbol{x}_p$. While this is a very fast way to update the particles, it is too crude an approximation for the particle impaction simulations. Instead, a cloud-in-cell approach is used, which means that the data is interpolated from the surrounding grid points. Hence, bi-linear interpolation (Eqs. (2.43)–(2.45)) is used for two-dimensional simulations, and tri-linear for three-dimensional runs.

If an IBM is used to represent the cylinder, particles very closer to the cylinder surface will not be surrounded by enough fluid-points to use the cloud-in-cell interpolation. Such particles are handled by the interpolation procedure described for mirror-points along surface normals that are surrounded by less than four fluid-points (see handling of point m in Figure 2.4(a)). The use of quadratic interpolation for the radial velocity component of particles in the immediate vicinity of the cylinder has an especially large effect on the accuracy of particle impaction simulations.

When the cylinder is represented by overset grids, the interpolation changes once the particles are within the fringe-point radius. For particles within this radius, data from the cylindrical grid is used, rather than from the background grid. The type of interpolation scheme is set by the Pencil Code user to be either tri-quadratic, bi-quadratic/linear, quadratic/bi-linear or tri-linear (for three-dimensional simulations). In the case of bi-quadratic/linear interpolation, the interpolation is linear in the z-direction. Alternatively, for quadratic/bi-linear interpolation, linear interpolation is used for θ and z-directions. The recommended settings are bi-quadratic/linear for radial and tangential velocity components (18 grid points used in 3D) and tri-linear for velocity in the z-direction and density (eight grid points used in 3D). The quadratic interpolation is Lagrangian, hence interpolation stencils:

$$\phi_p = L_0\phi_{i-1} + L_2\phi_i + L_3\phi_{i+1}\,, \tag{2.59}$$

with

$$\begin{aligned} L_1(x_p,x_{i-1},x_i,x_{i+1}) &= \left(\frac{x_p - x_i}{x_{i-1} - x_i}\right)\left(\frac{x_p - x_{i+1}}{x_{i-1} - x_{i+1}}\right), \\ L_1(x_p,x_{i-1},x_i,x_{i+1}) &= \left(\frac{x_p - x_{i-1}}{x_i - x_{i-1}}\right)\left(\frac{x_p - x_{i+1}}{x_i - x_{i+1}}\right), \\ L_2(x_p,x_{i-1},x_i,x_{i+1}) &= \left(\frac{x_p - x_{i-1}}{x_{i+1} - x_{i-1}}\right)\left(\frac{x_p - x_i}{x_{i+1} - x_i}\right), \end{aligned} \tag{2.60}$$

are applied in each quadratic direction. Points x_{i-1}, x_i, x_{i+1} are chosen such that $x_{i-1} < x_p < x_{i+1}$ and $\text{Min}\left(|x_p - xj|\right)\big|_{j=i-1,i,i+1} = |x_p - x_i|$. To avoid overshoots in the vicinity of the cylinder, the quadratic interpolation scheme for particles closer to the surface than the closest grid point[29] changes to Eq. (2.48) in the radial direction. All interpolation from the cylindrical grid is done in the computational space of radial and tangential coordinates.

Particle impaction

If a particle contacts the cylinder, the particle is removed from the simulation and an impaction is registered. Particle sticking mechanisms, bouncing, etc. are not considered. Although the particles in the simulations are point-particles, their finite radii do come into consideration when impaction is considered. For a particle to impaction on the cylinder, it is sufficient that the particle and cylinder radii overlap. This impaction mechanism allows for boundary interception, that is, impaction on the cylinder by the particle being intercepted by the cylinder due to the finite radii. Figure 2.6 illustrates this mechanism.

2.7. Parallelization

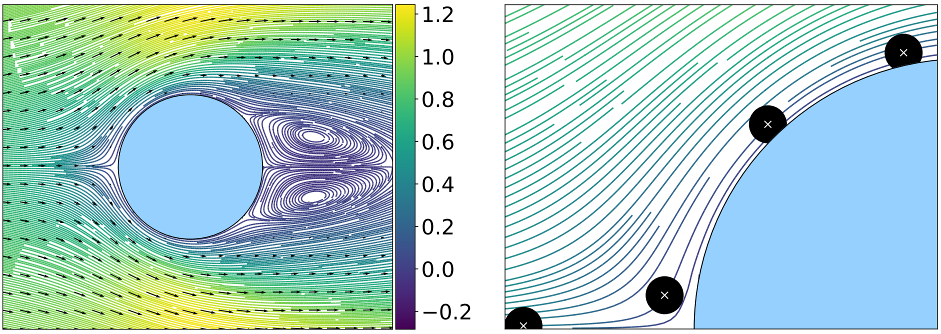
The Pencil Code is parallelized by domain decomposition (data-parallelism), using the Message Passing Interface (MPI) to communicate data between processors on distributed memory systems[30]. The domain decomposition is a straightforward parallelization where grid points in the flow domain are assigned a processor according

[29] Users may choose a larger distance than this for the special handling, such as the momentum thickness of a stagnation point flow [113]:

$$\delta_m \approx 0.20669 \frac{D}{\sqrt{2Re}}$$

as the limit for the special handling.

[30] Data-parallelism on GPUs is also possible, for parts of the code, but will not be considered here.



(a) Streamlines for $Re = 20$

(b) Particle impaction by interception

Figure 2.6.: Trajectories for particles that following the streamlines to a large extent. Without boundary interception, the particles in positions three and four (counting from left to right) will not be removed from the simulation, but pass the cylinder. (●) particle, (×) center of mass of particle. Mean flow from left to right, colouring for horizontal velocity component.

to location, and local flow variable arrays are constructed on each processor. A processor's array of flow variables is padded with three ghost-points on all sides, where copies of flow variables from neighboring processors are stored. These halos of ghost-points are updated at every sub-iteration of the Runge-Kutta scheme, such that the numerical solution is not affected by the parallelization.

With large particle simulations, an efficient parallelization is necessary. Typically, one of two methods is used: (1) particles are located on the processor that "owns" the part of the flow domain on which the particle is located (spatial particle parallelization) or (2) particles are distributed equally among processors, regardless of spatial position. While (2) has the advantage of load balancing of particles distributed unevenly in space, large amounts of data communication is a drawback (global flow arrays may be necessary). Spatial particle distribution (1) is the preferred method in the Pencil Code although a version of (2) called particle block domain decomposition is available (see Johansen et al. [49]).

Parallelization tests can be found in the manual for the Pencil Code [16]. Timing for up to 73 728 processors are included. Timing results show near linear weak scaling. Strong scaling was also considered, which follows an approximate second-order fit with very good scaling for $\geq 16^3$ grid-points per processor in three-dimensional simulations. Scaling for large simulations with a fixed grid (512^3 grid points) and a large amounts of particles in the flow (6.4×10^6) on 128–4096 cores showed a linear trend.

When overset grids are used, each processor will own a part of each grid (background and overset grids). This data layout is chosen to avoid idle processors during the update steps on different grids, as these cannot be updated simultaneously. In general, the domain covered by the background grid on one processor may or may not overlap with that same processor's domain on the cylindrical body-fitted grid. Hence, inter-processor communication may be necessary during inter-grid communication of interpolated flow variables in the overlapping region.

At present, the particle handling in (1) is not altered when overset grids are used. Hence, a particle is identified on a processor by (x,y,z)-coordinate, not (r,θ,z)-coordinate. This means that additional communication may be necessary when a particle enters the region covered by the body-fitted grid. For this reason, part of the disadvantage of particle parallelization in (2) is encountered even though (1) is used. Although significant, the increased data communication is not detrimental to the parallel simulations on overset grids, since:

- only parts of the body-fitted grid need to be communicated; that which is overlapped by the background grid of the processor in question.
- the communication needs only occur once per Runge-Kutta timestep on the background grid, as flow variables on the body-fitted grid are not updated during the sub-timesteps on the background grid.
- the total number of grid points on the body-fitted grid is much smaller than that on the background grid.

As an example of the latter point, consider two-dimensional rectangular domains with domain size $L_x \times L_y = 10D \times 20D$, used for particle-laden flows in Aarnes et al. [2]. Here, the ratio of grid points on the body-fitted mesh to the grid points on the background mesh is $1/10$. Even if a global flow array of the body-fitted grid[31] is communicated, it is still much smaller than communicating in the opposite direction (from the background to the body-fitted grid) or generating global Cartesian data arrays to use with particle handling (2).

[31] Note that such global arrays are not used. The worst case scenario in terms of communication is that in a two-dimensional simulation, parallelization would only occur in one direction that resulted in a split of the body-fitted grid region among two processors, such that a processor needed to receive (up to) half of the grid points from the body-fitted grid.

I am putting myself to the fullest possible use,
which is all I think that any conscious entity can
ever hope to do.

HAL 9000[32]

3 Contributions

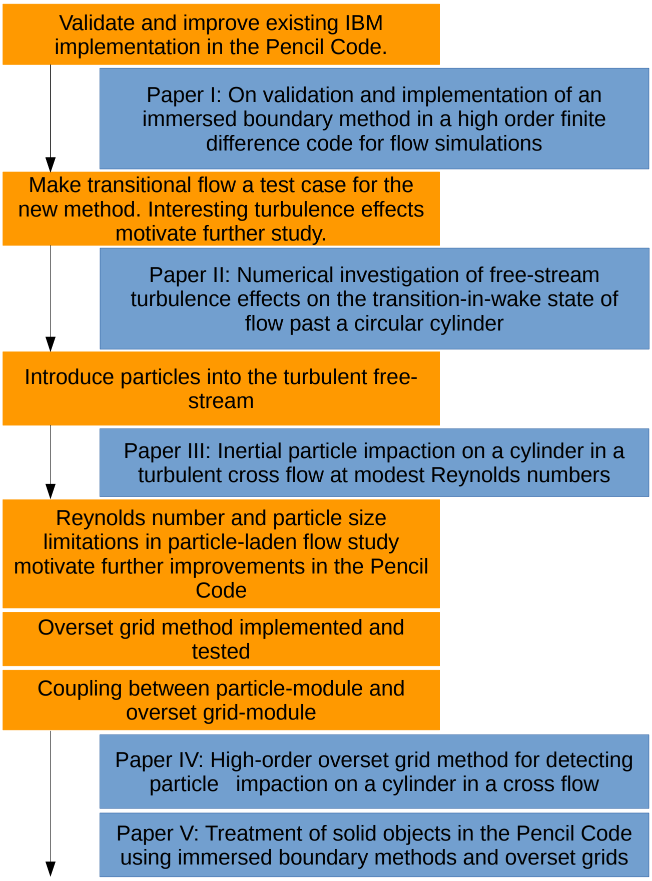
The main contributions to the present thesis consist of five manuscripts, which have been published in or submitted to peer-review journals with scopes appropriate for each of the manuscripts. Of the manuscripts, one has been published in conference proceedings, two have been published in international journals and two are under consideration for publication in international journals. In this chapter context for and summaries of the manuscripts are given.

3.1. Timeline

The five manuscripts cover quite a wide range of topics, all within the field of flow past a solid geometry. The range of topics reflects different phases of the work during the Ph.D. studies. The main motivation has always been to increase understanding of the physics of fluid and particle flows, but accuracy limitations in the simulations forced the alternation between research on physical problems and development of numerical methods. A rough time-line for the research is shown in Figure 3.1.

Roughly speaking, the five manuscript included in Chapter 5 can be categorized as three papers concerning numerical methods (Papers I, IV and V) and two concerning advancements in understanding of physical mechanisms in flows with free-stream turbulence (Papers II and III). Of the papers focusing on numerical

[32] From Stanley Kubrick's *2001: A Space Odyssey*, where the fictional character HAL 9000 is an artificial general intelligence [54].



Validate and improve existing IBM implementation in the Pencil Code.
Paper I: On validation and implementation of an immersed boundary method in a high order finite difference code for flow simulations
Make transitional flow a test case for the new method. Interesting turbulence effects motivate further study.
Paper II: Numerical investigation of free-stream turbulence effects on the transition-in-wake state of flow past a circular cylinder
Introduce particles into the turbulent free-stream
Paper III: Inertial particle impaction on a cylinder in a turbulent cross flow at modest Reynolds numbers
Reynolds number and particle size limitations in particle-laden flow study motivate further improvements in the Pencil Code
Overset grid method implemented and tested
Coupling between particle-module and overset grid-module
Paper IV: High-order overset grid method for detecting particle impaction on a cylinder in a cross flow
Paper V: Treatment of solid objects in the Pencil Code using immersed boundary methods and overset grids

Figure 3.1.: Timeline of Ph.D. work.

methods, Paper I [1] validates an immersed boundary method (IBM) implementation and improvements made to this method, Paper IV [2] introduces the overset grid method for flows past a cylinder, with and without particles in the flow, and Paper V [5] is a comparative study of IBM and overset grids for solid body representation in the Pencil Code. For the manuscripts focusing on physical phenomena, Paper II [4] is a study of a transitional flow under conditions of high-intensity free-stream turbulence, and Paper III [3] consists of direct numerical simulations and statistical analysis of particle impaction in laminar and turbulent flows.

Although updates to the existing immersed boundary method and development of the overset grids in the Pencil Code are documented in the manuscripts, extensive work done during the Ph.D. studies was only indirectly documented within this research. That work is related to continuous improvements of code efficiency, post-processing routines, parallelization, interpolation, etc. To demonstrate the amount of code development completed, consider the Fortran-module of the overset grid method alone (not counting any post-processing or coupling to other modules), which is more than 10^4 lines of code (and growing), and more than 26000 lines of code were submitted to the Git repository of the Pencil Code by the thesis author.

3.2. List of papers

PAPER I

Jørgen R. Aarnes, Nils E. L. Haugen and Helge I. Andersson (2015)

On validation and implementation of an immersed boundary method in a high order finite difference code for flow simulations

Published in MekIt'15: Eighth national conference on computational mechanics; 2015 May 18–19, Trondheim, Norway. Ed. by Skallerud, Bjørn and Andersson, Helge I. Barcelona, Spain: International Center for Numerical Methods in Engineering, pp. 1–21.

The authors' contribution to paper: *Jørgen R. Aarnes implemented the novel handling of ghost-point in the Pencil Code, with support in joint-coding sessions with Nils E. L. Haugen. Nils E. L. Haugen also contributed to the idea of the new method. Jørgen R. Aarnes performed all validation runs and wrote the paper. The process was supervised and the manuscript reviewed by Nils E. L. Haugen and Helge I. Andersson.*

PAPER II

Jørgen R. Aarnes, Helge I. Andersson and Nils E. L. Haugen (2018)

Numerical investigation of free-stream turbulence effects on the transition-in-wake state of flow past a circular cylinder

Published in Journal of Turbulence 19, 252–273.

The authors' contribution to paper: *Jørgen R. Aarnes performed the simulations, evaluated the results and wrote the paper. Helge I. Andersson suggested the topic for study, and both Nils E. L. Haugen and Helge I. Andersson contributed with supervision and suggestions for manuscript revision.*

PAPER III

Jørgen R. Aarnes, Nils E. L. Haugen and Helge I. Andersson (2018)

Inertial particle impaction on a cylinder in a turbulent cross-flow at modest Reynolds numbers

Submitted to journal

The authors' contribution to paper: *Jørgen R. Aarnes performed the simulations, evaluated the results and wrote the paper. Nils E. L. Haugen suggested the research topic and made major contributions to the statistical analysis part of the paper. The process was supervised and the manuscript reviewed by Nils E. L. Haugen and Helge I. Andersson.*

PAPER IV

Jørgen R. Aarnes, Nils E. L. Haugen and Helge I. Andersson (2018)

High-order overset grid method for detecting particle impaction on a cylinder in a cross flow

Submitted to journal, some revisions made after submission.

The authors' contribution to paper: *Jørgen R. Aarnes implemented and performed tests on the overset grid module, evaluated the results and wrote the paper. Code improvements and implementation details was discussed on many occasions with Nils E. L. Haugen. The process was supervised and the manuscript reviewed by Nils E. L. Haugen and Helge I. Andersson.*

PAPER V

Jørgen R. Aarnes, Tai Jin, Chaoli Mao, Nils E. L. Haugen, Kun Luo and Helge I. Andersson (2018)

Treatment of solid objects in the Pencil Code using an immersed boundary method and overset grids

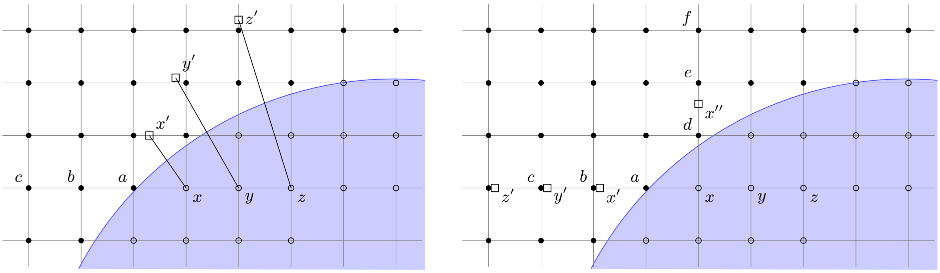
Published in Geophysical & Astrophysical Fluid Dynamics. Forthcoming in special issue titled On the Physics and Algorithms of the Pencil Code.

The authors' contribution to paper: *Jørgen R. Aarnes performed the tests and evaluated the results for the circular geometry part of the paper. Jørgen R. Aarnes and Tai Jin co-wrote the complex geometry section, for which tests were performed by Chaoli Mao and analyzed by Tai Jin. The process was supervised by Nils*

E. L. Haugen. Nils E. L. Haugen, Helge I. Andersson and Kun Luo reviewed the manuscript.

3.3. Summary of papers

PAPER I



(a) Orthogonal mirror-point method

(b) Cartesian mirror-point method

Figure 3.2.: The two IBM implementations. Ghost-points (○) set by corresponding mirror-points (□) along surface normal (a) or along grid lines in x- and y-direction (b) Mirror-points interpolated from surrounding fluid-points (●).

Solid boundary representation of a cylinder in a cross flow by the immersed boundary method (IBM) was validated by parametric studies of relevant physical and numerical properties that influence the simulations. The three non-dimensional flow variables: mean drag coefficient, root-mean-square lift coefficient and Strouhal number were computed from the simulation results, and compared for varying grid spacing, domain size (upstream length, downstream length, width), Mach number and Courant number. The parameter study yielded expected results, with drag, lift and Strouhal numbers approaching asymptotic values that correlate with the parameter variation. An exception to this was the mean drag coefficient computed for varying upstream and downstream length. The unexpected behavior was attributed to blockage effects. The parametric studies were concluded with a comparison to a large set of previously published data, showing good agreement with the present study.

In the grid refinement study, it was found that 64 grid points per cylinder diameter was necessary to resolve the flow with reasonable accuracy when $Re = 100$. To counter this strict grid requirement a new IBM method was suggested (see Figure

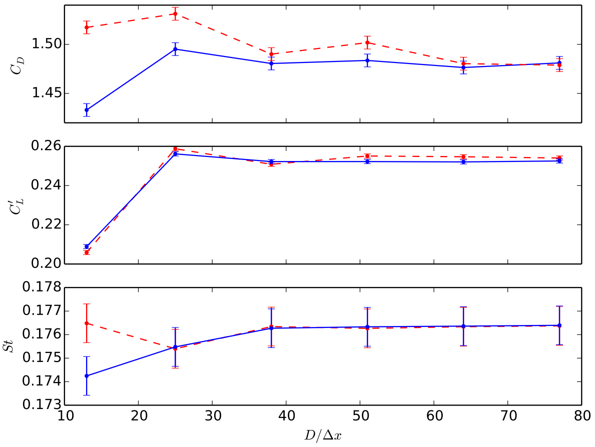


Figure 3.3.: Mean drag coefficient (C_D), root-mean-square lift coefficient (C_L') and Strouhal number (St) for different grid spacing and IBM: The Cartesian mirror point method (solid blue) and the orthogonal mirror point method (dashed red). Error bars indicate 1% deviation.

3.2). The previous IBM implementation in the Pencil Code identified mirror-points along orthogonal lines from the ghost-points through the solid surface. The new method instead used mirror-points along horizontal and vertical grid lines. The methods were contrasted as orthogonal and Cartesian mirror-point implementations of the immersed boundary method. With the new method, a grid spacing of 38 grid points per cylinder diameter was found to be sufficiently fine (see Figure 3.3. This is a significant improvement from the previous IBM implementation.

PAPER II

The transition-in-wake state of flow past a cylinder, during which the first three-dimensional aspects of the flow develop, was studied by direct numerical simulations (DNS). The effect of a turbulent free-stream (TFS) on the transition was investigated by inserting homogeneous isotropic turbulence at the inlet of the flow domain. The intensity of the decaying TFS varied from 3.6% to 12.2% (as it reached the cylinder), with integral scales 0.5, 0.75 and 1.0 times the cylinder diameter for the high intensity case. The results were compared with a laminar free-stream (LFS) case.

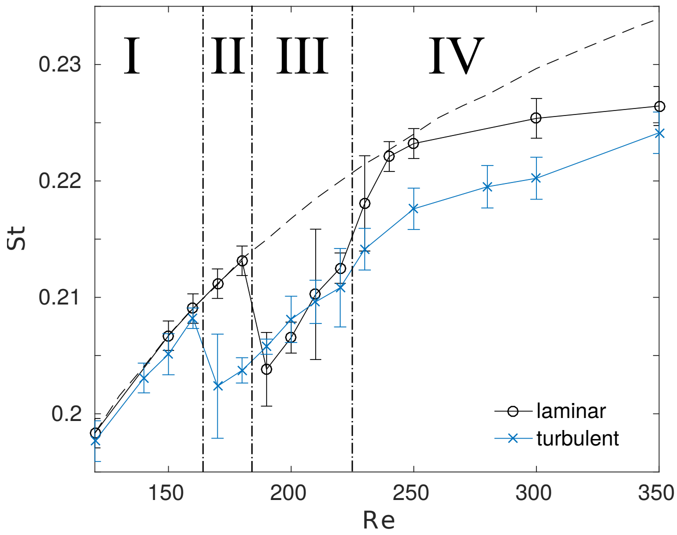


Figure 3.4.: Relationship between Strouhal number (*St*) and Reynolds number (*Re*) over the transitional domain (b) TFS flow compared to LFS case. -o- LFS; -x- TFS; - - 2D LFS.

Effects of the TFS were seen in two Reynolds number ranges (see Figure 3.4). At the lower end of the transition ($Re < 190$) the TFS perturbed the flow, which forced the onset of the transition at a lower Reynolds number than the corresponding LFS case. With a laminar free-stream, there is a sharply defined critical Reynolds number where the first three-dimensional effects in the wake were found at $Re \approx 190$. With a TFS, the critical Reynolds number was not sharply defined. This is not surprising, as this transition is subcritical, hence, flow disturbances may initiate the transition earlier. The shift due to the TFS was, however, to a lower *Re* than allowed by the hysteresis pattern of the subcritical instability seen in experiments with an LFS. Turbulence with different integral scales and intensities revealed further complexities at the onset of the transition. Quasi-stable states, in which the wake alternates between two- and three-dimensional vortex shedding, were observed. Such states are closely related to large scale vortex dislocations that develop in the wake. For a quasi-stable shedding state, the vortex dislocation breaks up before the wake reaches a saturated state of mode *A* instabilities.

The TFS also affected the upper part of the transitional regime ($Re \geq 250$). Mode *A* instabilities were simulated in the wake, such that mixed *A*–*B* instabilities were observed for $Re \geq 250$. This is in contrast to the LFS flow, where mode *A* instabilities die out in the upper part of the transitional regime, and pure mode *B*

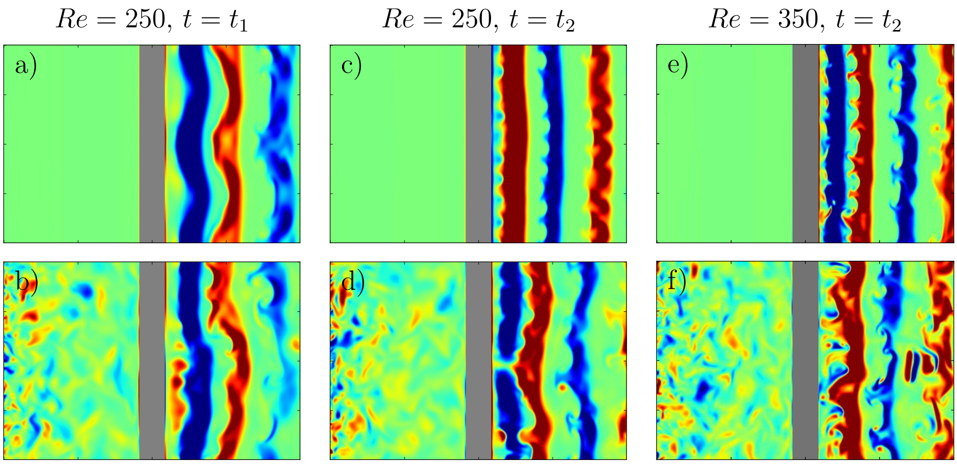


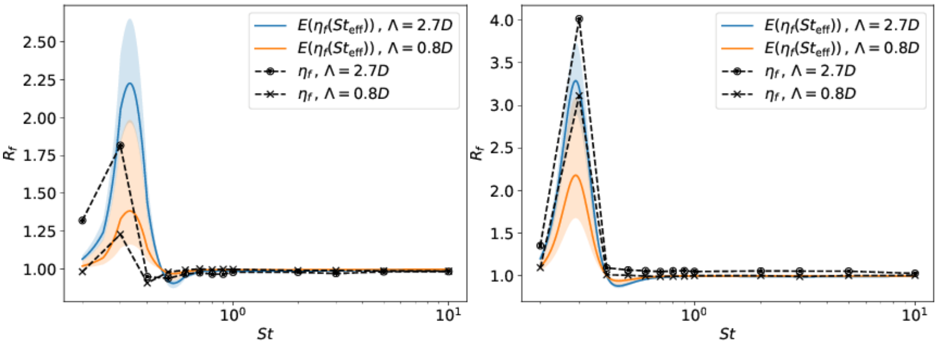
Figure 3.5.: Contours of instantaneous vorticity $\omega_z = [\boldsymbol{\nabla} \times \boldsymbol{u}]_z$ in a plane along the streamwise direction, through the cylinder axis, plotted for $Re = 250$ (left and middle) and $Re = 350$ (right) for flow simulations with LFS (top) and TFS (bottom). Contour plots for two time instants are given for $Re = 250$: t_1, early development of instabilities; t_2, asymptotic state ($t_2 \gg t_1$). Only the asymptotic state is shown for $Re = 350$.

instability states are seen in the wake (see Figure 3.5). The effects of stimulated existence of mode *A* instabilities in the wake (inhibiting a pure mode *B* wake) correlated with the turbulence intensity, and was strongest for the highest intensity TFS.

In accordance with the studies by Bloor [14] and Norberg [66], the effect of the turbulent free-stream on the transition-in-wake was an increased bandwidth of Reynolds numbers spanning the transition-in-wake. More precisely, the bandwidth of Reynolds numbers where mode *A* instabilities are sustained in the wake increased. Disturbances in form of turbulent fluctuations are therefore an additional source of scatter in measured critical Reynolds numbers, previously attributed to contamination from end conditions [118].

PAPER III

The effects of turbulence in the free-stream on front-side particle impaction on a cylinder were studied with direct numerical simulations and statistical analysis. A large number of inertial point particles ($\sim 10^7$) were inserted at random positions over the inlet of the flow simulations. The particles were distributed over Stokes



(a) $Re = 100$

(b) $Re = 400$

Figure 3.6.: Predicted and observed results for amplification factor (R_f) of impaction efficiencies resulting from turbulence of different integral scales interacting with particles in the flow.

numbers $0.2 \leq St \leq 10$, and the simulations were performed for moderate Reynolds numbers ($Re = 100$ and $Re = 400$). For each Reynolds number, three cases of particle impaction were simulated: LFS, TFS with integral scale $\Lambda = 0.8D$, and TFS with $\Lambda = 2.7D$. In addition to the simulations, expectation values of particle impaction affected by TFS were derived:

$$E(\eta(St_{\text{eff}})) \approx \eta(St) + \frac{\eta''(St)}{2}\sigma_{St}^2\,, \tag{3.1}$$

where St_{eff} is the effective Stokes number, $\eta''(St)$ is the second derivative of the impaction efficiency for laminar conditions and σ_{St}^2 is the variance of the effective Stokes number. With a turbulent free-stream St_{eff} is a stochastic value, $St_{\text{eff}} = (St \cdot U)/U_0$, where U is the flow velocity and U_0 is the velocity of the mean flow.

The effects of the TFS were large for particles with $St = 0.3$, in particular at $Re = 400$, where 3–4 times as many particles hit the cylinder (depending on the integral scale of the turbulence). For smaller ($St = 0.2$) and larger particles ($St \geq 0.4$) only a small effect relative increase in particle impaction was seen. Previously published results by Weber et al. [111] showed a cut-off Stokes number for particles that were affected by the TFS. A similar upper cut-off was reproduced, but using such a cut-off to identify which particles are affected by free-stream turbulence was regarded as inappropriate, when boundary interception was included as an impaction mechanism. There appears to be a peak Stokes number where the maximum effect of the turbulence on particle impaction exists, and a small region of St around this maximum quite large amplification of impaction occurs. This Stokes number is in the St-region where the dominant impaction mechanism on the cylinder changes from boundary interception to boundary stopping. Including impaction by boundary interception yields such a peak at $St_{\text{peak}} > St_{\text{crit}}$.

The effect of turbulence on particle impaction depended on the Reynolds number of the flow, and the integral scale and intensity of the turbulence. The turbulence intensity (T_i) decayed at a much faster rate for smaller integral scales and Reynolds numbers. Higher Reynolds number and larger integral scale (i.e., higher turbulence intensity) yields a larger amplification of impaction.

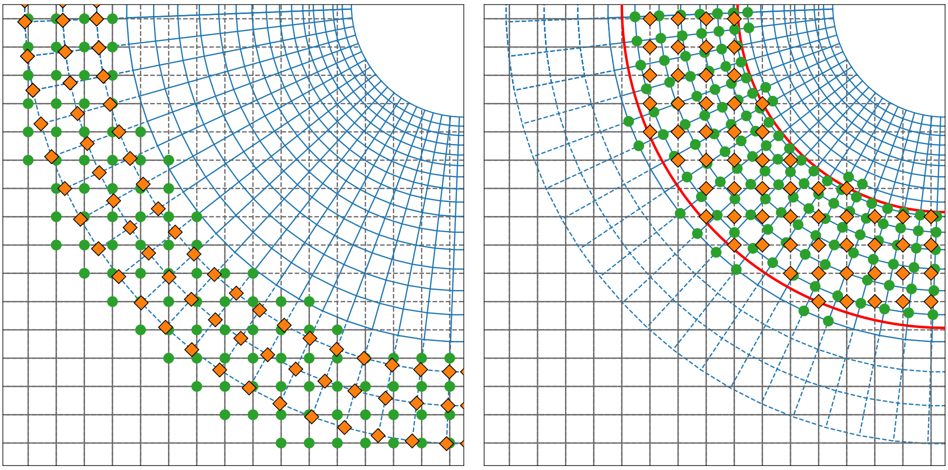
With $\eta''(St)$ approximated from the LFS simulations and σ_{St}^2 computed from decaying turbulence in a cylinder-free domain, expectation values of front-side particle impaction were computed and plotted together with the observations from the direct numerical simulations (coloured fields in Figure 3.6). The observed effect of the TFS on a limited range of Stokes numbers was reproduced by the predictions using Eq. (3.1). The fit was very good with regards to which Stokes numbers were affected by the turbulence, and it is indicative of large amplifications of the impaction efficiencies. The predictive power of Eq. (3.1) is good, and it is expected that more accurate approximations of $\eta''(St)$ and σ_{St}^2 will only make the predictions better. Such an expression can be useful for predictions of particle impaction in flows with free-stream turbulence, requiring only data from an LFS flow and the intensity of the decaying turbulence.

PAPER IV

The recent implementation of an overset grid method (see Figure 3.7) in the Pencil Code was described, and tests were performed to determine the formal order of accuracy and performance of the method for practical flow simulations. Particle impaction on a cylinder in a cross flow motivated the development of this method, and as such, this was the sample application of overset grids used within a complex flow problem. Unique features of the overset grid implementation were documented. These include local time-step restrictions, summation-by-parts boundary conditions and application to particle impactions.

The formal order of accuracy, P, was determined, by computing L_2-error norms from steady flow simulations ($Re = 20$) on grids with spacing refined stepwise by a factor $1/2$. Two methods of interpolation were compared, and both yielded $P \approx 2.5$ for all flow variables in most of the domain covered by the body-fitted cylindrical grid. The order of accuracy was highest in the immediate vicinity of the of the solid surface ($P \approx 5$ for the radial velocity component).

For unsteady flow at $Re = 100$, the simplest interpolation between grids (bi-linear



(a)

(b)

Figure 3.7.: Communication from background Cartesian grid to overset cylindrical grid (a) and back (b): Fringe-points (◆) receive data from surrounding donor-points (●) by interpolation. Dashed lines used to identify sections of the grids where variables are not computed by finite-differences (fringe-points and hole-points).

Lagrangian interpolation) performed better than the complex method (bi-quadratic spline interpolation). Grid refinement runs showed very rapid convergence towards grid independent solutions for central flow coefficients (drag, lift and Strouhal number). Differently sized overset grids were tested, and all showed good agreement with previously published data for the computed coefficients.

When applied to particle-laden flows, simulations reproduced previously published results over the entire range of Stokes numbers tested ($0.01 \leq St \leq 10$). The simulations using overset grids used much coarser meshes for the particle simulations than used in previous studies. Direct comparison for equally sized flow domains yielded a 90% reduction in the number of grid points required for two-dimensional simulations. The dramatic reduction in computational cost was exploited to efficiently investigate the effect of domain sizes and grid resolution on particle impaction. Increasing the domain size, from $L_x \times L_y = 6D \times 12D$ to $10D \times 20D$ (with mean flow along the y-direction) revealed considerable blockage effects on the smaller domain (see Figure 3.8). The effect of a larger flow domain (smaller blockage effect) significantly reduced particle impaction for the smallest particles. Together with grid refinement, the new result showed an over-prediction in published results by Haugen and Kragset [39] of a factor of 6.3 for the small-

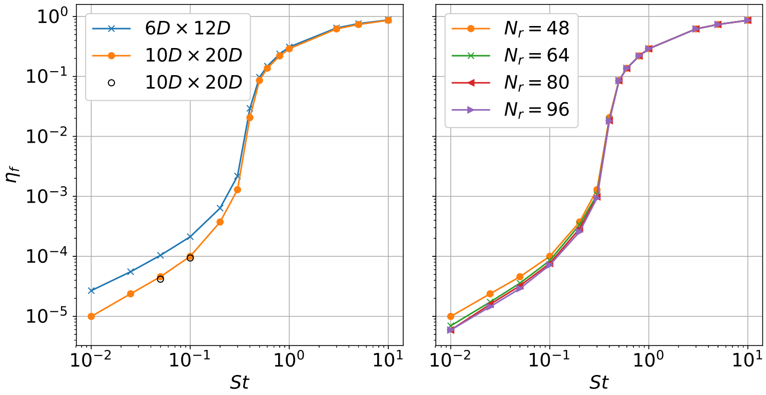


Figure 3.8.: Front side impaction efficiency (η_f) as a function of Stokes number (St) at Reynolds number 100 for different domain sizes (left) and grid resolutions (right).

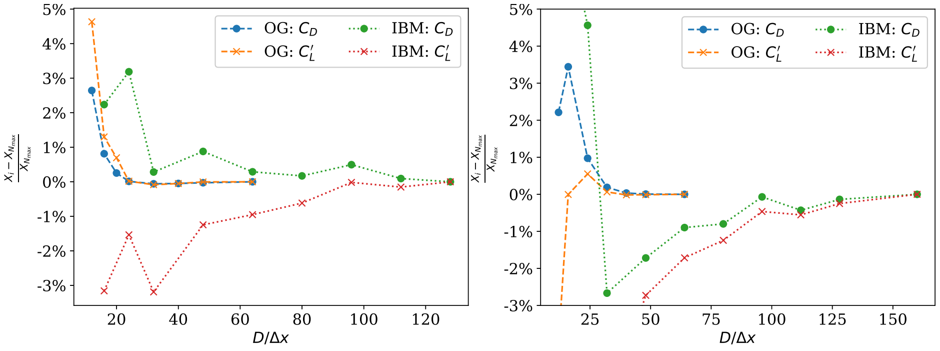
est Stokes number ($St = 0.01$). The over-prediction was most prominent for the smallest particles, and was reduced to a factor 2.8 for $St = 0.1$.

PAPER V

Two methods of solid body representation implemented in the Pencil Code, were compared for steady and unsteady flows past a solid object. The orthogonal mirror-point IBM and the newly implemented overset grid method were considered.

Grid refinement simulations for Reynolds numbers, $Re = 20$ (steady), $Re = 100$ (unsteady) and 400 (unsteady) revealed a large difference between resolutions necessary to reach grid independence. With overset grids, less than one fourth the number of grid points were necessary in each direction to reach grid independence, as compared to when IBM was used (see Figure 3.9). Because the time-step is proportional to the grid spacing, significantly larger time-steps could be used with the overset grid method. This was partially due to the local time-step restriction on the overset grids. With a sufficiently fine grid, both the IBM and overset grids could be used to compute flow coefficients (drag, lift and Strouhal number) that agreed well with previously published data.

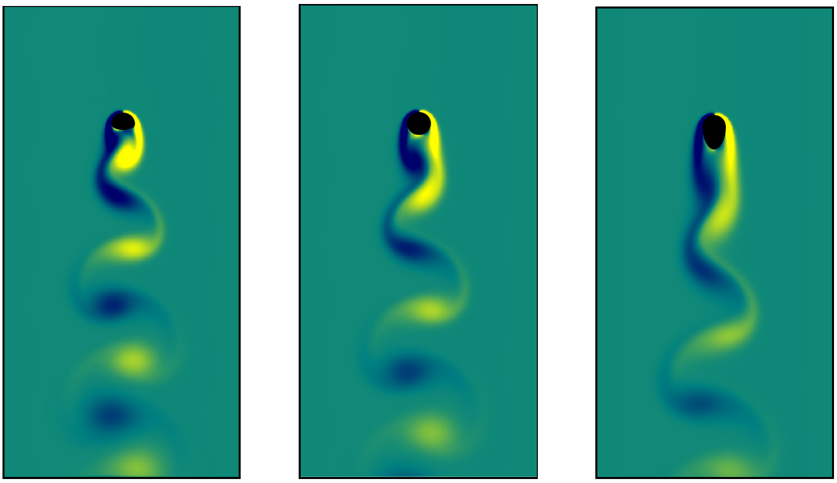
The advantages of IBM over overset grids were considered. Although much less efficient than overset grids, IBM is useful due to its straightforward implementation into an existing flow solver and the flexibility of the method. The latter trait was demonstrated, by resolving a non-circular boundary with IBM. A semi-circular/semi-elliptical cylinder was used as a test geometry, for which the ratio of



(a) $Re = 100$

(b) $Re = 400$

Figure 3.9.: Normalized values for mean drag coefficient (C_D) for flow with $Re =$ 100, mean drag coefficient and root-mean-square lift coefficient (C_L') for flow with $Re = 400$. Results are computed for grids with varying coarseness. The cylinder in the flow is represented either by the immersed boundary method (IBM) or with overset grids (OG).



(a) $R/b = 2.0$

(b) $R/b = 1.0$

(c) $R/b = 0.5$

Figure 3.10.: Flow visualization. Contours of instantaneous vorticity normal to the view plane plotted for three different geometries at $Re = 100$. Inflow at the top of plane.

circular radii R to elliptical axis b was varied (see Figure 3.10).

3.4. Other contributions

In addition to the included manuscripts and the developments in the Pencil Code, the author played an active role in the Pencil Code community and participated in a range of scientific conferences during the Ph.D. studies. This has also contributed towards the present thesis, and relevant works are therefore listed here.

Conference presentations

Improving the implementation of solid objects in a high order finite difference code for direct numerical simulations of turbulent flow

Presented at MekIT'15 – The 8th National Conference in Computational Mechanics (18–19 May, 2015), Trondheim, Norway.

The effect of turbulence on particle impaction on a cylinder in cross flow

Presented at 9th International Conference on Multiphase Flow (22–27 May, 2016), Florence, Italy.

The effect of free stream turbulence on the transition in wake

Presented at 11th European Fluid Mechanics Conference, (12–16 September, 2016), Seville, Spain.

Resolving bluff bodies in DNS using overset grids

Presented at MekIT'17 - 9th National Conference on Computational Mechanics (11–12 May, 2017), Trondheim, Norway.

Free-stream turbulence effects on the transition-in-wake state of the flow past a circular cylinder

Presented at 16th European Turbulence Conference (21–24 August, 2017), Stocholm, Sweden.

Particle-laden flow past a cylinder resolved with IBM and overset grids

Presented at 70th Annual Meeting of the American Physical Society Division of Fluid Dynamics (19–21 November, 2017), Denver, USA.

Pencil Code community

Participation without presentation at the 10th Pencil Code Meeting (7-11 July, 2014), Göttingen, Germany.

Validation and immersed boundary method improvements for fluid flow simulations

Presented at 11th Pencil Code Meeting (11–14 May, 2015), Trondheim, Norway.

Free-stream turbulence effects on the onset of the transition in the wake of a circular cylinder

Presented at 12th Pencil Code User Meeting (8–12 August, 2016), Graz, Austria.

Resolving bluff bodies in the Pencil Code using overset grids

Presented at 13th Pencil Code User Meeting (10–14 July, 2017), Newcastle, UK.

Whenever a theory appears to you as the only possible one, take this as a sign that you have neither understood the theory nor the problem which it was intended to solve.

Karl R. Popper[33]

4 Conclusion

In this chapter, concluding remarks on the different research topics are given. Future outlook with suggestions for further research and recommended extensions to the work presented in this thesis are also included.

4.1. Concluding remarks

A computational study of a cylinder in a cross flow has been presented. The focus of the study has been on both physical phenomena and numerical problems related to such a flow case.

The physical problems considered are the transition-in-wake state of the flow and particle impaction on the cylinder. For both of these flow problems, the novel contribution has been the investigations of how free-stream turbulence affects the different mechanisms at play, investigated by DNS with homogeneous isotropic turbulence inserted at the flow inlet for moderate Reynolds numbers. Under conditions of a medium to high-intensity turbulent free-stream the range of Reynolds number spanning the transition-in-wake state of the flow is increased. The disturbances in the flow trigger transition at a lower Reynolds number and that transition endures to a higher Reynolds number. Relevant mechanisms for this bandwidth increase are stimulation of mode *A* instabilities, intermittent vortex dislocations, quasi-stable shedding states and mixed *A*–*B* instabilities. For inertial particle impaction on a cylinder, the turbulence amplifies the particle impaction for Stokes numbers in the range where the dominating impaction mechanism changes from boundary

[33] In *Objective Knowledge: An Evolutionary Approach* [76].

stopping to boundary interception. The relative amplification of impaction is small for particles outside this Stokes number range.

The numerical problems considered were related to the accurate resolution of fluid and particle flow around a solid. The accurate resolution of the boundary layer around a solid cylinder was a crucial aspect of this research, and both immersed boundary methods and overset grid methods were employed to achieve sufficiently accurate results at reasonable computational costs. Efforts to improve the numerical representation of a solid cylinder by an IBM yielded a significant reduction in the necessary grid resolution. The particle-laden flow simulations were still limited to particles with a Stokes number larger than or equal to 0.2 for accurate representation of particle trajectories in the vicinity of the cylinder. For smaller particles, significantly finer grids would be necessary. This motivated the implementation of overset grids. The Pencil Code implementation of overset grids has shown great promise. Not only is the order of accuracy of the method higher than that of the IBM implementation, but a much more flexible grid stretching scheme and local time-restrictions provide large cost reductions in highly accurate flow simulations. The result is a method with outstanding performance for the simulation of a cylinder in a cross flow, both with and without particles in the flow.

4.2. Future outlook

During the research performed in this doctoral work, many new research problems were encountered. Some are direct extensions to the work performed, while others are new possibilities or ideas that have arisen during the research.

The immediate extension to the investigation of transitional flow is to perform simulations with lower intensity TFS. It is plausible that there is a cut-off for the turbulence intensity at which the onset of transition is sensitive to flow disturbances. Identifying a cut-off intensity can have applications to determining an acceptable level of disturbances in experimental studies of transitional flow states. A focused investigation into the mechanism of vortex dislocations in the wake is another topic for future research. The dislocations are attributed to self-excited mode A instabilities by Henderson [42], yet still occur in quasi-stable states where the wake does not settle into a mode A instability when the free-stream is turbulent. Does this mean that the self-excited A modes exist without other visible flow effects other than the intermittent vortex dislocations, when high-intensity disturbances force the transition?

For particle impaction under conditions of free-stream turbulence, an immediate extension is to perform simulations for many more Stokes numbers, in particular in the region where impaction is affected by the turbulence. In this way, the peak Stokes number for amplification can be more accurately identified, which can be applied in practical applications when impaction should be avoided (like in industrial boilers). Further studies should also aim to include a larger range of Stokes numbers and turbulent intensities. When the Stokes number range is increased (to include smaller particles), mechanisms like Brownian motion and thermophoresis should also be accounted for. Perhaps even more interesting is to increase the Reynolds numbers of such flows, such that simulations are performed under conditions that are more representative of real world applications. A combination of this and accounting for more than just inertial impactions should be the aim of future DNS investigations of cylinders in particle-laden cross flows. For higher Reynolds numbers, the backside impaction also becomes an interesting aspect of particle deposition.

These research topics, and many more, can be studied with the novel numerical methods implemented in the Pencil Code during these Ph.D. studies. The code is open-source and freely available for anyone to download from GitHub. At present, recent extensions to the overset grid module include handling of non-isothermal flows (energy equation included in the solver) and computation of thermophortic forces on particles in the flow.

The intent of this thesis is to provide useful information and inspiration to researchers in the field of fluid dynamics, who have the time and patience to investigate fundamental flow problems by direct numerical simulations. Together with advances made by researchers in experimental fluid dynamics and efforts by engineers to make use of fundamental research in the design of real-world applications, such computational studies are a source of continuous progress in the understanding of the physics of fluid and particle flows.

Let no one ignorant of geometry enter here.

Inscription, Plato's Acadamy[34]

A Navier-Stokes equations

The compressible Navier-Stokes equations are rarely written out in full. Although the short-hand notation is quite elegant, the Navier-Stokes equations written in an expanded format are useful when developing or understanding procedures in a computational fluid dynamics software. The expressions are given here. To understand how computations are performed in The Pencil Code, the Navier-Stokes equations: (Eqs. (2.6) and (2.7)) formulated term-by-term are most useful, as this is how the equations are solved by this particular software. Both the full form and the term-by-term form are given here. The expressions are based on and checked against material found in Cantwell [19], Griffiths [36], Bird et al. [13], and Quartapelle [80].

A.1. Cartesian coordinates

It is straightforward to write the Navier-Stokes equations in Cartesian coordinates. The full form of the continuity equation (Eq. (2.1)) is:

$$\frac{\partial \rho}{\partial t} + u_x \frac{\partial \rho}{\partial x} + u_y \frac{\partial \rho}{\partial y} + u_z \frac{\partial \rho}{\partial z} = -\rho \left(\frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} + \frac{\partial u_z}{\partial z} \right) . \tag{A.1}$$

The momentum equation, Eq. (2.2), in Cartesian coordinates is given by the set of equations:

[34] Tradition has it that the statement was engraved over the door of Plato's Academy [30].

$$
\begin{aligned}
\rho\left(\frac{\partial u_x}{\partial t}+u_x\frac{\partial u_x}{\partial x}+u_y\frac{\partial u_x}{\partial y}+u_z\frac{\partial u_x}{\partial z}\right) &= -\frac{\partial p}{\partial x}+\frac{\partial}{\partial x}2\mu\left(\frac{\partial u_x}{\partial x}-\frac{1}{3}\boldsymbol{\nabla}\cdot\boldsymbol{u}\right)\\
&\quad+\frac{\partial}{\partial y}\mu\left(\frac{\partial u_y}{\partial x}+\frac{\partial u_x}{\partial y}\right)+\frac{\partial}{\partial z}\mu\left(\frac{\partial u_z}{\partial x}+\frac{\partial u_x}{\partial z}\right)+F_x\,,
\end{aligned}
\tag{A.2}
$$

$$
\begin{aligned}
\rho\left(\frac{\partial u_y}{\partial t}+u_x\frac{\partial u_y}{\partial x}+u_y\frac{\partial u_y}{\partial y}+u_z\frac{\partial u_y}{\partial z}\right) &= -\frac{\partial p}{\partial y}+\frac{\partial}{\partial x}\mu\left(\frac{\partial u_x}{\partial y}+\frac{\partial u_y}{\partial x}\right)\\
&\quad+\frac{\partial}{\partial y}2\mu\left(\frac{\partial u_y}{\partial y}-\frac{1}{3}\boldsymbol{\nabla}\cdot\boldsymbol{u}\right)+\frac{\partial}{\partial z}\mu\left(\frac{\partial u_z}{\partial y}+\frac{\partial u_y}{\partial z}\right)+F_y\,,
\end{aligned}
\tag{A.3}
$$

$$
\begin{aligned}
\rho\left(\frac{\partial u_z}{\partial t}+u_x\frac{\partial u_z}{\partial x}+u_y\frac{\partial u_z}{\partial y}+u_z\frac{\partial u_z}{\partial z}\right) &= -\frac{\partial p}{\partial z}+\frac{\partial}{\partial x}\mu\left(\frac{\partial u_x}{\partial z}+\frac{\partial u_z}{\partial x}\right)\\
&\quad+\frac{\partial}{\partial z}\mu\left(\frac{\partial u_y}{\partial z}+\frac{\partial u_z}{\partial y}\right)+\frac{\partial}{\partial z}2\mu\left(\frac{\partial u_z}{\partial z}-\frac{1}{3}\boldsymbol{\nabla}\cdot\boldsymbol{u}\right)+F_z\,,
\end{aligned}
\tag{A.4}
$$

where the divergence $\left(\boldsymbol{\nabla}\cdot\boldsymbol{u}=\frac{\partial u_x}{\partial x}+\frac{\partial u_y}{\partial y}+\frac{\partial u_z}{\partial z}\right)$ is written in compact form to keep the equations from spanning too many lines.

Writing out the right-hand sides of Eqs. (2.6) and (2.7) is trivial in Cartesian coordinates. These are included here for completeness. The terms on the right hand side of the continuity equation are:

$$
\begin{aligned}
(\boldsymbol{u}\cdot\boldsymbol{\nabla})\,\rho &= u_x\frac{\partial\rho}{\partial x}+u_y\frac{\partial\rho}{\partial y}+u_z\frac{\partial\rho}{\partial z}\,,\\
\rho\boldsymbol{\nabla}\cdot\boldsymbol{u} &= \rho\left(\frac{\partial u}{\partial x}+\frac{\partial u}{\partial y}+\frac{\partial u}{\partial z}\right)\,.
\end{aligned}
\tag{A.5}
$$

The terms on the right hand side of Eq. (2.7) are split into components in in x, y and z-direction. This yields:

$$
\begin{aligned}
\left[(\boldsymbol{u}\cdot\boldsymbol{\nabla})\,\boldsymbol{u}\right]_x &= u_x\frac{\partial u_x}{\partial x}+u_y\frac{\partial u_x}{\partial y}+u_z\frac{\partial u_x}{\partial z}\,,\\
\left[(\boldsymbol{u}\cdot\boldsymbol{\nabla})\,\boldsymbol{u}\right]_y &= u_x\frac{\partial u_y}{\partial x}+u_y\frac{\partial u_y}{\partial y}+u_z\frac{\partial u_y}{\partial z}\,,\\
\left[(\boldsymbol{u}\cdot\boldsymbol{\nabla})\,\boldsymbol{u}\right]_z &= u_x\frac{\partial u_z}{\partial x}+u_y\frac{\partial u_z}{\partial y}+u_z\frac{\partial u_z}{\partial z}\,,
\end{aligned}
\tag{A.6}
$$

$$
\begin{aligned}
\left[c_s^2\boldsymbol{\nabla}\left(\ln\rho\right)\right]_x &= c_s^2\frac{1}{\rho}\frac{\partial\rho}{\partial x}\,,\\
\left[c_s^2\boldsymbol{\nabla}\left(\ln\rho\right)\right]_y &= c_s^2\frac{1}{\rho}\frac{\partial\rho}{\partial y}\,,
\end{aligned}
\tag{A.7}
$$

$$\left[c_s^2 \boldsymbol{\nabla}(\ln\rho)\right]_z = c_s^2 \frac{1}{\rho}\frac{\partial\rho}{\partial z},$$

$$\begin{aligned}
\left[\boldsymbol{\nabla}^2 \boldsymbol{u}\right]_x &= \frac{\partial^2 u_x}{\partial x^2} + \frac{\partial^2 u_x}{\partial y^2} + \frac{\partial^2 u_x}{\partial z^2}, \\
\left[\boldsymbol{\nabla}^2 \boldsymbol{u}\right]_y &= \frac{\partial^2 u_y}{\partial x^2} + \frac{\partial^2 u_y}{\partial y^2} + \frac{\partial^2 u_y}{\partial z^2}, \\
\left[\boldsymbol{\nabla}^2 \boldsymbol{u}\right]_z &= \frac{\partial^2 u_z}{\partial x^2} + \frac{\partial^2 u_z}{\partial y^2} + \frac{\partial^2 u_z}{\partial z^2},
\end{aligned} \tag{A.8}$$

$$\begin{aligned}
\left[\boldsymbol{\nabla}(\boldsymbol{\nabla}\cdot\boldsymbol{u})\right]_x &= \frac{\partial^2 u_x}{\partial x^2} + \frac{\partial^2 u_y}{\partial x \partial y} + \frac{\partial^2 u_z}{\partial x \partial z}, \\
\left[\boldsymbol{\nabla}(\boldsymbol{\nabla}\cdot\boldsymbol{u})\right]_y &= \frac{\partial^2 u_x}{\partial x \partial y} + \frac{\partial^2 u_y}{\partial y^2} + \frac{\partial^2 u_z}{\partial y \partial z}, \\
\left[\boldsymbol{\nabla}(\boldsymbol{\nabla}\cdot\boldsymbol{u})\right]_z &= \frac{\partial^2 u_x}{\partial x \partial z} + \frac{\partial^2 u_y}{\partial x \partial z} + \frac{\partial^2 u_z}{\partial z^2},
\end{aligned} \tag{A.9}$$

$$\begin{aligned}
\left[\boldsymbol{S}\cdot\boldsymbol{\nabla}(\ln\rho)\right]_x &= \left[\frac{\partial u_x}{\partial x} - \frac{1}{3}\left(\frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} + \frac{\partial u_z}{\partial z}\right)\right]\frac{1}{\rho}\frac{\partial\rho}{\partial x} \\
&\quad + \frac{1}{2}\left(\frac{\partial u_y}{\partial x} + \frac{\partial u_x}{\partial y}\right)\frac{1}{\rho}\frac{\partial\rho}{\partial y} + \frac{1}{2}\left(\frac{\partial u_z}{\partial x} + \frac{\partial u_x}{\partial z}\right)\frac{1}{\rho}\frac{\partial\rho}{\partial z}, \\
\left[\boldsymbol{S}\cdot\boldsymbol{\nabla}(\ln\rho)\right]_y &= \left[\frac{\partial u_y}{\partial x} - \frac{1}{3}\left(\frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} + \frac{\partial u_z}{\partial z}\right)\right]\frac{1}{\rho}\frac{\partial\rho}{\partial y} \\
&\quad + \frac{1}{2}\left(\frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x}\right)\frac{1}{\rho}\frac{\partial\rho}{\partial x} + \frac{1}{2}\left(\frac{\partial u_z}{\partial y} + \frac{\partial u_y}{\partial z}\right)\frac{1}{\rho}\frac{\partial\rho}{\partial z}, \\
\left[\boldsymbol{S}\cdot\boldsymbol{\nabla}(\ln\rho)\right]_z &= \left[\frac{\partial u_z}{\partial z} - \frac{1}{3}\left(\frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} + \frac{\partial u_z}{\partial z}\right)\right]\frac{1}{\rho}\frac{\partial\rho}{\partial z} \\
&\quad + \frac{1}{2}\left(\frac{\partial u_x}{\partial z} + \frac{\partial u_z}{\partial x}\right)\frac{1}{\rho}\frac{\partial\rho}{\partial x} + \frac{1}{2}\left(\frac{\partial u_y}{\partial z} + \frac{\partial u_z}{\partial y}\right)\frac{1}{\rho}\frac{\partial\rho}{\partial y}.
\end{aligned} \tag{A.10}$$

A.2. Cylindrical coordinates

The momentum equations in Cartesian coordinates have an identical structure in all three directions. This is not the case in cylindrical (polar) coordinates, where the equations become notably more complicated.

In cylindrical coordinates, the Navier-Stokes equations can be written out in the r, θ and z-directions by using the cylindrical coordinate form of the gradient, divergence, curl, Laplacian, vector Laplacian and advective operators for a scalar and vector field:

$$
\begin{aligned}
\boldsymbol{\nabla} f &= \frac{\partial f}{\partial r}\hat{\boldsymbol{r}} + \frac{1}{r}\frac{\partial f}{\partial \theta}\hat{\boldsymbol{\theta}} + \frac{\partial f}{\partial z}\hat{\boldsymbol{z}}\,,\\
\boldsymbol{\nabla}\cdot\boldsymbol{F} &= \frac{1}{r}\frac{\partial (rF_r)}{\partial r} + \frac{1}{r}\frac{\partial F_\theta}{\partial \theta} + \frac{\partial F_z}{\partial z}\,,\\
\boldsymbol{\nabla}\times\boldsymbol{F} &= \left(\frac{1}{r}\frac{\partial F_z}{\partial \theta} - \frac{F_\theta}{z}\right)\hat{\boldsymbol{r}} + \left(\frac{\partial F_r}{\partial z} - \frac{F_z}{r}\right)\hat{\boldsymbol{\theta}} + \frac{1}{r}\left(\frac{\partial (rF_\theta)}{\partial r} - \frac{F_r}{\theta}\right)\hat{\boldsymbol{z}}\,,\\
\boldsymbol{\nabla}^2 f &= \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial f}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2 f}{\partial \theta^2} + \frac{\partial^2 f}{\partial z^2}\,,\\
\boldsymbol{\nabla}^2 \boldsymbol{F} &= \left(\boldsymbol{\nabla}^2 F_r - \frac{1}{r^2}F_r - \frac{2}{r^2}\frac{\partial F_\theta}{\partial \theta}\right)\hat{\boldsymbol{r}} + \left(\boldsymbol{\nabla}^2 F_\theta - \frac{1}{r^2}F_\theta\right.\\
&\qquad \left. + \frac{2}{r^2}\frac{\partial F_r}{\partial \theta}\right)\hat{\boldsymbol{\theta}} + \left(\boldsymbol{\nabla}^2 F_z\right)\hat{\boldsymbol{z}}\,,\\
(\boldsymbol{G}\cdot\boldsymbol{\nabla}) f &= G_r\frac{\partial f}{\partial r} + \frac{G_\theta}{r}\frac{f}{\theta} + G_z\frac{\partial f}{\partial z}\,,\\
(\boldsymbol{G}\cdot\boldsymbol{\nabla}) \boldsymbol{F} &= \left[(\boldsymbol{G}\cdot\boldsymbol{\nabla}) F_r - \frac{G_\theta F_\theta}{r}\right]\hat{\boldsymbol{r}} + \left[(\boldsymbol{G}\cdot\boldsymbol{\nabla}) F_\theta + \frac{G_\theta F_r}{r}\right]\hat{\boldsymbol{\theta}}\\
&\qquad + \left[(\boldsymbol{G}\cdot\boldsymbol{\nabla}) F_z\right]\hat{\boldsymbol{z}}\,.
\end{aligned}
\tag{A.11}
$$

In Eq. (A.11), $\hat{\boldsymbol{r}}, \hat{\boldsymbol{\theta}}$ and $\hat{\boldsymbol{z}}$ are unit vectors in r, θ and z-directions, respectively. The scalar function $f = f(r,\theta,z)$, $\boldsymbol{F}$ and $\boldsymbol{G}$ are vector functions $\boldsymbol{F} = F_r(r,\theta,z)\hat{\boldsymbol{r}} + F_\theta(r,\theta,z)\hat{\boldsymbol{\theta}} + F_z(r,\theta,z)\hat{\boldsymbol{z}}$ (and similar for $\boldsymbol{G}$), and $\boldsymbol{\nabla}$ is a three-dimensional derivative operator in cylindrical coordinate space. With these, the continuity equation can be written as:

$$
\frac{\partial \rho}{\partial t} + u_r\frac{\partial \rho}{\partial r} + \frac{u_\theta}{r}\frac{\partial \rho}{\partial \theta} + u_z\frac{\partial \rho}{\partial z} = -\rho\frac{1}{r}\left(\frac{\partial (ru_r)}{\partial r} + \frac{1}{r}\frac{\partial u_\theta}{\partial \theta} + \frac{\partial u_z}{\partial z}\right)\,, \tag{A.12}
$$

and the momentum equations as:

$$
\begin{aligned}
\rho\left(\frac{\partial u_r}{\partial t} + u_r\frac{\partial u_r}{\partial r} + \frac{u_\theta}{r}\frac{\partial u_r}{\partial \theta} - \frac{u_\theta^2}{r} + u_z\frac{\partial u_r}{\partial z}\right) &= -\frac{\partial p}{\partial r}\\
&+ \frac{1}{r}\frac{\partial}{\partial r}\left[2r\mu\left(\frac{\partial u_r}{\partial r} - \frac{1}{3}\boldsymbol{\nabla}\cdot\boldsymbol{u}\right)\right] + \frac{1}{r}\frac{\partial}{\partial \theta}\left[\mu\left(r\frac{\partial}{\partial r}\left(\frac{u_\theta}{r}\right) + \frac{1}{r}\frac{\partial u_r}{\partial \theta}\right)\right]\\
&- \frac{2\mu}{r}\left[\frac{1}{r}\frac{\partial u_\theta}{\partial \theta} + \frac{u_r}{r} - \frac{1}{3}\boldsymbol{\nabla}\cdot\boldsymbol{u}\right] + \frac{\partial}{\partial z}\left[\mu\left(\frac{\partial u_r}{\partial z} + \frac{\partial u_z}{\partial r}\right)\right] + F_r
\end{aligned}
\tag{A.13}
$$

$$\begin{aligned}\rho\left(\frac{\partial u_\theta}{\partial t}+u_r\frac{\partial u_\theta}{\partial r}+\frac{u_\theta}{r}\frac{\partial u_\theta}{\partial \theta}-\frac{u_r u_\theta}{r}+u_z\frac{\partial u_\theta}{\partial z}\right)&=-\frac{1}{r}\frac{\partial p}{\partial \theta}\\&+\frac{1}{r^2}\frac{\partial}{\partial r}\left[r\mu\left(r^2\frac{\partial}{\partial r}\left(\frac{u_\theta}{r}\right)+\frac{\partial u_r}{\partial \theta}\right)\right]+\frac{1}{r}\frac{\partial}{\partial \theta}\left[2\mu\left(\frac{1}{r}\frac{\partial u_\theta}{\partial \theta}+\frac{u_r}{r}\right.\right.\\&\left.\left.-\frac{1}{3}\nabla\cdot\boldsymbol{u}\right)\right]+\frac{\partial}{\partial z}\left[\mu\left(\frac{\partial u_\theta}{\partial z}+\frac{1}{r}\frac{\partial u_z}{\partial \theta}\right)\right]+F_\theta\end{aligned}\tag{A.14}$$

$$\begin{aligned}\rho\left(\frac{\partial u_z}{\partial t}+u_r\frac{\partial u_z}{\partial r}+\frac{u_\theta}{r}\frac{\partial u_z}{\partial \theta}+u_z\frac{\partial u_z}{\partial z}\right)&=-\frac{\partial p}{\partial z}+\frac{\partial}{\partial r}\left[r\mu\left(\frac{\partial u_r}{\partial z}+\frac{\partial u_z}{\partial r}\right)\right]\\&+\frac{1}{r}\frac{\partial}{\partial \theta}\left[\mu\left(\frac{\partial u_\theta}{\partial z}+\frac{1}{r}\frac{\partial u_z}{\partial \theta}\right)\right]+\frac{\partial}{\partial z}\left[2\mu\left(\frac{\partial u_z}{\partial z}-\frac{1}{3}\nabla\cdot\boldsymbol{u}\right)\right]+F_z\end{aligned}\tag{A.15}$$

The term-by-term formulation of right hand sides of Eqs. (2.6) and (2.7) in cylindrical coordinates are:

$$\begin{aligned}\left[(\boldsymbol{u}\cdot\nabla)\boldsymbol{u}\right]_r&=u_r\frac{\partial u_r}{\partial r}+\frac{u_\theta}{r}\frac{\partial u_r}{\partial \theta}+u_z\frac{\partial u_r}{\partial z}-\frac{u_\theta^2}{r},\\\left[(\boldsymbol{u}\cdot\nabla)\boldsymbol{u}\right]_\theta&=u_r\frac{\partial u_\theta}{\partial r}+\frac{u_\theta}{r}\frac{\partial u_\theta}{\partial \theta}+u_z\frac{\partial u_\theta}{\partial z}-\frac{u_r u_\theta}{r},\\\left[(\boldsymbol{u}\cdot\nabla)\boldsymbol{u}\right]_z&=u_r\frac{\partial u_z}{\partial r}+\frac{u_\theta}{r}\frac{\partial u_z}{\partial \theta}+u_z\frac{\partial u_r}{\partial z},\end{aligned}\tag{A.16}$$

$$\begin{aligned}\left[c_s^2\nabla(\ln\rho)\right]_r&=c_s^2\frac{1}{\rho}\frac{\partial \rho}{\partial r},\\\left[c_s^2\nabla(\ln\rho)\right]_\theta&=c_s^2\frac{1}{\rho r}\frac{\partial \rho}{\partial \theta},\\\left[c_s^2\nabla(\ln\rho)\right]_z&=c_s^2\frac{1}{\rho}\frac{\partial \rho}{\partial z},\end{aligned}\tag{A.17}$$

$$\begin{aligned}\left[\nabla^2\boldsymbol{u}\right]_r&=\frac{\partial^2 u_r}{\partial r^2}+\frac{1}{r}\frac{\partial u_r}{\partial r}+\frac{1}{r^2}\frac{\partial^2 u_r}{\partial \theta^2}+\frac{\partial^2 u_r}{\partial z^2}-\frac{1}{r^2}u_r-\frac{2}{r^2}\frac{\partial u_\theta}{\partial \theta},\\\left[\nabla^2\boldsymbol{u}\right]_\theta&=\frac{\partial^2 u_\theta}{\partial r^2}+\frac{1}{r}\frac{\partial u_\theta}{\partial r}+\frac{1}{r^2}\frac{\partial^2 u_\theta}{\partial \theta^2}+\frac{\partial^2 u_\theta}{\partial z^2}-\frac{1}{r^2}u_\theta+\frac{2}{r^2}\frac{\partial u_r}{\partial \theta},\\\left[\nabla^2\boldsymbol{u}\right]_z&=\frac{\partial^2 u_z}{\partial r^2}+\frac{1}{r}\frac{\partial u_z}{\partial r}+\frac{1}{r^2}\frac{\partial^2 u_z}{\partial \theta^2}+\frac{\partial^2 u_z}{\partial z^2},\end{aligned}\tag{A.18}$$

$$\begin{aligned}\left[\nabla(\nabla\cdot\boldsymbol{u})\right]_r&=\frac{\partial^2 u_r}{\partial r^2}+\frac{1}{r}\frac{\partial^2 u_\theta}{\partial r\partial \theta}+\frac{\partial^2 u_z}{\partial r\partial z}+\frac{1}{r}\frac{\partial u_r}{\partial r}-\frac{u_r}{r^2}-\frac{1}{r^2}\frac{\partial u_\theta}{\partial \theta},\\\left[\nabla(\nabla\cdot\boldsymbol{u})\right]_\theta&=\frac{1}{r}\frac{\partial^2 u_r}{\partial r\partial \theta}+\frac{1}{r^2}\frac{\partial^2 u_\theta}{\partial \theta^2}+\frac{1}{r}\frac{\partial^2 u_z}{\partial \theta\partial z}+\frac{1}{r^2}\frac{\partial u_r}{\partial \theta},\end{aligned}\tag{A.19}$$

$$
\begin{aligned}
[\nabla(\nabla\cdot\boldsymbol{u})]_z &= \frac{\partial^2 u_r}{\partial r\partial z} + \frac{1}{r}\frac{\partial^2 u_\theta}{\partial\theta\partial z} + \frac{\partial^2 u_z}{\partial z^2} + \frac{1}{r}\frac{\partial u_r}{\partial z}, \\
[\boldsymbol{S}\cdot\nabla(\ln\rho)]_r &= \left[\frac{\partial u_r}{\partial r} - \frac{1}{3}\left(\frac{\partial u_r}{\partial r} + \frac{u_r}{r} + \frac{1}{r}\frac{\partial u_\theta}{\partial\theta} + \frac{\partial u_z}{\partial z}\right)\right]\frac{1}{\rho}\frac{\partial\rho}{\partial r} \\
&\quad + \frac{1}{2}\left(\frac{\partial u_\theta}{\partial r} - \frac{u_\theta}{r} + \frac{1}{r}\frac{\partial u_r}{\partial\theta}\right)\frac{1}{r\rho}\frac{\partial\rho}{\partial\theta} \\
&\quad + \frac{1}{2}\left(\frac{\partial u_r}{\partial z} + \frac{\partial u_z}{\partial r}\right)\frac{1}{\rho}\frac{\partial\rho}{\partial z}, \\
[\boldsymbol{S}\cdot\nabla(\ln\rho)]_\theta &= \frac{1}{2}\left(\frac{\partial u_\theta}{\partial r} - \frac{u_\theta}{r} + \frac{1}{r}\frac{\partial u_r}{\partial\theta}\right)\frac{1}{\rho}\frac{\partial\rho}{\partial r} \\
&\quad + \left[\frac{1}{r}\frac{\partial u_\theta}{\partial r} + \frac{u_r}{r} - \frac{1}{3}\left(\frac{\partial u_r}{\partial r} + \frac{u_r}{r} + \frac{1}{r}\frac{\partial u_\theta}{\partial\theta} + \frac{\partial u_z}{\partial z}\right)\right]\frac{1}{r\rho}\frac{\partial\rho}{\partial\theta} \\
&\quad + \frac{1}{2}\left(\frac{\partial u_\theta}{\partial z} + \frac{1}{r}\frac{\partial u_z}{\partial\theta}\right)\frac{1}{\rho}\frac{\partial\rho}{\partial z}, \\
[\boldsymbol{S}\cdot\nabla(\ln\rho)]_z &= \frac{1}{2}\left(\frac{\partial u_z}{\partial r} + \frac{\partial u_r}{\partial z}\right)\frac{1}{\rho}\frac{\partial\rho}{\partial r} + \frac{1}{2}\left(\frac{\partial u_\theta}{\partial z} + \frac{1}{r}\frac{\partial u_z}{\partial\theta}\right)\frac{1}{r\rho}\frac{\partial\rho}{\partial\theta} \\
&\quad + \left[\frac{\partial u_z}{\partial z} - \frac{1}{3}\left(\frac{\partial u_r}{\partial r} + \frac{u_r}{r} + \frac{1}{r}\frac{\partial u_\theta}{\partial\theta} + \frac{\partial u_z}{\partial z}\right)\right]\frac{1}{\rho}\frac{\partial\rho}{\partial z},
\end{aligned}
\tag{A.20}
$$

The term-by-term equations can be reduced to versions of the ordinary formulations of the Navier-Stokes equations, provided the fluid can be described by the ideal gas law (Eq. (2.5)) with constant speed of sound and constant kinematic viscosity.

The cylindrical terms of Eqs. (A.16)–(A.20) appear more unwieldy than the Cartesian counterparts, Eqs. (A.6)–(A.10). However, by formulating the Navier-Stokes equations in this way it becomes clear that regardless of the choice of coordinate system, the equations are functions of the fluid variables ρ and $\boldsymbol{u}$ and a set of *primitive* derivatives only. The primitive derivatives are:

$$
\frac{\partial\rho}{\partial x_i}, \frac{\partial u_{x_i}}{\partial x_i}, \frac{\partial^2 u_{x_i}}{\partial x_j\partial x_k}, \tag{A.21}
$$

where $x_i \in \{x, y, z\}$ for Cartesian coordinates and $x_i \in \{r, \theta, z\}$ for cylindrical coordinates (the same goes for x_j and x_k).

Bibliography

[1] Aarnes, J. R., Haugen, N. E. L., Andersson, H. I., On validation and implementation of an immersed boundary method in a high order finite difference code for flow simulations. In *MekIT'15: Eight National Conference on Computational Mechanics*, Skallerud, B., and Andersson, H. I., Eds.; Barcelona, Spain: International Center for Numerical Methods in Engineering (CIMNE): 2015, pp 1–21.

[2] Aarnes, J. R., Haugen, N. E. L., Andersson, H. I., (2018). High-order overset grid method for detecting particle impaction on a cylinder in a cross flow. *(Manuscript submitted for publication).*

[3] Aarnes, J. R., Haugen, N. E. L., Andersson, H. I., (2018). Inertial particle impaction on a cylinder in turbulent cross flow at modest Reynolds numbers. *(Manuscript submitted for publication).*

[4] Aarnes, J. R., Andersson, H. I., Haugen, N. E. L., (2018). Numerical investigation of free-stream turbulence effects on the transition-in-wake state of flow past a circular cylinder. *Journal of Turbulence 19*, 252–273.

[5] Aarnes, J. R., Jin, T., Mao, C., Haugen, N. E. L., Luo, K., and Andersson, H. I., (2018). Treatment of solid objects in the Pencil Code using an immersed boundary method and overset grids. *Geophysical & Astrophysical Fluid Dynamics 0*, 1–23.

[6] Abramowitz, M., and Stegun, I. A., *Handbook of Mathematical Functions: With Formulas, Graphs, and Mathematical Tables*; Dover Publications: Mineola, NY, 1964.

[7] Babkovskaia, N., Haugen, N., Brandenburg, A., (2011). A high-order public domain code for direct numerical simulations of turbulent combustion. *Journal of Computational Physics 230*, 1–12.

[8] Barkley, D., and Henderson, R. D., (1996). Three-dimensional Floquet stability analysis of the wake of a circular cylinder. *Journal of Fluid Mechanics 322*, 215–241.

[9] Barkley, D, Tuckerman, L., Golubitsky, M, (2000). Bifurcation theory for three-dimensional flow in the wake of a circular cylinder. *Physical Review E 61*, 5247–52.

[10] Beckmann, A., Mancini, M, Weber, R, Seebold, S, and Müller, M, (2016). Measurements and CFD modeling of a pulverized coal flame with emphasis on ash deposition. *Fuel 167*, 168–179.

[11] Benek, J. A., Buningt, P. G., Steger, J. L., A 3-D chimera grid embedding technique. In *AIAA 7th Computational Fluid Dynamics Conference*, AIAA Paper No. 1523, 1985.

[12] Berthelsen, P. A., and Faltinsen, O. M., (2008). A local directional ghost cell approach for incompressible viscous flow problems with irregular boundaries. *Journal of Computational Physics 227*, 4354–4397.

[13] Bird, R., Stewart, W., Lightfoot, E., *Transport Phenomena*, 2nd ed.; John Wiley and Sons: New York, NY, 2007.

[14] Bloor, M. S., (1964). The transition to turbulence in the wake of a circular cylinder. *Journal of Fluid Mechanics 19*, 290–304.

[15] Brandenburg, A., and Dobler, W., (2002). Hydromagnetic turbulence in computer simulations. *Computer Physics Communications 147*, 471–475.

[16] Brandenburg, A., et al. The Pencil Code: A High-Order MPI code for MHD Turbulence – User's and Reference Manual., Stockholm, Sweden: NORDITA. Updated February 2018. Available at https://github.com/pencil-code and http://pencil-code.nordita.org/.

[17] Brandenburg, A., (2001). The inverse cascade and nonlinear alpha-effect in simulations of isotropic helical hydromagnetic turbulence. *The Astrophysical Journal 550*, 824.

[18] Brandenburg, A., and Sarson, G. R., (2002). Effect of Hyperdiffusivity on Turbulent Dynamos with Helicity. *Physical Review Letters 88*, 055003.

[19] Cantwell, B. J., *Fundamentals of Compressible Flow*; Stanford Bookstore Custom Publishing: Stanford, CA, 2005.

[20] Causon, D. M., Ingram, D. M., Mingham, C. G., Yang, G, and Pearson, R. V., (2000). Calculation of shallow water flows using a Cartesian cut cell approach. *Advances in Water Resources 23*, 545–562.

[21] Chesshire, G., and Henshaw, W. D., (1990). Composite overlapping meshes for the solution of partial differential equations. *Journal of Computational Physics 90*, 1–64.

[22] Chicheportiche, J., and Gloerfelt, X., (2012). Study of interpolation methods for high-accuracy computations on overlapping grids. *Computers & Fluids 68*, 112–133.

[23] Clift, R., Grace, J. R., Weber, M. E., (1978). Bubbles. *Drops and Particles*.

[24] Crowe, C. T., Schwarzkopf, J. D., Sommerfeld, M., and Tsuji, Y., *Multiphase flows with Droplets and Particles*, 2nd ed.; CRC press: Boca Raton, FL, 2011.

[25] Cunningham, E, (1910). On the velocity of steady fall of spherical particles through fluid medium. *Proceedings of the Royal Society of London A 83*, 357–365.

[26] Davidson, P., *Turbulence: An Introduction for Scientists and Engineers*; Oxford University Press: Oxford, United Kingdom, 2014.

[27] Davies, C., (1945). Definitive equations for the fluid resistance of spheres. *Proceedings of the Physical Society 57*, 259.

[28] Dobler, W., Stix, M., Brandenburg, A., (2006). Magnetic field generation in fully convective rotating spheres. *The Astrophysical Journal 638*, 336–347.

[29] Douglas, P. L., and Ilias, S., (1988). On the deposition of aerosol particles on cylinders in turbulent cross flow. *Journal of Aerosol Science 19*, 451–462.

[30] Eves, H. W., *An Introduction to the History of Mathematics*, 6th ed.; Cengage Learning: Boston, MA, 1990.

[31] Fernández, D. C.D. R., Hicken, J. E., Zingg, D. W., (2014). Review of summation-by-parts operators with simultaneous approximation terms for the numerical solution of partial differential equations. *Computers & Fluids 95*, 171–196.

[32] Ferzuger, J. H., Large Eddy Simulations. In *Simulation and Modeling of Turbulent Flows*, Gatski, T. B., Hussaini, M. Y., Lumley, J. L., Eds.; Oxford University Press: New York, NY, USA, 1996; Chapter 3, pp 109–154.

[33] Fornberg, B., (1988). Generation of finite difference formulas on arbitrarily spaced grids. *Mathematics of Computation 51*, 699–706.

[34] Gaitonde, D. V., and Visbal, M. R., (2000). Padé-type higher-order boundary filters for the Navier-Stokes equations. *AIAA Journal 38*, 2103–2112.

[35] Gerrard, J. H., (1978). The wakes of cylindrical bluff bodies at low Reynolds number. *Philosophical Transactions of the Royal Society of London: A 288*, 351–382.

[36] Griffiths, D. J., *Introduction to Electrodynamics*, 3rd ed.; Prentice Hall: Upper Saddle River, NJ, 1999.

[37] Gustafsson, B., *High Order Difference Methods for Time Dependent PDE*; Springer-Verlag: Berlin, Germany, 2008.

[38] Haugen, N. E. L., and Brandenburg, A., (2006). Hydrodynamic and hydromagnetic energy spectra from large eddy simulations. *Physics of Fluids 18*, 075106.

[39] Haugen, N. E. L., and Kragset, S., (2010). Particle impaction on a cylinder in a crossflow as function of Stokes and Reynolds numbers. *Journal of Fluid Mechanics 661*, 239–261.

[40] Haugen, N. E. L., Kragset, S., Bugge, M., Warnecke, R., and Weghaus, M., (2013). MSWI super heater tube bundle: Particle impaction efficiency and size distribution. *Fuel Processing Technology 106*, 416–422.

[41] Heisenberg, W., *Physics and philosophy: The Revolution in Modern Science*; Harper & Row: New York, NY, 1958.

[42] Henderson, R. D., (1997). Nonlinear dynamics and pattern formation in turbulent wake transition. *Journal of Fluid Mechanics 352*, 65–112.

[43] Henderson, R. D., and Barkley, D., (1996). Secondary instability in the wake of a circular cylinder. *Physics of Fluids 8*, 1683–1685.

[44] Huang, L. Y., Norman, J. S., Pourkashanian, M, and Williams, A, (1996). Prediction of ash deposition on superheater tubes from pulverized coal combustion. *Fuel 75*, 271–279.

[45] Hussain, A. K.M. F., and Ramjee, V., (1976). Periodic wake behind a circular cylinder at low Reynolds numbers. *Aeronautic Quarterly 27*, 123–142.

[46] Ingham, D. B., Hildyard, L. T., Hildyard, M. L., (1990). On the critical Stokes' number for particle transport in potential and viscous flows near bluff bodies. *Journal of Aerosol Science 21*, 935–946.

[47] Ingram, D. M., Causon, D. M., Mingham, C. G., (2003). Developments in Cartesian cut cell methods. *Mathematics and Computers in Simulation 61*, 561–572.

[48] Israel, R, and Rosner, D. E., (1982). Use of a generalized Stokes number to determine the aerodynamic capture efficiency of non-Stokesian particles from a compressible gas flow. *Aerosol Science and Technology 2*, 45–51.

[49] Johansen, A., Klahr, H., Henning, T., (2011). High-resolution simulations of planetesimal formation in turbulent protoplanetary discs. *Astronomy & Astrophysics 529*, A62.

[50] Kasper, G., Schollmeier, S., Meyer, J., and Hoferer, J., (2009). The collection efficiency of a particle-loaded single filter fiber. *Journal of Aerosol Science 40*, 993–1009.

[51] Kim, J., Moin, P., Moser, R., (1987). Turbulence statistics in fully developed channel flow at low Reynolds number. *Journal of Fluid Mechanics 177*, 133–166.

[52] Kolmogorov, A. N., (1941). The local structure of turbulence in incompressible viscous fluid for very large Reynolds numbers (Russian). *Doklady Akademii Nauk SSSR 30*, 299–303.

[53] Kolmogorov, A. N., (1991). The local structure of turbulence in incompressible viscous fluid for very large Reynolds numbers (English translation). *Proceedings of the Royal Society of London A 434*, 9–13.

[54] Kubrick, S. (Producer), Kubrid, S. (Director), 2001: A space odyssey., United States: Metro-Goldwyn-Mayer Corp. 1968.

[55] Kuerten, J. G. M., (2016). Point-Particle DNS and LES of Particle-Laden Turbulent flow – a state-of-the-art review. *Flow, turbulence and combustion 97*, 689–713.

[56] Lee, F., and Lockwood, F., (1999). Modelling ash deposition in pulverized coal-fired applications. *Progress in Energy and Combustion Science 25*, 117 –132.

[57] Li, X., Zhou, H., Cen, K., (2008). Influences of various vortex structures on the dispersion and deposition of small ash particles. *Fuel 87*, 1379–1382.

[58] Linnick, M. N., and Fasel, H. F., (2005). A high-order immersed interface method for simulating unsteady incompressible flows on irregular domains. *Journal of Computational Physics 204*, 157–192.

[59] Lodato, G., Domingo, P., Vervisch, L., (2008). Three-dimensional boundary conditions for direct and large-eddy simulation of compressible viscous flows. *Journal of Computational Physics 227*, 5105 –5143.

[60] Marchioli, C., Physics and modelling of particle deposition and resuspension in wall-bounded turbulence. In *Particles in Wall-Bounded Turbulent Flows: Deposition, Re-Suspension and Agglomeration*, Minier, J.-P., and Pozorski, J., Eds.; Springer Verlag: Berlin, Germany, 2017, pp 151–208.

[61] Mattsson, K., and Nordström, J., (2004). Summation by parts operators for finite difference approximations of second derivatives. *Journal of Computational Physics 199*, 503–540.

[62] Mavriplis, D. J., (1997). Unstructured grid techniques. *Annual Review of Fluid Mechanics 29*, 473–514.

[63] Meakin, R., The chimera method of simulations for unsteady three-dimensional viscous flow. In *Computational Fluid Dynamics Review*, Hafez, M., and Oshima, K., Eds.; Wiley: New York, NY, 1995, pp 70–86.

[64] Mittal, R., and Iaccarino, G., (2005). Immersed Boundary Methods. *Annual Review of Fluid Mechanics 37*, 239–261.

[65] Noack, R. W., SUGGAR: A general capability for moving body overset grid assembly. In *17th AIAA Computational Fluid Dynamics Conference*, AIAA Paper 2005-5117, 2005.

[66] Norberg, C., In, Dept. Applied Thermodynamics and Fluid Mechanics, Chalmers University of Technology, Gothenburg, Sweden, 1987.

[67] Norberg, C., (1994). An experimental investigation of the flow around a circular cylinder: influence of aspect ratio. *Journal of Fluid Mechanics 258*, 287–316.

[68] Owen, S. J., A survey of unstructured mesh generation technology. In *Proceedings of 7th international Meshing Roundtable*; Sandia National Laboratories: Dearborn, MI, 1998, pp 239–267.

[69] Pärt-Enander, E., and Sjögreen, B., (1994). Conservative and non-conservative interpolation between overlapping grids for finite volume solutions of hyperbolic problems. *Computers and Fluids 23*, 551–574.

[70] Pérez, M. G., Vakkilainen, E., Hyppänen, T., (2016). Fouling growth modeling of kraft recovery boiler fume ash deposits with dynamic meshes and a mechanistic sticking approach. *Fuel 185*, 872–885.

[71] Peskin, C. S., (1972). Flow patterns around heart valves: A numerical method. *Journal of Computational Physics 10*, 252–271.

[72] Peskin, C. S., (2002). The immersed boundary method. *Acta Numerica 11*, 479–517.

[73] Plato, Cooper, J. M., Hutchinson, D. S., *Plato: complete works*; Hackett Publishing Company: Indianapolis, IN, 1997.

[74] Poinsot, T. J., and Lele, S., (1992). Boundary conditions for direct simulations of compressible viscous flows. *Journal of Computational Physics 101*, 104–129.

[75] Pope, S., *Turbulent Flows*; Cambridge University Press: Cambridge, United Kingdom, 2000.

[76] Popper, K. R., *Objective Knowledge: An Evolutionary Approach*; Clarendon Press: Oxford, United Kingdom, 1972.

[77] Posdziech, O., and Grundmann, R., (2001). Numerical simulation of the flow around an infinitely long circular cylinder in the transition regime. *Theoretical and Computational Fluid Dynamics 15*, 121–141.

[78] Press, W. H., *Numerical Recipes: The Art of Scientific Computing*, 3rd ed.; Cambridge University Press: Cambridge, United Kingdom, 2007.

[79] Provansal, M., Mathis, C., Boyer, L., (1987). Bénard-von Kármán instability: transient and forced regimes. *Journal of Fluid Mechanics 182*, 1–22.

[80] Quartapelle, L., *Numerical Solution of the Incompressible Navier-Stokes Equations*; Birkhäuser Verlag: Basel, Switzerland, 1993.

[81] Quirk, J. J., (1994). An alternative to unstructured grids for computing gas dynamic flows around arbitrarily complex two-dimensional bodies. *Computers & fluids 23*, 125–142.

[82] Reynolds, O., (1895). On the dynamical theory of incompressible viscous fluids and the determination of the criterion. *Philosophical Transactions of the Royal Society of London: A 186*, 123–164.

[83] Richardson, L. F., (1922). Weather Prediction by Numerical Process.

[84] Rogers, S. E., Suhs, N. E., Dietz, W. E., (2003). PEGASUS 5: An automated preprocessor for overset-grid computational fluid dynamics. *AIAA Journal 41*, 1037–1045.

[85] Roshko, A., (1954). On the development of turbulent wakes from vortex streets. *NASA Report*.

[86] Russell, B., *The Conquest of Happiness*; Liveright Publishing Corporation: New York, NY, 1930.

[87] Schiller, V. L., and Neumann, A, (1933). Uber die grundlegenden Berechnungen bei der Schwerkraftaufbereitung (German). *Zeitschrift des Vereines Deutscher Ingenieure 77*, 318–320.

[88] Schneiders, L., Hartmann, D., Meinke, M., and Schröder, W., (2013). An accurate moving boundary formulation in cut-cell methods. *Journal of Computational Physics 235*, 786–809.

[89] Schweers, E., Umhauer, H., Löffler, F., (1994). Experimental investigation of particle collection on single fibres of different configurations. *Particle & Particle Systems Characterization 11*, 275–283.

[90] Seo, J. H., and Mittal, R., (2011). A high-order immersed boundary method for acoustic wave scattering and low-Mach number flow-induced sound in complex geometries. *Journal of Computational Physics 230*, 1000–1019.

[91] Sherer, S. E., and Scott, J. N., (2005). High-order compact finite-difference methods on general overset grids. *Journal of Computational Physics 210*, 459–496.

[92] Smagorinsky, J., (1963). General circulation experiments with the primitive equations: I. The basic experiment. *Monthly Weather Review 91*, 99–164.

[93] Steger, J. L., Dougherty, F. C., Benek, J. A., A chimera grid scheme. In *Advances in Grid Generation*, Ghia, K. N., and Ghia, U., Eds., ASME FED-5, 1983, pp 59–69.

[94] Steger, J. L., and Benek, J. A., (1987). On the use of composite grid schemes in computational aerodynamics. *Computer Methods in Applied Mechanics and Engineering 64*, 301–320.

[95] Stokes, G. G., (1851). On the effect of the internal friction of fluids on the motion of pendulums. *Transactions of the Cambridge Philosophical Society 9 (II)*, 8–106.

[96] Strand, B., (1994). Summation by parts for finite difference approximations for d/dx. *Journal of Computational Physics 110*, 47–67.

[97] Svärd, M., Mattsson, K., Nordström, J., (2005). Steady-state computations using summation-by-parts operators. *Journal of Scientific Computing 24*, 79–95.

[98] Tannehill, J. C., Anderson, D., Pletcher, R. H., *Computational Fluid Mechanics and Heat Transfer*, 2nd ed.; Taylor & Francis: Philadelphia, PA, 1997.

[99] Taylor, C. C. W., and Huby, P. M., *Simplicius: On Aristotle Physics 1.3–4*; Bloomsbury Academic: London, United Kingdom, 2014.

[100] The Pencil Code [Computer software], Hosted at http://pencil-code.nordita.org; Stockholm, Sweden: NORDITA; Retrieved April 2018, from https://github.com/pencil-code.

[101] Thomas, L. H., Elliptic problems in linear difference equations over a network., tech. rep., Watson Scientific Computing Laboratory, Columbia University, New York, NY, 1949.

[102] Tritton, D. J., (1959). Experiments on the flow past a circular cylinder at low Reynolds numbers. *Journal of Fluid Mechanics 6*, 547–567.

[103] Tseng, Y.-H., and Ferziger, J. H., (2003). A ghost-cell immersed boundary method for flow in complex geometry. *Journal of Computational Physics 192*, 593–623.

[104] van der Hoef, M. A., van Sint Annaland, M, Deen, N., and Kuipers, J., (2008). Numerical simulation of dense gas-solid fluidized beds: a multiscale modeling strategy. *Annual Review of Fluid Mechanics 40*, 47–70.

[105] Versteeg, H. K., and Malalasekera, W., *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*, 2nd ed.; Pearson Education: Harlow, United Kingdom, 2007.

[106] Visbal, M. R., and Gaitonde, D. V., (1999). High-order-accurate methods for complex unsteady subsonic flows. *AIAA Journal 37*, 1231–1239.

[107] Visbal, M. R., and Gaitonde, D. V., (2002). On the use of higher-order finite-difference schemes on curvilinear and deforming meshes. *Journal of Computational Physics 181*, 155–185.

[108] Völkner, S., Brunswig, J., Rung, T., (2017). Analysis of non-conservative interpolation techniques in overset grid finite-volume methods. *Computers & Fluids 148*, 39–55.

[109] Von Kármán, T., *Aerodynamics: Selected Topics in the Light of Their Historical Development*; Dover Books on Aeronautical Engineering Series; Dover Publications: Mineola, NY, 1954.

[110] Wacławiak, K., and Kalisz, S., (2012). A practical numerical approach for prediction of particulate fouling in PC boilers. *Fuel 97*, 38–48.

[111] Weber, R., Schaffel-Mancini, N., Mancini, M., and Kupka, T., (2013). Fly ash deposition modelling: Requirements for accurate predictions of particle impaction on tubes using RANS-based computational fluid dynamics. *Fuel 108*, 586–596.

[112] Weber, R., Mancini, M., Schaffel-Mancini, N., and Kupka, T., (2013). On predicting the ash behaviour using Computational Fluid Dynamics. *Fuel Processing Technology 105*, 113–128.

[113] White, F. M., *Viscous Fluid Flow*, 3rd ed.; McGraw-Hill Education: New York, NY, 2006.

[114] White, F. M., and Chul, R. Y., *Fluid Mechanics*, 8th ed.; McGraw-Hill series in mechanical engineering; McGraw-Hill Education: New York, NY, 2016.

[115] Williamson, C. H. K., (1988). The existence of two stages in the transition to three-dimensionality of a cylinder wake. *Journal of Fluid Mechanics 31*, 3165–3168.

[116] Williamson, C. H. K., (1992). The natural and forced formation of spot-like 'vortex dislocations' in the transition of a wake. *Journal of Fluid Mechanics 243*, 393–441.

[117] Williamson, C. H. K., (1996). Three-dimensional wake transition. *Journal of Fluid Mechanics 328*, 345–407.

[118] Williamson, C. H. K., (1996). Vortex dynamics in the cylinder wake. *Annual Review of Fluid Mechanics 28*, 477–539.

[119] Williamson, C. H. K., and Roshko, A, (1990). Measurements of base pressure in the wake of a cylinder at low Reynolds numbers. *Zeitschrift fur Flugwissenschaften und Weltraumforschung 14*, 38–46.

[120] Williamson, J. H., (1980). Low-storage Runge-Kutta schemes. *Journal of Computational Physics 35*, 48–56.

[121] Xia, J., Luo, K., Fan, J., (2014). A ghost-cell based high-order immersed boundary method for inter-phase heat transfer simulation. *International Journal of Heat and Mass Transfer 75*, 302–312.

[122] Yilmaz, S, and Cliffe, K. R., (2000). Particle deposition simulation using the CFD code FLUENT. *Journal of the Institute of Energy 73*, 65–68.

[123] Yoo, C. S., Wang, Y., Trouvé, A., and Im, H. G., (2005). Characteristic boundary conditions for direct simulations of turbulent counterflow flames. *Combustion Theory and Modelling 9*, 617–646.

[124] Zang, Y., and Street, R. L., (1995). A composite multigrid method for calculating unsteady incompressible flows in geometrically complex domains. *International Journal for Numerical Methods in Fluids 20*, 341–361.

[125] Zdravkovich, M. M., (1990). Conceptual overview of laminar and turbulent flows past smooth and rough circular cylinders. *Journal of Wind Engineering & Industrial Aerodynamics 33*, 53–62.

[126] Zdravkovich, M. M., *Flow Around Circular Cylinders: Volume 1: Fundamentals*; Oxford University Press: Oxford, United Kingdom, 1997.

[127] Zhang, H. Q., Fey, U., Noack, B. R., König, M., and Eckelmann, H., (1995). On the transition of the cylinder wake. *Physics of Fluids 7*, 779–794.

Part II

One of the symptoms of an approaching nervous breakdown is the belief that one's work is terribly important.

Bertrand Russell[35]

5 Research papers

[35] In *The Conquest of Happiness* [86]

Paper I [1]

On validation and implementation of an immersed boundary method in a high order finite difference code for flow simulations

Jørgen R. Aarnes, Nils E. L. Haugen and Helge I. Andersson

In: MekIt'15: Eight national conference on computational mechanics, Trondheim (Norway). Barcelona (Spain): International Center for Numerical Methods in Engineering, pp. 1–21, 2015.

8. National Conference on Computational Mechanics
MekIT'15
B. Skallerud and H I Andersson (Eds)

ON VALIDATION AND IMPLEMENTATION OF AN IMMERSED BOUNDARY METHOD IN A HIGH ORDER FINITE DIFFERENCE CODE FOR FLOW SIMULATIONS

JØRGEN R. AARNES[1], NILS E. L. HAUGEN[1,2] AND HELGE I. ANDERSSON[1]

[1]Department of Energy- and Process Engineering
Norwegian University of Science and Technology (NTNU)
Kolbjørn Hejes vei 1b, 7491 Trondheim, Norway
e-mail: jorgen.r.aarnes@ntnu.no, web page: http://www.ntnu.no/ept

[2] SINTEF Energy Research
Kolbjørn Hejes vei 1c, 7491 Trondheim, Norway
e-mail: nils.e.haugen@sintef.no - Web page: http://www.sintef.no/home/sintef-energy

Key words: Computational Methods, Immersed Boundary, Computational Fluid Dynamics, Flow Past Cylinder, Validation, Compressible Flow

Abstract. An open source code using high-order finite differences for fluid flow simulations has been validated for use for flows past a stationary cylinder. The implementation of the immersed boundary method in the code has been considered. A new implementation is suggested, replacing orthogonal mirror points by mirror points along grid lines. This implementation resulted in more accurate computation of drag and lift coefficients, indicating increased accuracy of the computed flow close to the cylinder and allowing for coarser grids to be used in the simulations.

1 INTRODUCTION

As the cost of high computing power is ever decreasing, the use of high accuracy computations of fluid flow is no longer restricted to academia, but widespread in both research and engineering applications. The extent one trusts the results from computational fluid dynamic simulations can be illustrated by the way direct numerical simulations (DNS) of turbulence are often referred to as numerical experiments, rather than simulations or numerical computations. The firm belief in computational results is not unfounded. Indeed, numerical simulations compare very well with experimental results, and open for research on many scientific problems that cannot easily be studied experimentally.

A problem may, however, arise when developed software is applied to research for which it has not been thoroughly tested and validated. The value of numerical results that are seemingly trustworthy will quickly decrease if a small change of the case studied, e.g. a small increase of the domain size, has a large, unexpected influence on the computed results.

The aim of this paper is to present a thorough and systematic validation of the open source high order finite difference code known as the Pencil code [1] used to compute compressible flow around a circular cylinder at Reynolds number Re = 100, where

$$\mathrm{Re} = \frac{U_\infty D}{\nu}, \tag{1}$$

with free stream velocity U_∞, cylinder diameter D and kinematic viscosity ν. The flow problem is a well-known benchmarking case. The validation is presented in detail as the code used is open for use by other researchers, whom may easily compare their validation to the results presented here. Further, it is the intent of the authors to make use of this software for simulating turbulent two-phase flow and the importance of understanding how different parameters affect the results cannot be overstated.

The Pencil Code is an open source code where all routines can be inspected in detail. In the process of validation, a closer look is taken on the way the fluid-solid interface in the flow problem is represented. This is done by the immersed boundary method with discrete forcing and ghost cells as described in, e.g., [2, 3]. The code uses 6th order central difference on all fluid points, and a three point deep ghost zone is therefore necessary in representing the solid interface. More details on the implementation of the immersed boundary implementation and a suggested method of improvement can be found in Sec. 3.

2 VALIDATION

In conducting a thorough validation meant to lay the foundation for further work for a specific case, the effect of varying several different parameters has been studied. The parameters include both physical properties of the system (domain size, Mach number) and numerical properties (grid spacing, Courant number). For all validation runs the drag coefficient

$$C_D = \frac{2F_D}{\rho U_\infty^2 D}, \tag{2}$$

the lift coefficient

$$C_L = \frac{2F_L}{\rho U_\infty^2 D}, \tag{3}$$

and the Strouhal number

$$St = \frac{fD}{u}, \tag{4}$$

are computed, and used to compare the effect of tuning different parameters. The coefficients are expressed in terms of the free stream velocity U_∞, density ρ, cylinder diameter D, vortex shedding frequency f, drag force F_D and lift force F_L. The drag and lift forces are computed on a given number of forcepoints close to the cylinder surface. These forces

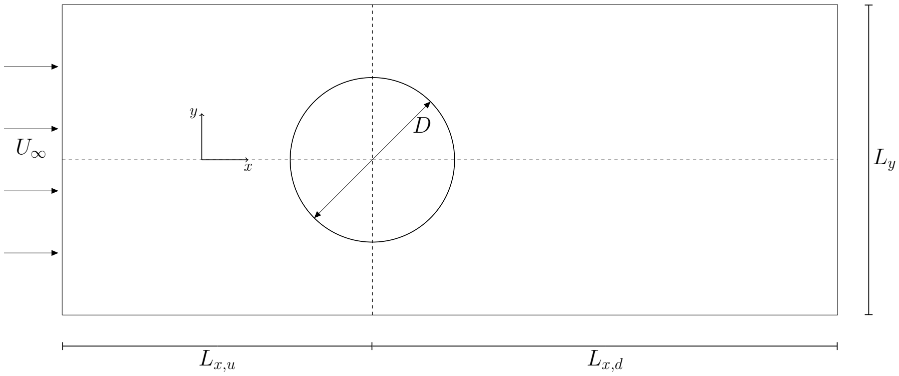


Figure 1: Definition of physical set-up. Cylinder is not to scale.

vary with time, as the Reynolds number is sufficiently high to cause vortex shedding. Therefore, the mean drag coefficient and the root-mean-square lift coefficient are computed and used in the validation. The using root-mean-square values of the lift coefficient is standard practice. The mean value should be close to zero, when rms value is not used.

The physical set-up is a two-dimensional rectangular domain of length L_x and width L_y, as show in Fig. 1. The length can be split into the upstream length, $L_{x,u}$ and the downstream length, $L_{x,d}$, which need not be equal. Partially reflecting Navier-Stokes characteristic boundary conditions (NCSBC) (see [4]) are used at both the inlet and the outlet, with constant inflow velocity U_∞ at the inlet. In the spanwise direction periodic boundary conditions are used. One could argue that free-slip walls should also be tested, but as this is not appropriate for the author's further work with this set up it is not the focus of this study.

For all the validation runs the following properties are not altered: $D = 0.1$ m, $U_\infty = 1.0$ m/s and $\nu = 1.0 \cdot 10^{-3}$ m^2/s. All lengths in the system are non-dimensionalized in terms of the cylinder diameter D.

The results from the validations runs are plotted with error bars, such that the effect of the different parameters are more easily compared. The area spanned by an error bar is 1% (0.5% above and 0.5% below a point) of the “correct” value of the drag, lift and Strouhal number. Note that the correct value is not obtainable analytically, and the results found by Qu et al. [5] using body conformal grids and very large domains are used for this purpose. For the largest domain in [5] (200D×200D) the values are $C_D = 1.310$, $C_L' = 0.2151$ and $St = 0.1647$ for incompressible flow. An alternative would be to use the asymptotical values at very large domains with body conformal grids, found by Posdziech and Grundermann [6]. They do, however, not include root-mean-square lift in their results.

2.1 Grid refinement

The first parameter that is validated is the grid spacing. The grid is Cartesian, and the convective time step is computed for each time step by

$$\Delta t = C_{CFL} \frac{(\Delta x)_{\min}}{(|\mathbf{u}| + c_s)_{\max}}, \tag{5}$$

where C_{CFL} is the Courant number, $(\Delta x)_{min} = \min(\Delta x, \Delta y, \Delta z)$, $\mathbf{u}$ is the fluid velocity at a grid point and c_s is the sound speed. Hence, the time-step is proportional to the smallest grid spacing in the system. This motivates choosing the grid spacing equal in the x- and y-direction, $\Delta x = \Delta y$. The remaining parameters (the domain size, Mach number and Courant number) are chosen rather arbitrarily, some of them perhaps overly cautiously others not strict enough, as will be shown in the validation to come. The reason for the rather arbitrary choice of parameters is the lack of a detailed validation in literature, for the specific flow problem computed with a similar software. The grid refinement results are purely of qualitative interest, to find a sufficiently fine grid, not to find quantitative information about the drag, lift and Strouhal number. The upstream and downstream length are chosen equal, and the length and width of the system are set to $20D$ and $10D$, respectively. The flow is weakly compressible with a Mach number set to 0.05. The Courant number is set to 0.4.

In Fig. 2 it can be seen that the resolution has little effect on the Strouhal number. The effect on the rms lift and the drag coefficient are comparable for $D/\Delta x \geq 25$ (for $D/\Delta x < 25$, however, the rms lift has a large drop, making the similarity between C_D and C_L' less than obvious in the figure). The oscillatory behaviour for both these quantities for $D/\Delta x < 64$ is an unwanted effect. Thus, the grid spacing should be chosen such that $D/\Delta x \geq 64$ to avoid this. Hence, quite a large number of grid points are necessary to compute accurate results, especially when the domain size is increased (assuming that the grid refinement is close to independent of the domain size). The use of a stretched grid should therefore be considered in order to reduce the resolution without using a coarser grid close to the cylinder surface.

The magenta curve in Fig. 2 depicts the mean lift coefficient for this grid refinement study. This is included for completeness, to confirm that the mean lift is indeed close to zero. The error bars in the figure are the same ones that are used for the root-mean-square lift coefficient. The results show a tendency towards negative mean coefficients for coarse grids, which may indicate symmetry breaking in the computations. This tendency is, however, not obvious as the grid spacing is decreased. Apart from this, it can be seen that the mean lift coefficient is not affected much by decreasing the grid spacing beyond $D/\Delta x \geq 25$. This indicates that rms lift coefficient is a more appropriate measure of accuracy in the validation runs. The mean lift coefficient will not be included beyond this.

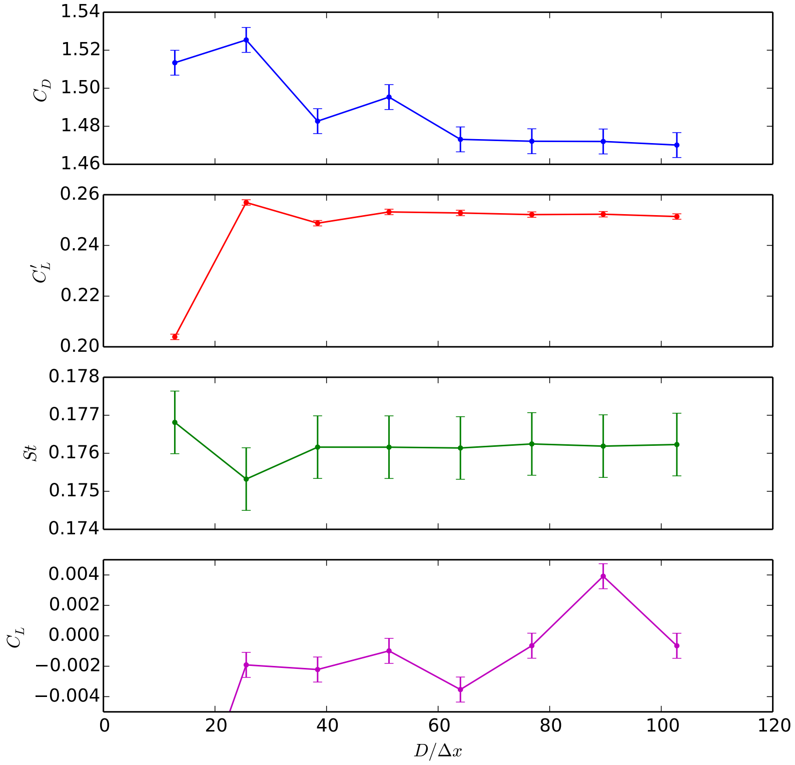


Figure 2: Computed mean drag coefficient, root-mean-square lift coefficient, Strouhal number and mean lift coefficient for different grid spacing.

2.2 Mach number

The Mach number, defined here as

$$\mathrm{Ma} = \frac{U_\infty}{c_s}, \tag{6}$$

has a large impact on the time-step in the simulations of compressible fluid flow. As the Mach number is reduced to approach the incompressible limit, the time step is also reduced. It is therefore necessary to consider the qualitative impact of the Mach number on the results, such that the effects are known for Mach numbers that are convenient to use in the simulations. All Mach numbers considered here are for subsonic flow. The domain size and Courant number is equal to that in the grid refinement simulations. The grid spacing is set to $D/\Delta x = 64$.

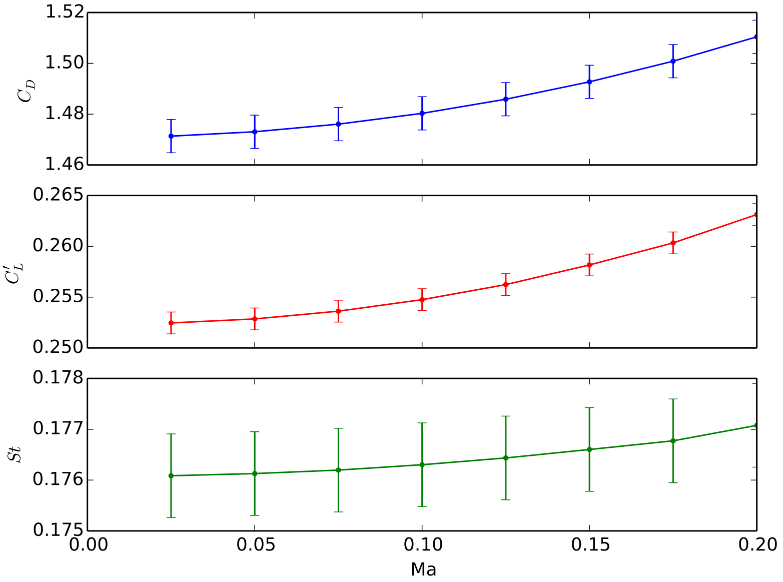


Figure 3: Computed mean drag coefficient, root-mean-square lift coefficient and Strouhal number for varying Mach number.

Figure 3 depicts the results from runs with varying Mach number. It can be seen that once again the Strouhal number is much less affected by the parameter variation, than the drag and lift coefficients. In contrast to the grid refinement results, the results in Fig. 3 vary in a smooth, non-oscillatory fashion, approaching the incompressible limit as the Mach number is decreased. This is as expected, as for most flows no important changes is observed when the Mach number is reduced from 0.2 to 0.01 [7] (not reduced below 0.025 in the present computations). From the results, one should expect less than 1% deviation from the incompressible limit, when the Mach number is set to 0.1. Thus, setting Ma = 0.1 is a reasonable trade-off between accuracy and computational cost.

2.3 Domain size

While both grid refinement and varying the Mach number is fairly straightforward, looking at the effect of changing the domain size requires some more thought. Not only should one expect that the three parameters $L_{x,u}$, $L_{x,d}$ and L_y affect the computed results, when varied one at the time, one might also expect that the variations are not independent (as assumed with grid spacing and Mach number variations). Hence, varying a single parameter at a time is not necessarily enough to fully understand how the domain size affect the computed quantities.

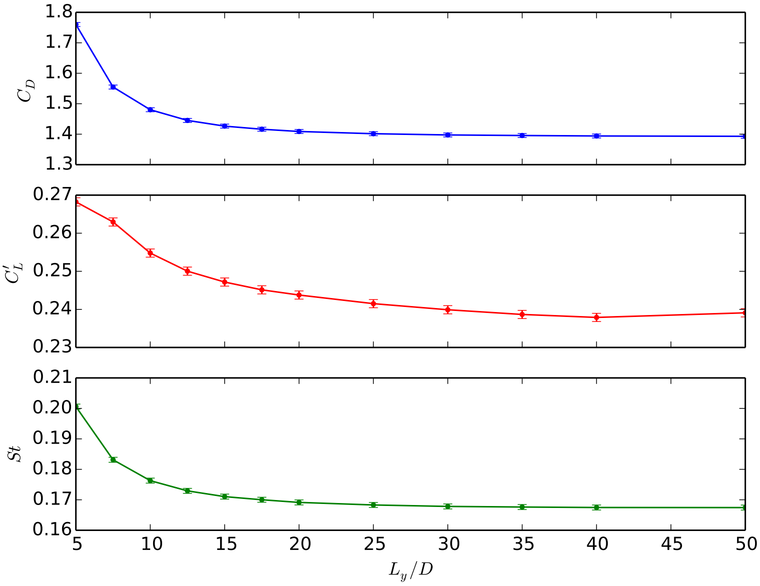


Figure 4: Computed mean drag coefficient, root-mean-square lift coefficient and Strouhal number for varying domain width.

Consider first the variation of system width, where the results are depicted in Fig. 4. These results are computed with $D/\Delta x = 64$, $C_{CFL} = 0.4$, Ma $= 0.1$ and $L_{x,u} = L_{x,d} = 10D$. It is clear that the width of the system has a large impact on the computed coefficients, and it is the first parameter this far that has more than a modest impact on the Strouhal number. This far the system width has been set to $L_y = 10D$. The results in Fig 4 indicates that this is, by far, a too narrow domain to obtain results that quantitatively accurate. However, as the results, in a similar manner as the results from the Mach number variation, drop quite smoothly towards an asymptotic limit, the qualitative behaviour in the system may be satisfyingly resolved with such a domain width.

The computed drag coefficient, lift coefficient and Strouhal number for varied streamwise lengths are depicted Fig. 5, where the blue and red curves represent the results for varied upstream and downstream lengths, respectively. The results are computed with $D/\Delta x = 64$, $C_{CFL} = 0.4$, Ma $= 0.1$ and $L_y = 10D$. When the upstream length is varied

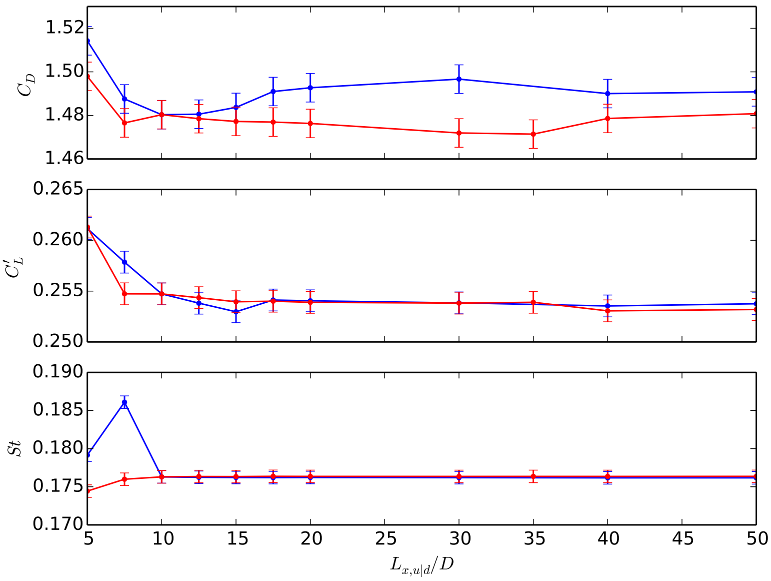


Figure 5: Computed mean drag coefficient, root-mean-square lift coefficient and Strouhal number for varying upstream length (blue) and downstream length (red).

the downstream length is set to $10D$, and vice versa. Hence, the curves overlap exactly at the point where the varied parameter is $10D$. The curves also overlap to a large extent for the computed Strouhal number for $L_{x,u}, L_{x,d} \geq 10D$, which is robust to changes of the upstream and downstream lengths.

The way the mean drag and rms lift coefficient vary, on the other hand, is not as expected. Both show a significant drop as the upstream and downstream lengths are increased from $5D$ to $10D$. This trend does not, however, extend towards an asymptotic limit, as was the case for increasingly large domain width. Consider the mean drag coefficient for increasingly large downstream lengths (red curve in top window of Fig. 5). The curve shows some oscillations for small values of $L_{x,d}$, before it decreases steadily. For $L_{x,d} > 35D$, however, there is an unexpected increase in the computed drag coefficient. This increase is not only apparent in the results computed at $L_{x,d} = 40D$, there is a further increase for $L_{x,d} = 50D$. The reason for this behaviour, and the similar behaviour for the variations with upstream length, is not obvious. It is probable that it is related to some blocking effect due to the domain being too small for the flow problem, either in the spanwise or the streamwise direction, or both.

Figure 6 depicts the von Kármán vortex street at close time intervals for two different

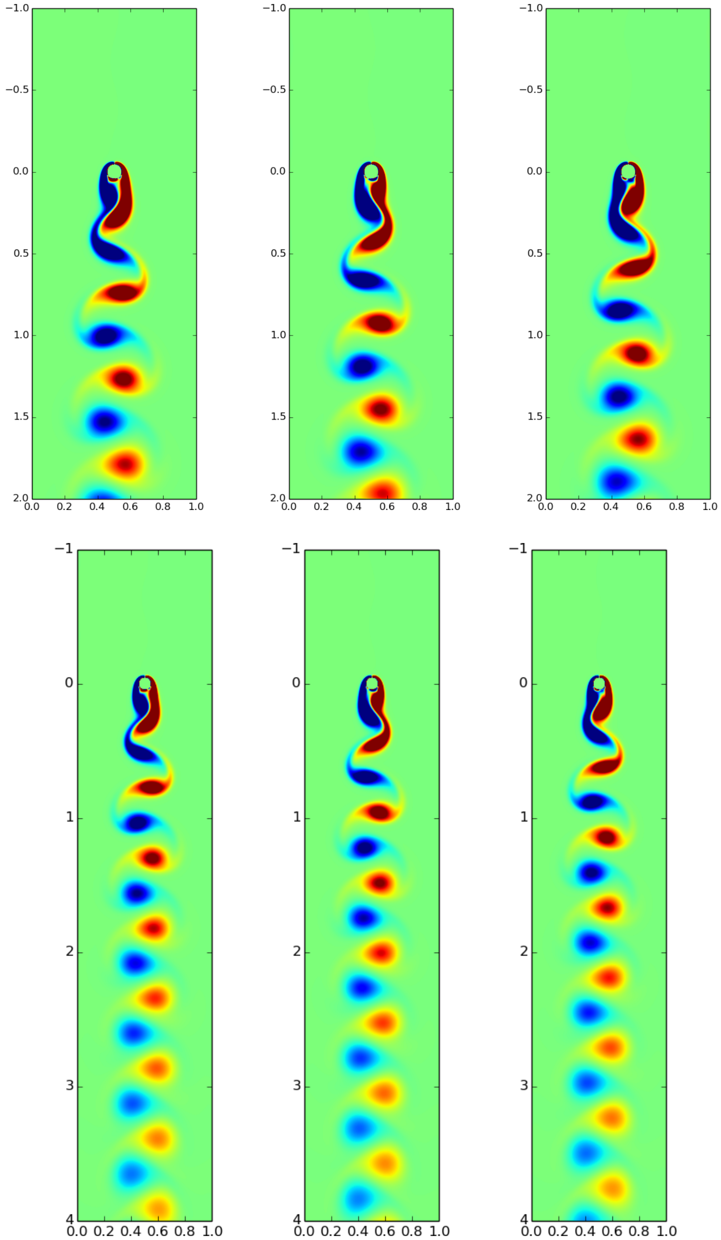


Figure 6: Vorticity component in z-direction for downstream lengths $L_{x,d} = 20D$ (top) and $L_{x,d} = 40D$ (bottom).

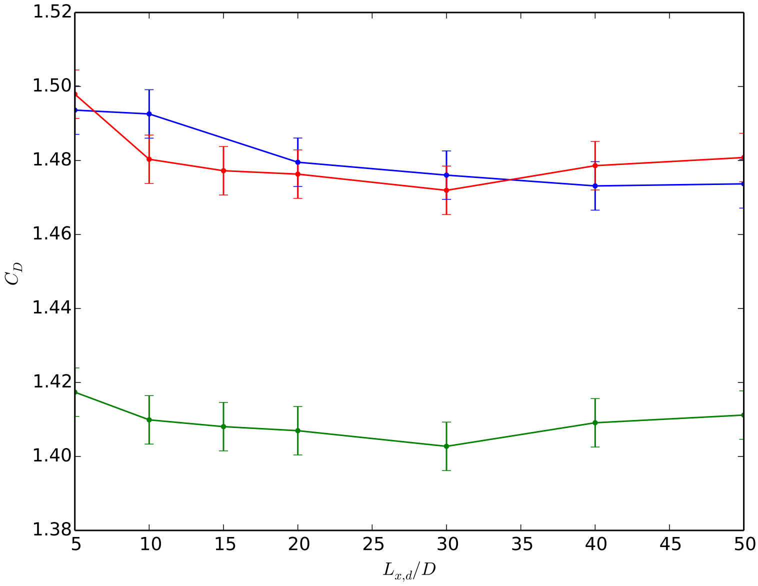


Figure 7: Computed mean drag coefficient for varying downstream length, for three parameter configurations: $L_{x,u} = 10D$ and $L_y = 10D$ (red), $L_{x,u} = 20D$ and $L_y = 10D$ (blue), $L_{x,u} = 10D$ and $L_y = 20D$ (green).

downstream lengths, $L_{x,d} = 20D$ and $L_{x,d} = 40D$. The snapshots do not indicate that the vortices leave the domain differently in the two different simulations and there is no sign of reflections that distort the results. Thus, it is unlikely that the this is the reason for the unexpected results in Fig. 5.

To understand the behaviour of the computed drag coefficient in Fig. 5 a few more validation runs for varying domain size are performed. The downstream length is varied for three different parameter configurations. One with $L_{x,u} = 10D$ and $L_y = 10D$ (the same as before), one with $L_{x,u} = 20D$ and $L_y = 10D$ and one with $L_{x,u} = 10D$ and $L_y = 20D$. The results are seen in Fig. 7.

Several aspects of the domains influence on the computed mean drag coefficient are apparent in the figure. Consider first the difference between the red and the green curves. The red curve is equal to the computed drag for varying downstream length in Fig. 5, while the green curve depicts the results with doubled domain width. The two curves are, qualitatively, very similar; the bottom curve is only shifted downwards by approximately 0.075. This shift, towards a more accurate result for the computed drag coefficient, can also be seen in Fig. 4, when comparing the computed mean drag coefficient for $L_y = 10D$

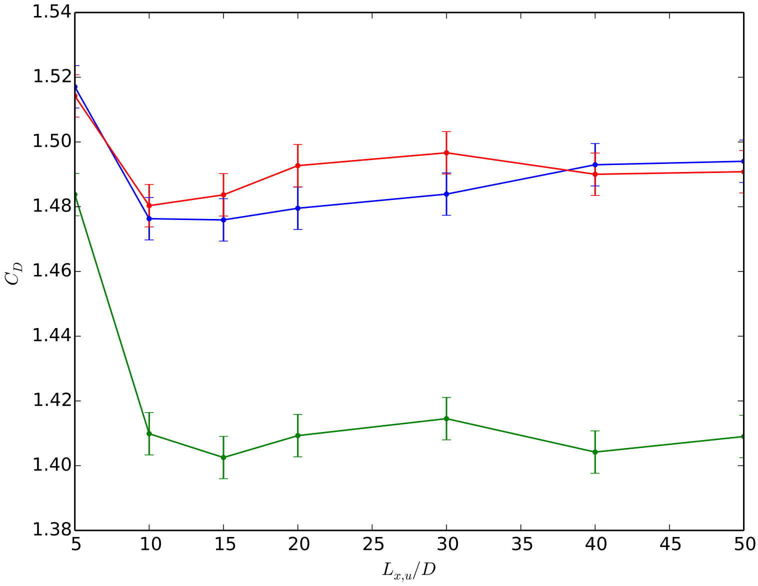


Figure 8: Computed mean drag coefficient for varying upstream length, for three parameter configurations: $L_{x,d} = 10D$ and $L_y = 10D$ (red), $L_{x,d} = 20D$ and $L_y = 10D$ (blue), $L_{x,d} = 10D$ and $L_y = 20D$ (green).

and $L_y = 20D$. This confirms that the blockage effects from a narrow width has a large impact on the quantitative results and a very modest impact on the qualitative results.

Comparing the red curve with the blue curve, for which the upstream length has been doubled shows that a possible reason for the unexpected jagged form of the drag coefficient computed with $L_{x,u} = 10D$ is a blockage effect due to a too short upstream length. This is in accordance with [6] that advise against using domain with smaller upstream lengths than $20D$. The blue curve shows the expected behaviour of dropping towards an asymptotic value for increasingly large downstream lengths. Thus, to achieve the expected qualitative behaviour of increasing the domain in the downstream direction the upstream direction must be larger than some threshold value.

The effects are somewhat similar for the computed drag coefficient with varying upstream lengths, for different L_y and $L_{x,d}$. Figure 8 depict the results for varied upstream length, for three parameter configurations. One equal to the one in Fig. 5 (red), one with $L_{x,d} = 20D$ and $L_y = 10D$ (blue), and one with $L_{x,d} = 10D$ and $L_y = 20D$ (blue). Again, the green and red curves are qualitatively very similar, indicating independent effects of increasing upstream length and domain width. The blue curve is more smooth. Beyond

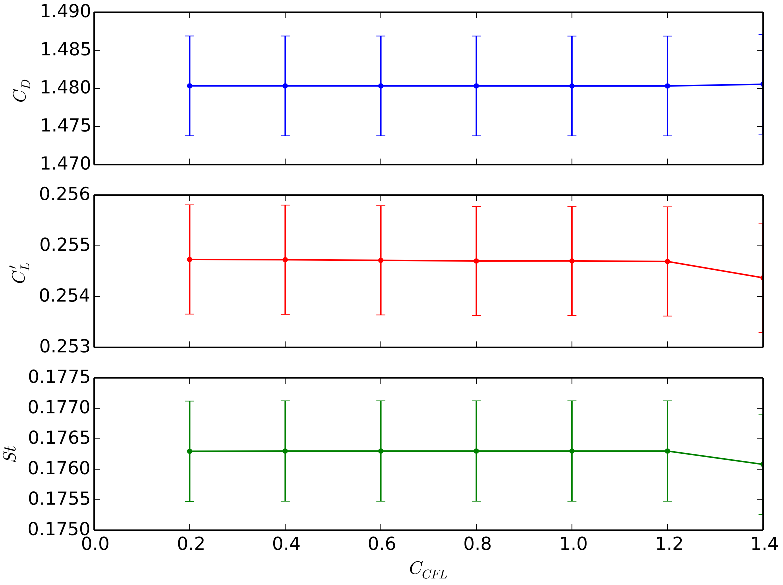


Figure 9: Computed mean drag coefficient, root-mean-square lift coefficient and Strouhal number for different Courant numbers.

the value for the smallest upstream length ($L_{x,u} = 5D$) the computed drag coefficient increases towards a limiting value for increasing upstream length for the configuration with $L_{u,d} = 20D$.

2.4 Courant number

The Courant number is a numerical constant that sets restrictions on the time step (Eq. (5)) to maintain stability of the explicit time discretization. One often tries to avoid a larger Courant number than one, which is related to the stability limit found by von Neumann analysis of the one-dimensional advection equation discretized by the upwind scheme. An even lower Courant number is often used chosen, to ensure that the simulations are well within the stability region. This is not necessarily a good idea, as the time step will become small. Indeed, for higher order discretization schemes a Courant number larger than unity may be permitted (as shown in, e.g., [8] where $C_{CLF} = 1.43$ is the stability criteria).

Figure 9 depict the computed drag and lift coefficients and the Strouhal number for different Courant numbers. The constant parameters are $D/\Delta x = 64$, Ma $= 0.1$, $L_{x,u} =$

$L_{x,d} = 10D$ and $L_y = 10D$. It can be seen that the Courant number has no significant effect on the computed quantities for $C_{CLF} \leq 1.2$. This does not mean that $C_{CLF} = 1.2$ is the optimal choice of Courant number for all simulations with this set-up. If, e.g., forced turbulence is introduced or the Reynolds number is increased such that turbulence is generated in the flow, the small scale structures will have a significant impact on the result. This may influence the choice of Courant number as the Courant number influences the small scale error.

With $C_{CLF} = 1.4$ the computations are close to becoming unstable, and the computed lift and drag show some deviations from sinusoidal behaviour. A run with $C_{CLF} = 1.6$ was also initiated, but did not complete successfully, as instability produced NaNs in the solution.

2.5 Comparison to data sets from previous studies

The impact of variation of different parameters of interest has been studied in detail. To conclude the validation part of this paper a few runs for a large domain have been computed, and are compared to previous studies. Table 1 show computed values for mean drag coefficient, rms lift coefficient and Strouhal number, from several different previous studies as well as the present one. The extent the previous studies are comparable to the present one varies from study to study. All studies are assumed grid independent, and only the domain size is given for comparison. While some studies use the rms lift as a control parameter, others use the amplitude of the lift. This amplitude has been scaled by 0.707 to an approximate rms-value. This may be done as the lift coefficient is a smooth sinusoidal-like function with zero mean value.

The present results have been computed on a rectangular domain of width $L_y = 60D$, upstream length $L_{x,u} = 20D$ and downstream length $L_{x,d} = 40D$. The flow is compressible with Mach number Ma $= 0.1$ and the Courant number is set to $C_{CLF} = 1.0$. The grid is equidistant with grid spacing $\Delta x = D/60$. In addition, two numerical parameters have been altered, as compared to the previous runs. One is a scaling parameter used in the computations of the drag force and lift force, and one is a parameter related to the fluid points very close to the cylinder surface. The latter (*linear_close_interpolate*) will be discussed further in Sec. 3. Default values in the software have been used for these parameters, this far. The change is made before computing the runs for the large domain and is based on previous experience with high accuracy runs for this flow problem.

The studies in Tab. 1 represent a wide spread of computational fluid dynamic methods for computing the flow past a cylinder. The top seven and the present study use the immersed boundary method to represent the cylinder in the flow, yet they use quite different domain sizes. The eight remaining do not use the immersed boundary method, but different body conformal grids. Finite-volume, finite-difference, finite element, spectral element and lattice-boltzmann methods are represented in the table. Only [14], [19] and the present study are for compressible flows, while the remaining are computations for incompressible flows. The study [14] uses the Pencil Code, the same software that is used

Table 1: Comparison with data sets from previous studies. Root-mean-square lift coefficients marked by a superscript star ($^{(*)}$) denote lift coefficient amplitudes scaled to root-mean-square values. The non-rectangular grids are marked as circular inlet/C-type (◖) or circular/O-grid (○). Domains where the cylinder is not centred have both upstream and downstream length given.

	$[(L_{x_u} + L_{x_d}) \times L_y]/D^2$	C_D	C'_L	St
Lai & Peskin [9]	$(6.1 + 20.5) \times 26.6$	1.4473	$0.233^{(*)}$	0.165
Kim, Kim & Choi [10]	70×100	1.33	$0.22^{(*)}$	0.165
Su, Lai & Lin [11]	$(13.4 + 16.5) \times 16.7$	1.40	$0.240^{(*)}$	0.168
Pan [12]	60×60	1.32	$0.226^{(*)}$	0.16
Tseng & Ferziger [3]	32×16	1.42	0.29	0.164
Noor, Chern & Horng [13]	–	1.4	–	0.167
Haugen & Kragset [14]	70×35	1.328	–	0.166
Park et al. [15]	$(50 + 20) \times 100$, ◖	1.33	$0.235^{(*)}$	0.165
Shi et al. [16]	100, ○	1.331	–	0.1650
Mittal [17]	100×100	1.322	0.226	0.1644
Stålberg et al. [18]	160, ○	1.32	$0.233^{(*)}$	0.166
Posdziech & Grundmann [6]	$(20 + 50) \times 40$ ◖	1.3504	$0.234^{(*)}$	0.1667
Posdziech & Grundmann [6]	$(40 + 50) \times 80$ ◖	1.3321	$0.229^{(*)}$	0.1650
Li et al. [19]	100×100	1.336	–	0.164
Qu et al. [5]	60×60	1.326	0.2191	0.1660
Present	$(20 + 40) \times 60$	1.334	0.227	0.1658

in the present study, but with a different domain size and resolution. The studies [5] and [6] are intended to compute the coefficients to high accuracy, for very large domains, meant to give qualitative results useful for, e.g., benchmarking purposes. Only the domains most similar to the domain used in the present study are included in Tab. 1. Two of the domains in [6] are close to the present domain size, and both are included in the table.

It is seen that the computed results are in very good agreement with previous studies. A few of the studies using the immersed boundary method are for smaller domains than the high accuracy runs in this study. These results compare well with the results for similar domain sizes, as depicted in Figs. 4–7. The validations performed does not only show how the different parameters affect the solution, but also that the solution is in good agreement with literature.

The results computed with the Pencil Code can be regarded as highly accurate. The results are, however, achieved by using a very fine grid, something that may prove troublesome if the computed case is to be generalized to three dimensions. The critical point when regarding the grid spacing in the set-up is the resolution of the points closest to the cylinder surface. Therefore, the way the fluid equations are solved here is studied in detail to find possible improvements to the boundary layer representation.

3 IMMERSED BOUNDARY METHOD

The immersed boundary method was introduced by Peskin in the 1970s to model flow around heart valves [20]. Today it is a class of methods that represent a boundary immersed in a flow with non-body conformal grids [2]. As the grid does not conform to the solid boundary, incorporating the boundary conditions require a modification to the equations in the vicinity of the boundary. In the Pencil Code, this is done by what is called a discrete forcing approach. In this way of implementing the immersed boundaries the flow equations are first discretized on a Cartesian grid without regard to the immersed boundary, and then forcing terms that represent the boundaries' effect on the flow are introduced. For details on the grid representation close to the boundaries the reader is referred to [14]. Only the main outline will be repeated here.

3.1 Orthogonal mirror point method

To avoid special handling of the grid points close to the solid surface, a ghost-cell zone is constructed inside the solid surface. The ghost-cell zone is three grid points deep, contrary to the much used single ghost-point in discrete forcing immersed boundary methods (see, e.g., [3]). Three points are necessary to make the sixth order central differencing scheme applicable at the grid points close to the solid surface as well as far from the boundaries. The ghost-cells are given values by using mirror points outside the solid surface. Figure 10 illustrates the relation between mirror points and ghost points for three ghost points.

The boundary conditions at the cylinder surface is no-slip and impearmeability for the velocity, and zero gradient in the radial direction for the density. The way the mirror points are set up using lines orthogonal to the interface makes satisfying these boundary conditions fairly straightforward. However, the distance from a computed fluid point close to the immersed boundary to the mirror points used to generate ghost points necessary to compute the flow variables at the fluid point can be large. This may have a negative influence on the accuracy of the computations. This effect will be greatest on the grid points nearest to the cylinder surface, as they make use of three grid points inside the solid geometry. This is illustrated in Fig 10, where the relation between the fluid point a close to the cylinder surface and the position of the mirror point used to set the value at the ghost points d can be seen. It is possible to handle these points as exceptions. A parameter *limit_close_linear* in the Pencil Code lets the user choose the distance from the cylinder that qualifies for special handling – using interpolation between the surface and the neighbouring point to compute the flow variables, rather than the high order finite difference method (detail in [14]). The default distance is half a grid cell (thus fluid point a in Fig. 10 would be handled as an exception). This distance is used in all the validation runs in Sec. 2, except for the simulations compared with previous studies (Tab. 1), where a larger distance has been used (this gave favourable results).

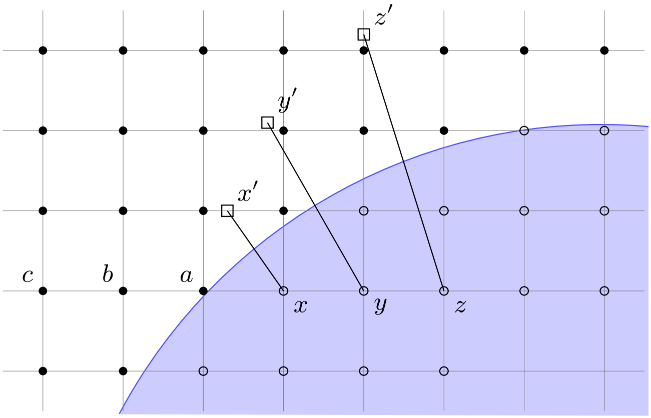


Figure 10: Ghost points set by corresponding mirror points at orthogonal distance from the solid surface. Fluid points (a,b,c; ●), ghost points (x,y,z; ○) and a mirror points (x',y',z'; □).

3.2 Cartesian mirror point method

A new implementation of the immersed boundary method in the Pencil Code is suggested. The method makes use of the Cartesian grid by setting local mirror points along the grid lines. This minimises the distance from a fluid point to the mirror points used when setting the ghost-cell values. The method is quite straightforward for the velocity components in the flow, but requires special handling of the density function to satisfy a zero gradient in the radial direction. In addition some ghost points will have to be computed several times – once for each direction for which they are to be used. The concept is illustrated in Fig. 11.

Consider a computation of the fluid point b. The ghost points x and y are needed (in addition to three fluid points to the right and the left fluid point a) in the 6th order central differencing scheme. The mirror points used in the computation of the fluid velocities in the ghost-points x and y are x' and y', respectively. It can be seen that the mirror points x' and y' are much closer to b than the corresponding mirror points in the orthogonal mirror point method (Fig. 10). This is expected to reduce the error in the ghost point representation in the central differencing scheme, as behaviour of the fluid velocities at the ghost points are more strongly coupled to the fluid point velocity if the mirror points are in close proximity to said grid point.

The fluid velocity vector in the mirror points is computed by interpolation between the two nearest grid points and the surface point on this grid line. Hence, when computing x' in Fig. 11 the points b and a are used, as well and the surface point to the right of

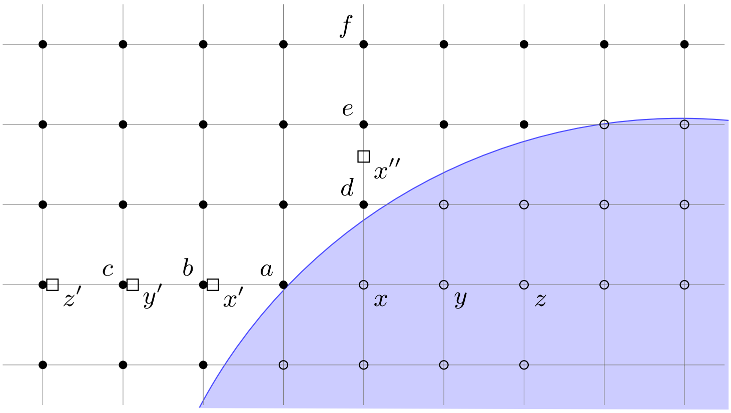


Figure 11: Ghost points set by corresponding mirror points along the grid lines in x- and y-direction. Fluid points (a,...,f; ●), ghost points (x,y,z; ○) and a mirror points (x',y',z',x''; □).

a. When computing y' points c, b and the surface point to the right of a is used in the interpolation, etc. The surface points can be used for the velocity interpolation since non-slip and impearmeability conditions at the surface require all velocity components to be zero there. Such conditions do not apply to the density, for which all that is known is that the radial component of the density gradient is zero at the surface. Due to this, the new immersed boundary method is more of a hybrid method than a pure Cartesian grid method. The orthogonal mirror points shown in Fig. 10 are constructed in this scheme as well, yet only to set the density in the ghost-points.

The mirror point x'' that is included in Fig. 11 is used when computing the velocity vector in x for computation of the y-components of d, e and f. This drawback, the necessity to compute the value in some ghost points several times, is not expected to have a large impact on the computational cost of the immersed boundary implementation.

This new immersed boundary implementation is expected to improve the accuracy of the computed flow variables, such that high accuracy results can be computed with a coarser grid than that used in the validation runs. If these expectations are met, this will not only negate the added cost of decomposition and additional ghost points computations, but reduce the overall simulation time significantly.

3.3 Results

Figure 12 depict mean drag coefficient, rms lift coefficient and Strouhal number for varying grid size computed with the two different implementations of the immersed bound-

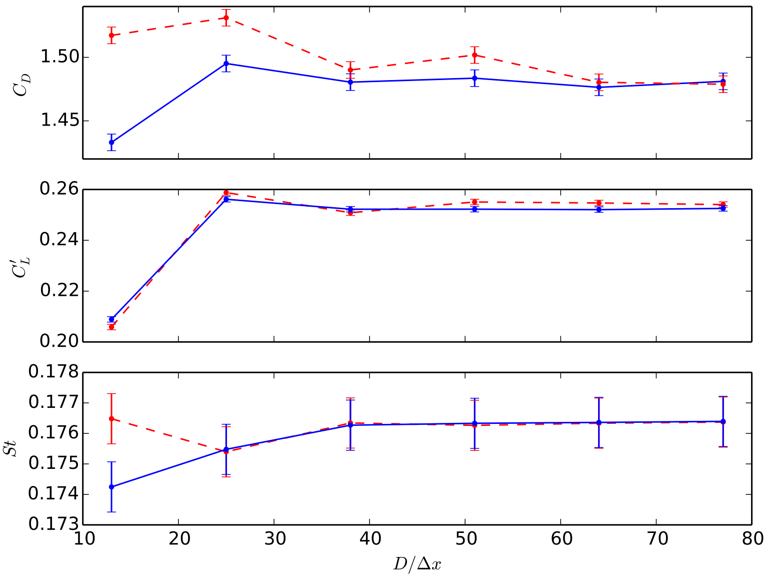


Figure 12: Computed mean drag coefficient, root-mean-square lift coefficient and Strouhal number for different different grid spacing. Two different implementations of the immersed boundary method have been used to get the results. The Cartesian mirror point method (blue) and the orthogonal mirror point method (red, dashed).

ary method in the Pencil Code. The blue and red curves show the results for the Cartesian mirror point method (see Sec. 3.2) and the orthogonal mirror point method (see Sec. 3.1), respectively. The computations have been carried out with $L_y = 10D$, $L_{x,u} = L_{x,d} = 10D$, $Ma = 0.1$ and $C_{CFL} = 0.9$. The scaling parameter and the parameter related to the treatment of fluid points close to the solid surface (both mentioned in Sec. 2.5) have been set to the same values as those used in Sec. 2.1–2.4.

It is seen that performing a grid refinement study with the new implementation of the immersed boundary method yields less oscillatory behaviour when going from one grid size to the next. Further, it can be seen that all three computed quantities, C_D, C'_L and St reach a point where decreasing the grid spacing has only a small effect. As a matter of fact, the variations in the computed results, for all three quantities, is within the error bounds for $D/\Delta x \geq 38$. This is a substantial improvement over the grid refinement performed in Sec. 2.1, indicating that the new immersed boundary method does indeed yield more accurate ghost points within the cylinder. The extra overhead in the computations, due to some ghost points being computed several times, has a negligible impact on the computational time compared to increasing the grid spacing by a factor of 1.7, when going from $D/\Delta x = 64$ to $D/\Delta x = 38$ (recall that the grid spacing not only affects the spatial resolution, but also the time step used in the computations).

4 CONCLUSION

Thorough validation for the case of computing flow past a stationary cylinder has been conducted. In the validations, the effect of the grid spacing, Mach number, domain size and Courant number on the computed mean drag coefficient, root-mean-square lift coefficient and Strouhal number was studied.

Out of the different parameters tuned in during the validation, only the grid spacing and the domain size behaved somewhat unexpected. The computed results showed an oscillatory behaviour for different grid sizes, resulting in the choice of a quite fine grid for the remaining validation runs ($D/\Delta x = 64$). When considering the domain size, the upstream and downstream lengths showed dependent behaviour. This indicated that a minimum length in the streamwise direction is not itself sufficient in computing high accuracy results, the cylinder placements in the domain must be such that the upstream and downstream lengths are both sufficiently large ($> 20D$). Of the remaining parameters, both the domain width and the Mach number indicated smooth decrease towards an asymptotic value for the computed quantities. The Courant number had very little effect on the computed values, for $C_{CFL} \leq 1.2$.

The results computed with the Pencil Code compared very well to previous studies. The computations of high accuracy results were, however, very computationally costly, on the fine grid used in the computations. This motivated a closer look at the flow computations closest to the solid cylinder, where the immersed boundary method with a ghost-zone is used in the Pencil Code.

A new implementation of the immersed boundary method was suggested. The method improves on the way the ghost-points inside the cylinder are computed, by making use of the grid lines in computing the fluid flow on corresponding mirror points. Such an improvement greatly reduced the oscillatory behaviour seen in the grid refinement results, indicating higher accuracy of the boundary layer around the cylinder. The increased accuracy allows for an increase in the grid spacing of a factor of 1.7 with negligible impact on the accuracy.

The presented study lays a solid foundation for further work on flows past a circular cylinder using the high order finite difference Pencil code. As the impact of many central parameters is well understood, the choice of these parameters – which is always a compromises between accuracy and computational cost – can be made without too much guesswork and hand-waving arguments. Further, the improved accuracy close to the solid surface will be beneficial in the authors' further study with this set-up, and for other users of the open source code that include immersed solid objects in their flow simulations.

Acknowledgements

The research leading to these results has received funding from the Research Council of Norway under the FRINATEK Grant 231444.

REFERENCES

[1] The Pencil Code. http://www.nordita.org/software/pencil-code (2015).

[2] Mittal, R. and Iaccarino, G. Immersed Boundary Methods. *Annu. Rev. Fluid Mech.* (2005) **37**:239–261.

[3] Tseng, Y.-H. and Ferziger, J. H. A ghost-cell immersed boundary method for flow in complex geometry. *J. Comput. Phys.* (2003) **192**:593–623.

[4] Poinsot, T. J. and Lele, S. Boundary conditions for direct simulations of compressible viscous flows. *J. Comput. Phys.* (1992) **101**:104–129.

[5] Qu, L., Norberg, C., Davidson, L., Peng, S.-H. and Wang, F. Quantitative numerical analysis of flow past a circular cylinder at Reynolds number between 50 and 200. *J. Fluids Struct.* (2013) **39**:347–370.

[6] Posdziech, O. and Grundmann, R. A systematic approach to the numerical calculation of fundamental quantities of the two-dimensional flow over a circular cylinder. *J. Fluids Struct.* (2007) **23**:479–499.

[7] Pletcher, R. H., Tannehill, J. C. and Anderson, D. *Computational fluid mechanics and heat transfer.* CRC Press (2012).

[8] Moin, P. and Mahesh, K. DIRECT NUMERICAL SIMULATION: A Tool in Turbulence Research. *Annu. Rev. Fluid Mech.* (1998) **30**:539–578.

[9] Lai, M.-C. and Peskin, C. S. An Immersed Boundary Method with Formal Second-Order Accuracy and Reduced Numerical Viscosity. *J. Comput. Phys.* (2000) **160**:705–719.

[10] Kim, J., Kim, D. and Choi, H. An Immersed-Boundary Finite-Volume Method for Simulations of Flow in Complex Geometries. *J. Comput. Phys.* (2001) **171**:132–150.

[11] Su, S.-W., Lai, M.-C. and Lin, C.-A. An immersed boundary technique for simulating complex flows with rigid boundary. *Comput. & Fluids* (2007) **36**:313–324.

[12] Pan, D. An Immersed Boundary Method on Unstructured Cartesian Meshes for Incompressible Flows with Heat Transfer. *Numer. Heat Transfer, Part B: Fundamentals* (2006) **49**:277–297.

[13] Noor, D. Z., Chern, M.-J. and Horng, T.-L. An immersed boundary method to solve fluidsolid interaction problems. *Comput. Mech.* (2009) **44**:447–453.

[14] Haugen, N. E. L. and Kragset, S. Particle impaction on a cylinder in a crossflow as function of Stokes and Reynolds numbers. *J. Fluid Mech.* (2010) **661**:239–261.

[15] Park, J., Kwon, K. and Choi, H. Numerical solutions of flow past a circular cylinder at Reynolds numbers up to 160. *KSME Int. J.* (1998) **12**:1200–1205.

[16] Shi, J. M., Gerlach, D., Breuer, M., Biswas, G. and Durst, F. Heating effect on steady and unsteady horizontal laminar flow of air past a circular cylinder. *Phys. Fluids* (2004) **16**:4331–4345.

[17] Mittal, S. Excitation of shear layer instability in ow past a cylinder at low Reynolds number. *Int. J. for Numer. Methods Fluids* (2005) **49**:1147–1167.

[18] Stålberg, E., Brüger, A., Lötstedt, P., Johansson, A. V. and Henningson, D. S. High order accurate solution of flow past a circular cylinder. *J. Sci. Comput.* (2006) **27**:431–441.

[19] Li, Y., Zhang, R., Shock, R. and Chen, H. Prediction of vortex shedding from a circular cylinder using a volumetric Lattice-Boltzmann boundary approach. *Eur. Phys. Journal: Special Top.* (2009) **171**:91–97.

[20] Peskin, C. S. Flow patterns around heart valves: A numerical method. *J. Comput. Phys.* (1972) **10**:252–271.

Paper II [4]

Numerical investigation of free-stream turbulence effects on the transition-in-wake state of flow past a circular cylinder

Jørgen R. Aarnes, Helge I. Andersson and Nils E. L. Haugen

Journal of Turbulence, Vol 19, pp. 252–273, 2018.

JOURNAL OF TURBULENCE, 2018
VOL. 19, NO. 3, 252–273
https://doi.org/10.1080/14685248.2017.1418085

Taylor & Francis
Taylor & Francis Group

Numerical investigation of free-stream turbulence effects on the transition-in-wake state of flow past a circular cylinder

J. R. Aarnes[a], H. I. Andersson[a] and N. E. L. Haugen[a,b]

[a]Department of Energy and Process Engineering, Norwegian University of Science and Technology, Trondheim, Norway; [b]SINTEF Energy Research, Trondheim, Norway

ABSTRACT

We investigate how a turbulent free-stream (TFS) affects the transition-in-wake state of flow past a cylinder. Direct numerical simulations of a decaying TFS flow past a cylinder have been performed, where the intensity and integral scale of the TFS was varied. Distinct effects of the TFS are observed for Reynolds numbers in the lower (Re $<$ 190) and upper (Re $\geq$ 250) ranges of the transitional regime. For a flow with a laminar free-stream (LFS), the onset of the transition-in-wake is observed at critical Reynolds number 190, where the first three-dimensional effects develop in the wake (mode *A* instabilities). A TFS perturbs the flow, forcing the onset of the transition at a lower Reynolds number and inhibiting a sharply defined critical Reynolds number. Quasi-stable states, where the wake alternates between two- and three-dimensional vortex shedding, are observed for Re $<$ 190. These states are closely related to intermittent vortex dislocations in the wake. In the upper Re part of the transition-in-wake regime, mode *B* instabilities dominate the wake of the LFS flow. A TFS stimulates the existence of mode *A* instabilities, resulting in mixed *A*–*B* instabilities in the wake for Re $\geq$ 250. This effect correlates with the turbulence intensity.

ARTICLE HISTORY
Received 28 June 2017
Accepted 28 November 2017

KEYWORDS
Isotropic turbulence; chaos and fractals; direct numerical simulation

1. Introduction

The unsteady vortex shedding and the transition to turbulence in flows past a circular cylinder have been an actively studied field of fluid mechanics during the last century. The flow problem has a generic nature and direct relevance to engineering applications.

The flow past a circular cylinder is steady at sufficiently low Reynolds numbers. The Reynolds number of the flow is defined as $\mathrm{Re} = \bar{U}D/\nu$, where $\bar{U}$ is the mean velocity of the flow, D is the diameter of the cylinder and ν is the kinematic viscosity. At Re $\simeq$ 47 the flow becomes unsteady by a Hopf bifurcation (see, e.g. [1]), which can be observed as oscillations in the wake of the cylinder. As the Reynolds number is increased, the characteristic von Kármán vortex street develops. The flow pattern of alternating vortices being shed from the cylinder is purely two-dimensional. With a further increase in the Reynolds number, the first three-dimensional effects can be observed in the flow. At this point, the flow past

CONTACT J. R. Aarnes jorgen.r.aarnes@ntnu.no
This article was originally published with errors. This version has been corrected/amended. Please see Erratum (http://dx.doi.org/10.1080/14685248.2018.1446614)

the cylinder is in the transition-in-wake state, which spans the range of Reynolds numbers: $(180 - 200) < \mathrm{Re} < (350 - 400)$ [2,3]. Following the transition-in-wake state of the flow, increasing Re leads to further break-up of the flow through several transitional regimes, before it can be characterised as fully turbulent.

The focus of this study is on the transition-in-wake state of the flow, and the effects that a turbulent free-stream (TFS) may have on this particular transition. Although it was noted above that this transition starts at $180 < \mathrm{Re} < 200$, there is considerable variation in the reported critical point where this transition first occurs (see [3], and references therein). We will henceforth denote this point the critical Reynolds number, abbreviated as Re_c for flow with laminar free-stream (LFS) and $\mathrm{Re}_{c,\,\mathrm{TFS}}$ for the TFS case. This is a *secondary instability* of the flow (opposed to the *primary instability* at $\mathrm{Re} \simeq 47$ [4]), and other authors may use Re_2, $\mathrm{Re}_{\mathrm{tr}}$, etc. to denote the onset of this instability. We will not consider the primary instability any further and therefore find Re_c to be an appropriate notation for the Reynolds number at onset of the transition-in-wake state of the flow.

By Floquet stability analysis, Barkley and Henderson [4] could identify the onset of transition at $\mathrm{Re}_c = 189$ and Henderson and Barkley [5] further proved that the transition is subcritical, explaining the hysteresis effect observed experimentally [6]. For $\mathrm{Re} \gtrsim 189$, three-dimensional modes with spanwise length of approximately four cylinder diameters develop in the flow. These mode A instabilities [6,7] are the dominant flow features in the spanwise direction in the first part of the transition-in-wake regime, along with spot-like *vortex dislocations*. The vortex dislocations are large-scale intermittent structures that grow downstream of the cylinder. These dislocations were discovered experimentally by Williamson [8], and similar features have been reproduced in numerical simulations (see [9,10]). Zhang et al. [9] call these structures vortex adhesion, due to vortices evidently adhering to the cylinder over many shedding periods, and found them to be self-sustained in the range $160 < \mathrm{Re} < 230$. Henderson [10] points out that the spot-like dislocations must be generated by the mode A instability, and concludes that a nonlinear interaction between self-excited modes in the A-band is responsible for the appearance of large-scale structures in the wake. Here, the A-band refers to the different possible wavelengths of the mode A instability. The observed spanwise wavelength of the three-dimensional pattern developing in the wake corresponds to the most unstable wavelength of the A-band. The wavelength is $\lambda_A \approx 4D$ at Re_c, decreasing somewhat as Re is increased (details in [4], Figure 12). Through high accuracy numerical simulations, Posdziech and Grundmann [11] reproduced the transition-in-wake with excellent agreement with experimental results. By using a spanwise domain length equal to the most unstable mode A instability the authors claim to exclude vortex dislocations in the wake, and conclude therefore that the role of such dislocations are overrated in experimental studies of the transitional regime.

When the Reynolds number is further increased, a new mode of instabilities develops in the wake: mode B instabilities. These instabilities are streamwise structures with spanwise length of approximately one cylinder diameter, dominating the flow at $\mathrm{Re} \gtrsim 260$ (see [7] and references therein). At Reynolds numbers between 210 and 220 the mode A and mode B instabilities start to coexist in the wake. The transition from a flow dominated by one mode to dominance of the other is gradual, with energy in the flow shifting continuously from the larger to the smaller instabilities (see, e.g. [12]) over a range of Reynolds numbers. Note that unlike the transition where mode A instabilities first occur, the second transition in the transition-in-wake state of the flow is supercritical [12].

The onset of the three-dimensional transition-in-wake can be identified by a sharp drop in the Strouhal number as Re is increased. The Strouhal number is defined as St $= fD/\bar{U}$, where f is the vortex shedding frequency. The drop in St is observed at the onset of the subcritical instability where the first mode A instabilities develop in the cylinder wake. As mode B instabilities start to develop alongside of the mode A instabilities, the shedding frequency is gradually increased towards that of the two-dimensional shedding state. As already mentioned, there is a considerable scatter in the reported critical Reynolds number where this St drop occurs. This is especially prominent in experimental results, which can be seen by comparing data from e.g. Gerrard [13], Williamson [6] and Norberg [14], where Re_c-values between 140 and 200 are reported. This point should be kept in mind when studies where the effect of free-stream turbulence upstream of the cylinder in the transitional regime are considered. The effect of TFS may have been overrated by some authors and underrated by others in previous studies, due to other disturbances (e.g. contamination from end conditions).

In studies performed in two different wind tunnels, Bloor [15] found a significant effect of TFS on the critical Reynolds number. The experiments were performed with low turbulence intensity, $T_i = u_{\text{rms}}/\bar{U} = 0.03\%$ and 1% (where u_{rms} is the root-mean-square value of the three-dimensional velocity fluctuations). Instabilities developed in the wake for Re between 160 and 200, where the higher Re consistently corresponded to the experimental set-up with the lowest turbulence intensity. How large the contribution of, e.g. end conditions was on the onset of transition in these experiments is hard to say, but it cannot be ruled out as an influencing parameter in the experiments, since the different turbulence intensities correspond to different experimental rigs in this study. Hussain and Ramjee [16] studied the effect of TFS on the shedding frequency at Reynolds numbers in the range of the von Kármán vortex street. The experiments were performed for $60 < \text{Re} \lesssim 160$ and T_i up to 8%. No effect of the TFS on the flow was found. This result has been used to argue that the transition-in-wake state of the flow is insensitive to TFS [17]. This is not a valid argument, since Hussain and Ramjee [16] only considered Re below the Re_c reported in the majority of studies on the subject. In a study by Norberg [18], flow past a cylinder for a large Re span was investigated, and the flow was deemed 'rather insensitive' to low intensity, free-stream turbulence ($T_i = 1.4\%$) for $50 < \text{Re} < 10^3$. Norberg [18] did, however, find an increase in the relative bandwidth of Re spanning the transitional regime, thereby supporting the observations by Bloor [15].

Beyond the investigations mentioned here, there exist few studies on the effect of TFS on the transition-in-wake state of the flow past a circular cylinder. To the best of our knowledge, no such studies have been performed using direct numerical simulations (DNS) as the main research tool. An advantage of DNS is that the end conditions, turbulence intensity, etc. can be kept exactly the same in all the simulations. This makes it an ideal tool for further investigations of the effect of TFS on the transition-in-wake, since the effects of the turbulence can be isolated from other disturbances of the flow.

In this paper, a DNS study on the effect of TFS on the transition-in-wake regime of a circular cylinder is presented. The aim of the study is to determine if and how the transition with TFS differs from a transition with LFS. The entire transition-in-wake regime is considered, by using $120 \leq \text{Re} \leq 350$. The structure of the paper is the following: in Section 2, the framework and validation of the numerical simulations are described. Section 3 is a presentation of the main results, with the discussion split into the different Reynolds number spans where effects from the TFS are seen. In Section 4, TFS flows with different turbulence properties (intensity, scale and length of the turbulence production domain) are considered, before conclusions are drawn in Section 5.

2. Methodology

To perform the simulations, the high-order finite-difference code for compressible hydrodynamic flows known as The Pencil Code (see [19,20]) has been used. The governing equations of the flow are the continuity equation:

$$\frac{\mathrm{D}\rho}{\mathrm{D}t} = -\rho\nabla\cdot\boldsymbol{u}\,, \tag{1}$$

and the momentum equation:

$$\rho\frac{\mathrm{D}\boldsymbol{u}}{\mathrm{D}t} = -\nabla p + \nabla\cdot(2\mu\boldsymbol{S})\ , \tag{2}$$

where ρ, t, $\boldsymbol{u}$ and p are the density, time, velocity vector and pressure, respectively. The rate of strain tensor is given by

$$\boldsymbol{S} = \frac{1}{2}\left(\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^{T}\right) - \boldsymbol{I}\left(\frac{1}{3}\nabla\cdot\boldsymbol{u}\right), \tag{3}$$

where $\boldsymbol{I}$ is the identity matrix. The pressure is computed by the ideal gas law, $p = c_s^2\rho$, where c_s is the speed of sound. The flow is isothermal and weakly compressible (the Mach number is 0.1 in all simulations).

2.1. Numerical method

The equations are discretised with sixth-order central-differences in space and a third-order Runge–Kutta scheme in time. An equidistant Cartesian mesh is used to simplify the process of inserting turbulence upstream of the circular cylinder. The cylinder is situated in the centre of the flow domain (details in Section 2.3), and an immersed boundary method is used to resolve the cylinder boundary.

The immersed boundary method, introduced by Peskin in the 1970s to model flow around heart valves [21], is a class of methods that represent a boundary immersed in a flow with non-body conformal grids (see review article by Mittal and Iaccarino [22] and references therein). As the grid does not conform to the solid boundary, incorporating the boundary conditions requires a modification of the governing equations in the vicinity of the boundary. In our implementation, a discrete forcing approach is used, in which the boundary conditions at the cylinder are enforced through ghost points inside the cylinder. The advantage of this approach is that it allows direct control over the numerical accuracy, stability and discrete conservation properties of the solver.

The seven-point finite-difference stencil used for the sixth-order method is not altered near the surface. Rather, a three-point deep ghost point zone is constructed inside the

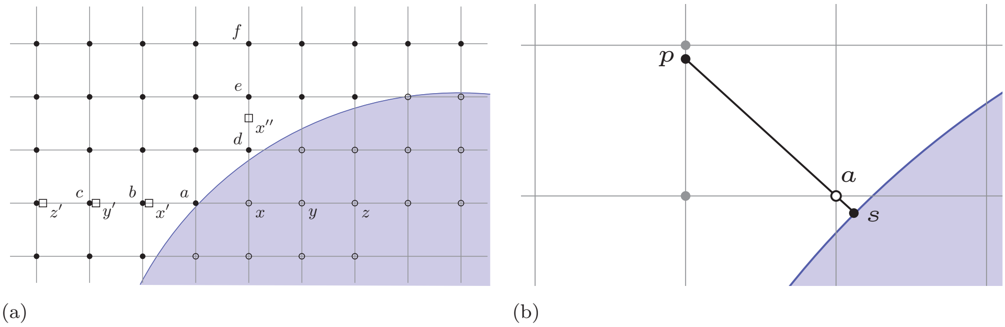


Figure 1. Immersed boundary method. Ghost points (x, y, z; ○) used in the central-difference stencils of fluid points (a–c; ●) are set by mirror points (x', y', z'; □). The mirror points are interpolated along grid lines. When (x; ○) is used to compute vertical velocity components of fluid points (d–f; ●) the corresponding mirror point is (x''; □). For a fluid point (a; ○) very close to the cylinder surface, interpolation between a surface point (s; ●) and the intersection between the surface normal and the first grid line at (p; ●) is used. (a) Ghost and mirror points for velocity computation; (b) special handling of fluid points close to the surface.

solid geometry (unlike the more common single ghost point discretisation, see, e.g. [23]). To resolve the boundary conditions set at the cylinder surface, the ghost points are assigned values from corresponding mirror points. These are no-slip and impermeability for the velocity, and zero gradient in the radial direction for the density. The latter condition can be derived from the ideal gas law and the boundary layer approximation $\left(\frac{\partial p}{\partial n} = 0, \text{ where } n \text{ is the wall normal direction}\right)$ for an isothermal flow.

Ghost points and mirror points used for velocity are computed along Cartesian grid lines; see Figure 1(a). Note that ghost point x is set by mirror point x' when used to compute the horizontal velocity component at fluid points a–c, and by mirror point x'' when used to compute the vertical velocity component at fluid points d–f. Ghost points used to resolve the density gradient are set from mirror points in the radial direction, since only the radial gradient is known (details in [24]). Computation along radial lines minimises the distance between a ghost point and its corresponding mirror point. The use of mirror points along grid lines, however, ensures that the mirror points are closer to the fluid points where the corresponding ghost points are used in the finite-difference stencils. This has been found advantageous, as spurious effects that arise due to the effective delocalised dependency in the finite-differences are reduced. Further, interpolation along grid lines is cheap and implementation of higher-order Lagrangian interpolation is trivial. Here, quadratic interpolation is used along the grid lines. The point where a grid line intersects the surface is always included in an interpolation stencil, thus extrapolation is avoided. Tri-linear interpolation is used to compute the density mirror points.

Special handling is used for fluid points very close to the solid surface: if a fluid point is closer to the surface than some predefined cut-off (e.g. $0.7\Delta x$), a value is explicitly assigned to the fluid point based on an interpolation along the surface normal, as shown in Figure 1(b). The fluid point a is computed by linear interpolation of the values at the surface point s and at the grid intersection point p. The intersection point p is itself interpolated from its nearest neighbours along the grid line.

Table 1. Comparison of mean drag coefficient C_D and Strouhal number St for a range of resolutions and Reynolds numbers for a two-dimensional domain with $L_x = 10D$ and $L_y = 20D$.

	Re = 100		Re = 200		Re = 300		Re = 350	
Resolution	C_D	St	C_D	St	C_D	St	C_D	St
160 × 320	1.49	0.174	1.48	0.204	1.53	0.218	1.55	0.223
240 × 480	1.49	0.176	1.46	0.206	1.46	0.206	1.52	0.225
320 × 640	1.46	0.176	1.42	0.206	1.44	0.220	1.45	0.225
400 × 800	1.47	0.176	1.43	0.205	1.46	0.220	1.47	0.224
480 × 960	1.47	0.176	1.43	0.205	1.45	0.219	1.46	0.224
640 × 1280	1.46	0.176	1.43	0.205	1.46	0.219	1.47	0.224
800 × 1600	1.46	0.176	1.43	0.205	1.46	0.219	1.47	0.223

2.2. Grid resolution

Grid refinement has been performed to determine the necessary resolution to capture the flow phenomena in the transition-in-wake. The number of equidistant grid points on a two-dimensional flow domain with dimensions $L_x \times L_y = 10D \times 20D$ was varied from 160 × 320 to 800 × 1600 for Reynolds numbers from 100 to 350. The inflow velocity and cylinder diameter were held constant, while the viscosity was used as a control parameter to set the Reynolds number.

From the grid refinement study, a resolution of $D/\Delta x = D/\Delta y = 40$ (i.e. a 400 × 800 grid) is found to be sufficient for Reynolds numbers spanning the transition-in-wake state of the flow (see Table 1). For simplicity, the same resolution is used for all Reynolds numbers, although a coarser grid would be sufficient at the lower Reynolds numbers. The laminar boundary layer around the cylinder is thinnest at the front stagnation point. This (non-dimensional) boundary layer thickness is $\tilde{\delta} = \delta/D = 2.4\sqrt{\nu/(BD^2)}$, where $B \approx 4\bar{U}/D$ for weakly compressible flows [25]. Hence, with dimensionless grid spacing $\Delta\tilde{x} = \Delta\tilde{y} = 0.025$, the boundary layer is at minimum 4.8 grid points deep for Re = 100 ($\tilde{\delta} = 0.12$) and 2.6 grid points deep for Re = 350 ($\tilde{\delta} = 0.064$). In practice, this guarantees that at most one of the grid points used for interpolation of the mirror point farthest from the surface (z' in Figure 1(a)) is (barely) outside the boundary layer for Re = 350. All other interpolation grid points, and all mirror points, are guaranteed to be inside the boundary layer at this resolution. This is due to the special handling of grid points closer to the surface than $0.7\Delta x$, ensuring that a ghost point is not used if it is farther than $2.3\Delta x$ from the surface.

The stability requirement for the Runge–Kutta method for a weakly compressible flow imposes a very strict limit on the time step. An adaptable time step is used, which results in the dimensionless time step $\Delta\tilde{t} = \Delta t\bar{U}/D \leq 1.3 \times 10^{-3}$ after the development of the von Kármán vortex street. To verify that this time step is adequate, two-dimensional flow simulations were performed on the 400 × 800 grid, with fixed time stepping. Reducing the time step to $\Delta\tilde{t} \leq 1.0 \times 10^{-4}$ had a negligible impact on the flow simulation, increasing C_D and St only by 0.015% and 0.06%, respectively.

For DNS, it is not sufficient to resolve the boundary layer around the cylinder, all scales of the TFS (spatial and temporal) must also be resolved. Fortunately, at the low to moderate Reynolds numbers spanning the transition-in-wake regime, the Kolmogorov length scale $\eta = (\nu^3/\varepsilon)^{1/4}$, where ε is the average energy dissipation rate, is larger than the necessary resolution required to resolve the boundary layer, for the turbulence considered in this research. The Kolmogorov time scale, $\tau_\eta = (\nu/\varepsilon)^{1/2}$, is substantially larger than the

Table 2. Comparison of the Kolmogorov microscales with spatial and temporal resolution for the TFS with integral scale $\Lambda/D \approx 1$ for Re = 100 and Re = 350. Non-dimensional properties given by $\tilde{\varepsilon} = \varepsilon D/\bar{U}^3$, $\tilde{\eta} = \eta/D$, $\tilde{\tau}_\eta = \tau_\eta \bar{U}/D$. The grid is equidistant with $\Delta x = \Delta y$.

Re	$\tilde{\varepsilon}_{\max}$	$\tilde{\eta}$	$\tilde{\tau}_\eta$	$\eta/\Delta x$	$\tau_\eta/\Delta t$
100	0.0161	0.089	0.788	3.55	606
350	0.0116	0.038	0.497	1.51	382

required temporal resolution imposed by the stability requirement (details are found in Table 2).

2.3. Computational domain

The computational domain consists of two rectangular boxes. In the first box, isotropic turbulence is generated by external forcing on given wave numbers (details in [26]). This domain (left box in Figure 2, henceforth called the turbulence domain) has periodic boundary conditions in all directions and is equal in size to the flow domain (more on this in Section 4.3). Once the forced turbulence is statistically stationary, slices of flow quantities from the turbulence domain are added to the inlet of the flow domain (right box in Figure 2). The flow domain has mean inlet velocity $\bar{\mathbf{U}} = (\bar{U}, 0, 0)$. Since the mean velocity in the turbulent box is zero, this mean inlet velocity is not affected by the added turbulent velocities. Hence, the inlet velocity is $\mathbf{U} = \bar{\mathbf{U}} + \mathbf{u}'$, where $\mathbf{u}'$ is updated at every time step using data from the turbulence domain. To avoid outflows at the inlet, the turbulence intensity is limited such that $\max |u'_x| \leq \bar{U}$. Navier–Stokes characteristic boundary conditions, which is a formulation that uses one-dimensional characteristic wave relations to allow acoustic waves to pass through the boundaries, are used both at the inlet and at the outlet of the flow domain [27]. The boundaries normal to the mean flow direction are periodic.

The size of the flow domain is $(L_x, L_y, L_z) = (12D, 6D, 8D)$. The size of the domain is chosen to be sufficiently large to resolve the flow phenomena qualitatively. Consider the

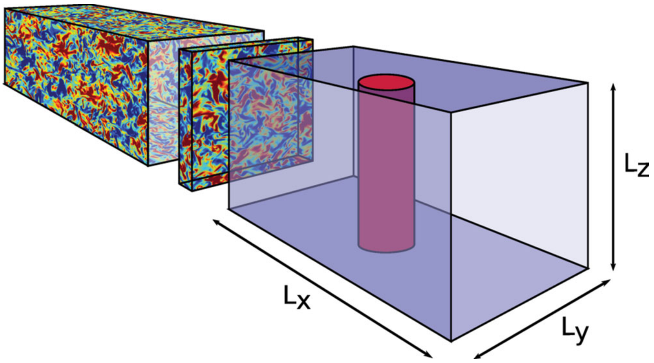


Figure 2. Computational domain, split into two rectangular boxes. Left box for turbulence generation and right box for flow domain. Thin slices of velocity data are taken from the turbulence domain and added to the velocity on the inlet of the flow domain, illustrated here by a thin rectangular box between the two domains. The slice thickness and cylinder diameter are not to scale.

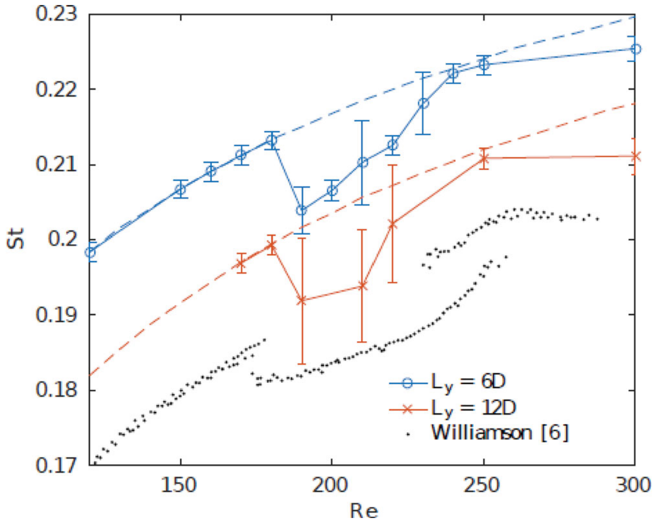


Figure 3. Comparison of St–Re relationships for flows with laminar free-stream conditions, for Reynolds numbers spanning over the transition-in-wake regime. $L_y = 6D$: -o- 3D, _ _ 2D; $L_y = 12D$: -x- 3D, _ _ 2D; • data from experiments by Williamson [6].

solid blue curve with o-markers in Figure 3, depicting the Strouhal number as a function of Reynolds number for the flow in the transitional regime with LFS. The results are compared with experimental results from Williamson [6] (black dots). Results from two-dimensional simulations, where the transition-in-wake does not occur, are also included in the figure (dashed blue line). The qualitative match between the three-dimensional simulations and the experimental results is good, but the quantitative match is poor. A discrepancy between the oblique vortex shedding results by Williamson [6] and the parallel vortex shedding from the DNS is expected, as parallel vortices are shed at a somewhat higher frequency than oblique vortices (see [28]). The mismatch between the experimental and DNS results is, however, too large to be attributed to this effect alone. It is well known that a limited L_y will generate blockage effects in the flow [29,30]. Previous validation runs show that L_y is the domain size parameter that has the greatest influence on the computed Strouhal number [31]. The blockage ratio is $D/L_y = 0.167$, and significantly larger St values are found in the simulations than in the experimental data. The blockage effect, however, does not affect the value of Re_c or the increase in St in the upper Re part of the transition. Doubling L_y, from $6D$ to $12D$, and thus approaching the recommended distance of $8D$ from the lateral boundaries to the cylinder, as suggested by Behr et al. [29], reduces the St values by ≈9% (red curve with x-markers in Figure 3). Since the quantitative value of St is not of interest to us, $L_y = 6D$ is used in the subsequent simulations in order to keep the computational costs manageable. Similarly to Thompson et al. [32], we believe that the restricted domain does not alter the essential physics underlying the development and interaction of the three-dimensional structures. Note that even with this limited domain width, the typical cost of running a simulation resulting in a single point in Figure 3 is 5.4×10^3 CPU hours. For simulations with TFS the typical simulation time is approximately 40% longer.

Two-dimensional simulations with varying upstream and downstream lengths have been performed, and the results showed that a cylinder situated in the centre of a $12D$ long

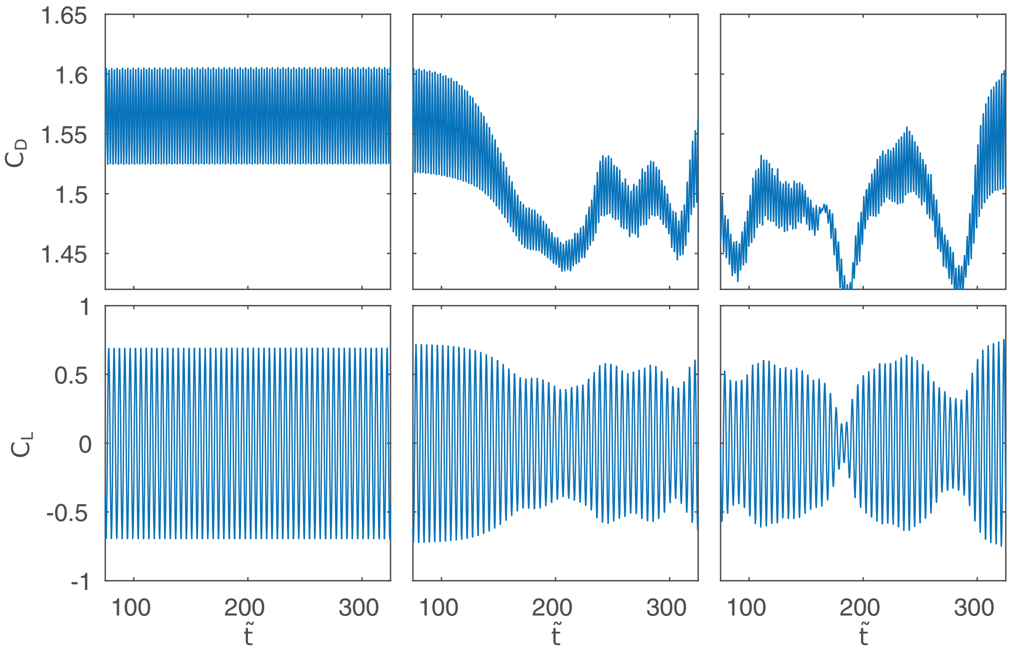


Figure 4. The computed instantaneous drag and lift coefficients, C_D and C_L, respectively, for Reynolds numbers 180 (left), 190 (middle) and 210 (right) as a function of the dimensionless time $\tilde{t} = t\bar{U}/D$.

flow domain gives accurate results for the shedding frequency. The simulations were performed for Re = 100, with upstream and downstream lengths varied up to $50D$ each. The Strouhal number St was barely affected by these lengths, as long as the lengths were above a certain threshold. Results from validation runs with $L_x = 20$ and Re spanning over the transition-in-wake region (with the cylinder in the centre of the domain) are not included in Figure 3 since they are indistinguishable from the corresponding results with $L_x = 12D$.

The spanwise length L_z is chosen sufficiently large to allow room for at least two wavelengths of the mode A instabilities inside the flow domain at Re_c. The choice of the spanwise length is based on the findings of Posdziech and Grundmann [11], that an L_z twice as large as the most unstable wavelength of the mode A instabilities ($\lambda_A \lesssim 4D$) guarantees an accurate reproduction of the measurements.

2.4. Measuring the shedding frequency

There is a large variation in the size of the error bars in Figure 3. This stems from the nature of the shedding of vortices in the transitional regime, particularly for Reynolds numbers where several vortex dislocations occur.

Let us consider as an illustration Figure 4, depicting the drag and lift coefficients for Re = 180, 190 and 210. For the two-dimensional vortex shedding (Re = 180) the coefficients vary sinusoidally with a constant shedding frequency. For $\text{Re} > \text{Re}_c$, the oscillations diverge from a sinusoidal behaviour. The sharpest amplitude drops in the drag and lift coefficients correspond to time instants where vortex dislocations develop in the cylinder wake.

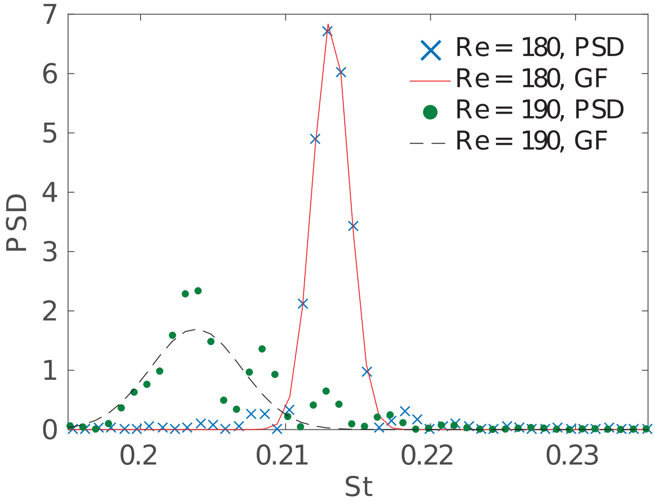


Figure 5. Power spectral density (PSD) as a function of Strouhal number with curve fitted Gaussian distribution (GF), for LFS flow at Re = 180 (PSD:×, GF:—) and Re = 190 (PSD:●, GF:- - -).

The vortex dislocations also affect the vortex shedding frequency, i.e. strong intermittent vortex dislocations result in a drop in St over a short time interval.

To find the shedding frequency from the lift data, a Lomb–Scargle power spectral density (PSD) estimate [33] is computed, followed by a Gaussian curve fitting. The Lomb–Scargle periodogram is preferred rather than a Fourier transform, due to uneven time sampling from the adaptable time step in the simulations. A curve fitting in frequency space is necessary since the Strouhal number varies with time in the transitional regime (particularly in regions with intermittent vortex dislocations). A Gaussian fit is used since the strongest peaks in frequency space generate a normal-like distribution (admittedly skewed for $\mathrm{Re} > \mathrm{Re}_c$); see Figure 5. The error bars in the St–Re curves in Figures 3, 6, 10, 12 and 13 are standard deviations of the curve-fitted Gaussian distributions in frequency space. Although using a moving average over extrema would suffice for computing St in the LFS simulations, for flow with a TFS high-frequency noise from the turbulence makes measurements of shedding frequency without the transformation to frequency space hard and inaccurate. Another advantage of the Gaussian fitting (compared to, e.g. setting St equal to the dominant frequency of the PSD) is that the standard deviation of the fitted curve provides information about the St span at each Re. In particular, this brings attention to simulations that develop vortex shedding that alternates between two- and three-dimensional vortex shedding. These quasi-stable states (see Section 4) are easily overlooked by using the dominating shedding frequency directly.

The simulations are run for 50–120 shedding periods after the vortex shedding has developed. The simulations in which the strongest vortex dislocations appear are run for the longest time to avoid an incorrect influence of the dislocations on St. Imagine, for instance, that only 30 periods were included for Re = 210, i.e. terminating the simulation at $\tilde{t}$ between 200 and 250. This would give an excessive influence of the strong vortex dislocation close to $\tilde{t} = 200$ on the flow statistics. For all but a few simulations, the flow is considered to be developed after a time corresponding to approximately 20 shedding periods. The exceptions are for Re very close to the critical Reynolds number, where the flow needs longer time to develop before the measurements start.

3. Effects of high-intensity free-stream turbulence on the transition-in-wake

Turbulence with intensity $T_i = 25\%$ and integral scale $\Lambda = D$ is inserted at the inlet of the flow domain for simulations with Reynolds numbers ranging from 120 to 350. The high intensity turbulence decays in the flow direction, and reaches the cylinder surface with an intensity of 10.5%–13.5% (highest decay of T_i for the lowest flow Re). Figure 6 depicts the resulting St–Re relationship, together with results from the LFS simulations. For the lowest and intermediate Re, corresponding to Regions I and III in the figure, the difference between St for the two cases is negligible. In Regions II and IV, this is not the case. In this section, we will discuss the two regions where the TFS and LFS results differ, before considering the effect of varying the properties of the TFS (intensity, scale and length of the turbulence domain) in Section 4.

3.1. Mode A instabilities and the onset of three-dimensionality in the wake

The drop in shedding frequency at the onset of three-dimensionality in the wake is known to be closely related to the mode *A* instabilities and vortex dislocations in the wake of the cylinder for flows with LFS conditions. This is confirmed in our simulations by looking at instantaneous plots of the vorticity ($\vec{\omega} = \nabla \times \vec{u}$) in a plane along the flow direction and the cylinder axis (the *xz*-plane).

The wake at Re = 180 is purely two-dimensional, as seen by the lack of cross-stream vorticity in Figure 7(a). At Re = 190, mode *A* instabilities (with $\lambda_A \approx 4D$) have developed in the wake; see Figure 7(c–f). A vortex dislocation is seen in Figure 7(e,f), obscuring the observations of mode *A* instabilities.

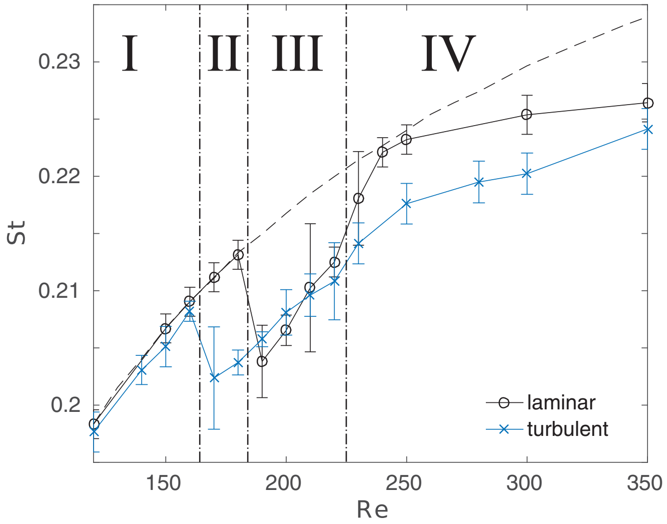


Figure 6. Comparison of St–Re relationships for flows with and without free-stream turbulence introduced upstream of the cylinder, for Reynolds numbers spanning over the transition-in-wake regime: -o- LFS; -x- TFS; - - 2D LFS.

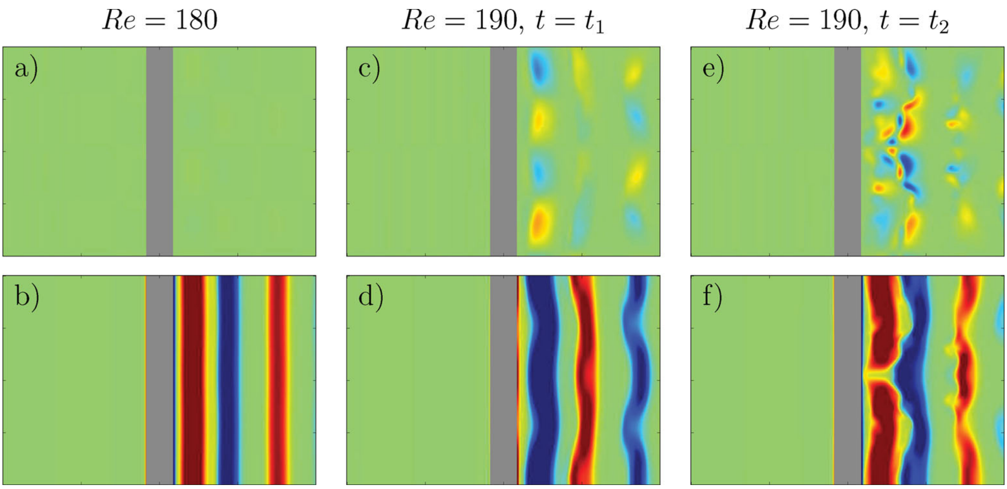


Figure 7. Laminar free-stream. Contours of instantaneous vorticity ω_y (top) and ω_z (bottom) in a plane along the streamwise direction, through the cylinder axis, plotted for Re = 180 (left) and Re = 190 (middle and right). Top and bottom frames correspond to the same time instant. Contour plots for two time instants are given for Re = 190: t_1, time instant without vortex dislocation; t_2, time instant with vortex dislocation.

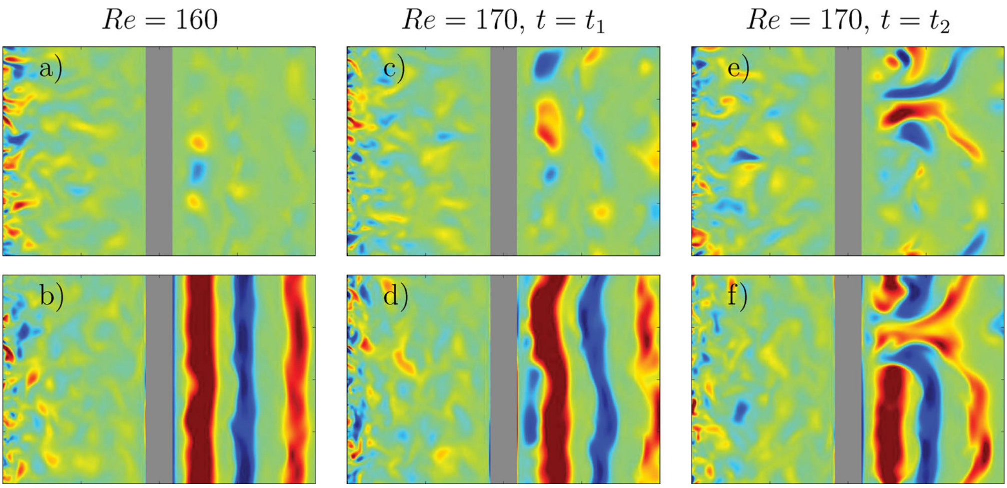


Figure 8. Turbulent free-stream. Contours of instantaneous vorticity ω_y(top) and ω_z (bottom) in a plane along the streamwise direction, through the cylinder axis, plotted for Re = 160 (left) and Re = 170 (middle and right). Top and bottom frames correspond to the same time instant. Contour plots for two time instants are given for Re = 170: t_1, time instant with vortex dislocation beginning to develop; t_2, time instant with fully developed vortex dislocation.

The drop in St occurs at lower Re for the TFS simulations as compared to the LFS, hence, $\mathrm{Re}_{c,\,\mathrm{TFS}} < \mathrm{Re}_c$ (see Region II of Figure 6). Based on the results with LFS, one would expect this to be due to mode A instabilities with vortex dislocations occurring at lower Reynolds numbers as a result of the TFS in the flow. Figure 8 shows instantaneous contours for the TFS flow in Region II and confirms this to some degree: The inherent three-dimensionality in the TFS results in three-dimensional flow effects even at $\mathrm{Re} < \mathrm{Re}_{c,\,\mathrm{TFS}}$, distorting the vorticity plots. This is apparent as $\omega_y \neq 0$ in the near wake in Figure 8(a) although St has not dropped from the two-dimensional mode (Re = 160 data point in Figure 6). Despite this 'noise', coherent flow structures can be observed in the contour plots. Let us compare Figure 8(c,d) to Figure 7(c,d), depicting ω_y and ω_z for TFS flow with $\mathrm{Re} > \mathrm{Re}_{c,\,\mathrm{TFS}}$ and LFS flow with $\mathrm{Re} > \mathrm{Re}_c$, respectively. Vortex dislocations do not dominate the near wake at the specific time instant of these snapshots. While a mode A instability is clearly visible in the LFS contours, this mode is less clear in the TFS case. There is an alternation between positive and negative ω_y in Figure 8(c), similar to the variation in the LFS flow, and the wavy structures in Figure 8(d) appear to have a wavelength close to $4D$. The mode A instability is present, but the view is obscured by a vortex dislocation beginning to develop in the upper half of the frames.

Strong vortex dislocations develop at Re = 170, as can be seen in Figure 8(e,f). The pattern is similar to the experimental flow visualisation by Zhang et al. (see [9], Figure 5), where the term vortex adhesion is used to describe this phenomenon. The development in time, from Figure 8(d–f), appears to be the development of a large-scale structure, and similar vortex dislocation patterns occur repeatedly after the initialisation of the simulation for this Re. The snapshot in Figure 8(f) is among the clearest observations of a vortex dislocation appearing to adhere to the cylinder at this Reynolds number.

Henderson [10] reported that spot-like disturbances (i.e. vortex dislocations) always developed from small perturbations at sufficiently large $\mathrm{Re} > \mathrm{Re}_c$ and $L_z > \lambda_A$, and that these large-scale structures only appear in a natural transition after mode A instability. The results of the present study can hardly be considered as a natural transition, as the TFS acts as a strong perturbation on the wake instability; the shift of the critical Reynolds number to a lower value is a direct consequence of the TFS. Yet, similar to the observations by Henderson [10], the vortex dislocations in the wake first occur for $\mathrm{Re} > \mathrm{Re}_{c,\,\mathrm{TFS}}$. The strong vortex dislocation and inherent three-dimensionality of the TFS obscure the view of mode A instability in the flow. Still, the similarity between the drop in St in Region II of Figure 6 for the two flow cases, the presence of vortex dislocations in Figure 8(e,f) and the vorticity snapshot in Figure 8(d), are strong indications of the existence of mode A instabilities at Re = 170 for the TFS flow past the cylinder.

3.2. Mixed instability modes in the upper part of the transition

In moving from Region III to Region IV of Figure 6 the results for TFS and LFS diverge. The difference in shedding frequency between the two cases remains almost constant for $250 \leq \mathrm{Re} \leq 300$, before decreasing somewhat for Re = 350.

The detailed analysis by Henderson [10] revealed a spatio-temporal chaos in the transition to a wake dominated by mode B instabilities at Re = 265. Since mode B instabilities are slower to develop than mode A instabilities (in development of the mode B dominated near wake) there exists a temporal region dominated by mode A instabilities and a temporal region of mixed mode A and B instabilities even at Re above the threshold where mode B is dominant. During this development the shedding frequency is reduced by $\Delta f \in (-0.004, -0.014)$ (corresponding to $\Delta \mathrm{St} \in (-0.04, -0.14)$) as compared to the frequency for the 2D shedding state (see [10], Figure 12).

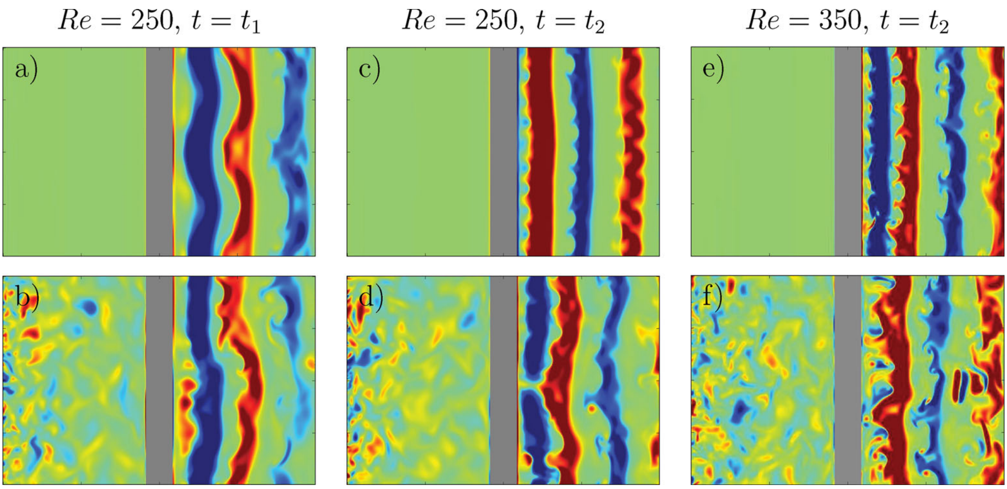


Figure 9. Contours of instantaneous vorticity ω_z in a plane along the streamwise direction, through the cylinder axis, plotted for Re = 250 (left and middle) and Re = 350 (right) for flow simulations with LFS (top) and TFS (bottom). Contour plots for two time instants are given for Re = 250, t_1, early development of instabilities; t_2, asymptotic state ($t_2 \gg t_1$). Only the asymptotic state is shown for Re = 350.

Figure 9 depicts snapshots of spanwise vorticity ω_z in the central xz-plane, with and without TFS, at two different Reynolds numbers in Region IV. The snapshots are taken at times that show the early development of the wake (Figure 9(a,b)) and at times where the wake has had time to reach an asymptotic state (Figure 9(c,f)). After a temporal region of mode A dominated flow (Figure 9(a)), the mode B dominated near wake develops for the LFS flow. At $t_2 \gg t_1$ the mode B instability is clearly visible, with a spanwise wavelength of approximately $1D$ (see Figure 9(c,e)). The spatio-temporal chaos [10] is present in the development of mode B instabilities, but comes to an end as the wake settles in a state dominated by mode B instabilities. For the flow with TFS mode B instabilities have barely started to develop at Re = 250. The vorticity in the near wake (Figure 9(d)) reveals that the flow is more similar to that at the onset of the transition (Figure 8(d)) than with the flow with LFS at Re = 250. Thus, the mode A dominated state at t_1 (Figure 9(b)) is not temporal when the free-stream is turbulent. At Re = 350, smaller scale instabilities have developed, with vortex fingers that adhere strongly to the cylinder before the vortices are distorted and dissipate downstream; see Figure 9(f). The flow is more similar to the LFS case at this Re, although larger-scale disturbances are still present in the near wake.

From the vorticity plots and the reduced shedding frequency, it is evident that the disturbances from the TFS suppress the transition to a near-wake dominated by mode B instabilities. The asymptotic state with no discernible large-scale pattern reported by Henderson [10] is not reached with a high-intensity TFS. Rather, the TFS stimulates the existence of mode A instabilities, resulting in the mixed state of mode A and B instabilities at higher Re when the free-stream is turbulent than when it is laminar.

Williamson [7] discussed measurements by Prasad and Williamson [34], where manipulation of the end conditions in experiments made vortex dislocations at Re > 260 possible. By means of this disturbance, the jump to higher St in Region IV of Figure 6 did not occur. A low frequency curve, similar to our TFS results, is shown for Re up to 400 in Williamson

[7]. This instability state was called B^* instabilities, where the asterisk denotes vortex dislocations in the flow. It is likely that end conditions might have a similar effect as the TFS, and that the B^* instabilities reported by Williamson [7] are the mixed A–B instabilities we observe in the present study. Hence, the asterisk denoting vortex dislocations is somewhat misleading. Dislocations are a product of interacting self-exiting mode A instabilities, present due to disturbances in the flow, and not unrelated to the flow instabilities and simply forced by end conditions.

4. Effects of varying free-stream turbulence properties on the transition-in-wake

So far, the turbulence inserted at the inlet of the flow domain has been of high intensity and with the integral scale equal to the cylinder diameter. The turbulence has been generated in a turbulence domain equal in length to the flow domain. In this section, we shall consider the effect (if any) that varying these properties of the TFS has on the St–Re relationship in the transition-in-wake state of the flow.

4.1. *The effect of the intensity of the TFS*

Figure 10 depicts the development of St for a span of Re for flows with TFS with four different turbulence intensities: $T_i = 25\%$, 13.5%, 8.1% and 4.8%, henceforth denoted high, medium high, medium low and low turbulence intensity. At Re = 200, the turbulence intensity decays to 12.2%, 7.4%, 5.2% and 3.6%, respectively, before reaching the cylinder. Lower Re gives a more rapid decay, e.g. the high intensity turbulence ($T_i = 25\%$) decays to 11.1% for Re = 150 and 13.5% for Re = 350, before reaching the cylinder surface.

Consider first the upper part of the St–Re relationship for varying T_i, as seen in Figure 10(b). The effect of the turbulence intensity in the part of the transition-in-wake, where the role of mode B instabilities becomes important, is perhaps what one intuitively would expect: A lower T_i yields a St–Re relationship that is more similar to what is found for LFS in the range $250 \leq \text{Re} \leq 350$. The correlation between T_i and St can be understood in light of our discussion in Section 3.2. The TFS stimulates the mode A instabilities in

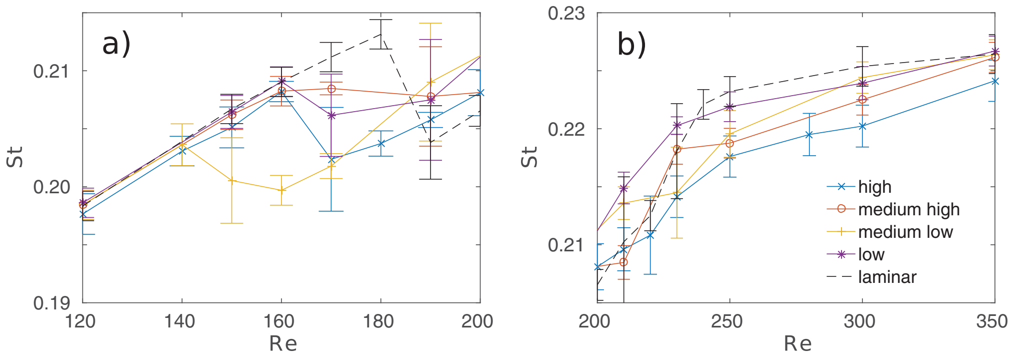


Figure 10. Comparison of St–Re relationships for flows with TFS with different turbulence intensities T_i on the inlet of the flow domain: -x- 25%; -o- 13.5%; -+- 8.1%; _*_ 4.8%; - - LFS.

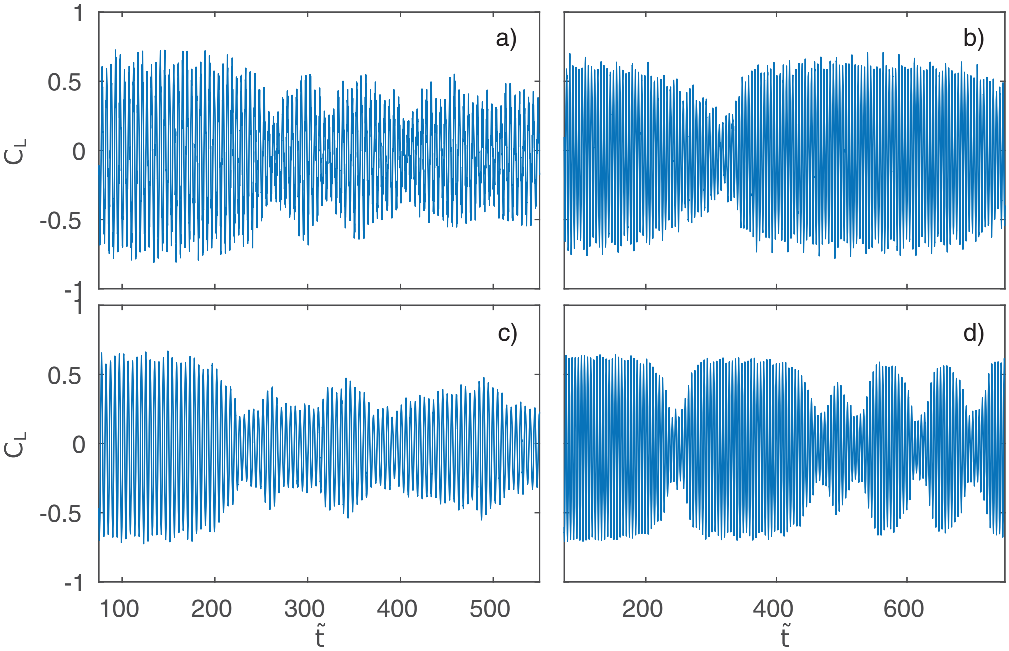


Figure 11. Instantaneous lift coefficient C_L of the cylinder as a function of the dimensionless time $\tilde{t} = t\bar{U}/D$, for flows with Re = 170 and free-stream turbulence upstream of the cylinder. The intensity T_i of the turbulence at the inlet of the flow domain is: (a) 25%; (b) 13.5%; (c) 8.1%; (d) 4.8%.

the mixed *A*–*B* instability mode (inhibiting the development of pure mode *B* in the near wake). The larger the effect of the mode *A* instabilities, the lower the shedding frequency. While the high intensity TFS in Section 3 almost completely suppresses the mode *B* instabilities, the results for moderate T_i reflect a more even mix between the two modes for Re = 250 and a wake dominated by mode *B* instabilities for Re = 350.

Consider now the more complex situation at the onset of three-dimensionality in the wake, seen in Figure 10(a). At Re = 170, all TFS simulations have St below the LFS flow (which is two-dimensional for this Reynolds number), yet there is a scatter in the TFS data. For the $T_i = 13.5\%$ and $T_i = 4.8\%$ cases, St is surprisingly high, indicating that the wakes have not fully developed to three-dimensional flows. Further, the early drop in St for the case with medium low turbulence intensity contradicts a possible simple correlation between high T_i and low $\mathrm{Re}_{c,\,\mathrm{TFS}}$ for our T_i range. This reflects the complicated nature of the instability in the transition-in-wake state of the flow. To gain insight into the mechanisms in play here, we take a more detailed look at the temporal development of the flow at Re = 170.

Figure 11 depicts the instantaneous lift coefficient of the cylinder at Re = 170 for the four different TFS intensities. For $T_i = 25\%$ and $T_i = 8.1\%$ the transition to a three-dimensional wake happens at $200 < \tilde{t} < 300$, where the amplitude of C_L drops significantly in Figure 11(a,c). The frequency of the oscillations decreases together with the drop in amplitude. A similar drop in C_L also occurs for the $T_i = 13.5\%$ and $T_i = 4.8\%$ cases at $200 < \tilde{t} < 400$, but C_L returns quickly to its previous amplitude; see Figure 11(b,d). This

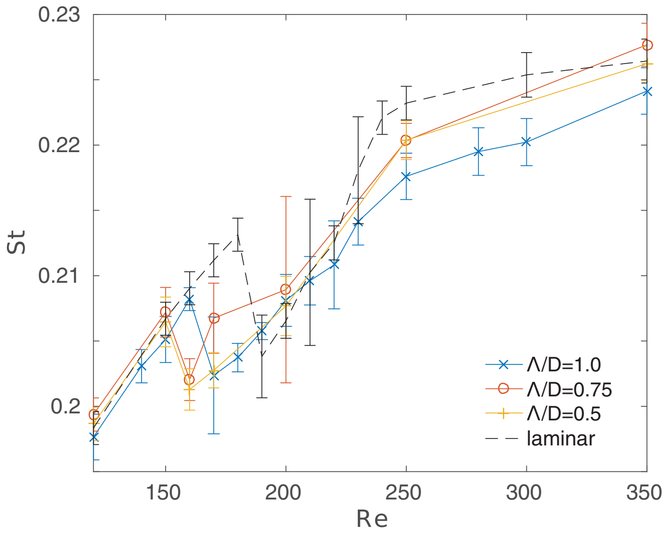


Figure 12. Comparison of St–Re relationships for flows with TFS with $T_i = 25\%$ and different integral scales: -x- $\Lambda/D = 1.0$; -o- $\Lambda/D = 0.75$; -+- $\Lambda/D = 0.5$; - - LFS.

behaviour can be explained in terms of a single vortex dislocation that is unable to saturate the wake in a mode A instability state. When the vortex dislocation disperses, the wake resumes two-dimensional vortex shedding. Such a drop also occurs at $\tilde{t} \approx 800$ for the $T_i = 13.5\%$ case, again followed by a rebound to the two-dimensional vortex street. The low-intensity TFS case (Figure 11(d)) shows several short periods of three-dimensionality in the wake for $\tilde{t} > 400$. A saturated state of mode A instabilities is not reached, and the wake alternates between two- and three-dimensional flow. The result of these quasi-stable shedding modes is a St in-between frequencies computed for three-dimensional flows (high and medium low intensity) and the two-dimensional flow (LFS) at Re = 170. The instantaneous shedding frequency varies between a high and a low mode as the wake alternates between two- and three-dimensional states.

The results for varying turbulence intensities suggest that the TFS destabilises the wake when Re is near the critical Reynolds number, resulting in possible alternations between two- and three-dimensional flow. Furthermore, the onset of three-dimensionality at Re = 150 for $T_i = 8.1\%$ and quasi-stable transition at Re = 170 for $T_i = 13.5\%$ and $T_i = 4.8\%$ suggest that $\mathrm{Re}_{c,\,\mathrm{TFS}}$ is not sharply defined when large disturbances are present in the flow. This is in accordance with the discussion by Williamson [3] on scattered Re_c data reported in experiments. In our cases, however, the disturbance is a TFS, not a contamination in the form of vortex dislocations excited by the end conditions in the experiments.

4.2. The effect of the integral scale of the TFS

Figure 12 depicts the St–Re relationship in the transition-in-wake regime for flows with TFS with different integral scales. The turbulence intensity is approximately 25% at the inlet for each case. However, as small-scale turbulence decays significantly faster than large-scale turbulence, the turbulence intensities for the different flow cases are no longer equal when the turbulence reaches the cylinder. For Re = 200, T_i is 12.2%, 7.2% and 6.7% at the front side of the cylinder for the turbulence with $\Lambda/D = 1.0$, $\Lambda/D = 0.75$ and $\Lambda/D = 0.5$, respectively.

The deviations between the St–Re relationships in Figure 12 are similar to those discussed in the previous subsection. Again, the complicated nature of the onset of the transition-in-wake is seen, particularly by considering the case with $\Lambda/D = 0.75$ for low Re. At Re = 160, three-dimensionality has developed in the flow, but at Re = 170 a quasi-stable shedding mode with St alternating between results for large-scale TFS and LFS occurs. The behaviour is a result of interactions of the TFS with the wake. At Re = 160, conditions are favorable for a saturated state of mode *A* dominating the flow, while at Re = 170 they are not. The results show that the quasi-stable state can occur for $\mathrm{Re} > \mathrm{Re}_{c,\,\mathrm{TFS}}$. For very long simulations, we expect that alternation between arbitrarily long periods of two- and three-dimensional flow in the wake will appear in several of the simulations with TFS. At present, we cannot conclude that a flow in a quasi-stable shedding state eventually will end up in a saturated three-dimensional state.

In the upper part of the transition-in-wake regime, the deviation between the curves in Figure 12 is in accordance with the observations in Section 4.1. The low intensity turbulence associated with the shorter integral scales results in a mixed *A*–*B* instability at Re = 250, whereas the wake is dominated by mode *B* instability at Re = 350.

4.3. The effect of the length of the turbulence domain

As described in Section 2.3, slices of forced turbulence from the turbulence domain are inserted on the inlet of the flow domain, from where the inflow turbulence is convected past the cylinder. The validity of introducing turbulence in a flow in such a way has not yet been addressed. Therefore, a short validation section is included here, with a focus on the effect of the size of the domain used for turbulence generation. The interesting length of the turbulence domain is the length corresponding to the streamwise direction in the flow domain (denoted by L_{xT}). This is the only free length parameter of the turbulence domain (the other lengths must match the size of the flow domain inlet).

Inserting turbulence from slices taken from a rectangular domain into the flow domain will introduce an artificial periodicity. This is because the same turbulence re-enters the inlet several times over the time span that the wake flow is investigated, due to the cycling over slices along L_{xT}. The artificial period introduced is the time it takes to iterate through all the slices along the *x*-direction in the turbulence domain, i.e. the time between each re-occurrence of the same slice at the inlet of the flow domain. The frequency introduced from the turbulence domain is $f_T = \bar{U}/L_{xT}$. Non-dimensionalising this frequency with the characteristic length (cylinder diameter *D*) and characteristic velocity in the flow domain yields $\tilde{f}_T = D/L_{xT}$. For a turbulence domain equal in length to the flow domain (12*D*), $\tilde{f}_T = 0.083$, which is close enough to the Strouhal number in the transitional regime to not be neglected prior to further investigations.

Figure 13 depicts results from flow simulations with TFS generated in domains with $L_{xT} = 12D$ and $L_{xT} = 24D$. Turbulence of two different integral scales are included for both domain sizes. Doubling L_{xT} does not affect the general trends in the St–Re relationship, but a deviation occurs for Re = 170 for the cases with $\Lambda/D = 1.0$. The deviation is in the behaviour at Re close to $\mathrm{Re}_{c,\,\mathrm{TFS}}$, discussed in Sections 4.1 and 4.2. Unlike the flow with TFS

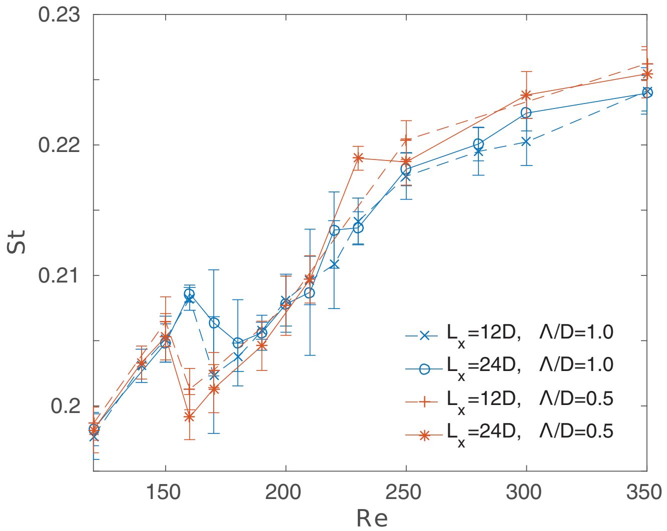


Figure 13. Comparison of St–Re relationships for flows with TFS, with turbulence generated in domains of different size and with different integral scales. Dashed lines for $L_{xT} = 12D$ and solid lines for $L_{xT} = 24D$.

generated in the $L_{xT} = 12D$ domain, flow from the simulations with $L_{xT} = 24D$ and $\Lambda/D = 1.0$ enters a quasi-stable state, alternating between two- and three-dimensional shedding modes, at Re = 170. The resulting St is therefore in-between the results for TFS and LFS at this Re. This is not an effect related to the artificial periodicity, but rather of a different realisation of the same turbulence having less favourable conditions for a saturated mode *A* dominated flow for this particular simulation. The effect of the turbulence domain length is negligible.

5. Conclusions

In the DNS presented here, free-stream turbulence has been inserted upstream of a circular cylinder for Reynolds numbers spanning over the entire transition-in-wake state of the flow. Distinct effects of the turbulence on the transition-in-wake are observed in two particular Re regions. These regions (denoted as Regions II and IV in Figure 6) have Re near the onset of three-dimensionality in the flow and Re where mode *B* instabilities dominate the wake for flows with LFS, respectively.

With TFS, the critical Reynolds number, identified by the sudden drop in shedding frequency due to the three-dimensional flow effects, is not sharply defined. This is not surprising, as the instability at the onset of the transition-in-wake state of the flow is subcritical, hence, disturbances in the flow can shift the point where the transition occurs. The shift due to the TFS is, however, to lower Re than allowed by the hysteresis pattern seen in experiments for the subcritical instability. With TFS, we observe $150 \lesssim \mathrm{Re}_{c,\,\mathrm{TFS}} \lesssim 170$ (as compared to $\mathrm{Re}_c \simeq 190$ for LFS), with intermittent vortex dislocations present in the wake for $\mathrm{Re} > \mathrm{Re}_{c,\,\mathrm{TFS}}$. Quasi-stable shedding states, in which the wake alternates between two- and three-dimensional vortex shedding, are observed in several of our simulations for $\mathrm{Re} \approx \mathrm{Re}_{c,\,\mathrm{TFS}}$. In such cases, the shedding frequency is reduced when vortex dislocations occur in the wake. The vortex dislocations do not, however, lead to a saturated state of mode *A* instabilities. Rather, the wake either returns to a state of two-dimensional vortex shedding, until the next vortex dislocation occurs, or stays in a three-dimensional shedding state for some time before returning to the two-dimensional state.

Variations of the intensity and integral scale of the TFS revealed the complexity of the onset of the transition-in-wake state of the flow. For turbulence intensity between 3.6% and 12.2%, no correlation between intensity and $\mathrm{Re}_{c,\,TFS}$ was found. Surprisingly, the lowest $\mathrm{Re}_{c,\,TFS}$ was seen for T_i in-between the two extremes. A reduction of the integral scale of the turbulence resulted in a lower $\mathrm{Re}_{c,\,TFS}$. Unfortunately, the integral scale of the turbulence could not be isolated from the intensity as small-scale turbulence decays faster than the large scales in the flow domain. The results are therefore inconclusive in regard to the effect of the integral scale.

In the discussion by Williamson [3] regarding the scatter in Re_c data reported from experiments, Williamson concluded that (in absence of other effects) the scatter can be accounted for by contamination from end conditions. The scatter of the critical Reynolds number in our simulations show another possible source of Re_c variation. The disturbances in the flow from the TFS were large enough to perturb the flow and initiate the transition at a significantly lower Re than for a corresponding flow with LFS and identical end conditions. Hence, in the absence of other effects (including contamination from end conditions), the scatter in Re_c can be ascribed to turbulence in the free-stream.

The second range of Re, in which an effect from the TFS is seen, is the region that for LFS is dominated by mode *B* instabilities. A lower shedding frequency is observed for the simulations with TFS in this region. This is due to the TFS stimulating the existence of mode *A* instabilities, thereby inhibiting the development of a near wake dominated by mode *B* instabilities. Simulations with varying turbulence intensities revealed a correlation between this effect and T_i. A lower T_i means a closer relationship between the TFS flow and the LFS flow for $\mathrm{Re} \geq 250$.

The lowest intensity case of our DNS has turbulence with T_i that has decayed to approximately 3% before it reaches the cylinder for $\mathrm{Re} \approx \mathrm{Re}_c$. An interesting extension of this study would be to find a cut-off for the intensity of the TFS that forces the transition.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

We would like to acknowledge that this research has received funding from the Research Council of Norway (Norges Forskningsråd) under the FRINATEK Grant [grant number 231444]. This research was supported in part with computational resources provided by UNINETT Sigma2 AS [project number NN9405K], [project number NN2649K].

ORCID

J. R. Aarnes http://orcid.org/0000-0002-5899-2597

References

[1] Provansal M, Mathis C, Boyer L. Bénard-von Kármán instability: transient and forced regimes. J Fluid Mech. 1987;182:1–22.

[2] Zdravkovich MM. Flow around circular cylinders: volume 1: fundamentals. Oxford (UK): Oxford University Press; 1997.

[3] Williamson CHK. Vortex dynamics in the cylinder wake. Ann Rev Fluid Mech. 1996;28:477–539.

[4] Barkley D, Henderson RD. Three-dimensional Floquet stability analysis of the wake of a circular cylinder. J Fluid Mech. 1996;322:215–241.

[5] Henderson RD, Barkley D. Secondary instability in the wake of a circular cylinder. Phys Fluids. 1996;8:1683–1685.

[6] Williamson CHK. The existence of two stages in the transition to three-dimensionality of a cylinder wake. J Fluid Mech. 1988;31:3165–3168.

[7] Williamson CHK. Three-dimensional wake transition. J Fluid Mech. 1996;328:345–407.

[8] Williamson CHK. The natural and forced formation of spot-like 'vortex dislocations' in the transition of a wake. J Fluid Mech. 1992;243:393–441.

[9] Zhang HQ, Fey U, Noack BR, et al. On the transition of the cylinder wake. Phys Fluids. 1995; 7: 779–794.

[10] Henderson RD. Nonlinear dynamics and pattern formation in turbulent wake transition. J Fluid Mech. 1997;352:65–112.

[11] Posdziech O, Grundmann R. Numerical simulation of the flow around an infinitely long circular cylinder in the transition regime. Theor Comp Fluid Dyn. 2001;15:121–141.

[12] Barkley D, Tuckerman L, Golubitsky M. Bifurcation theory for three-dimensional flow in the wake of a circular cylinder. Phys Rev E. 2000;61:5247–5252.

[13] Gerrard JH. The wakes of cylindrical bluff bodies at low Reynolds number. Phil Trans R Soc A. 1978;288:351–382.

[14] Norberg C. An experimental investigation of the flow around a circular cylinder: influence of aspect ratio. J Fluid Mech. 1994;258:287–316.

[15] Bloor MS. The transition to turbulence in the wake of a circular cylinder. J Fluid Mech. 1964;19:290–304.

[16] Hussain AKMF, Ramjee V. Periodic wake behind a circular cylinder at low Reynolds numbers. Aeronaut Q. 1976;27:123–142.

[17] Zdravkovich MM. Conceptual overview of laminar and turbulent flows past smooth and rough circular cylinders. J Wind Eng Ind Aerodyn. 1990;33:53–62.

[18] Norberg C. Effects of Reynolds number and a low intensity freestream turbulence on the flow around a circular cylinder. Gothenburg, Sweden: Dept. Applied Thermodynamics and Fluid Mechanics, Chalmers University of Technology; 1987. 87/2.

[19] Brandenburg A, Dobler W. Hydromagnetic turbulence in computer simulations. Comput Phys Commun. 2002;147:471–475.

[20] pencil-code.nordita.org [Internet]. Stockholm (SE): NORDITA; [updated 2017 June 12]. Available from: https://github.com/pencil-code

[21] Peskin CS. Flow patterns around heart valves: a numerical method. J Comput Phys. 1972;10:252–271.

[22] Mittal R, Iaccarino G. Immersed boundary methods. Annu Rev Fluid Mech. 2005;37:239–261.

[23] Tseng YH, Ferziger JH. A ghost-cell immersed boundary method for flow in complex geometry. J Comput Phys. 2003;192:593–623.

[24] Haugen NEL, Kragset S. Particle impaction on a cylinder in a crossflow as function of Stokes and Reynolds numbers. J Fluid Mech. 2010;661:239–261.

[25] White FM. Viscous fluid flow. 3rd ed. Boston (MA): McGraw-Hill Higher Education; 2006.

[26] Haugen NEL, Brandenburg A. Hydrodynamic and hydromagnetic energy spectra from large eddy simulations. Phys Fluids. 2006;18:075106.

[27] Poinsot TJ, Lele SK. Boundary conditions for direct simulations of compressible viscous flows. J Comput Phys. 1992;101:104–129.

[28] Williamson CHK. Defining a universal and continuous Strouhal-Reynolds number relationship for the laminar vortex shedding of a circular cylinder. Phys Fluids. 1988;31:2742–2744.

[29] Behr M, Hastreiter D, Mittal S, et al. Incompressible flow past a circular cylinder: dependence of the computed flow field on the location of the lateral boundaries. Comput Methods Appl Mech Eng. 1995; 123: 309–316.

[30] Anagnostopoulos P, Iliadis G, Richardson S. Numerical study of the blockage effects on viscous flow past a circular cylinder. Int J Numer Meth Fluids. 1996;22:1061–1074.

[31] Aarnes JR, Haugen NEL, Andersson HI. On validation and implementation of an immersed boundary method in a high order finite difference code for flow simulations. In: Skallerud B, Andersson HI, editors. MekIt'15: eight national conference on computational mechanics; 2015 May 18–19; Trondheim (Norway). Barcelona (Spain): International Center for Numerical Methods in Engineering; 2015. p. 1–21.

[32] Thompson M, Hourigan K, Sheridan J. Three-dimensional instabilities in the wake of a circular cylinder. Exp Therm Fluid Sci. 1996;12:190–196.

[33] Scargle JD. Studies in astronomical time series analysis. II. Statistical aspects of spectral analysis of unevenly spaced data. Astrophys J. 1982;263:835–853.

[34] Prasad A, Williamson CHK. Three-dimensional effects in turbulent bluff body wakes. Exp Therm Fluid Sci. 1997;14:9–16.

Paper III [3]

Inertial particle impaction on a cylinder in a turbulent cross-flow at modest Reynolds numbers

Jørgen R. Aarnes, Nils E. L. Haugen and Helge I. Andersson

Manuscript submitted for publication.

Inertial particle impaction on a cylinder in turbulent cross-flow at modest Reynolds numbers

Jørgen R. Aarnes[a], Nils E. L. Haugen[a,b], Helge I. Andersson[a]

[a] *Department of Energy and Process Engineering, Norwegian University of Science and Technology, Trondheim, Norway*
[b] *SINTEF Energy Research, Trondheim Norway*

Abstract

Particle impaction on a cylinder in a cross-flow is investigated with the use of Direct Numerical Simulations, with focus on the effect of free-stream turbulence on the front-side impaction efficiency. The turbulence considered is high-intensity homogeneous isotropic turbulence, introduced upstream of the cylinder in moderate Reynolds number flows. It is found that the free-stream turbulence leads to a significant increase in the number of particles that impact the cylinder for certain Stokes numbers (St). The peak amplification of impaction is observed at $St = 0.3$, for different integral scales and Reynolds numbers. This peak is related to a change in impaction mechanism, from boundary stopping to boundary interception, and it will therefore dependent on the size of the particles as well as the Stokes number. The effect of the free-stream turbulence on impaction decreases rapidly when the difference between a particle's Stokes number and Stokes number at peak impaction increases. Using statistical analysis, an expectation value of predicted effects of free-stream turbulence on particle impaction is derived. This expression predicts the observed impaction amplification to a good degree, particularly in terms of which Stokes numbers that are affected by the turbulence.

1. Introduction

Particle laden fluid flows are common both in nature and in a large number of industrial applications. In many applications the impaction between

*Corresponding authors: Jørgen Røysland Aarnes
Email address: jorgen.r.aarnes@ntnu.no (Jørgen R. Aarnes)

particles and solid objects entrained in the fluid flow is central. Such impactions may lead to the buildup of a deposition layer on the solid-fluid interface or to erosion of the solid object. In industry, these mechanisms are typically found in filters, furnaces, industrial boilers and ventilation systems.

A simple cylindrical geometry may well approximate the solid objects in several of the applications mentioned above, and a cylinder in a cross-flow has become a benchmark case for particle impaction. Inertial particle impaction on a cylinder in a cross-flow can be split into three different impaction modes, based on what drives the trajectories of the convected particles during impact. The modes are classical impaction (particle inertia driven trajectory), boundary stopping (boundary layer driven trajectory) and boundary interception (mass center of particles do not come in contact with the cylinder) (cf. Haugen and Kragset, 2010; Weber et al., 2013b). The mode is determined by the Stokes number (with some dependence on Reynolds number). Here we define the Stokes number as the ratio of the particle Stokes time to the timescale of the fluid flow around the solid object ($St = \tau_p/\tau_f$). The Reynolds number is $Re = U_0 D/\nu$, where U_0 is the free-stream velocity, D is the cylinder diameter and ν is the kinematic viscosity. Generally, classical impaction occurs for Stokes numbers $St \gtrsim 0.9$, boundary stopping for $0.2 \lesssim St \lesssim 0.9$ and boundary interception for $St \lesssim 0.2$. Note that the boundary stopping mode is partly overtaken by the classical impaction mode for high Reynolds numbers (see Haugen and Kragset, 2010).

Potential flow theory can be used to calculate the particle impaction efficiencies as a function of particle size (see Israel and Rosner, 1982). The method is well accepted for Stokes numbers larger than unity. Rotational and viscous flow effects are not resolved with a potential flow approximation. These are particularly important for transport of particles with small Stokes numbers, as these follow the flow to a large extent. Further, the approximation assumes infinitesimal particles sizes, and will therefore not account for boundary interception. Consequently, no impaction occurs for particles with $St < St_{\mathrm{crit}} = 1/8$ (Ingham et al., 1990) (note that viscous effects will increase the value of St_{crit}, hence St_{crit} is larger for smaller Reynolds numbers (see Phillips and Kaye, 1999)).

Other approaches to finding the impaction efficiencies, using experimental (Schweers et al., 1994; Kasper et al., 2009), numerical (Yilmaz et al., 2000; Li et al., 2008; Haugen and Kragset, 2010; Haugen et al., 2013) and phenomenological modeling (Huang et al., 1996) are found in literature. With few exceptions, these studies regard smooth laminar flow past one or more cylinders. It is, however, reasonable to expect that the influence of any turbulence in the flow is highly relevant for the rate of particle impaction in the mentioned applications. The turbulence in said applications can be due to, e.g., transitional eddies in the free shear layers of the cylinder at high Reynolds numbers, combustion upstream of the cylinder, wall turbulence for a cylinder in a confined space, or similar sources. Either way, one would expect the velocity fluctuations to affect the inertial particle impactions on the cylinder surface.

An exception to the experimental studies on particles in smooth laminar flow, is the measurements by Douglas and Ilias (1988) on the effect of turbulence on particle impaction. In that study the cylinder was situated within a channel such that the turbulence was generated by the channel walls. The results showed increased impaction efficiencies when turbulence was present in the flow, and eddy diffusion was considered as a prime mechanism leading to this. The scatter in the data was, however, quite large for small Stokes numbers.

Numerical studies on particle impaction in turbulent flows are often limited by the fact that they use Reynolds Average Navier-Stokes modeling of the turbulence (see review by Weber et al., 2013a). One such study was performed by Weber et al. (2013b), in which it was found that the turbulence played a minor role for particles with Stokes number larger than $St_{\rm crit}$ and had the effect of increasing the particle impactions below this critical point. However, as pointed out by the authors of said study, the predictable power of these CFD-simulations is limited by the lack of rigorous testing of the particle tracking procedure in use. Further limitations are introduced from the modeling and time-averaging of the flow, as seen by the factor two difference in impaction efficiency for small particles when switching between turbulence modeled by the k-ϵ model and Reynolds Stress Models.

In the current study, we consider the effect of turbulence in the flow on the particle impaction efficiencies on a circular cylinder by using Direct Numerical Simulations (DNS), i.e., three-dimensional simulations where all scales of the turbulence (spatial and temporal) are resolved. The turbulence is not generated in the flow past the cylinder, as the Reynolds numbers in our study are too low for this. Rather, free-stream turbulence is inserted upstream of the cylinder and decays as it is convected by the mean flow. This typically emulates the flow regime above a thermal incinerator or boiler for solid fuels. As particles impact they deposit on the cylinder surface. The aim of this study is to understand the way the turbulence affects the inertial impaction mechanism of the particles.

The structure of the paper is the following: In Section 2 the governing flow equation and particle equations are given, and the simulation set-up is described. Following this, different ways to predict the effects of turbulence on the particle impaction is described in Section 3. Results from simulations with and without free-stream turbulence are presented in Section 4, and compared to predicted amplification of particle impaction, before conclusions are drawn in Section 5.

2. Methodology

2.1. Governing equations

The equations describing the flow are the equation for continuity,

$$\frac{\mathrm{D}\rho}{\mathrm{D}t} = -\rho \boldsymbol{\nabla} \cdot \boldsymbol{u} \,, \tag{1}$$

and momentum,

$$\rho \frac{\mathrm{D}\boldsymbol{u}}{\mathrm{D}t} = -\boldsymbol{\nabla} p + \boldsymbol{\nabla} \cdot (2\mu \boldsymbol{S}) \ , \tag{2}$$

where ρ, t, $\boldsymbol{u}$ and p are the density, time, velocity vector and pressure, respectively, $\mu = \rho\nu$ is the dynamic viscosity and

$$\frac{\mathrm{D}}{\mathrm{D}t} = \frac{\partial}{\partial t} + \boldsymbol{u} \cdot \boldsymbol{\nabla} \tag{3}$$

is the substantial derivative operator. The rate of strain tensor is given by:

$$\boldsymbol{S} = \frac{1}{2}\left(\boldsymbol{\nabla}\boldsymbol{u} + (\boldsymbol{\nabla}\boldsymbol{u})^T\right) - \boldsymbol{I}\left(\frac{1}{3}\boldsymbol{\nabla} \cdot \boldsymbol{u}\right) \ , \tag{4}$$

where $\boldsymbol{I}$ is the identity matrix. The pressure is computed by the isothermal ideal gas law, $p = c_s^2 \rho$, where c_s is the speed of sound. The flow is isothermal and weakly compressible.

Particles in the flow are point particles. A Lagrangian formalism is used to track the particles. A particle's velocity and position is described by:

$$\frac{\mathrm{d}\boldsymbol{v}_p}{\mathrm{d}t} = \frac{\boldsymbol{F}_{D,p}}{m_p} \,, \tag{5}$$

$$\frac{\mathrm{d}\boldsymbol{x}_p}{\mathrm{d}t} = \boldsymbol{v}_p \,, \tag{6}$$

where $\boldsymbol{v}_p$, $\boldsymbol{x}_p$ and m_p are the velocity, center of mass position and mass of the particle, respectively. The drag force acting upon a particle is given by:

$$\boldsymbol{F}_{D,p} = \frac{1}{2C_c}\rho C_{D,p} A_p \left|\boldsymbol{u} - \boldsymbol{v}_p\right| \left(\boldsymbol{u} - \boldsymbol{v}_p\right) \ , \tag{7}$$

where $A_p = \pi d_p^2/4$ is the cross sectional area of the particle and

$$C_c = 1 + \frac{2\lambda}{d_p}\left(1.257 + 0.4e^{(-1.1d_p/2\lambda)}\right) \ , \tag{8}$$

is the Stokes-Cunningham factor with parameters set for air for a particle with diameter d_p (Cunningham, 1910; Davies, 1945) . The mean free path λ accounts for the fact that for very small particles, the surrounding medium can no longer be regarded as a continuum but, rather, distinct particles. The particle drag coefficient is given by:

$$C_{D,p} = \frac{24}{Re_p}\left(1 + 0.15Re_p^{0.687}\right) \ . \tag{9}$$

The expression is valid for particles with particle Reynolds number $Re_p = d_p \left|\boldsymbol{v_p} - \boldsymbol{u}\right| / \nu \lesssim 1000$, which is the case for all particles in the present study. Thus, the particle drag force can be written as:

$$\boldsymbol{F}_{D,p} = \frac{m_p}{\tau_p}\left(\boldsymbol{u} - \boldsymbol{v}_p\right) \ , \tag{10}$$

where

$$\tau_p = \frac{S d_p^2 C_c}{18\nu(1 + f_c)} \tag{11}$$

is the particle response time, $f_c = 0.15Re_p^{0.687}$ and $S = \rho_p/\rho$. Note that Eq. (11) equals the Stokes time in the limit $C_c = 1$ and $Re_p \ll 1$. The Stokes number ($St = \tau_p/\tau_f$) is defined with a fluid time scale

$$\tau_f = \frac{D}{2U_0} \ . \tag{12}$$

An alternative convention is to define the Stokes number without a factor two in the denominator of the fluid time scale.

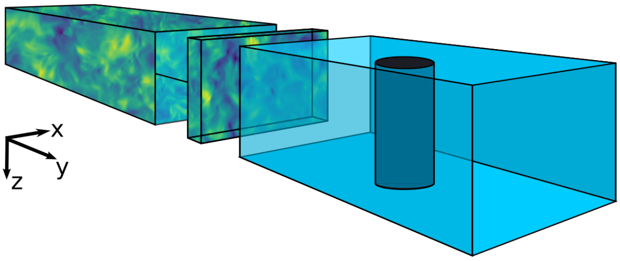


Figure 1: Computational domain, split into two rectangular boxes. Left box for turbulence generation and right box for flow domain. Thin slices of velocity data are taken from the turbulence domain and added to the inlet of the flow domain, illustrated here by a thin rectangular box between the two domains. The slice thickness and the cylinder in the flow domain are not to scale.

2.2. Simulations

The simulations were performed with the high-order finite-difference code for compressible hydrodynamic flows known as The Pencil Code (see Brandenburg and Dobler, 2002; The Pencil Code, 2018). The equations describing the flow are discretized with a memory efficient third-order Runge-Kutta method in time (developed by Williamson, 1980) and sixth-order central differencing in space. A three-point deep ghost-zone immersed boundary method was used to resolve the cylinder surface, with ghost-points set by no-slip and impermeability boundary conditions along grid lines (with quadratic interpolation) and zero gradient of the density along the surface normal (with linear interpolation). For details on the numerical method, and grid and time independence tests, see Aarnes et al. (2018).

The fluid velocities used to update the particle trajectories were interpolated from surrounding grid points by tri-linear Lagrangian interpolation. In the immediate vicinity of the cylinder surface, quadratic interpolation was used to compute the velocity component normal to the surface.

2.2.1. Fluid flow and turbulence generation

The computational domain consists of two rectangular boxes: a turbulence production domain and a flow domain (see Fig. 1). The production domain is a periodic box in which homogeneous isotropic turbulence was generated by external forcing in random directions over a selected range of wave numbers. The strength of the forcing and the wavenumber range are free parameters which determine the turbulence intensity and length scale, respectively. For details, see Brandenburg (2001); Haugen and Brandenburg (2006); Aarnes et al. (2018). The flow domain is a rectangular box with periodicity in two direction (x- and z-direction), and a mean flow in the third direction (y-direction). Navier-Stokes characteristic boundary conditions (NSCBC) were used both at the inlet and at the outlet of the flow domain. This boundary formulation is a formulation that makes use of one-dimensional characteristic wave relations to allow acoustic waves to pass through the boundaries (Poinsot and Lele, 1992). The NSCBC implementation in the Pencil Code uses modifications suggested by Yoo et al. (2005) and Lodato et al. (2008) to account for transversal flow effects, e.g. for turbulent flow at the inlet. A circular cylinder is placed in the center of the flow domain, with axis along the z-direction. Simulations were performed for flow in the unsteady vortex shadding regime, with Reynolds numbers $Re = 100$ and $Re = 400$.

The size of both the turbulence production domain and the flow domain is $(L_x, L_y, L_z) = (6D, 12D, 4D)$, where D is the cylinder diameter. The domain size was set sufficiently large to resolve the three-dimensional phenomena in the flow (the wavelength of instabilities in the cylinder wake is approximately 1D in the z-direction at $Re = 400$ (Williamson, 1996)). Validation runs have been performed, and a uniform grid spacing of $D/\Delta x = 40$ and $D/\Delta x = 64$ for flow with $Re = 100$ and $Re = 400$, respectively, was deemed sufficiently accurate to fully resolve all scales of the free-stream turbulence and the boundary layer around the cylinder. The grid spacing did, however, limit the particle Stokes numbers which could be used to $St \geq 0.2$. Smaller particles require a finer resolution in order to be represented accurately in the vicinity of the cylinder, as the interpolation of fluid velocities for use in Eq. (10) is sensitive to the grid spacing. Very small, tracer-like particles are particularly sensitive to incorrect estimation of forces from the fluid on the particle, as being brought marginally closer to the surface may result in interception rather than no interception. Note that grid spacings $\Delta x = \Delta y = \Delta z$ were used, to ensure that the spatial Kolmogorov microscale (η_{Kol}) was resolved in all directions ($\Delta x/\eta_{\mathrm{Kol}} = 3.48, 1.92$ for $Re = 100, 400$). The strict advective time step restriction ($\Delta t \leq C_u \Delta x / \left(|\boldsymbol{u}|_{max} + c_s\right)$, where C_u is the advective Courant number) due to the low Mach number ($Ma = 0.1$) , ensured several hundred time steps per (temporal) Kolmogorov microscale.

Once the forced turbulence was statistically stationary, slices of flow quantities from the turbulence domain were added to the inlet of the flow do-

Table 1: Turbulence intensity T_i at different positions along the streamwise direction in a $12D$ long domain, with no cylinder present in the flow.

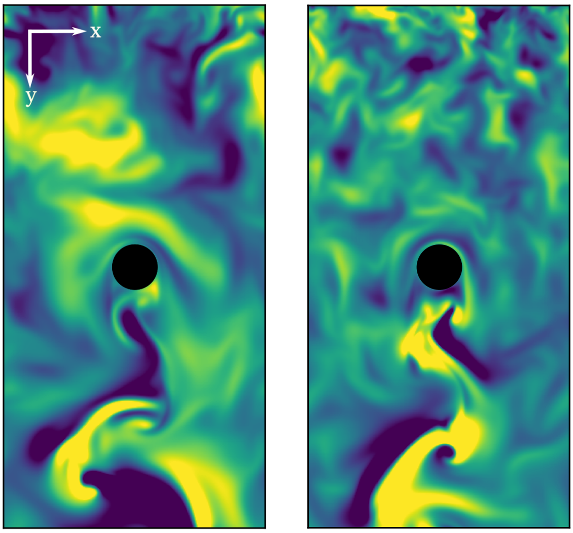
Position	$Re = 100$		$Re = 400$	
(downstream)	$\Lambda = 2.7D$	$\Lambda = 0.8D$	$\Lambda = 2.7D$	$\Lambda = 0.8D$
0	0.230	0.222	0.232	0.225
3D	0.172	0.090	0.203	0.143
6D	0.152	0.061	0.183	0.106
9D	0.139	0.048	0.169	0.086

main. The inlet velocity in the flow domain $\boldsymbol{U} = \langle \boldsymbol{U_0} \rangle + \boldsymbol{u'}$, with mean flow $\boldsymbol{U_0} = (0, U_0, 0)$ and the velocity flucuations $\boldsymbol{u'}$. The velocity fluctuations at the inlet were updated at every time step using data from the turbulence domain. The mean velocity was not affected by the insertion of turbulence at the inlet, since the turbulence was isotropic. In the flow domain, the turbulence decayed when convected downstream towards and past the circular cylinder. The decay rate of the turbulence depends on its intensity ($T_i = u_{\mathrm{rms}}/\langle \boldsymbol{U} \rangle$ where u_{rms} is the is the root-mean-square value of the three-dimensional velocity fluctuations) and integral scale (Λ), and the Reynolds number of the flow. Turbulence with integral scale $\Lambda = 2.7D$ and $\Lambda = 0.8D$ was used, with intensity 0.22–0.23% at the inlet. For details on the turbulence decay, see Tab. 1, where T_i at selected points along the streamwise direction in a cylinder-free flow domain is listed, for the different Reynolds numbers and integral scales. Instantaneous velocity contours of the decaying turbulence at $Re = 400$ can be seen in Fig. 2.

2.2.2. Particles

Particles with Stokes numbers ranging from 0.2 to 10 were inserted at random positions within a thin box covering the inlet of the flow domain. The velocity of an inserted particle was initialized to the mean inlet velocity. Particles were removed from the simulations either when hitting the cylinder or when reaching the outlet boundary.

In each simulation, a large number of particles (1.0×10^7) over a selection of Stokes numbers were inserted at a constant rate of approximately $(1.25 \times 10^6)/\tau_K$, where τ_K is the shedding period of the von Kármán eddy street. This ensured that the particles interacted with the flow past the cylinder over several shedding cycles. The variation in Stokes number was achieved by varying the particle radii. For a given Stokes and Reynolds number, the



(a) $\Lambda \approx 2.7D$

(b) $\Lambda \approx 0.8D$

Figure 2: Contours of velocity component normal to the plane of view, in a plane along the streamwise direction, perpendicular to the cylinder axis, with flow from top to bottom. High intensity free-stream turbulence with integrals scale Λ inserted at the top and convected downstream. Flow Reynolds number is 400. Dark blue (light yellow) corresponds to negative (positive) values.

particle diameter can be determined by:

$$\frac{d_p}{D} = 3\sqrt{\frac{St}{SRe}}\,. \tag{13}$$

In all simulations the ratio of particle to fluid density $S = \rho_p/\rho = 1000$.

The distribution of particles with respect to Stokes numbers was non-uniform. Since the impaction probability is a lot lower for particles with small St than high St, more particles were needed in the low than in the high St range in order to obtain good statistics of the impaction rate. To illustrate this, consider that even with 14% of the inserted particles having $St = 0.2$ and only 2.3% having $St = 10$, approximately 250 times more

Table 2: Typical distribution of particles in a simulation. Stokes number greater or equal 2.0 include $St = 2.0, 3.0, 5.0$ and 10, each containing approximately 2.3% of the particles in the simulation.

St	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	≥ 2.0
N_{St}/N_{tot} (%)	14	13	12	12	9	9	9	6	6	9

particles with $St = 10$ impact the cylinder than with $St = 0.2$. The particle distribution for a typical simulations is given in Tab. 2.

3. Estimated effects of turbulence on particle impaction

When turbulence is introduced at the inlet of the flow domain, there are several possible ways it can affect the particle trajectories and impaction.

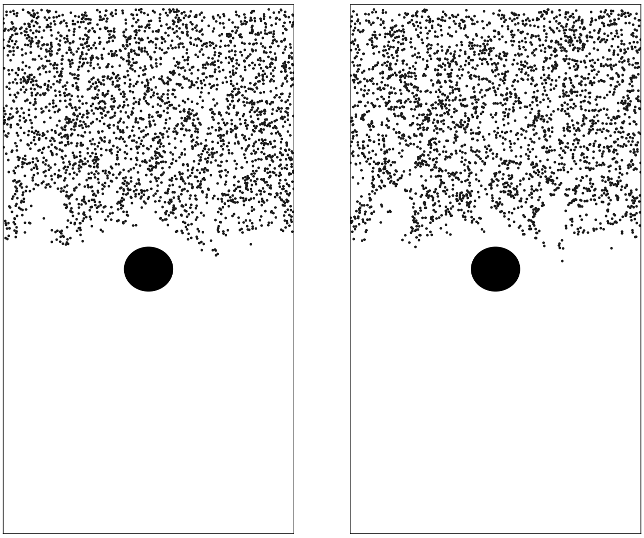
3.1. The turbulence based Stokes number

The turnover time of an integral scale turbulent eddy is given as

$$\tau_e = \Lambda / u_{\text{rms}} \,. \tag{14}$$

With this time scale a turbulence based Stokes number can be defined as $St_e = \tau_p/\tau_e$. If the eddy turnover time is much shorter than the particle Stokes time ($St_e \gg 1$) the particles are too heavy to be affected by the turbulent eddies, i.e, for large St_e the turbulence has a negligible effect on the particles. In this work, $St_e < 1.5$ for $\Lambda = 0.8D$ and $St_e < 0.5$ for $\Lambda = 2.7D$ at the inlet, so large scale effects cannot be disregarded for any of the particle sizes in use.

Alternatively, the turbulent Stokes number can be defined as $St_\eta = \tau_p/\tau_\eta$, where $\tau_\eta = \sqrt{\nu/\varepsilon}$ is the Kolmogorov time scale (with the average energy dissipation rate ε). It is known that particles with $\tau_\eta \lesssim \tau_p \lesssim \tau_e$ will experience preferential concentration (Yoshimoto and Goto, 2007), i.e particles will cluster if $St_\eta \lesssim 1 \lesssim St_e$. The clustering has been found in low vorticity regions (Squires and Eaton, 1991). In the present study, $0.3 \lesssim St_\eta < 13.8$ ($0.1 < St_\eta \lesssim 6.6$) for $Re = 400$ ($Re = 100$). Hence, some preferential concentration should be expected, in particular for $Re = 400$ flow with particles that have particle Stokes numbers $St \gtrsim 0.7$ ($St_\eta = 0.96$ when $St = 0.7$ for this Reynolds number). Consider Fig. 3 for an illustration of preferential concentration for $St = 0.2$ and $St = 0.8$ particles. A higher degree of



(a) $St = 0.2$

(b) $St = 0.8$

Figure 3: Particles convected towards the cylinder in $Re = 400$ flow with free-stream turbulence with integral scale $\Lambda = 0.8D$. The thickness of the 2D-slice (in z-direction) is $12\eta_{\text{Kol}}$.

clustering can be seen for the larger particles (clustering takes some time to develop, and is mainly seen for particles that have been convected at least half of the upstream length). Note that corresponding plots for $Re = 100$ show no notable clustering.

3.2. Large scale turbulence

The impaction efficiency $\eta = N_{\text{impact}}/N_{\text{init}}$ in a laminar flow is determined by the Stokes number. Here, N_{impact} is the number of particles impacting on the cylinder surface and N_{init} is the number of particles initially positioned such that their finite radii would overlap with the cylinder at some point if convected with the mean flow in the direction of the cylinder. When $St_e \lesssim 1$, the particle trajectories will be affected by the turbulent eddies. Consequently, the particle velocities will deviate from the mean flow velocity. When the scale of the turbulence is not small compared to the size of the cylinder, this yields a modified Stokes number, which will then also give a change in the particle impaction efficiency.

The Stokes number $St = \tau_p/\tau_f$ is by definition proportional to the mean fluid flow velocity since $\tau_f = D/2U_0$. With turbulence present in the flow, the magnitude of the flow velocity U (in general different from the mean flow velocity) is stochastic. Thus, the Stokes number is also a stochastic variable, effectively different from the *laminar* St, expressed by the uniform fluid velocity U_0. The effective Stokes number can be expressed as

$$St_{\text{eff}} = St\frac{U}{U_0}\,. \tag{15}$$

As St_{eff} is a linear function of U, its variance becomes

$$Var(St_{\text{eff}}) = \sigma_{St}^2 = \left(\frac{St}{U_0}\right)^2 Var(U)\,. \tag{16}$$

Since U_0 is constant, the variance of St_{eff} is zero when $U = U_0$. The expectation value of the Stokes number is $E(St_{\text{eff}}) = St$, since U fluctuates symmetrically around the mean velocity in homogeneous isotropic turbulence.

We may re-write the effective Stokes number to $St_{\text{eff}} = St + \Delta_{St}$, where $\Delta_{St} = ((U - U_0)/U_0)St$ is the fluctuation in the Stokes number due to $U \neq U_0$. Taylor expanding the impaction efficiency yields

$$\eta(St_{\text{eff}}) = \eta(St) + \eta'(St)\Delta_{St} + \frac{\eta''(St)}{2}\Delta_{St}^2 + \mathcal{O}(\Delta_{St}^3)\,. \tag{17}$$

The expectation value of the impaction efficiency can now be found:

$$E(\eta(St_{\text{eff}})) = \eta(St) + \eta'(St)E(\Delta_{St}) + \frac{\eta''(St)}{2}E(\Delta_{St}^2) + E(\eta(\mathcal{O}(\Delta_{St}^3))\,. \tag{18}$$

Neglecting higher order terms and using that $\sigma_{St}^2 = E(\Delta_{St}^2) - E(\Delta_{St})^2 = E(\Delta_{St}^2)$, due to the symmetry of velocity fluctuations around the mean, yields a simple expression for the expectation value of particle impaction efficiency:

$$E(\eta(St_{\text{eff}})) \approx \eta(St) + \frac{\eta''(St)}{2}\sigma_{St}^2. \tag{19}$$

When there is no turbulence $\Delta_{St} = 0$ and the expectation value of the impaction efficiency is $E(\eta(St_{\text{eff}}) = \eta(St)$. In other words, it equals the impaction efficiency in the laminar case, as anticipated.

In order to use Eq. (19) to predict front-side particle impaction efficiencies with turbulence present, values for $\eta_f(St)$, $\eta''_f(St)$ and σ^2_{St} are required. There are different ways to find or approximate these variables, and we suggest the following: Results for front-side impaction efficiencies, η_f, are readily available in literature (see, e.g. Haugen and Kragset, 2010) and from our recent simulations. Second derivatives of these measurements can be found by curve fitting to existing results, and finding derivatives of the fitted curves. To find the effective Stokes number, the variance of U is needed. This value can be computed straightforwardly by considering the sample variance of the velocity field of the decaying turbulence. This can be done either in the flow domain, upstream of the cylinder, or in a cylinder-free domain, where the decaying turbulence is considered free from any obstructions in the flow. The latter approach is used in this study. We will return to this subject in Section 4.2.

As the front side impaction is also dependent on the Reynolds number, a Reynolds number dependent expectation value expression can be derived. With the same procedure for derivation as above, we get:

$$E(\eta(Re_{\text{eff}})) \approx \eta(Re) + \frac{\eta''(Re)}{2}\sigma^2_{Re}\,, \tag{20}$$

where $Re_{\text{eff}} = Re(U/U_0)$ and $\sigma^2_{Re} = Var(Re_{\text{eff}})$. To use this expression, $\eta(Re)$ and (approximations of) $\eta''(Re)$ are needed. In the present study, simulations were only performed for two different Reynolds numbers, which is insufficient to compute $\eta(Re)$ and $\eta''(Re)$. We will therefore only focus on the predictions of turbulence effects based on the effective Stokes number.

3.3. Small scale turbulence

If the integral scale of the turbulent eddies is very small, the eddies may penetrate into the boundary layer around the cylinder. If this happens, the particles can impact on the cylinder surface due to turbophoresis, i.e. the transport of particles from areas of high turbulent intensity to areas of low turbulent intensity (related to much larger spatial scales than preferential concentration), which could have a significant effect on the impaction efficiency. Let us call this “impaction by external turbophoresis” (see Mitra et al., 2018, and references therein for a study of non-channel-flow turbophoresis). For a weakly compressible flow, the thickness of the cylinder's front-side boundary layer (stagnation layer) is given by

$$\delta \approx 2.4\sqrt{\nu/B}\,, \tag{21}$$

where $B \approx 4U_0/D$ (White, 2006). Hence,

$$\delta = \frac{\kappa D}{\sqrt{Re}}\,; \tag{22}$$

where κ is a constant approximately equal to 1.2. In order to have impaction by external turbophoresis the turbulent integral scale Λ must fulfill the inequality

$$\delta > \Lambda\,. \tag{23}$$

If the Navier-Stokes equations are transformed into spectral space, the dissipative term of the velocity is $\nu k^2 \tilde{\boldsymbol{u}}$, where $\tilde{\boldsymbol{u}}$ is the Fourier transform of the fluctuating velocity vector. It is apparent that small scale turbulence, that is, turbulence with large wavenumber k, will decay very fast. If the turbulence generated upstream of the cylinder shall survive all the way down to the cylinder, the timescale of the turbulence decay, given by $\tau_\nu = \Lambda^2/\nu$, must be longer than the convective time scale $\tau_c = L_c/U_0$, where L_c is the distance that the decaying turbulence is convected before reaching the cylinder (upstream length of the flow domain in the present case). Hence, from the inequality ($\tau_\nu > \tau_c$), and the definitions of τ_ν and τ_c the following must be true if the turbulence shall survive until it reaches the cylinder:

$$\Lambda > \frac{\sqrt{\gamma} D}{\sqrt{Re}}, \tag{24}$$

where $\gamma = L_c/D$. Combining Eqs. (22) - (24) yields

$$\sqrt{\gamma} < \frac{\Lambda}{D}\sqrt{Re} < \kappa. \tag{25}$$

For the inequality to be satisfied, we must have $\sqrt{\gamma} < \kappa$. Hence, unless very small upstream domain lengths are used, the inequality in Eq. (25) is a contradiction. Consequently, impaction by external turbophoresis, where the integral scale of the turbulence is small enough to penetrate the boundary layer around the cylinder, is not practically feasible if the source of turbulence is not very close to the cylinder. An example where the source is indeed close to the cylinder is a cylinder placed in a turbulent channel flow as in the study by Douglas and Ilias (1988). This is a situation which is encountered in many industrial applications, and it introduces some extra parameters into the study. Due to the increased parameter space this application is not considered here; instead, this work focuses solely on decaying isotropic turbulence introduced upstream of the cylinder.

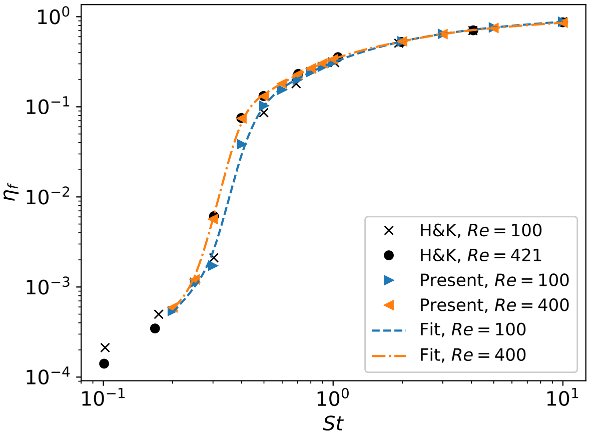


Figure 4: Front side impaction efficiency η_f for laminar free-stream flow. Present results computed in a three-dimensional domain at $Re = 100$ and 400, compared to results by Haugen and Kragset (2010) computed at $Re = 100$ and 421 in a two-dimensional domain.

4. Results and discussion

As mentioned the impaction efficiency $\eta = N_{\text{impact}}/N_{\text{init}}$, where N_{init} is the total number of particles inserted over a cylinder area projected onto the flow inlet. The impaction efficiency can be considered as a sum of two parts, $\eta = \eta_f + \eta_b$, where η_f and η_b are the front- and back-side impaction efficiencies, respectively. The bulk of the particles that impact the cylinder at low and moderate Reynolds numbers do so on the front-side, and we focus only on this part of the impaction efficiency here. For backside impaction to become significant, larger Reynolds numbers and smaller particles than used in this study must be considered.

To verify that the particle impaction results are sufficiently accurate, results from a flow without disturbances from free-stream turbulence are compared with results from a two-dimensional impaction study in literature. The comparison can be seen in Fig. 4, where it is apparent that the earlier results are reproduced very well by the present DNS.

From Fig. 4 it is also clear that the three-dimensional effects present in the flow at $Re = 400$ do not have a noticeable impact on the front-side impaction efficiency (this would be seen as discrepancy between the present (3D) and literature (2D) results for $Re = 400$). This is not surprising, as the

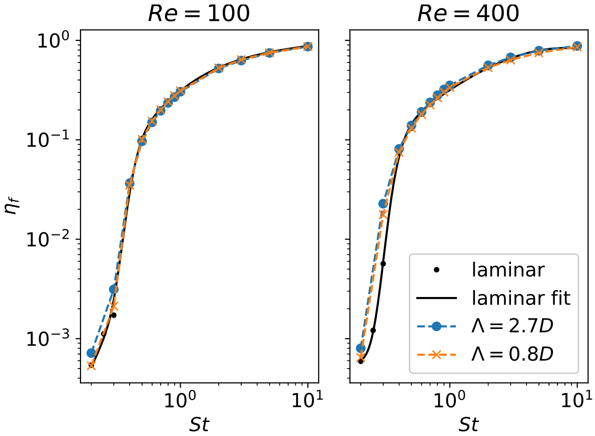


Figure 5: Front side impaction efficiency η_f. Results for laminar and turbulent free-stream, the latter with integral scale $\Lambda = 2.7D$ or $0.8D$. Results for Reynolds numbers $Re = 100$ and 400.

flow is in the upper part of the transition-in-wake regime at this Reynolds number, where tree-dimensional effects occur in the wake of the cylinder, not on the front-side.

4.1. Simulations with free-stream turbulence

High-intensity turbulence was inserted at the inlet an decayed when convected downstream. The intensity of the turbulence was 22%−23% at the inlet in all simulations with free-stream turbulence, and the decay was strongly dependent on integral scale and Reynolds number (see Section 2.2 for details).

Consider Fig. 5, depicting the impaction efficiencies with and without a turbulent free-stream. A few changes are apparent in the results: For certain Stokes numbers an increase in η_f can be seen, for both Reynolds numbers. The increase is largest for the largest integral scale turbulence. Further, the effect of the turbulence appears to be larger for $Re = 400$ than for $Re = 100$. These observation are not surprising, as larger effects are expected from the turbulence with higher turbulence intensity. Another, perhaps more unexpected consequence of the turbulence, is that the increase in impaction is largest at $St \leq 0.3$. This is in the Stokes number-region where the dominant impaction mechanism changes from boundary stopping to boundary interception (as the Stokes number is decreased).

To gain further insight into the effect of the turbulence on the front-side impaction, the impaction results from particles subjected to a turbulent free-stream are normalized with corresponding results from laminar free-stream flow. The resulting amplification factor:

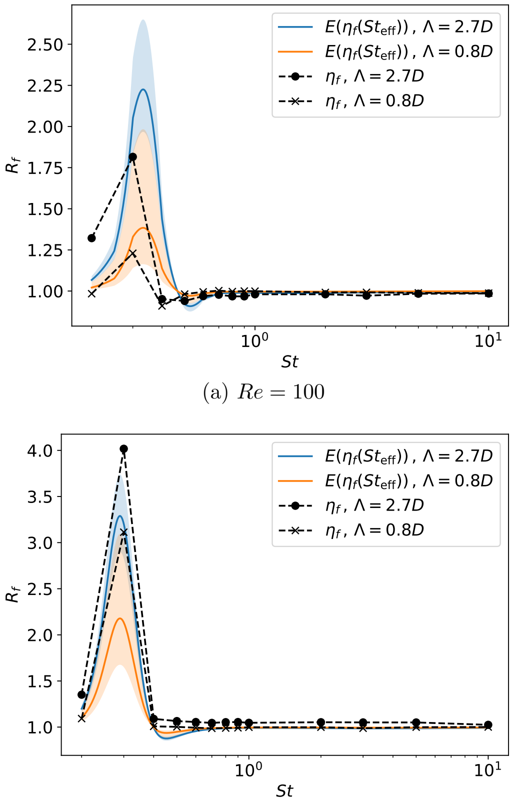
$$R_f = \frac{\eta_{f,\mathrm{turb}}}{\eta_{f,\mathrm{lam}}}\,, \tag{26}$$

is a direct measure of increase or decrease of impactions due to the free-stream turbulence. The amplification factor is plotted together with predicted values in Fig. 6. The predicted values will be considered in Section 4.2.

Figure 6(a) depicts R_f as a function of Stokes number for $Re = 100$ for the two different turbulence cases considered. It is evident that the turbulence significantly increased impaction of particles in the lower range of the Stokes number domain. This is in accordance with the findings of Weber et al. (2013b), where only impaction below a critical Stokes number, St_{crit}, was affected by turbulence. For $Re = 106$, Weber et al. (2013b) computed $St_{\mathrm{crit}} = 0.324$, which is in agreement with the $Re = 100$ simulation results, showing only a small effect of the turbulence for $St \geq 0.4$. For particles with $St \geq 0.4$ we observe a small decrease in the impaction for particles in the free-stream turbulence, with R_f approaching one as the Stokes number is increased.

Contrary to the results of Weber et al. (2013b), a distinct peak of amplification occurs at $St = 0.3$. For the larger integral scale turbulence the amplification is approximately 1.8, an increase from 0.17% impaction to 0.31% impaction. This peak can be attributed to the change in impaction mechanism, from boundary stopping to boundary interception, near this Stokes number. When particles are affected by the turbulence, they experience a change in their effective Stokes number. A small increase (decrease) in velocity will give a slightly higher (lower) Stokes number. If we take St_{peak} to be the St where the impaction mechanism changes (see Sec. 4.2) and consider the laminar impaction curves in Fig. 4, a particle with Stokes number $St_{\mathrm{peak}} + \Delta_{St}$ will have a smaller chance to impact if $\Delta_{St} < 0$ (boundary interception dominating), but a much higher change to impact if $\Delta_{St} > 0$ (boundary stopping dominating). With the mean of Δ_{St} being zero, this will result in an average increase in impaction efficiency near St_{peak}. Note that without including the boundary interception mechanism in the simulations (as in Weber et al., 2013b) the amplification $R_f \to \infty$ as $St \to St_{\mathrm{crit}}$, and the distinct amplification peak will not be observable.

The results for $Re = 400$ (see Fig. 6(b)), are similar to those for the lower



(a) $Re = 100$

(b) $Re = 400$

Figure 6: Predicted and observed results for amplification factor of impaction efficiencies resulting from turbulence of different integral scales interacting with particles in the flow.

Reynolds number. Again, the amplification is greatest for particles with Stokes numbers ≤ 0.3, and there is a distinct peak in R_f at $St = 0.3$. Yet, a few differences between the results at $Re = 100$ and $Re = 400$ can be noted: the turbulence gave rise to a larger amplification effect in the latter case, and the difference between the effect from the two turbulence realizations is smaller. Neither of these findings are very surprising, if one considers that the intensity of the turbulence decayed slower in the flow domain when the Reynolds number was increased. In addition to these aspects, some amplification of the impact of larger particles is observed at $Re = 400$. This effect ($R_f = 1.02$–1.06 for $St \geq 0.5$, decreasing with increasing St) is only seen for the highest intensity case ($\Lambda = 2.7D$). In a study by Homann et al. (2016), the amplification of particle impaction on a sphere in a turbulent flow converged to unity as a power law of St (decreasing with increased St). This trend was most clear for very strong turbulent fluctuations ($T_i = 0.60$ and $T_i = 1.18$). A similar relationship can be investigated for impaction on a cylinder, but higher intensity turbulence than used in the present study is required before conclusions can be drawn. Note that impaction by boundary interception was not included in the study by Homann et al. (2016), and including such a mechanism may affect a power law trend.

The results at $Re = 100$ showed agreement with the CFD-study by Weber et al. (2013b) in terms of a cut-off Stokes number above which the turbulence played a minor role in terms of amplifying particle impaction. Weber et al. (2013b) did not report a critical Stokes number for $Re = 400$, but St_{crit} can be approximated to 0.25 for $Re = 400$, by interpolating from their reported critical Stokes numbers. Using this value as a cut-off Stokes number is, however, not consistent with our observations of effects of the free-stream turbulence for $Re = 400$. On the contrary, the peak in turbulence effect on impaction is found at the same Stokes number for $Re = 400$ as for $Re = 100$, which is above the approximated St_{crit} for $Re = 400$. The critical Stokes number as defined by Weber et al. (2013b) was intended as a limit for when zero-sized particles impact ($St > St_{\text{crit}}$) or do not impact ($St < St_{\text{crit}}$) the cylinder in a laminar free-stream flow. From our simulations it appears that this critical Stokes number is related to a peak amplification, when boundary interception (particles with finite radii) is included. The $St = 0.3$ particles in the $Re = 400$ flow are expected to be near this peak Stokes number. Hence, a significant amplification at $St = 0.3$ is not surprising, and an even larger amplification is expected if particles with $St = St_{\text{peak}}$ were considered. Investigating this would require, first, a determination of St_{peak}.

4.2. Predictions of free-stream turbulence effects on particle impaction

In all simulations, the integral scale of the free-stream turbulence was of the same order of magnitude as the cylinder diameter. The free-stream turbulence is such that $\mathrm{St}_e = \tau_p/\tau_e \lesssim 1$, hence, the particles should be expected to be convected by the larger turbulent eddies if the flow domain was free of any obstacles.

To be able to predict effects of the turbulence on the particle impaction, data from the turbulence-free simulations was used to compute the unknowns $\eta(St)$ and $\eta''(St)$ in Eq. (19). Flow data from simulations with a turbulent free-stream decaying in a cylinder-free domain was used to approximate σ_{St}. In the following we explain how this was done in practice.

First, the cfit-tool with smoothing splines (for $Re = 100$ results) and rational fractions (for $Re = 400$ results) in MATLAB (software version MATLAB, 2016) was used to fit curves to the impaction data computed for flow with a laminar free-stream. This yielded expressions for $\eta(St)$ in the entire Stokes number range covered by the sample of particle sizes. Different curve fitting was used for the two data sets, as this yielded the best possible fit for each set. This fitting was done in logarithmic space, hence, exponentials of the fitted results are the impaction efficiencies (displayed in Fig. 4). From a fitted function $f(\log(St)) = \log(\eta(St))$, second derivatives were found analytically by:

$$\begin{aligned}\eta''(St) &= \frac{\mathrm{d}^2}{\mathrm{d}(St)^2}\exp\left[f(\log(St))\right] \\ &= \frac{e^f}{St^2}\left[\frac{\mathrm{d}^2 f}{\mathrm{d}(\log(St))^2} + \left(\frac{\mathrm{d}f}{\mathrm{d}(\log(St))}\right)^2 - \frac{\mathrm{d}f}{\mathrm{d}(\log(St))}\right]. \qquad (27)\end{aligned}$$

Finally, the variance of the upstream velocity field, necessary to compute σ_{St} by Eq. (16), was needed. Since the Reynolds numbers were moderate, the turbulence decayed quite rapidly. Hence, the variance of the velocity depended strongly on where in the flow domain the velocity was measured. Since we are interested in the stochastic property of the velocity due to the turbulence only, the variance was computed in a flow domain without a cylinder present. To obtain relevant values for the computations, the variance was averaged at each grid point in time and moving average were computed in the streamwise direction. The moving averages were computed over a $0.5D$ thick slab in the streamwise direction, starting a distance $0.5D$ downstream of the inlet, and continuing to where the center of the cylinder would be positioned if the cylinder was present. Mean, maximum and minimum values of the moving averages were found and used as prediction and prediction bounds of $Var(U)$.

The amplification factors predicted by inserting the computed $\eta(St)$, $\eta''(St)$ and σ_{St} in Eq. (19) can be seen as the colored areas in Fig. 6. Quite large error bounds are used, due to the way the variance was computed (as noted in the previous paragraph). The predictions fit well with the DNS impaction results. In particular, the strong amplification of the impaction efficiencies close to $St = 0.3$ is predicted by the statistical analysis. From Eq. (19) the observed amplification of particle impaction is not surprising for the Stokes numbers where the impaction curves are concave up. Haugen and Kragset (2010) showed that the impaction efficiency is close to linear in the boundary interception mode. Hence, the second derivative of $\eta(St)$ with respect to Stokes number will be approximately zero in this region. Based on the expression for predicted impaction we would therefore not expect a significant effect of turbulence on inertial impaction in the boundary interception regime (but a strong effect in the region where impaction mechanism changes from boundary stopping to boundary interception).

The predictions (Fig. 6) show that the amplification is greatest for particles with $St = 0.33$ for $Re = 100$ and $St = 0.29$ for $Re = 400$. We identify these Stokes numbers as measures for St_{peak} for the particle sizes used in the present cases. Note that St_{peak} is close to, but somewhat larger than, the critical Stokes numbers for its respective Reynolds number. This is in accordance with the suggested explanation that the boundary interception becomes the dominant impaction mechanism at St_{peak}. With zero-sized particles, $\eta(St) \to 0$ as $St \to St_{\text{crit}}$. With finite-sized particles, $\eta(St)$ decreases towards zero in the same way, when the Stokes number is reduced in the boundary stopping impaction regime. The trend is interrupted for $St = St_{\text{peak}} > St_{\text{crit}}$, for which a significant amount of particles impact by boundary interception. Boundary interception is dependent on the particle radii, and, hence, so is St_{peak}. Consider, e.g. particles with half the density ρ_p of the particles in the present study, with all other parameters (except size) unchanged. The particles will have a 40% larger radius than particles in the present study (see Eq. (11)). Consequently, impaction by boundary interception is more likely to occur, and a larger St_{peak} is expected.

For the larger particles ($St \gtrsim 0.5$) the predictions by Eq. (19) show a convergence to unity from below (impaction curve $\eta(St)$ is concave down, yielding $\eta''(St) < 0$). The results from turbulence simulations yielded $\eta(St > 0.5) \simeq 1$, approached from below, for $\Lambda = 0.8D$ turbulence for both Reynolds numbers. A similar trend is found for $\Lambda = 2.7D$ turbulence for $Re = 100$, yet here approximately 3%–1% decrease ($R_f \approx 0.97$–0.99) was measured for $St = 0.6$–10 (R_f closer to 1 for higher Stokes numbers). Only with the large scale turbulence for $Re = 400$ was the trend $R_f \to 1$ from below contradicted. As mentioned in Sec. 4.1, amplification of 2%–6% for $St = 0.5$–10

(smaller amplification for higher Stokes numbers) was observed for this case. This contradicts the predictions by Eq. (19), yet is in accordance with observations by Homann et al. (2016) that found increased impaction (on a sphere) for all St when the free-stream was turbulent. Nevertheless, we have seen that the predictive expression fits well with our data for $St < 0.5$. The inability to predict amplification of impaction at higher Stokes numbers for $Re = 400$ and $\Lambda = 2.7D$ may be due to assumptions made when deriving Eq. (19) being invalid for such cases. One possibility is that the preferential concentration of these particles in low vorticity zones affected the velocity fluctuations experienced by the particles. Although the velocity fluctuations of the flow field were symmetric (yielding $E(\Delta_{St}) = 0$), preferential concentration may have lead particles to experience a non-symmetric velocity field. Consequently, the odd derivatives of $\eta(St)$ in (Eq. (18)) should not cancel in the Taylor expansion of $\eta(St_{\mathrm{eff}})$. Note also, that the higher order terms in the Taylor expansion cannot be neglected if large particles are affected by turbulence with very high intensity, since this may yield $\sigma_{St} > 1$. Consider e.g., that turbulence with intensity decaying such that $Var(U)/U_0^2 \geq 1$ yields $\sigma_{St} \geq 1$ for $St \geq 1$. This was not the case here, but should be considered by other researchers who would like to use Eq. (19) to predict effects of turbulence on particle impaction.

5. Conclusion

The effect of free-stream turbulence on inertial particle impaction for a large range of Stokes numbers has been investigate by DNS. The following can been concluded from the study:

(a) The effect that free-stream turbulence has on the impaction rate of particles on a cylinder depends largely on the particle Stokes number. There is a peak Stokes number for which the largest amplification of particle impaction on the cylinder occurs. For the sample of Stokes numbers used in the present study, $St = 0.3$ was closest to the peak Stokes number, in both $Re = 100$ and $Re = 400$ flow. The relative increase in particle impaction decreases fast when the difference between a particle's Stokes number and the peak Stokes number increases.

(b) The peak Stokes number is related to where the impaction mechanism changes from being dominated by boundary stopping ($St > St_{\mathrm{peak}}$)

to boundary interception ($St < St_{\text{peak}}$). At which Stokes number impaction by boundary interception becomes the dominating impaction mechanism depends on the size of the particles for a given Stokes number. Hence, St_{peak} is not a fixed value depending only on St. The value will depend on the parameters that make up the Stokes number.

(c) The effect of turbulence on particle impaction is also dependent on the Reynolds number of the flow, and the intensity and integral scale of the turbulence. The Reynolds number and the integral scale of the turbulence determine how fast the intensity of the turbulence decays in the flow domain. Higher Reynolds number and larger integral scale (i.e., higher turbulence intensity) yields a larger amplification of impaction. In the present study, the Reynolds number and integral scale effects cannot be distinguished from the turbulence intensity effects. Hence the study is inconclusive in regards to how these parameters individually will effect the amplification of impaction.

(d) Using St_{crit} as a measure of for which particles with $St < St_{\text{crit}}$ experience the largest effects of the free-stream turbulence on impaction is not appropriate when boundary interception is included as an impaction mechanism. For finite-sized particles, the peak Stokes number, where the maximum effect of the turbulence on particle impaction is found, will always be larger than St_{crit}.

(e) A quite simple expression for the expected turbulence effects on impaction was derived:

$$E(\eta(St_{\text{eff}})) \approx \eta(St) + \frac{\eta''(St)}{2}\sigma_{St}^2.$$

Comparison of predicted effects from this expression with DNS results was favorable, in particular in the lower Stokes number region.

Based on our conclusions, we suggest three topics of further research. Firstly, the larges impaction was observed at $St = 0.3$. This is slightly below the predicted peak for $Re = 100$ and slightly above for $Re = 400$. Increasing the St-resolution in this region, that is, performing additional simulations with a large number of particles distributed among Stokes numbers ranging from 0.2 to 0.4 could to a greater extent pinpoint a the Stokes number where the maximum turbulence effect on impaction occurs. This would be useful in a further validation of the model for predicting particle impaction in a turbulent flow. This would also increase the accuracy of the fitted laminar $\eta_f(St)$ curves, giving better predictions based on expectation values. Secondly, the Reynolds number should be increased both to see if the Stokes number of the peak is shifted, and to further investigate the effect of the turbulence integral scale when separated from intensity. Along with this, the inlet turbulence intensity should be varied for a fixed integral scale, such that distinct effects of T_i can be identified. Thirdly, extending this study to much smaller particle sizes would make it possible to consider if effect of turbulence diminishes for very small particles and, further, to explore turbulence effects on back-side impaction. The back-side impaction phenomenon has previously been found by Haugen and Kragset (2010) to occur for small particles ($St \lesssim 0.13$), for $Re \geq 400$. For very small particles effects of thermophoresis should be taken into account, as recent studies (Beckmann et al., 2016; García Pérez et al., 2016) have found that such mechanisms to be very important for particle transport and deposition of small particles.

Acknowledgements

We would like to acknowledge insightful discussions with Steinar Kragset and Tore Flåtten regarding the predictive power of laminar particle impaction results on impaction efficiencies in a turbulent flow. We would also like to acknowledge that this research is funded by the Research Council of Norway (Norges Forskingsråd) under the FRINATEK Grand [grant number 231444]. The research was supported in part with computational resources provided by UNINETT Sigma2 AS [project numbers NN9405K, NN2649K].

References

Aarnes, J., Andersson, H., Haugen, N., 2018. Numerical investigation of free-stream turbulence effects on the transition-in-wake state of flow past a circular cylinder. Journal of Turbulence 19 (3), 252–273.

Beckmann, A. M., Mancini, M., Weber, R., Seebold, S., Müller, M., 2016. Measurements and CFD modeling of a pulverized coal flame with emphasis on ash deposition. Fuel 167, 168–179.

Brandenburg, A., 2001. The inverse cascade and nonlinear alpha-effect in simulations of isotropic helical hydromagnetic turbulence. The Astrophysical Journal 550 (2), 824.

Brandenburg, A., Dobler, W., 2002. Hydromagnetic turbulence in computer simulations. Computer Physics Communications 147 (1-2), 471–475.

Cunningham, E., 1910. On the velocity of steady fall of spherical particles through fluid medium. Proceedings of the Royal Society of London A 83 (563), 357–365.

Davies, C., 1945. Definitive equations for the fluid resistance of spheres. Proceedings of the Physical Society 57 (4), 259.

Douglas, P. L., Ilias, S., 1988. On the deposition of aerosol particles on cylinders in turbulent cross flow. Journal of aerosol science 19 (4), 451–462.

García Pérez, M., Vakkilainen, E., Hyppänen, T., 2016. Unsteady CFD analysis of kraft recovery boiler fly-ash trajectories, sticking efficiencies and deposition rates with a mechanistic particle rebound-stick model. Fuel 181, 408–420.

Haugen, N. E. L., Brandenburg, A., 2006. Hydrodynamic and hydromagnetic energy spectra from large eddy simulations. Physics of Fluids 18 (7), 075106.

Haugen, N. E. L., Kragset, S., 2010. Particle impaction on a cylinder in a crossflow as function of Stokes and Reynolds numbers. Journal of Fluid Mechanics 661, 239–261.

Haugen, N. E. L., Kragset, S., Bugge, M., Warnecke, R., Weghaus, M., 2013. MSWI super heater tube bundle: Particle impaction efficiency and size distribution. Fuel Processing Technology 106, 416–422.

Homann, H., Guillot, T., Bec, J., Ormel, C., Ida, S., Tanga, P., 2016. Effect of turbulence on collisions of dust particles with planetesimals in protoplanetary disks. Astronomy & Astrophysics 589, A129.

Huang, L., Norman, J., Pourkashanian, M., Williams, A., 1996. Prediction of ash deposition on superheater tubes from pulverized coal combustion. Fuel 75 (3), 271–279.

Ingham, D. B., Hildyard, L. T., Hildyard, M. L., 1990. On the critical stokes' number for particle transport in potential and viscous flows near bluff bodies. Journal of Aerosol Science 21 (7), 935–946.

Israel, R., Rosner, D. E., 1982. Use of a generalized stokes number to determine the aerodynamic capture efficiency of non-stokesian particles from a compressible gas flow. Aerosol Science and Technology 2 (1), 45–51.

Kasper, G., Schollmeier, S., Meyer, J., Hoferer, J., 2009. The collection efficiency of a particle-loaded single filter fiber. Journal of Aerosol Science 40 (12), 993–1009.

Li, X., Zhou, H., Cen, K., 2008. Influences of various vortex structures on the dispersion and deposition of small ash particles. Fuel 87 (7), 1379–1382.

Lodato, G., Domingo, P., Vervisch, L., 2008. Three-dimensional boundary conditions for direct and large-eddy simulation of compressible viscous flows. Journal of Computational Physics 227 (10), 5105 – 5143.

MATLAB, 2016. Matlab and statistics toolbox release 2016a. MathWorks, Inc., Massachusetts, United States.

Mitra, D., Haugen, N. E. L., Rogachevskii, I., 2018. Turbophoresis in forced inhomogeneous turbulence. The European Physical Journal Plus 133 (2), 35.

Phillips, C., Kaye, S., 1999. The influence of the viscous boundary layer on the critical stokes number for particle impaction near a stagnation point. Journal of aerosol science 30 (6), 709–718.

Poinsot, T. J., Lele, S., 1992. Boundary conditions for direct simulations of compressible viscous flows. Journal of Computational Physics 101 (1), 104–129.

Schweers, E., Umhauer, H., Löffler, F., 1994. Experimental investigation of particle collection on single fibres of different configurations. Particle & Particle Systems Characterization 11 (4), 275–283.

Squires, K. D., Eaton, J. K., 1991. Preferential concentration of particles by turbulence. Physics of Fluids A: Fluid Dynamics 3 (5), 1169–1178.

The Pencil Code, 2018. (pencil-code.nordita.org [Internet]). Stockholm (SE): NORDITA; [updated January 2018]. Available from: https://github.com/pencil-code.

Weber, R., Mancini, M., Schaffel-Mancini, N., Kupka, T., 2013a. On predicting the ash behaviour using computational fluid dynamics. Fuel Processing Technology 105, 113–128.

Weber, R., Schaffel-Mancini, N., Mancini, M., Kupka, T., 2013b. Fly ash deposition modelling: Requirements for accurate predictions of particle impaction on tubes using RANS-based computational fluid dynamics. Fuel 108, 586–596.

White, F. M., 2006. Viscous Fluid Flow, 3rd Edition. McGraw-Hill Higher Education Boston.

Williamson, C. H. K., 1996. Vortex dynamics in the cylinder wake. Annual Review of Fluid Mechanics 28, 477–539.

Williamson, J., 1980. Low-storage runge-kutta schemes. Journal of Computational Physics 35 (1), 48–56.

Yilmaz, S., Cliffe, K., et al., 2000. Particle deposition simulation using the CFD code FLUENT. Journal of the Institute of Energy 73 (494), 65–68.

Yoo, C. S., Wang, Y., Trouvé, A., Im, H. G., 2005. Characteristic boundary conditions for direct simulations of turbulent counterflow flames. Combustion Theory and Modelling 9 (4), 617–646.

Yoshimoto, H., Goto, S., 2007. Self-similar clustering of inertial particles in homogeneous turbulence. Journal of Fluid Mechanics 577, 275–286.

Paper IV [2]

High-order overset grid method for detecting particle impaction on a cylinder in a cross flow

Jørgen R. Aarnes, Nils E. L. Haugen and Helge I. Andersson

Manuscript submitted for publication.

High-order overset grid method for detecting particle impaction on a cylinder in a cross flow

Jørgen R. Aarnes[a], Nils E. L. Haugen[a,b], Helge I. Andersson[a]

[a] *Department of Energy and Process Engineering, Norwegian University of Science and Technology, 7034 Trondheim, Norway*
[b] *SINTEF Energy Research, 7465 Trondheim Norway*

Abstract

An overset grid method was used to investigate the interaction between a particle-laden flow and a circular cylinder. The overset grid method is implemented in the Pencil Code, a high-order finite-difference code for compressible flow simulation. High-order summation-by-part operators were used at the cylinder boundary, and both bi-linear Lagrangian and bi-quadratic spline interpolation was used to communicate between the background grid and the body-conformal cylindrical grid. The performance of the overset grid method was assessed to benchmark cases of steady and unsteady flows past a cylinder. For steady flow at low Reynolds number, high-order accuracy was achieved for velocity components. Results for flow in the vortex shedding regime showed good agreement to the literature. The method was also applied to particle-laden flow simulations, where spherical point particles were inserted upstream of the cylinder. These inertial particles were convected towards and (possibly) past the cylinder. The simulations reproduced data from the literature at a significantly reduced cost, revealing that the previously published DNS data is less accurate than assumed for particles with very small Stokes numbers.

Keywords: overset grids, particle-laden flow, high-order, finite-difference, particle impaction, compressible fluid dynamics

[*] Corresponding author: Jørgen Røysland Aarnes
Email address: `jorgen.r.aarnes@ntnu.no` (Jørgen R. Aarnes)

1. Introduction

A common flow problem in numerical simulations is flow past a bluff body. Obstructions in the flow include (but are not limited to) spheres, flat plates, circular, rectangular or elliptical cylinders, triangles, spheroids, and complex geometries made out of a combination of these shapes. Particle-laden flows interacting with such obstacles are important for a range of applications. Whether the goal is to maximize the particle extraction from the flow, as for filter applications, or to minimize particle attachment on the object to avoid an insulating layer, as for biomass boilers, understanding the mechanics of inertial particles helps improve design, and hence, the efficiency of said applications. Accurate prediction of particle behavior in the vicinity of bluff bodies requires highly accurate boundary layer representation within numerical simulations. Finding the numerical method best suited to this task is not trivial, and can have a huge impact on both the efficiency and accuracy of simulations.

1.1. Representing solid objects in the flow

For generic shapes (cylinders, spheres, plates, etc.) body-fitted structured meshes are commonly used to accurately resolve the solid boundary. These methods use grids that conform to the solid (or solids) immersed in the flow and to other physical boundaries of the domain (inlet/outlet, walls, etc.). Depending on the domain geometry, this may require some deformation of the grid to conform to the boundaries, in addition to the mapping of the flow domain onto a simple computational domain. The result may lead to unnecessary local variations in the grid and rather time consuming grid generation [1]. Alternatively, unstructured meshes can be applied to resolve the solid boundaries in the flow. Unstructured meshes provide the highest flexibility in adapting a mesh to the flow problem, and are a good alternative for complex geometries when finite-volume or finite-element formulations of the governing equations are used [2]. Among the disadvantages of such grids are much larger storage requirements [3] and the need for intricate mesh generation techniques [4].

An alternative to body-fitted grids are non-conforming (typically Cartesian) meshes, where a solid in the flow is represented by a change in the fluid equations in the vicinity of the solid boundary. One such method, which has gained vast popularity over the last decades, is the immersed boundary method (IBM). This method (or rather, this class of methods) was originally developed to model flow around heart valves [5] by allowing for the representation of bluff bodies in the flow without using a body-conformal grid. A simple Cartesian grid can be used, where the boundary conditions (the sharp interface) of the bluff body are incorporated into the solver by a modification of the equations in the vicinity of the boundary (see review article by Mittal & Iaccarino [6] and references therein for details). This makes IBMs very flexible for representing bluff bodies, and particularly well-suited to complex geometries, where the use of body-fitted structured meshes is limited. A caveat to the IBM is the difficulty in achieving high-order accuracy near boundaries that do not conform to physical boundaries. For complex geometries this may be regarded as a necessary loss in order to be able to represent the boundary. For flow past simple geometric, bodies other methods may be more suitable, especially when the accuracy in the vicinity of the surface is a major concern.

Roughly ten years after the development of the IBM, a method of multiple grids *overset* on top of one another was proposed to represent solids in a flow (see [7, 8, 9]). Overset grids, or Chimera methods, employ body-conformal grids at the bluff bodies, but the body-conformal grids do not extend to the domain boundaries. Instead, a non-conforming background grid (typically uniform Cartesian) is used, and updated flow information within overlapping grid regions is communicated between grids at every time step. In this way, the flow simulation is split into multiple sub-simulations, one for each grid, and the boundaries of one grid are updated with information from the other grids. The background grid is used to compute the general flow field outside the smaller body-fitted grids, and the communication between grids is done through interpolation.

Overset grid methods have the advantage of being highly accurate at the solid-fluid interface. This is due to the use of body-fitted grids in these regions, and the flexibility in grid stretching made possible when several grids are used. At the same time, no grid deformation is necessary to conform to domain boundaries, due to the use of an appropriate non-conformal background grid. If the domain is circular, a cylindrical grid can be used as a background grid, if rectangular, a Cartesian grid, etc.

The communication between the grids is the limiting factor in terms of the accuracy of overset grid methods. In general, the interpolation of flow variables is detrimental to mass conservation (although conservative, mass correcting overset grid methods do exist for finite-volume codes, see e.g. [10, 11]). Using high-order interpolation between grids have proved beneficial in regards to the overall accuracy and stability of the overset grid method for both finite-difference and finite-volume implementations [12, 13, 14]. While advantageous in terms of accuracy, high-order interpolation techniques have the disadvantage of increases in complexity, inter-processor communication and floating-point operations, when compared to low-order interpolation schemes. Furthermore, straightforward extension to high-order interpolation, typically from second-order to fourth-order Lagrangian interpolation, does not guarantee a better solution. Possible overshoots in the interpolation polynomials may have a devastating impact on the interpolation accuracy. The applied interpolation scheme should therefore be evaluated for the specific flow problem at hand. For overset grid implementations, several interpolation schemes are available. In this study two such schemes are compared: bi-linear Lagrangian interpolation and bi-quadratic spline interpolation. Together with high-order low-pass filtering, the resulting computations were both stable and accurate. This topic is further discussed in Section 2.

If several body-fitted grids overlap, the overset grid computations become increasingly difficult, particularly in regards to the communication between the different grids. For the purposes of this paper, the discussion is limited to a single body-fitted grid on top of a Cartesian background grid. For more general discussion on overset grids, see [15] or [16].

1.2. Particle impaction

When considering particle deposition on a surface, two mechanisms are required for a particle to deposit. The particle must first impact the surface, that is, it must physically contact the surface, and then it must adhere to the surface. Only the first of these two mechanisms will be the focus of this study. Hence, all particles that come into contact with the bluff body are considered to have been absorbed by it. Further, only inertial impaction is considered. Any other particle impaction mechanisms including Brownian motion, thermophoresis and turbulent diffusion are omitted. Note that this is not an acceptable omission in non-isothermal flows, where the effects of temperature will be large on small particles (see [17, 18]).

The impaction efficiency $\eta = N_{imp}/N_{ins}$ is a measure of the cylindrical object's ability to capture the particles that are initially incident on the cylinder. The number of impacting particles is given by N_{imp}, while N_{ins} is the count of particles with a center of mass that is initially moving in the direction of the solid object. Note that following this convention may lead to $\eta > 1$, even if no forces act on the particles, since a particle may follow a path close enough to be intercepted by the object, due to its finite size, even though the center of mass does not hit the object.

A fluid flow will be deflected by the object, and particles in the flow will experience a drag force. This force will accelerate the particles along the fluid trajectory, leading particles away from the bluff body. The particle Stokes number, $St = \tau_p/\tau_f$, where τ_p and τ_f are particle and fluid time scales, respectively (details in Section 4), can be considered a measure of particle inertia. Hence, particles with a small Stokes number follow the flow to a larger extent than particles with a large Stokes number. By using potential flow theory to compute the flow past a circular cylinder, Israel & Rosner [19] determined a relation for the impaction efficiency as a function of the Stokes number. The predictions by Israel & Rosner are inaccurate in predicting particle impactions for flows where the viscous boundary layer of the cylinder plays an significant role. This is because potential flow theory assumes inviscid flow. In particular, this theory is insufficient at predicting particle impactions for particles with small Stokes numbers, and for moderate Reynolds number flows. Here, the Reynolds number is defined as $Re = U_0D/\nu$, where U_0 is the mean flow velocity, D is the diameter of a cylinder (the bluff body in the flow) and ν is the kinematic viscosity. Haugen & Kragset [20] performed simulations using the Pencil Code to compute inertial particle impaction on a cylinder in a crossflow for different Stokes and Reynolds numbers. Later, Haugen et al. [21] performed a similar study on a flow with multiple cylinders, in order to emulate impaction on a super-heater tube bundle. The impaction efficiencies obtained by Haugen & Kragset [20] have been used as benchmarking results, but were limited to moderate Reynolds numbers and two-dimensional flows. Part of the reason for this limitation is the use of an immersed boundary method that requires a very fine grid to achieve the required accuracy.

1.3. Present

The purpose of this paper is to introduce an overset grid method applicable to compressible particle-laden flows past a circular cylinder, and to assess its performance in benchmarking cases and a true particle-laden flow simulation. The method has been implemented in the open source compressible flow solver known as the Pencil Code [22, 23], with the aim to improve the accuracy in the vicinity of the cylinder and to reduce the computational cost of particle-laden flow simulations.

The structure of the paper as follows: In Section 2 the equations governing the flow and the bluff body representation are described. An assessment of the accuracy of the method for steady and unsteady flow past a cylinder is given in Section 3. In Section 4 the capabilities of the overset grid method are demonstrated by simulating particle-laden flows interacting with a bluff body at a moderate Reynolds number. The results and the computational costs are compared with those of Haugen & Kragset [20], before concluding remarks are given in Section 5.

2. Methodology

2.1. Governing equations

The governing equations of the flow are the continuity equation:

$$\frac{\mathrm{D}\rho}{\mathrm{D}t} = -\rho\boldsymbol{\nabla}\cdot\boldsymbol{u}\,, \tag{1}$$

and the momentum equation:

$$\rho\frac{\mathrm{D}\boldsymbol{u}}{\mathrm{D}t} = -\boldsymbol{\nabla}p + \boldsymbol{\nabla}\cdot\left(2\mu\boldsymbol{S}\right)\,, \tag{2}$$

where ρ, t, $\boldsymbol{u}$ and p are the density, time, velocity vector and pressure, respectively, and $\mu = \rho\nu$ is the dynamic viscosity. The compressible strain rate tensor $\boldsymbol{S}$ is given by:

$$\boldsymbol{S} = \frac{1}{2}\left(\boldsymbol{\nabla}\boldsymbol{u} + (\boldsymbol{\nabla}\boldsymbol{u})^T\right) - \boldsymbol{I}\left(\frac{1}{3}\boldsymbol{\nabla}\cdot\boldsymbol{u}\right)\,, \tag{3}$$

where $\boldsymbol{I}$ is the identity matrix. The pressure is computed using the isothermal ideal gas law, $p = c_s^2\rho$, where c_s is the speed of sound. The flow is isothermal and weakly compressible, with a Mach number of ~ 0.1 for all simulations. With a constant speed of sound (for the isothermal case) and a constant kinematic viscosity, the momentum equation to be solved on the overset grids is:

$$\frac{\mathrm{D}\boldsymbol{u}}{\mathrm{D}t} = -c_s^2\boldsymbol{\nabla}\left(\ln\rho\right) + \nu\left(\boldsymbol{\nabla}^2\boldsymbol{u} + \frac{1}{3}\boldsymbol{\nabla}\left(\boldsymbol{\nabla}\cdot\boldsymbol{u}\right) + 2\boldsymbol{S}\cdot\boldsymbol{\nabla}\left(\ln\rho\right)\right)\,. \tag{4}$$

The governing equations were discretized with sixth-order finite-differences in space and a third-order memory efficient Runge-Kutta scheme in time [24].

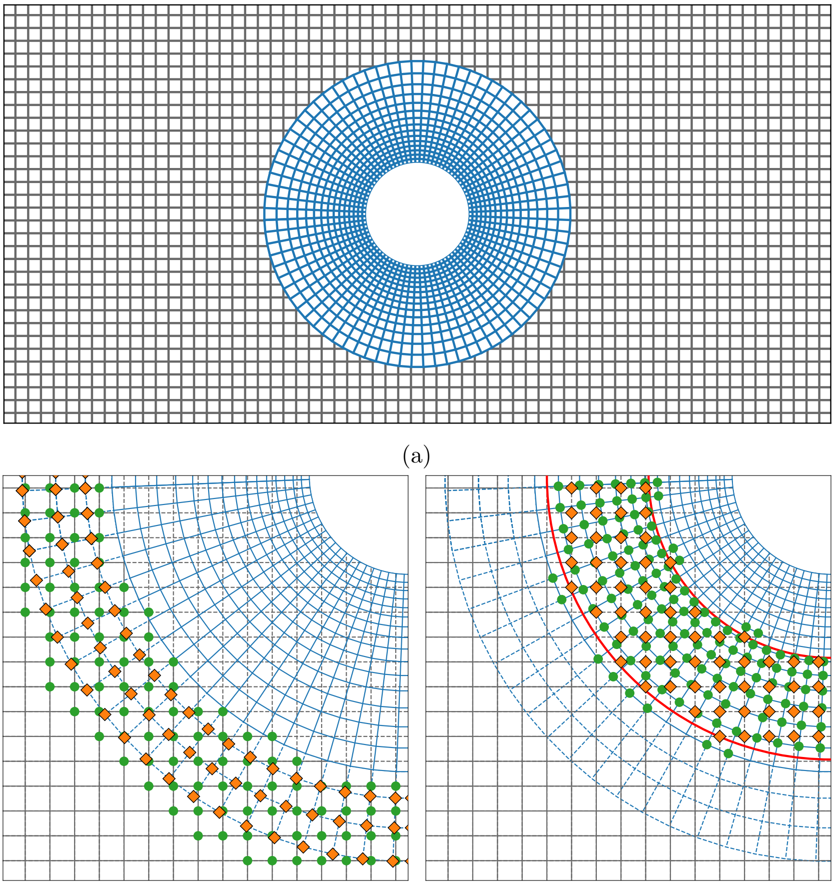
The flow was simulated on a rectangular domain with an inlet at the bottom and flow in the vertical direction. The circular cylinder was situated in the center of the domain, with the following boundary conditions: no-slip and impermeability for velocity, and zero gradient in the radial direction for the density. The latter condition was derived from the ideal gas law and the boundary layer approximation $\left(\frac{\partial p}{\partial n} = 0\,, \text{where } n \text{ is the wall normal direction}\right)$ for an isothermal flow. Navier-Stokes characteristic boundary conditions were used both at the inlet and at the outlet of the flow domain. This boundary condition is a formulation that makes use of one-dimensional characteristic wave relations to allow acoustic waves to pass through the boundaries [25, 26]. The remaining domain boundaries were periodic.

2.2. Overset grids

To resolve the flow domain using an overset grid method, a cylindrical coordinate grid was body-fitted to the cylinder, and a uniform Cartesian grid was used as the background grid (see Fig. 1a). The cylindrical grid was streched in the radial direction. In the region where fluid data is communicated between grids, it is beneficial that the grids have similar spacing. Grid stretching enables similar grid spacing in the interpolation region and a much finer grid near the cylinder surface.

The compressibility of the flow lead to a strict stability limit for the Runge-Kutta method, imposing a very small time step in the simulations. Because the overset grids is effectively solving two different flow problems, coupled only by the communication between the grids, there us flexibility in the choice of time step. Choosing a time step on the background grid that is small enough to guarantee stability for the Cartesian grid spacing, and choosing a smaller time step on the cylindrical grid reduces the overall computational cost significantly. The cylindrical grid time step must be a multiple of the background grid time step to ensure that the computations on each grid are synchronized. An implicit solver may be beneficial, if the grid spacing near the cylinder is several orders of magnitude smaller than that of the background grid, but this is beyond the scope of this study.

The communication between the grids in the overset grid simulation was completed in two stages for each time step of the background grid. At each stage of the communication, the required flow properties were interpolated from donor-points to fringe-points. Each grid requires a zone of fringe-points at least three points deep, such that the seven point central difference stencil



(a)

(b)

(c)

Figure 1: Overset grid method: (a) Cylinder grid on top of background grid (fringe-points of cylinder grid and background grid points within cylinder grid radius not shown). (b) Communication between grids, interpolation from Cartesian donor-points to cylindrical fringe-points. (c) Communication between grids, interpolation from cylindrical donor-points to Cartesian fringe-points. Four donor-points (●) surround each fringe-point (◆) in bi-linear interpolation. Dashed lines used where variables are not computed by finite-differences (fringe-points and hole-points).

could be used without any special handling of points adjacent to the fringe-points. For a curvilinear grid, the fringe-points were simply the three outer points at each radial grid line (see Fig. 1b). For the Cartesian grid, the fringe-points must be identified, typically during pre-processing, in order to include all grid points within a fixed area in the region covered by both the Cartesian and the cylindrical grid. This is set by an inner and outer radius defining the interpolation region, see red lines enclosing fringe-points on the Cartesian grid in Fig. 1c. Cartesian grid points that are closer to the solid than the inner radius of the fringe-point zone (or inside the solid), are hole-points. The hole-points are not used in the computations.

In the overset grid method implemented in the Pencil Code, there is no overlap between the two interpolation regions of Figs. 1b and 1c. That is, no fringe-points are used as donor points. Hence, the interpolation is explicit, not implicit [16]. Note that if the bluff body enclosed by the body-fitted grid was moving, the cost of inter-grid communication would be significantly increased due to the cost related to identifying new fringe and donor-points on the background grid at each new position of the bluff body.

At present, two types of interpolation are implemented for overset grid communication in the Pencil Code: bi-linear Lagrangian interpolation and bi-quadratic spline interpolation. Both methods have the advantage of avoiding oscillations in the interpolation interval, which is a common problem for high-order interpolation. The Lagrangian interpolation is a second-order accurate scheme, while the spline interpolation is third-order accurate. The illustration of donor-points and fringe-points in Figs. 1b and 1c is for Lagrangian interpolation, where each fringe-point on one grid is interpolated from the 2×2 surrounding donor-points of the other grid. For spline interpolation, a zone of the 3×3 closest grid points are used as donor-points for interpolation of each fringe-point. Note that the interpolation is bi-linear or bi-quadratic in both two- and three-dimensions. This is due to the Cartesian and cylindrical grid having a shared z-plane, hence no interpolation is required in the z-direction.

At the solid-fluid interface, summation-by-parts finite-difference operators are used to enhance stability for unsteady flow simulations (an unsteady wake develops for $Re > 47$). These operators are third-order accurate for the sixth-order centered finite-difference method. Details on these operators can be found in [27] (first derivatives) and [28] (second derivatives).

The centered finite-difference schemes are non-dissipative, which can be detrimental due to the potential growth of high-frequency modes, leading to numerical instability. To some extent, the summation-by-parts boundary conditions suppress such instabilities, but are not sufficient to suppress all oscillations in the solution on the curvilinear stretched grid. In particular, such oscillations are prominent in the density field. The detrimental effect of the high-frequency modes increases as the grid spacing decreases, which may lead to diverging solutions as the grid is refined. To suppress the high-frequency modes, a high-order low-pass filter is used on the curvilinear portion of the overset grid. The filter is a 10th order Padé filter, with boundary stencils of 8th and 6th order. On the interior of the domain, the filter is given by:

$$\alpha_f \hat{\phi}_{i-1} + \hat{\phi}_i + \alpha_f \hat{\phi}_{i+1} = \sum_{n=0}^{N} \frac{\alpha_n}{2} (\phi_{i+n} + \phi_{i-n}) \,, \tag{5}$$

where $\hat{\phi}_k$ and ϕ_k are components k of the filtered and unfiltered solution vectors, respectively, α_f is a free parameter ($|\alpha_f| \leq 0.5$) and α_n are fixed parameters dependent only on α_f [29]. Boundary stencils can be found in Gaitonde and Visbal [30]. The Padé filter is implicit, and requires the solution of a tri-diagonal linear system at grid points in the radial direction, and a cyclic tri-diagonal system at every grid point in the direction tangential to the surface. The free parameter a_f was set to 0.1, where filtering the solution once per Cartesian time step was found to be sufficient for a stable and accurate solution.

3. Performance

3.1. Assessment of accuracy

The spatial accuracy of the overset grid method was examined by simulating a steady flow past a circular cylinder at a Reynolds number of 20. A domain of size $L_x \times L_y = 10D \times 10D$ was used. The diameter of the curvilinear, body-fitted grid (henceforth called the cylinder grid) was three times the cylinder diameter.

An indicative measure of the accuracy of the method can be found by computing solutions on several grid refinement levels, and using the finest grid as the "correct solution" when computing two-norm errors. The grids used in this accuracy assessment are listed in Tab. 1. An odd number of grid points was used in the directions that were not periodic, in order to have grid points that are aligned at each refinement level. A fixed (dimensionless) time step $\Delta t = 0.25 \times 10^{-5}$ was used for the Cartesian grid computations

Table 1: Grid refinement levels used in the assessment of accuracy of the overset grid method.

Refinement level	Cylinder grid $N_r \times N_\theta$	Cartesian grid $N_x \times N_y$
0	17×80	80×81
1	33×160	160×161
2	65×320	320×321
3	129×640	640×641

at all refinement levels. The small time step ensured that there was no violation of diffusive or advective time step restrictions on any of the grids. These restrictions are $\Delta\tau \leq C_\nu \Delta\chi_{min}^2/\nu$ and $\Delta\tau \leq C_u \Delta\chi_{min}/\left(|\boldsymbol{u}| + c_s\right)$, respectively, where $\Delta\tau$ is the dimensional time step, $\Delta\chi_{min}$ is the smallest grid spacing in any direction, and C_ν and C_u are the diffusive and advective Courant numbers, respectively.

Hyperbolic sine functions were used for the stretching in the radial direction. The grid stretching parameters were set such that the ratio between the grid spacing normal and tangential to the surface was approximately one, both in the vicinity of the solid surface and in the interpolation region in the outer part of the cylinder grid. Furthermore, the number of grid points in the Cartesian and cylindrical grids were chosen in order to have similar grid spacings in the region of inter-grid interpolation. The resulting local time step on the cylindrical grid was $\Delta t_c = 0.2\Delta t$.

The main objective of the method, is to compute a very accurate boundary layer around the cylinder. This is crucial for the application to particle impaction simulations in Section 4 and in future studies. The L_2-error norms of flow variables are therefore considered along strips tangential to the cylinder surface as close as possible to the surface, in order to get an indication of the accuracy of the scheme in the boundary layer. Figure 2 depicts L_2-error norms of the density and the normal and tangential velocity components (with respect to the cylinder surface), computed with the two different interpolation methods. The norms were computed along a strip around the cylinder, at the grid point closest the cylinder for the refinement level 0 (this corresponds to the 2nd point from the cylinder for refinement level 1, 4th for level 2, etc.). Note that the computations with spline interpolation did not fully converge to a stable solution at the coarsest grid level, as indicated by

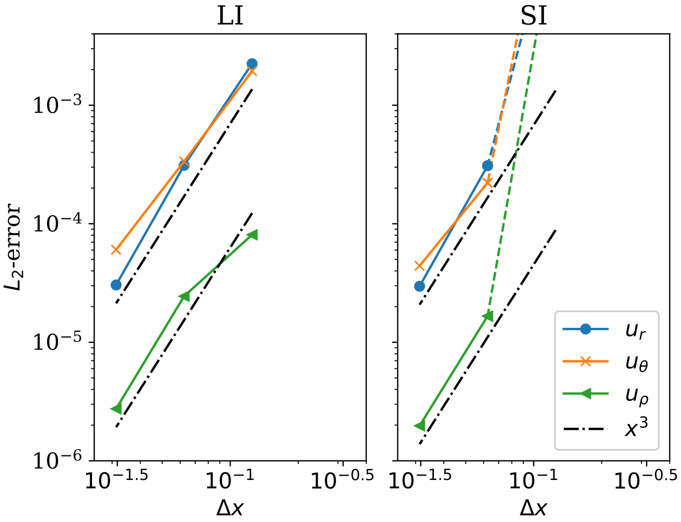


Figure 2: L_2-error norms of u_r, u_θ and ρ at varying refinement levels at the grid point closest to the cylinder surface (for the coarsest grid). Results are for the computations with bi-linear Lagrangian interpolation (LI) and bi-quadratic spline interpolation (SI), with Δx (non-dimensional) grid spacing on the Cartesian grid.

the dashed lines between the first refinement results in Fig. 2.

For both interpolation methods computation of the density was third-order, the radial velocity component was between third- and fourth-order and the tangiantal velocity component was between second- and third-order accurate, at the grid point closest to the surface on the coarsest grid. The results suggest that the difference in accuracy between the interpolation methods is negligible in the immediate vicinity of the cylinder.

For a more detailed picture of the formal order of accuracy of the overset grid method, consider Fig. 3. This figure depicts the formal order of accuracy P, of the density and velocity components, computed along strips at increasing distance from the cylinder boundary (cylinder boundary at $r_c = 0.5$). The computations are based on the assumption that the L_2-error norm on a grid with grid spacing Δx can be expressed as $L_2(\Delta x) \sim \Delta x^p$, such that the order of accuracy P can be computed by:

$$P = \frac{\log(L_2(\Delta x)/L_2(\Delta x/2))}{\log 2} \,. \tag{6}$$

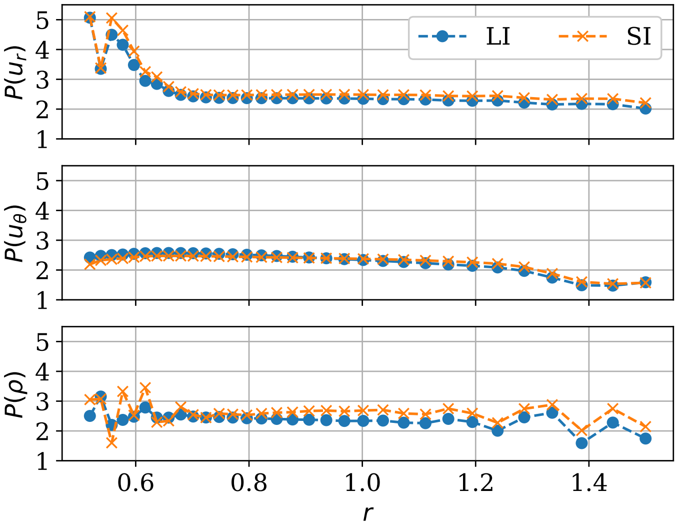


Figure 3: Formal order of accuracy of velocity components computed along strips tangential to the cylinder surface at non-dimensional radial position r, for upper refinement levels for flow with $Re = 20$ with Lagrangian interpolation (upper) and spline interpolation (lower).

In principle, the spline interpolation scheme is third-order accurate while the Lagrangian interpolation a second-order accurate method. The effect of using the different methods of interpolation can be seen in Fig. 3. Some effect of the interpolation is seen, when considering the entire flow domain considered by the cylindrical grid. The difference is, however, much smaller than the difference in accuracy between the two interpolation schemes. The difference in order of accuracy of the radial velocity computations is 0.02–0.56, for which spline interpolation yielded the highest order (median $P = 2.49$ with spline interpolation, $P = 2.42$ with Lagrangian interpolation). A similar difference can be seen for the density. For the tangential velocity, on the other hand, there is no obvious best method. In the vicinity of the cylinder surface the difference between the Lagrangian and spline interpolation is negligible. Notice that the radial velocity component was computed with a very high order of accuracy, $P \approx 5$ in this region. This is significantly more accurate than the more conservative suggestion of radial velocity accuracy between third- and fourth-order, seen in Fig. 2 (the results in Fig. 2 correspond to the second point from the left in Fig. 3).

The consideration of formal order of accuracy shows that the overset grid method is a high-order method ($P > 2$). No obvious distinction among the two interpolation schemes was found by this analysis, although the spline interpolation appeared to have a marginally higher accuracy for the density and radial velocity component.

3.2. Unsteady flow

The L_2-error norms are suggestive of the formal accuracy of the numerical method, but do not reveal the in-use accuracy of the method for simulations in the unsteady flow regime. This must be determined, before arriving at a full-blown simulation of a particle-laden flow interacting with a cylinder in this flow regime.

A grid refinement study was performed for $Re = 100$, where unsteady vortex shedding developed in the cylinder wake. A domain with $L_x \times L_y = 10D \times 20D$ was used, with the cylinder in the center of the domain. The resulting mean drag coefficient (C_D), root-mean-square lift coefficient (C'_L) and Strouhal number (Str) were computed. The drag and lift coefficients were computed using the pressure and shear forces on the cylinder, $\boldsymbol{F_p}$ and $\boldsymbol{F_s}$, respectively, as given by:

$$\boldsymbol{F_p} = -\int p\big|_{r_c} \boldsymbol{dA} \approx -\hat{\boldsymbol{r}} h r_c \Delta\theta \sum_{i=1}^{N_\theta} p(r_c, \theta_i) \ , \tag{7}$$

$$\boldsymbol{F_s} = \int \boldsymbol{\sigma}\big|_{r_c} dA \approx \hat{\boldsymbol{\theta}} \nu h r_c \Delta\theta \sum_{i=1}^{N_\theta} \rho(r_c, \theta_i) \left.\frac{\partial u}{\partial r}\right|_{(r_c, \theta_i)} \ , \tag{8}$$

where h is the height of the cylinder and $\boldsymbol{\sigma}$ is the shear stress. With flow in the y-direction, the drag and lift forces, F_D and F_L, were found by taking the sum of the pressure and shear forces in y- and x-direction, respectively. These forces can be used to calculate the drag and lift coefficients as follows:

$$C_D = \frac{F_D}{\frac{1}{2}\rho_0 U_0^2 A} \ , \tag{9}$$

$$C_L = \frac{F_L}{\frac{1}{2}\rho_0 U_0^2 A} \ , \tag{10}$$

Table 2: Grid refinement levels used in the grid refinement study for flow past a cylinder at $Re = 100$ with two differently sized cylindrical grids. Grid spacing Δr non-dimensionalized by the cylinder diameter.

Refinement	Δr_{min}	$r_{cg} = 3r_c$		$r_{cg} = 5r_c$	
level	$\times 10^{-2}$	$N_r \times N_\theta$	$N_x \times N_y$	$N_r \times N_\theta$	$N_x \times N_y$
0	4.1	16 × 80	80 × 160	24 × 80	50 × 100
1	2.7	24 × 120	120 × 240	36 × 120	76 × 152
2	2.0	32 × 160	160 × 320	48 × 160	100 × 200
3	1.6	40 × 200	200 × 400	60 × 200	128 × 256
4	1.3	48 × 240	240 × 480	72 × 240	150 × 300
5	0.97	64 × 320	320 × 640	96 × 320	200 × 400
6	0.77	80 × 400	400 × 800	120 × 400	256 × 512
7	0.64	96 × 480	480 × 960	144 × 480	306 × 612
8	0.48	128 × 640	640 × 1280	192 × 640	408 × 816

where ρ_0 and U_0 are free-stream values of the density and velocity, respectively, and $A = 2hr_c$ is the projected frontal area of the cylinder. The Strouhal number is simply the shedding frequency, non-dimensionalized by the free-stream velocity and cylinder diameter.

A grid refinement study of the unsteady flow was performed with both Lagrangian and spline interpolation on two differently sized overset grids. One had a cylindrical grid with diameter $3D$ (the same size that was used in the assessment of accuracy for the $Re = 20$). The other had a size $5D$. Hence, there was a factor two difference in the radial length ($L_r = r_{cg} - r_c$, where r_{cg} is the outer cylinder grid radius) of the two cylindrical grids. At each refinement level, the smallest spacing in the radial direction was the same for the two different overset grids, and the stretching properties were the same as that in the $Re = 20$ flow simulations (approximately quadratic cells in the vicinity of the surface and the interpolation region, and approximately equal grid spacing on the Cartesian and curvilinear grid in the interpolation region). Hence, the outer grid spacing on the larger cylindrical grid was larger than the outer grid spacing of the smaller cylindrical grid. Thus, a coarser Cartesian grid could used for the overset grid with the larger cylinder grid. This, in turn, allowed for a larger time step on the background grid, but required more sub-cycles on the cylindrical grid for each Cartesian time step. Details on the grids used in this refinement study are listed in Tab. 2.

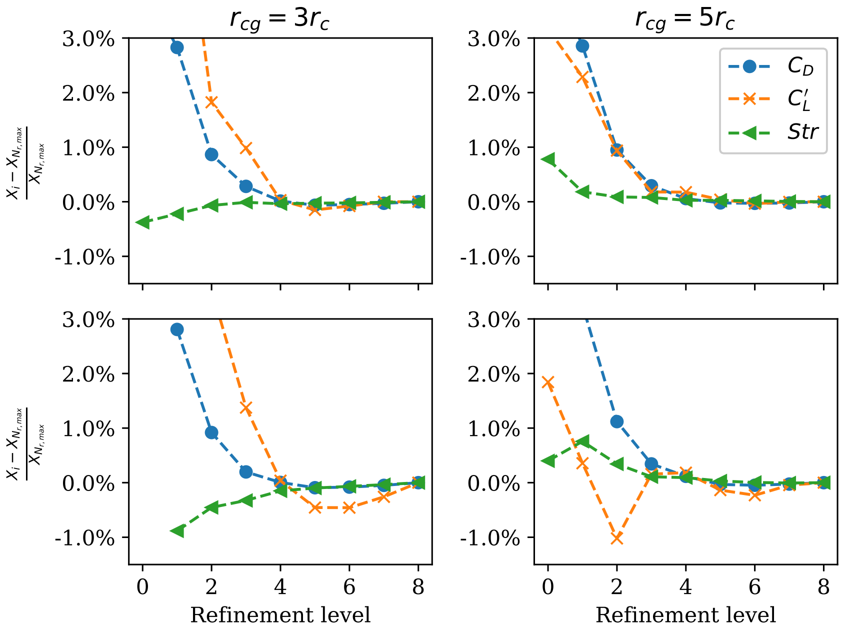


Figure 4: Normalized values for mean drag coefficient (C_D), rms-lift coefficient (C'_L) and Strouhal number (Str) for $Re = 100$ computed at different refinement levels (see Tab. 2) for overset grids with two sizes of radii for the body-fitted cylindrical grid, r_{cg}. Results are given for computations with Lagrangian interpolation (upper) and spline interpolation (lower).

Results for the grid refinement at $Re = 100$ can be seen in Fig. 4 and Tab. 3. In Fig. 4, the dimensionless drag and lift coefficients, and the Strouhal number have been normalized by the result computed at the finest grid. Hence, the plots depict the relative deviation from the result at grid refinement level eight from Tab. 2. The values of the coefficients computed at this refinement level, for each of the four cases, are given in Tab. 3.

The dimensionless numbers converged quite rapidly for all of the tested cases. The best performance for grid independency was achieved with Lagrangian interpolation. Yet, even the poorest result, the computation of the rms-lift coefficient at the smaller of the two cylindrical grids with spline interpolation deviated less than 0.5% from the finest grid result, for grid refinement level ≥ 4. Some deviation is also seen in the lift coefficient of results

Table 3: Mean drag coefficient (C_D), rms-lift coefficient (C_L') and Strouhal number (Str) for $Re = 100$ computed at a domain $L_x \times L_y = 10D \times 20D$ with two different overset grids. The resolution is given by the finest refinement levels in Tab. 2, and both Lagrangian (LI) and spline interpolation (SI) cases are considered.

	$r_{cg} = 3r_c$		$r_{cg} = 5r_c$	
Coefficient	LI	SI	LI	SI
C_D	1.460	1.458	1.461	1.461
C_L'	0.2509	0.2450	0.2527	0.2522
Str	0.1763	0.1762	0.1763	0.1763

computed on the large cylindrical grid with spline interpolation (less than 0.24% for refinement level ≥ 4). For the cases where Lagrangian interpolation was used for inter-grid communication, the deviation from results at refinement levels four to seven from the finest grid result is less than 0.15% for all coefficients. (if only drag and Strouhal number are considered, the deviation is less than 0.064% for these cases).

The difference between results obtained with quadratic spline and linear Lagrangian interpolation was particularly clear for the smaller cylinder grid. With a larger grid, it is not surprising that the effects from interpolation were reduced. Nevertheless, the best results on the larger grid were also achieved with Lagrangian interpolation. For the steady flow simulations the spline interpolation yielded results with somewhat higher order of accuracy. The sub-par performance of this interpolation for unsteady simulations may be due to the overshoots in this non-linear interpolation, or perhaps, a larger mass loss during interpolation. No further speculation is conducted here, but note that the Lagrangian interpolation outperformed the spline interpolation for simulations of unsteady flow.

By considering the grid independent solutions in Tab. 3, used to normalize the grid refinement results, two particular factors could be noted. Firstly, by comparing the results for the two different interpolation schemes on the domain where the cylindrical grid has $r_{cg} = 5r_c$, it is evident that the computed drag, lift and Strouhal number were independent of the inter-grid communication. This in contrast to the $r_{cg} = 3r_c$ results, but in accordance with an intuitive understanding of the problem: the farther away from the cylinder boundary the interpolation is performed, the less it affects computation of quantities at the boundary. Note, however, that even though the drag and lift forces were computed at the boundary, these coefficients were also dependent on the flow upstream and downstream of the cylinder. The results therefore suggest that the flow surrounding the cylinder was negligibly impacted by interpolation method selected when the larger r_{cg} was used for the cylinder grid.

By comparing the results for C_D and C'_L on the differently sized cylinder grids, computed with Lagrangian interpolation, the dependency on cylinder grid size was found to be small. There was a small difference in the computed lift coefficient (somewhat higher for the larger cylinder grid). Although the results are grid independent, neither of the values are quantitatively accurate for the drag or lift of a cylinder in a cross flow at $Re = 100$. The small difference in computed lift may be due to blockage effects or interpolation errors propagating across periodic boundaries.

To confirm that the grid independent solutions yielded accurate flow predictions a simulation was also conducted on a large domain, $L_x \times L_y = 50D \times 50D$, for the two different grid sizes used above. Since Lagrangian interpolation had the best performance for the unsteady flow simulations, only this interpolation procedure was used. The grid spacing corresponding to grid refinement level five in Tab. 2 was used on the large square domain. The computed flow quantities showed good agreement with previous studies performed on similar domain sizes (see Tab. 4). Note that for the simulations on a large square domain there was a negligible difference between the results from the different overset grid simulations. Because a smaller domain was used in the next section (with width $6D$), the smallest cylinder grid (with $r_{cg} = 3r_c$) was selected. Using a larger cylinder grid will reduce the total number of grid points in the simulations, which is a major advantage on large domains. On small flow domains, overset grids with interpolation regions very close to each side of the periodic boundary should not be used, as this may cause spurious interpolation errors to propagate across the boundary.

4. Particle deposition on a circular cylinder in a laminar cross flow

Direct numerical simulations with a large number of particles suspended in the flow have been performed to assess the performance of overset grids on a more complex and demanding simulation than the simple flow past a cylinder at low Reynolds numbers.

The particle deposition simulations are based on the study by Haugen & Kragset [20], where particle-laden flow simulations were performed over a

Table 4: Comparison with previous studies. The studies were performed on domains with streamwise length $60 \leq L_x/D \leq 100$ and spanwise length $40 \leq L_x/D \leq 100$, and the present study has $L_x = L_y = 50D$. Results from the present study are for domains covered by two differently sized overset grids, with inter-grid interpolation performed by bi-linear Lagrangian interpolation. The asterisk on some values of C_L' denotes where only the amplitude of the lift was given. The asterisk mark a lift amplitude scaled by $1/\sqrt{2}$ to get the root-mean-square lift coefficient, a valid scaling for the sinusoidal-like lift coefficient (with mean value zero).

	C_D	C_L'	St
Li et al. [31]	1.336	–	0.164
Posdziech & Grundmann [32]	1.350	$0.234^{(*)}$	0.167
Pan [33]	1.32	$0.23^{(*)}$	0.16
Qu et. al. [34]	1.326	0.219	0.166
Present, $r_{cg} = 3r_c$	1.346	0.235	0.166
Present, $r_{cg} = 5r_c$	1.346	0.234	0.166

range of Reynolds numbers on a moderately sized flow domain ($6D \times 12D$). The analysis is not repeated here, but a brief introduction to the method used for particle representation and deposition will be presented. The particle-laden flow simulations were performed on a domain exactly the same size as in Haugen & Kragset[20], with Reynolds number 100.

4.1. Particle equations

The particles are tracked using a Lagrangian formalism, where the particle velocity and position are described by:

$$\frac{\mathrm{d}\boldsymbol{v}_p}{\mathrm{d}t} = \frac{\boldsymbol{F}_{D,p}}{m_p}\,, \tag{11}$$

$$\frac{\mathrm{d}\boldsymbol{x}_p}{\mathrm{d}t} = \boldsymbol{v}_p\,, \tag{12}$$

where $\boldsymbol{v}_p$, $\boldsymbol{x}_p$ and m_p are the velocity, center of mass position and mass of the particle, respectively. The force $\boldsymbol{F}_{D,p}$ acting upon a spherical particle is the drag force:

$$\boldsymbol{F}_{D,p} = \frac{1}{2C_c}\rho C_{D,p} A_p \left|\boldsymbol{u} - \boldsymbol{v}_p\right| \left(\boldsymbol{u} - \boldsymbol{v}_p\right)\,, \tag{13}$$

where $A_p = \pi d_p^2/4$ is the cross sectional area of the particle and

$$C_c = 1 + \frac{2\lambda}{d_p}\left(1.257 + 0.4e^{(-1.1d_p/2\lambda)}\right) , \tag{14}$$

is the Stokes-Cunningham factor (with parameters set for air) for a particle with diameter d_p. The mean free path $\lambda = 67\,\text{nm}$ accounts for the fact that for very small particles, the surrounding medium can no longer be regarded as a continuum but rather as distinct particles. The particle drag coefficient is given by:

$$C_{D,p} = \frac{24}{Re_p}\left(1 + 0.15Re_p^{0.687}\right) , \tag{15}$$

for particle Reynolds number $Re_p = d_p\left|\boldsymbol{v}_p - \boldsymbol{u}\right|/\nu \lesssim 1000$. With this, the particle drag force can be re-written as

$$\boldsymbol{F}_{D,p} = \frac{m_p}{\tau_p}\left(\boldsymbol{u} - \boldsymbol{v}_p\right) , \tag{16}$$

where

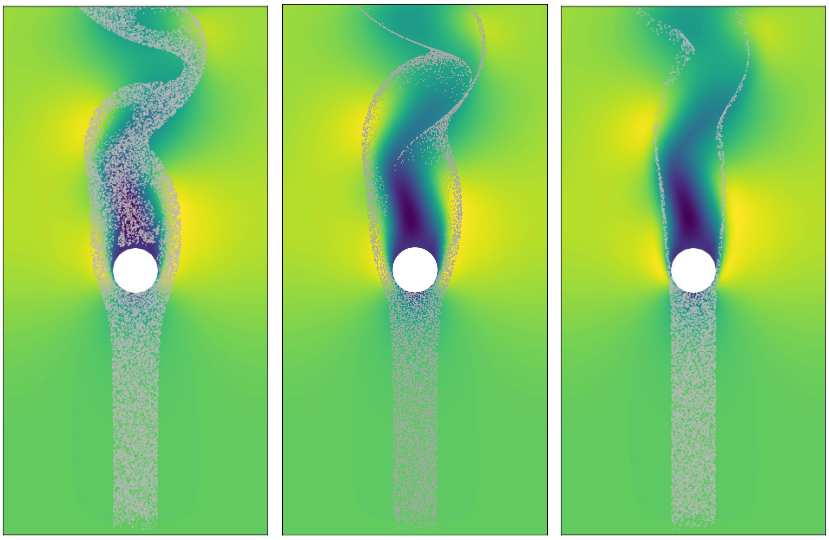
$$\tau_p = \frac{Sd_p^2C_c}{18\nu(1+f_c)} \tag{17}$$

is the particle response time, with $f_c = 0.15Re_p^{0.687}$ and $S = \rho_p/\rho$. Note that this is Stokes drag in the limit $C_c = 1$ and $f_c = 0$. Using the convention of [20], the Stokes number ($St = \tau_p/\tau_f$) is defined with a fluid time scale:

$$\tau_f = \frac{D}{2U_0} . \tag{18}$$

The fluid velocity was interpolated from surrounding grid points by bilinear interpolation on the Cartesian grid and bi-quadratic interpolation on the curvilinear grid. The order of the interpolation is higher on the curvilinear grid as the velocity components (the radial, in particular) are close to quadratic near the cylinder surface. For three-dimensional simulations, linear interpolation is used for the velocity component along the z-direction (the cylinder's spanwise direction) on all grids.

For particles very close to the cylinder surface, special handling was used to interpolate the radial component of the fluid velocity. Very close to the cylinder refers to within one grid point from the surface, or alternatively,



(a) $St = 0.1$ (b) $St = 1.0$ (c) $St = 10$

Figure 5: Particle-laden flow interacting with a circular cylinder at $Re = 100$. An unhindered particle will cross the flow domain, from the inlet (bottom) to the outlet (top) in approximately two shedding periods at this Reynolds number. Contours of the streamwise velocity component make up the background.

within the pre-calculated momentum thickness of the boundary layer. The special handling in use for particles at such positions was a quadratic interpolation that guarantees no overshoots. Since all velocities are zero at the surface, this was achieved by:

$$u_{r,p} = u_{i,g} \left(\delta r_p / \delta r_g\right)^2 \ , \tag{19}$$

where $u_{r,p}$ and $u_{r,g}$ are radial velocity components at the position of the particle and at the position of the interception between a surface normal and the first grid line away from the surface, respectively. The distances δ_p and δ_g are from the surface to the particle and to said grid line, respectively.

4.2. Particle impaction

After the flow developed into periodic vortex shedding, particles were inserted continuously at the inlet. The particles were inserted randomly, as a

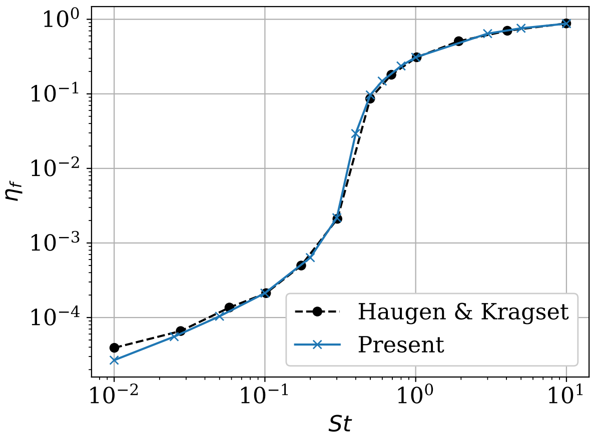


Figure 6: Front side impaction efficiency (η_f) as a function of Stokes number (St) for Reynolds number 100. Present results compared to a previous study by Haugen & Kragset [20]

homogeneous distribution over a rectangular cross-section encompassing particle trajectories that could impact the cylinder. From here the particles were convected downstream, and removed from the flow either by impacting the cylinder or by reaching the outlet (see Fig. 5). An impaction was registered (and the particle removed) if the distance between the cylinder surface and the particle's center of mass was less than or equal to $d_p/2$. Every particle impaction simulation was run until all particles were removed from the flow. In total 1.1×10^7 particles were inserted, with Stokes numbers of 0.01–10, and a progressive particle distribution with respect to particle radius.

The impaction efficiency ($\eta = N_{imp}/N_{ins}$) can be split into front (η_f) and back side impaction (η_b). At the low Reynolds number flow in this study, backside impaction rarely occurred so and front side impaction was the focus. Figure 6 depicts the particle front side impaction, compared to literature results. The results were computed with grid spacing defined as refinement level four in Tab. 2, for the $r_{cg} = 3r_c$ case with Lagrangian interpolation. With the $L_x \times L_y = 6D \times 12D$ domain a grid $(N_r \times N_\theta) + (N_x \times N_y) = (48 \times 240) + (144 \times 288)$ was used. The results from Haugen & Kragset [20] were computed on an equidistant grid with 512×1024 grid points, using an immersed boundary method to resolve the cylinder surface.

The particle impaction results from this study agree very well with the results from the literature, even though the results of the present study were computed on a grid with only 10.1% as many grid points as used by Haugen & Kragset [20]. An additional efficiency improvement was achieved, due to using a time step that was 3.5 times larger. This was possible because of the time step's proportionality to the grid spacing and the local time step restrictions, though some extra work was necessary at each time step (computation on two grids, communication of data, filtering on cylinder grid, etc.). Note that for very small particles, the time step can also be restricted by the particle time scale; that is, the time step must be small enough to resolve the time-dependent particle equations. Particles are updated only at the Cartesian time step.

4.3. Investigating the accuracy of the computed impaction efficiencies

The coarseness of the grid used in the computation of particle impaction efficiencies allow for the assessment of the assumptions that must be made in order to regard these impaction results as quantitatively accurate. The assumptions are, firstly, that blockage effects from the limited domain (with $L_x \times L_y = 6D \times 12D$) have a negligible impact on the particle impaction. Secondly, it was assumed that the coarsest resolution where grid independency of drag and lift coefficients was reached was sufficiently fine for the particle simulations, i.e., that that the transport of the particles was dependent on an accurate flow field only.

A critical assessment of these assumptions led to the expectation of a higher impaction result for particles on a domain where the blockage effect is large, due to a squeezing of the flow field and, consequently, less particles being directed away from the cylinder. In particular, this is expected to affect particles that follow the flow to a large extent, i.e. particles with small Stokes numbers. Further, the flow velocities at particle positions are not only dependent on an accurately computed flow field, but also on accurate interpolation. The latter aspect can be very sensitive to grid spacing, even if the flow is resolved accurately. Haugen & Kragset [20] used linear interpolation to compute flow velocity at particle positions, except within the grid point closest to the surface, where an expression similar to that of Eq. 19 was used. Linear interpolation of velocities that are proportional to $-(\delta r)^2$ (as the upstream flow field at the centerline through the cylinder is) will lead to a systematic over-estimation. Hence, an over-prediction in particle impaction can be expected from their results. What is important to determine in this

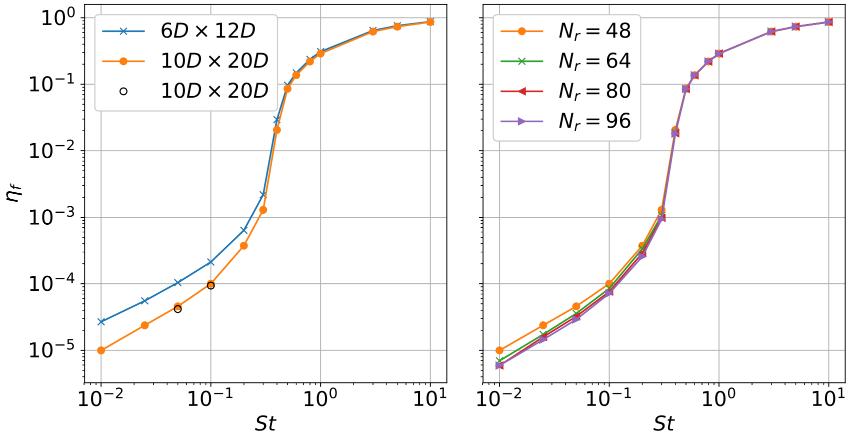


Figure 7: Front side impaction efficiency (η_f) as a function of Stokes number (St) for Reynolds number 100 for different domain sizes (left) and grid resolutions (right).

respect, is how large this possible over-prediction is, and for what particle sizes it occurs.

To investigate the accuracy of the computed impaction efficiencies particle-laden flow simulations were conducted at a larger domain size, $L_x \times L_y = 10D \times 20D$, as used in the grid independence study of Section 3.2. For this larger domain several refined grids were used. These utilized refinement levels 4–7 in Tab. 2, with $r_{cg} = 3r_c$. Thus, from 48 (coarsest) to 96 (finest) grid points were used in the radial direction on the cylindrical grid, and the background grid was refined accordingly. The number of inserted particles was 1.1×10^7, where 7×10^6 were particles with $St \leq 0.1$. The results are seen in Fig. 7.

Very few of the smallest particles deposit on the cylinder. To get enough particle impaction at the smallest Stokes numbers for reliable statistics, particles with $St \leq 0.1$ were only inserted over a region covering one tenth of the cylinder's projected area, at its centerline. The inserted particle count was scaled correspondingly (multiplied by ten) during post-processing. No small Stokes number particles inserted outside the insertion area would be expected to hit the cylinder. To confirm this, a simulation was conducted with particles with $St = 0.05$ and 0.1, inserted over the whole projected cylinder area. The results are included as black circles (o) in Fig. 7. The difference of impaction efficiency among particles inserted by the two different methods was negligible.

From Fig. 7 it is clear that the blockage effect from the limited domain size had a significant effect on the particle impaction efficiencies. For $St \leq 0.5$ this effect was larger than 10%, and increased as the Stokes number decreased. The largest difference in impaction efficiencies was seen at $St = 0.01$, where 2.7 times more impaction occurred for the smallest domain size. The resolution played a smaller, but not insignificant, role in the impaction efficiencies. Increasing from the coarsest grid, with $N_r = 48$, to $N_r = 64$, noticeably reduced the impaction efficiencies. The reduction was more than 10% for $St \leq 0.3$. A further refinement of the grid had a small effect, which was negligible for $N_r \geq 80$. Comparing the results from the larger domain with $N_r = 80$ to those by Haugen & Kragset [20] suggests that Haugen & Kragset found a qualitatively correct result, but have somewhat quantitatively over-predicted the particle impaction, in particular in the boundary interception region (where $St \lesssim 0.3$). For the smallest Stokes number ($St = 0.01$) the over-prediction is of approximately a factor 6.3. At $St = 0.1$ this factor is 2.8. The previously published results agree with the new results for $St \geq 0.5$.

5. Concluding remarks

In this work, a high-order overset grid method has been presented. The method uses high-order finite-difference discretization to solve the compressible Navier-Stokes equations on several grids, and communicates necessary flow data between the grids by linear or quadratic interpolation. Unique to the overset grid implementation described here, is the use of local time step restriction and summation-by-parts finite-difference operators. The relaxed time stepping restriction on the coarser grid is very efficient for a weakly compressible flow, while the summation-by-parts operators enhance numerical stability together with the use of Padé filtering. The purpose of developing the method was to compute particle impaction on a cylinder in a cross flow, and for this purpose a body-fitted cylindrical grid is an appropriate choice to resolve the boundary layer around the cylinder with high accuracy.

An investigation of the formal order of accuracy of the overset grid implementation revealed that high-order accuracy was indeed reached. Flow variables where computed with median order $P \approx 2.5$, regardless of the use of bi-linear interpolation or bi-quadratic interpolation for communication. Near the surface, the radial velocity component reached an accuracy of fifth-order. For unsteady flow, the method converged rapidly to grid independent solutions for the essential flow variables (drag, lift and Strouhal number). For these computations, using bi-linear interpolation was beneficial, yielding the most rapid convergence to grid independent solutions as the grid was refined. Using a larger cylindrical grid, with a radius five times as large as the cylinder radius, decreased the effect of the inter-grid interpolation.

When applied to the problem of inertial particles impacting on a cylinder, impaction efficiencies of previously published results were reproduced at a significantly reduced computational cost. A coarser background grid was utilized to resolve the flow, which yielded both a much smaller number of grid points (90% reduction in 2D) and the possibility of using a larger time step.

A critical assessment of the particle impaction results revealed that the limited domain size had a significant impact on particle impaction, particularly for the smaller Stokes numbers. Further, although the flow was deemed grid independent, using a finer grid, and thus a more accurate interpolation of flow velocity, reduced the number of particles that hit the cylinder. The resulting impaction curves suggested that particle impaction has been overestimated in the previous studies, in particular for very light particles where impaction occurs by boundary interception.

The overset grid method implementation in the Pencil Code is now ready for three-dimensional simulations, and DNS studies of particle impactions on a cylinder with Reynolds number for real-world application (a factor 10-20 larger than the investigation here, for industrial boilers) is within reach. However, even with the highly accurate and efficient method presented here, increasing the Reynolds number and computing three-dimensional flow will be computationally costly. The magnitude of the Reynolds numbers that can be considered will largely depend on the Stokes numbers of the particles, and the acceptable accuracy when particle impaction efficiencies are computed. If the focus is not just qualitative trends, but quantitatively accurate results, a careful assessment of grid independence is recommended (not just for flow variables, but for the particle impaction itself), and great care is required when selecting the domain size and setting up the simulations.

6. Acknowledgements

We would like to acknowledge the valuable discussions on numerical methods and code development with our colleagues Prof. Bernhard Müller and

Dr. Ehsan Khalili.

The work was supported by the Research Council of Norway (Norges Forskningsråd) under the FRINATEK Grant [grant number 231444]. Partial funding was provide by the GrateCFD project, which was funded by: LOGE AB, Statkraft Varme AS, EGE Oslo, Vattenfall AB, Hitachi Zosen Inova AG and Returkraft AS together with the Research Council of Norway through the ENERGIX program [grant number 267957/E20]. Computational resources were provided by UNINETT Sigma2 AS [project numbers NN9405K and NN2649K].

References

[1] H. K. Versteeg, W. Malalasekera, An Introduction to Computational Fluid Dynamics: The Finite Volume Method, 2nd Edition, Pearson Education, Harlow, United Kingdom, 2007.

[2] D. Mavriplis, Unstructured grid techniques, Annual Review of Fluid Mechanics 29 (1) (1997) 473–514.

[3] J. C. Tannehill, D. Anderson, R. H. Pletcher, Computational Fluid Mechanics and Heat Transfer, 2nd Edition, Taylor & Francis, Philadelphia, PA, 1997.

[4] S. J. Owen, A survey of unstructured mesh generation technology., in: Proceedings of 7th international meshing roundtable, 1998, pp. 239–267.

[5] C. S. Peskin, Flow patterns around heart valves: A numerical method, Journal of Computational Physics 10 (2) (1972) 252–271.

[6] R. Mittal, G. Iaccarino, Immersed Boundary Methods, Annual Review of Fluid Mechanics 37 (1) (2005) 239–261.

[7] J. L. Steger, F. C. Dougherty, J. A. Benek, A chimera grid scheme, in: K.N. Ghia and U. Ghia, eds., Advances in Grid Generation, ASME FED-5, 1983, pp. 59–69.

[8] J. A. Benek, P. G. Buningt, J. L. Steger, A 3-d chimera grid embedding technique, in: AIAA 7th Computational Fluid Dynamics Conference, AIAA Paper No. 1523, 1985.

[9] J. L. Steger, J. A. Benek, On the use of composite grid schemes in computational aerodynamics, Computer Methods in Applied Mechanics and Engineering 64 (1-3) (1987) 301–320.

[10] E. Pärt-Enander, B. Sjögreen, Conservative and non-conservative interpolation between overlapping grids for finite volume solutions of hyperbolic problems, Computers & Fluids 23 (3) (1994) 551–574.

[11] Y. Zang, R. L. Street, A composite multigrid method for calculating unsteady incompressible flows in geometrically complex domains, International Journal for Numerical Methods in Fluids 20 (5) (1995) 341–361.

[12] S. E. Sherer, J. N. Scott, High-order compact finite-difference methods on general overset grids, Journal of Computational Physics 210 (2) (2005) 459–496.

[13] J. Chicheportiche, X. Gloerfelt, Study of interpolation methods for high-accuracy computations on overlapping grids, Computers & Fluids 68 (2012) 112–133.

[14] S. Völkner, J. Brunswig, T. Rung, Analysis of non-conservative interpolation techniques in overset grid finite-volume methods, Computers & Fluids 148 (2017) 39–55.

[15] R. Meakin, The chimera method of simulations for unsteady three-dimensional viscous flow, in: M. Hafez, K. Oshima (Eds.), Computational Fluid Dynamics Review, Wiley, New York, NY, 1995, pp. 70–86.

[16] G. Chesshire, W. D. Henshaw, Composite overlapping meshes for the solution of partial differential equations, Journal of Computational Physics 90 (1) (1990) 1–64.

[17] M. García Pérez, E. Vakkilainen, T. Hyppänen, Unsteady CFD analysis of kraft recovery boiler fly-ash trajectories, sticking efficiencies and deposition rates with a mechanistic particle rebound-stick model, Fuel 181 (2016) 408–420.

[18] A. Beckmann, M. Mancini, R. Weber, S. Seebold, M. Müller, Measurements and CFD modeling of a pulverized coal flame with emphasis on ash deposition, Fuel 167 (2016) 168–179.

[19] R. Israel, D. E. Rosner, Use of a generalized stokes number to determine the aerodynamic capture efficiency of non-stokesian particles from a compressible gas flow, Aerosol Science and Technology 2 (1) (1982) 45–51.

[20] N. E. L. Haugen, S. Kragset, Particle impaction on a cylinder in a crossflow as function of Stokes and Reynolds numbers, Journal of Fluid Mechanics 661 (2010) 239–261.

[21] N. E. L. Haugen, S. Kragset, M. Bugge, R. Warnecke, M. Weghaus, MSWI super heater tube bundle: Particle impaction efficiency and size distribution, Fuel Processing Technology 106 (2013) 416–422.

[22] A. Brandenburg, W. Dobler, Hydromagnetic turbulence in computer simulations, Computer Physics Communications 147 (1-2) (2002) 471–475.

[23] The Pencil Code, (pencil-code.nordita.org [Internet]). Stockholm (SE): NORDITA; [updated 2018 May 01]. Available from: https://github.com/pencil-code.

[24] J. Williamson, Low-storage runge-kutta schemes, Journal of Computational Physics 35 (1) (1980) 48–56.

[25] T. J. Poinsot, S. Lele, Boundary conditions for direct simulations of compressible viscous flows, Journal of Computational Physics 101 (1) (1992) 104–129.

[26] C. S. Yoo, Y. Wang, A. Trouvé, H. G. Im, Characteristic boundary conditions for direct simulations of turbulent counterflow flames, Combustion Theory and Modelling 9 (4) (2005) 617–646.

[27] B. Strand, Summation by parts for finite difference approximations for d/dx, Journal of Computational Physics 110 (1) (1994) 47–67.

[28] K. Mattsson, J. Nordström, Summation by parts operators for finite difference approximations of second derivatives, Journal of Computational Physics 199 (2) (2004) 503–540.

[29] M. R. Visbal, D. V. Gaitonde, High-order-accurate methods for complex unsteady subsonic flows, AIAA Journal 37 (10) (1999) 1231–1239.

[30] D. V. Gaitonde, M. R. Visbal, Padé-type higher-order boundary filters for the navier-stokes equations, AIAA Journal 38 (11) (2000) 2103–2112.

[31] Y. Li, R. Zhang, R. Shock, H. Chen, Prediction of vortex shedding from a circular cylinder using a volumetric Lattice-Boltzmann boundary approach, European Physical Journal: Special Topics 171 (1) (2009) 91–97.

[32] O. Posdziech, R. Grundmann, A systematic approach to the numerical calculation of fundamental quantities of the two-dimensional flow over a circular cylinder, Journal of Fluids and Structures 23 (3) (2007) 479–499.

[33] D. Pan, An immersed boundary method on unstructured cartesian meshes for incompressible flows with heat transfer, Numerical Heat Transfer, Part B: Fundamentals 49 (3) (2006) 277–297.

[34] L. Qu, C. Norberg, L. Davidson, S.-H. Peng, F. Wang, Quantitative numerical analysis of flow past a circular cylinder at Reynolds number between 50 and 200, Journal of Fluids and Structures 39 (2013) 347–370.

Paper V [5]

Treatment of solid objects in the Pencil Code using an immersed boundary method and overset grids

Jørgen R. Aarnes, Tai Jin, Chaoli Mao, Nils E. L. Haugen, Kun Luo and Helge I. Andersson

Published online in Geophysical & Astrophysical Fluid Dynamics, 2018.
Forthcoming in special issue: *On the Physics and Algorithms of the Pencil Code*.

GEOPHYSICAL & ASTROPHYSICAL FLUID DYNAMICS
https://doi.org/10.1080/03091929.2018.1492720

Taylor & Francis
Taylor & Francis Group

Treatment of solid objects in the Pencil Code using an immersed boundary method and overset grids

Jørgen R. Aarnes[a], Tai Jin[b,c], Chaoli Mao[c], Nils E. L. Haugen[a,d], Kun Luo[c] and Helge I. Andersson[a]

[a]Department of Energy and Process Engineering, Norwegian University of Science and Technology, Trondheim, Norway; [b]Department of Mechanical Engineering, University College London, London, UK; [c]State Key Laboratory of Clean Energy Utilization, Zhejiang University, Hangzhou, People's Republic of China; [d]SINTEF Energy Research, Trondheim, Norway

ABSTRACT
Two methods for solid body representation in flow simulations available in the Pencil Code are the immersed boundary method and overset grids. These methods are quite different in terms of computational cost, flexibility and numerical accuracy. We present here an investigation of the use of the different methods with the purpose of assessing their strengths and weaknesses. At present, the overset grid method in the Pencil Code can only be used for representing cylinders in the flow. For this task, it surpasses the immersed boundary method in yielding highly accurate solutions at moderate computational costs. This is partly due to local grid stretching and a body-conformal grid, and partly due to the possibility of working with local time step restrictions on different grids. The immersed boundary method makes up the lack of computational efficiency with flexibility in regard to application to complex geometries, due to a recent extension of the method that allows our implementation of it to represent arbitrarily shaped objects in the flow.

ARTICLE HISTORY
Received 3 April 2018
Accepted 20 June 2018

KEYWORDS
Pencil code; immersed boundary method; overset grids; compressible fluid dynamics; complex geometries

1. Introduction

Fluid flow in a domain that contains an immersed solid object is a common case in computational fluid dynamics. Obstructions in the flow include (but are not limited to) cylinders, spheres, flat plates, rectangular or elliptical cylinders and spheroids, triangles, and complex geometries made out of a combination of these. Finding a method to represent such objects in the best possible way in simulations is not a trivial task, and the method used is often chosen specifically to the problem at hand.

For many generic shapes, such as cylinders, spheres and plates, body-fitted structured meshes are commonly used to represent the object(s) in the flow. Body-fitted structured meshes conform to the object(s) in the flow domain and to the domain's other physical boundaries (inlet, outlet, walls, etc.). Depending on the flow domain and object in the flow, this may require a deformation of the grid to conform to domain boundaries, in

CONTACT Jørgen. R. Aarnes jorgen.r.aarnes@ntnu.no; Tai Jin tai.jin@ucl.ac.uk

addition to the mapping procedures to map the grid in the flow domain to a simple computational domain. This may result in a grid with unnecessary local variations of the grid (e.g. a grid that is denser than necessary in certain areas of the domain) and time-consuming grid generation (Versteeg and Malalasekera 2007). A popular alternative to such meshes, particularly when the shape of the flow domain or objects in the flow domain is more complex, is unstructured meshes. Unstructured meshes provide the highest flexibility in grid adaptation to a particular flow geometry and is a good alternative for complex geometries when finite-volume or finite-element formulations of the governing equations are used (Mavriplis 1997). Disadvantages of such grids are much larger storage requirements than for structured grids (Tannehill *et al.* 1997), the need for intricate mesh generation techniques (Owen 1998) and the difficulty in achieving high order of accuracy.

By other choices of grid methods, the object(s) in the flow and the flow domain can be represented without the grid conforming to the object(s). Typically this is done by using a Cartesian grid, with a modification in either the flow equations or the grid cells in the immediate vicinity of the solid object(s). Popular methods of this type include immersed boundary methods (IBMs) (Peskin 1972, 2002; Mittal and Iaccarino 2005) and cut-cell methods (Quirk 1994; Causon *et al.* 2000; Schneiders *et al.* 2013). These methods differ in that the IBM uses a Cartesian grid in the entire flow domain, while in cut-cell methods grid cells are "cut" near the objects or domain boundaries that do not conform to the grids, and the flow equations are solved on the new, modified cells (Ingram *et al.* 2003). Due to this cell cutting, care must be taken such that the cut cells do not become too small, since this may be a potential source of numerical instabilities.

For the IBM, rather than modifying the grid cells near the solid object, the boundary conditions on the solid are imposed directly in the flow equations. This is done either by a continuous or a discrete forcing technique. In both cases a body-force, present due to non-conforming boundaries in the flow, is introduced in the Navier–Stokes equations. This is done either before discretisation (continuous forcing) or after (discrete forcing) (Mittal and Iaccarino 2005). The latter of these is the preferred method for IBM used to represent rigid boundaries. One development of the discrete forcing method is to treat the immersed boundary as a sharp interface and to impose the boundary conditions directly. This is done by using a combination of ghost-points inside the solid and mirror/image-points in the flow domain (set by interpolation) to reconstruct the solid (Tseng and Ferziger 2003; Berthelsen and Faltinsen 2008). An advantage with this approach is that the boundary conditions are handled without any added force in the flow equations, hence, the method can easily be implemented in an existing flow solver. The disadvantage is the accuracy reduction in the vicinity of the surface, although recent developments show that some of the challenges related to high-order accurate reconstructions of velocities near the surface can be overcome (Seo and Mittal 2011; Xia *et al.* 2014). Furthermore, finite-difference IBMs are, in general, not mass conserving. Finite-volume approaches with cut-cell methodology are appropriate if mass and momentum conservation need to be guaranteed (Mittal and Iaccarino 2005).

About 10 years after the emergence of the first IBM, a new method was proposed to represent solids in the flow by using several grids overset one another (Steger *et al.* 1983; Benek *et al.* 1985). Such overset grid methods (often called Chimera methods) employ body-conformal grids at the boundaries of objects in the flow, but the grids do not extend to the physical boundaries of the domain. Rather, a background grid (typically uniform

Cartesian) is used, and updated flow information of overlapping grid regions is communicated between grids at every time step. Note that special overset grid methods without background grids exist, like yin-yang grids where two identical component grids are used to cover a spherical surface, thus avoiding very small grid cells close to poles of the spherical geometry (Kageyama and Sato 2004).

The flow domain resolved with overset grids may contain a single grid overlapping another, or several grids overlapping necessitating a priority of communication and computation of solutions of the different grids (Steger and Benek 1987; Chesshire and Henshaw 1990). For complex configurations, this may require extensive preprocessing for fixed objects (Rogers *et al.* 2003) or intricate grid handling at run time for moving bodies (Noack 2005). Overset grid methods are, in general, not mass conserving, since interpolation is necessary between grids overset one another (although exceptions do exist, for finite-volume implementations of overset grids, see Pärt-Enander and Sjögreen 1994 and Zang and Street 1995). The interpolation is done from donor-points on one grid to fringe-points on another. Many different interpolation procedures have been explored for this purpose, and several studies have found that using high-order interpolation between grids is beneficial in regard to the overall accuracy and stability of flow computations (Sherer and Scott 2005; Chicheportiche and Gloerfelt 2012; Völkner *et al.* 2017).

In the high-order compressible flow solver known as the Pencil Code (The Pencil Code 2018), solid objects in the flow can be represented by different schemes. This makes it possible to compare different surface representations not only for the same flow problems, but for simulations where the same finite-difference discretisation, time integration, communication procedures, etc., are used. The purpose of this paper is to perform such a comparison for solids represented by a ghost-points IBM and overset grids. The performance of these surface representations is assessed in terms of computational cost and accuracy for a common benchmarking case. The flow case used is the frequently appearing fluid mechanics problem of flow past a circular cylinder. Furthermore, we wish to shed light on an advantage of the IBM implementation in the Pencil Code by simulating flow past a complex geometry. The complex geometry used as an example case is a combination of a semi-circular and a semi-elliptical cylinder.

The structure of the paper is as follows. In section 2, the governing flow equations and the two methods for solid object representation are described, with details on their implementation in the Pencil Code included. Performance of the different methods for the flow past a cylinder in both the steady regime and the unsteady vortex shedding regime is compared in section 3. Following this, we elaborate on the extension of the IBM to complex geometries in section 4, before concluding remarks are made in section 5.

2. Methodology

2.1. Governing equations

The governing equations of the flow are the continuity equation

$$\frac{\mathrm{D}\rho}{\mathrm{D}t} + \rho\nabla\cdot\boldsymbol{u} = 0, \tag{1}$$

and the momentum equation

$$\rho \frac{\mathrm{D}\boldsymbol{u}}{\mathrm{D}t} = -\nabla p + \nabla\cdot\boldsymbol{u} = 0, \tag{2}$$

where $\rho, t, \boldsymbol{u}, p$ and μ are the density, time, velocity vector, pressure and dynamic viscosity ($\mu = \rho\nu$, with kinematic viscosity ν), respectively, and

$$\frac{\mathrm{D}}{\mathrm{D}t} = \frac{\partial}{\partial t} + \boldsymbol{u}\cdot\nabla \tag{3}$$

is the material derivative operator. The compressible rate of strain tensor $\boldsymbol{S}$ is given by

$$\boldsymbol{S} = \tfrac{1}{2}(\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^{\mathrm{T}}) - \boldsymbol{I}(\tfrac{1}{3}\nabla\cdot\boldsymbol{u}), \tag{4}$$

where $\boldsymbol{I}$ is the identity matrix. The pressure is computed by the isothermal ideal gas law, $p = c_s^2\rho$, where c_s is the speed of sound. With a constant speed of sound (for the isothermal case) and a constant kinematic viscosity, the momentum equation (2) can be rewritten as

$$\frac{\mathrm{D}\boldsymbol{u}}{\mathrm{D}t} = -c_s^2\nabla(\ln\rho) + \nu(\nabla^2\boldsymbol{u} + \tfrac{1}{3}\nabla(\nabla\cdot\boldsymbol{u}) + 2\boldsymbol{S}\cdot\nabla(\ln\rho)), \tag{5}$$

which is the form solved in the computations performed in this study.

2.2. Numerical methods

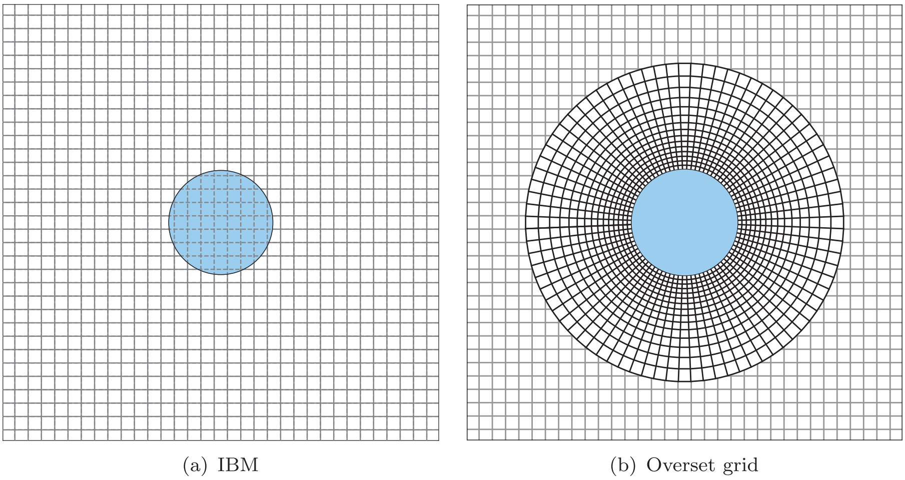
The governing equations (1) and (5) are discretised with sixth-order finite differences in space and a third-order memory efficient Runge–Kutta scheme in time (Williamson 1980).

Many different types of domain and enforcements of boundary conditions are available in the Pencil Code. For simplicity, we consider a domain with a uniform mean flow, using Navier–Stokes characteristic boundary conditions (NSCBC) on both the inlet and outlet of the flow domain, and periodic boundary conditions in all other directions. The NSCBC is a formulation developed by Poinsot and Lele (1992) that makes use of one-dimensional characteristic wave relations to allow acoustic waves to pass through the boundaries.

We place an object in the flow domain by representing it with one of the two available methods in the Pencil Code. The grids used in each of the methods are quite different. An illustration of the grids for IBM and overset grid representation of a circular solid can be seen in figure 1. For both methods of solid body representation, we use boundary conditions of no-slip and impermeability for velocity components, and zero gradient for density in the direction normal to the surface, on the solid's surface. The latter condition can be derived from the ideal gas law and the boundary layer approximation for pressure normal to the boundary ($\partial p/\partial r = 0$, White 2006). In the remainder of this section, details of how the different boundary representations are implemented in the Pencil Code are given.

2.2.1. Ghost-zone immersed boundary method

In the illustration of a circular object in a flow domain represented by an IBM (figure 1a), the intersections of solid grid lines represent fluid-points, where the governing equations are solved, while the intersection points of dashed grid lines are grid points inside an immersed solid (solid-points). At the solid-points, the governing equations are not solved.


(a) IBM

(b) Overset grid

Figure 1. Solid object representation on a uniform Cartesian grid, by either an immersed boundary method or an overset grid approach. Principle illustration of mesh in each of the cases. (a) IBM and (b)overset grid.

Rather, some points are used as ghost-points for the fluid solver and some are unused points. As mentioned, the boundary conditions of a solid object may be imposed directly on the flow variables by the ghost-point immersed boundary method. The IBM in the Pencil Code is such a ghost-point method. An uncommon feature of the IBM implementation in the Pencil Code is that a several points deep ghost-zone is used rather than a single strip of ghost-points inside the solid object. This ensures that the sixth-order finite-difference stencils can be used without any modifications in the vicinity of a solid object. The overhead related to computation of additional layers of ghost-points is negligible when compared to the computational cost of the fluid solver itself.

As sixth-order central differencing is used, three points on each side of a grid point are necessary to update the solution. This is illustrated in figure 2(a), where stencils of fluid-points f_1, f_2 and f_3 will include grid points within the solid object to update the horizontal gradients of the velocity components and density. Point f_1 will need information from f_2, f_3 and g_1, f_2 will need information from f_1, g_1 and g_2, etc. (in addition to information from fluid-points to the left). The points in the ghost-zone, g_1, g_2 and g_3, are set using corresponding mirror-points m_1, m_2 and m_3, respectively. With no-slip and impermeability for velocity and zero gradient for density, the relationship between a ghost point g and mirror-point m is simply

$$\boldsymbol{u}(g) = -\boldsymbol{u}(m), \quad \rho(g) = \rho(m). \tag{6a,b}$$

Note that a second-order accurate method to set the Neumann boundary condition has recently been implemented, but will not be described here. Details, and testing of different boundary conditions, can be found in Luo *et al.* (2016).

In general, the mirror-points do not coincide with grid points or grid lines, and need therefore to be interpolated from surrounding points. This is done by bilinear Lagrangian interpolation in 2D and trilinear Lagrangian interpolation in 3D. For mirror-points close to the surface, one or more of the surrounding grid points may be inside the solid object

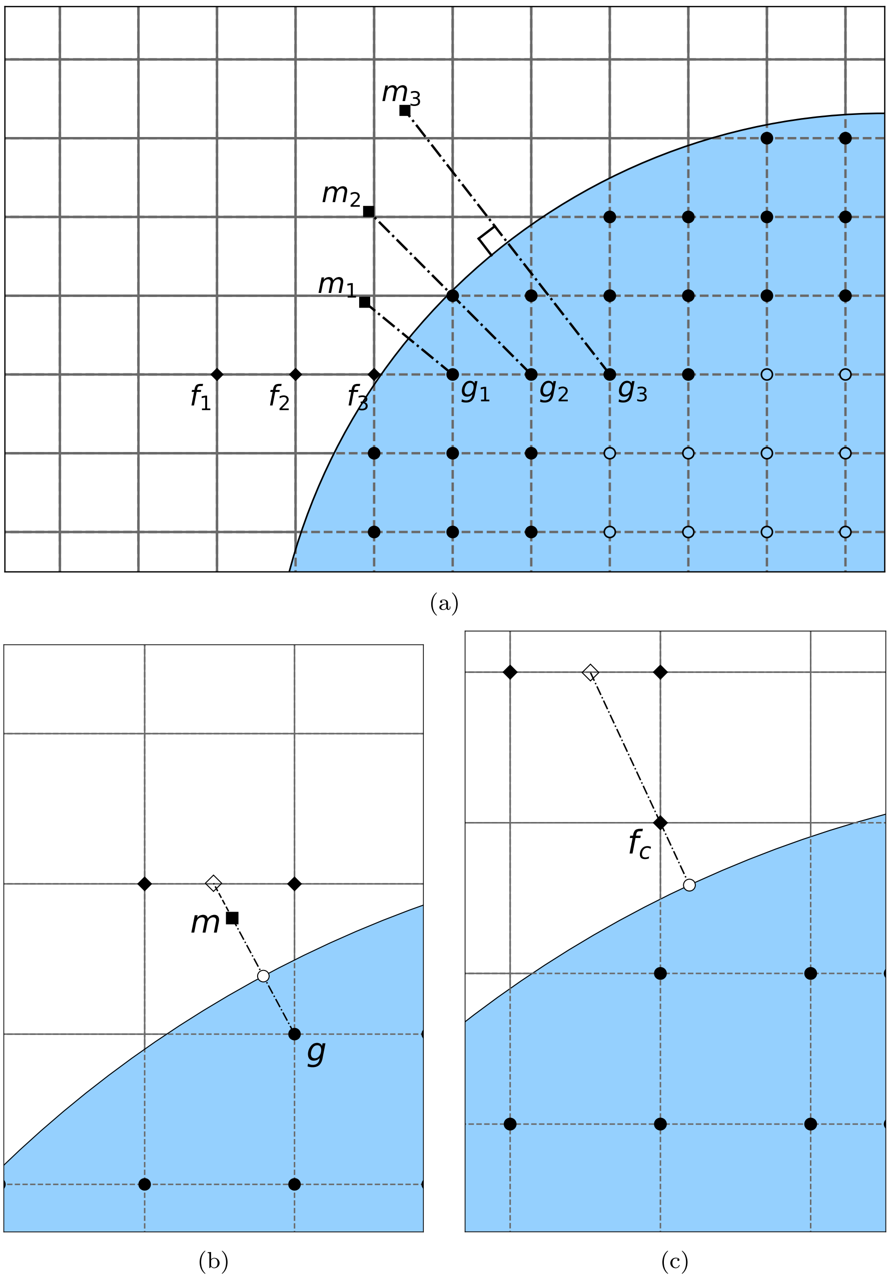


Figure 2. Immersed boundary method. (a) A zone of ghost-points (g_i; ●) is used when computing fluid-points (f_i; ◆). Ghost points are set from corresponding mirror/image-points in the flow domain (m_i; ■) found along lines orthogonal to the solid surface. The mirror-points are interpolated from surrounding fluid-points. (b) If a mirror-point is too close to the surface to be surrounded only by fluid-points, the values at the points where the orthogonal line intercepts the surface (◯) and the first grid line (◇) are used in interpolation. (c) A fluid-point (f_c; ◆) very close to the surface is set by direct interpolation along the surface normal points by using values at the interception with the surface (◯) and the closest grid line (◇). The interpolation point at the intersection with the closest grid line in (b) and (c) is set by interpolation from the closest fluid-points along the grid line (◆). (a) Ghost-zone immersed boundary method, (b)mirror-point close to surface and (c)fluid-point close to surface.

(see figure 2 b). Rather than using four surrounding fluid-points, these mirror-points are set by interpolation using the boundary intercept point (that is, the point where the surface normal through the ghost point intercepts the solid's boundary) and the point where the surface normal intercepts the first grid line outside the solid. Data is first interpolated to the interception point of the surface normal with the first grid line from neighbouring fluid-points (◆ in figure 2b) by linear Lagrangian interpolation. The velocity components normal to the surface are expected to scale as Δr^2 when approaching the surface, where Δr is the distance from the boundary. To interpolate velocity in a mirror-point near the surface, the velocity is decomposed into cylindrical components, and the radial component is computed by

$$u_{r,m} = u_{r,GI}\left(\frac{\Delta r_m}{\Delta r_{GI}}\right)^2, \tag{6}$$

where $u_{r,GI}$ is the radial velocity at the grid line interception point, and Δr_m and Δr_{GI} are the distances from the mirror-point and grid line interception point to the boundary interception point, respectively. Remaining velocity components are obtained by linear interpolation. No special handling is used for density.

Special handling is used for fluid-points very close to the surface of an object. This is done to avoid spurious errors due to de-localisation dependencies in the finite-difference stencils, a detrimental effect that occurs when flow variables quite far from a grid point is indirectly used in the update of said grid point. To see this, consider in figure 2(a) that the horizontal velocity component of grid points surrounding m_3 will affect f_3, since g_3 is set by m_3. Rather than computing flow variables in a fluid-point close to the surface in the usual way, by using the finite-difference stencils, they are set directly by interpolation, as seen in figure 2(c). The interpolation procedure for these grid points is the same as for mirror-points very close to the boundary, as described earlier. Note that this type of special handling is only possible for variables with a Dirichlet boundary condition on the surface.

An alternative to setting mirror-point positions using surface normals is to use mirror-points along grid lines. This simplifies interpolation (making all interpolation one-dimensional, along grid lines) and has proven promising in reducing the errors due to de-localisation dependencies. In such an approach, a ghost point can have several values for each flow variable that has a Dirichlet boundary condition, one used in horizontal and one in vertical finite-difference stencils (as would be the case for g_1, g_2 and g_3 in figure 2a). Flow variables with Neumann boundary conditions are set from mirror-points along surface normals as in the method described earlier. In this study, we will stick to the more mature method of using mirror-points from surface normals. Details on the alternative grid-line ghost-zone IBM can be found in Aarnes *et al.* (2018b).

2.2.2. Local-time restricted overset grids

Unlike the representation of solid bodies with most methods (IBM, body-fitted structured or unstructured grids), codes using overset grids require splitting of the flow solver, as one solver is needed for each grid. We limit this study to a single grid on top of a background grid, for a more general discussion see Chesshire and Henshaw (1990) or Meakin (1995). Yin-yang grids are not considered (although a yin-yang grid implementation exists in the Pencil Code).

For a flow with a solid object represented using a body-confined grid over a Cartesian background grid, the governing equations are, in principle, solved for two different flow domains, one with and one without a solid object present in the flow. At least one boundary in each domain is set by interpolating flow variables from another grid, and in this way the presence of the solid affects the flow on all grids. Admittedly, this makes overset grids somewhat more unwieldy than IBMs.

To consider this more systematically, let us regard a fluid time step as split into four parts: (1) solution of the governing equations on the background grid, (2) communication of data from the background grid to the body-fitted grid, (3) solution of the governing equations on the body-fitted grid, (4) communication of data from the body-fitted grid to the background grid. The solution step on the body-fitted grid requires the implementation of a Navier–Stokes solver applicable to the type of grid that is used to resolve the bluff body's boundary. For our case of a cylinder in a cross flow, a Navier–Stokes solver applicable to cylindrical coordinates is necessary.

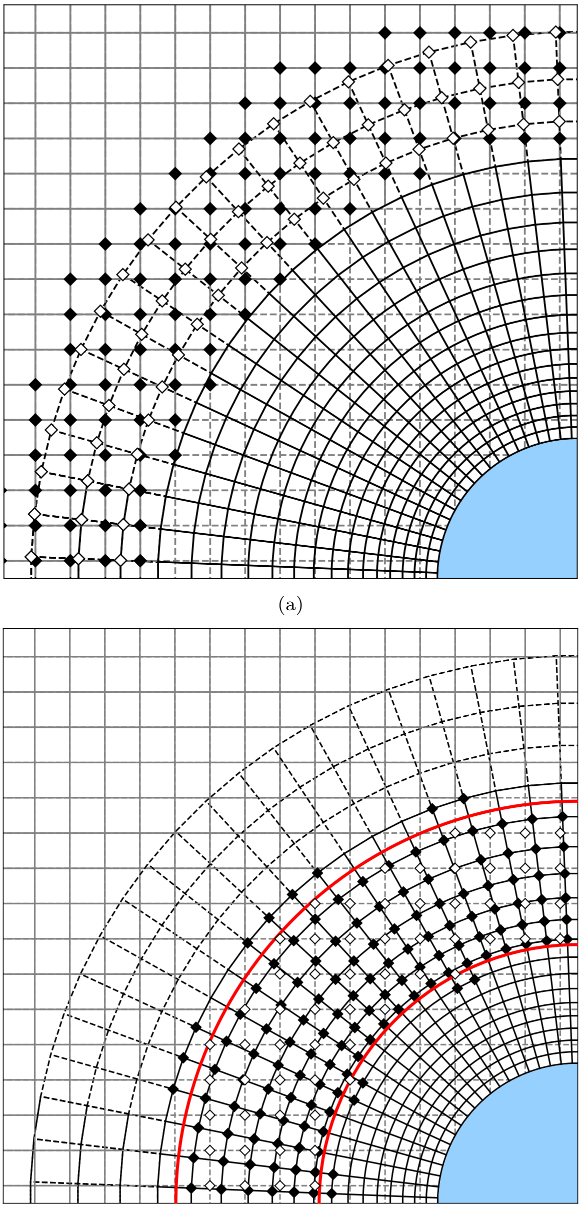
In the illustration of the mesh used in the (cylindrical) overset grid method (figure 1 b), there are no grid points inside the circular object. Strictly speaking, the background Cartesian grid is present in the entire domain (also inside the limits of the curvilinear mesh and inside the solid object), but only few points inside the curvilinear mesh are used, and not a single Cartesian grid point inside the solid is (or should ever be) used.

The points of the background grid that are in use inside the curvilinear mesh are mostly fringe-points. Fringe-points are fluid-points set by interpolation from nearby donor-points on the overlapping grid, rather than computed using discretisation of the governing equations. Donor-points are computed in the same way as an ordinary fluid-point, unless the interpolation is implicit, meaning that a fringe-point may be used as a donor-point (Chesshire and Henshaw 1990), which is not the case here. A third class of points found on overset grids are hole-points. These are unused grid points, typically found outside of the fringe of a grid (like Cartesian grid points far inside the curvilinear grid). Fringe-, donor- and hole-points on an overset grid generated to represent a circular object are seen in figure 3.

During the first stage of the inter-grid communication, data is sent to the outermost grid points of the body-fitted grid (figure 3 a). At the second stage, data is sent back to Cartesian fringe-points (figure 3b). The fringe-points on the Cartesian grid are identified during pre-processing, where points within a distance from the solid, illustrated by red circular curves in figure 3(b), are selected.

For moving objects, fringe-point locations on the background grid must be re-calculated every time step. A several layers thick zone of fringe-points is used on both grids to enable the use of the sixth-order centred stencils at the outer edges of each grid, equivalent to the use of a ghost-zone for the IBM.

In the illustration in figure 3, each fringe-point is surrounded by four donor-points, as necessary in bilinear Lagrangian interpolation. If higher order interpolation is desired the amount of donor points for each fringe points must be increased accordingly. Such an increase is straightforward for overset grids, unlike for immersed boundary methods where more intricate interpolation stencils are needed for high-order interpolation to avoid using grid points that are inside the bluff body. Note, however, that a straightforward extension from second- to third-order interpolation (or higher) does not guarantee a better solution. This is due to possible overshoots in the interpolation polynomials. High-order Lagrangian


(a)

(b)

Figure 3. Overset grid method. Interpolation between grids, from donor-points (◆) to fringe-points (◇). The outer points on the curvilinear grid are by default set as fringe-points, while on the Cartesian grid a zone of fringe-points are identified during pre-processing. Cartesian grid points closer to the solid than the inner diameter setting the fringe-point zone are hole-points. Intersections of solid grid lines represent regular fluid-points where finite-difference stencils are used to update the flow variables. At dashed cylindrical grid lines, all intersections are fringe-points. At dashed Cartesian grid lines, intersections may be regular fluid-points, fringe-points or hole-points. Fringe-points are identified in-between the inner and outer interpolation zone radius (red circular lines) set according to selected interpolation scheme. Cartesian grid points closer to the solid than the inner interpolation radius are hole-points. (colour online). (a) Interpolation: Cartesian to curvilinear grid and (b) Interpolation: Curvilinear to Cartesian grid.

interpolation and quadratic splines are implemented for overset grid interpolation in the Pencil Code. For simplicity, we will restrict ourselves to bilinear Lagrangian interpolation here.

At the solid–fluid interface, we use summation-by-parts (SBP) finite-difference operators to enhance stability of the solution. This means modifying the finite-difference stencils in the nine grid points closest to the surface (including the surface point) along each radial grid line, to asymmetric stencils (one-sided at the surface). The order of accuracy for the SBP operators are third order for a sixth-order finite-difference method. Details on these operators can be found in Strand (1994) (first derivatives) and Mattsson and Nordström (2004) (second derivatives).

A novel feature of the overset grid implementation in the Pencil Code is how the restrictions on the time step are handled. The advective and diffusive time step restrictions are $\Delta t \leq C_\nu \Delta \chi_{\min}^2/\nu$ and $\Delta t \leq C_u \Delta \chi_{\min}/(|\boldsymbol{u}| + c_s)$, respectively, where Δt is the time step, $\Delta \chi_{\min}$ the smallest grid spacing in any direction, and C_ν and C_u are the diffusive and advective Courant numbers, respectively. For a weakly compressible flow, we typically require a very short time step, increasingly so if grid stretching is used in order to have a fine grid in the vicinity of the solid object. However, when overset grids are used these restrictions are no longer global restrictions on the time step, but local. Hence, by performing several time steps on the body-fitted grid for each time step on the background grid, the efficiency of the code may be greatly improved. In particular, this allows for a very fine resolution close to the surface (on the body-fitted grid) without using small time steps for flow far from the surface (on the background grid).

For all the overset grid computations in the present study, the diameter of the cylindrical grid is three times that of the solid cylinder it is fitted to. For a consideration of the extent of the domain covered by the body-fitted grid, the reader is referred to Aarnes et al. (2018a).

2.2.3. A note on dissipation

The centred finite-difference schemes used for discretisation of the governing equations are non-dissipative. This can cause problems due to the potential growth of high-frequency modes, leading to numerical instability.

To some extent, the summation-by-parts boundary conditions suppress such instabilities that are due to boundary conditions when overset grids are used, but these boundary stencils are not sufficient to suppress all oscillations in the solution when grid stretching is used on the curvilinear grid. Such oscillations are most prominent in the density field. The detrimental effect of the high-frequency modes increases as the grid spacing decreases and may lead to diverging solutions as the grid is refined. To suppress the high-frequency modes, a high-order low-pass filter is used on the curvilinear part of the overset grid. The filter is a 10th-order Padé filter, with boundary stencils of 8th and 6th orders. On the interior of the domain, the filter is given by

$$\alpha_f \hat{\phi}_{i-1} + \hat{\phi}_i + \alpha_f \hat{\phi}_{i+1} = \sum_{n=0}^{N} \frac{\alpha_n}{2} (\phi_{i+n} + \phi_{i-n}), \tag{7}$$

where $\hat{\phi}_k$ and ϕ_k are components k of the filtered and unfiltered solution vectors, respectively, α_f is a free parameter ($|\alpha_f| \leq 0.5$) and α_n are fixed parameters dependent only on α_f (details in Visbal and Gaitonde 1999). Boundary stencils can be found in Gaitonde and

Visbal (2000). The Padé filter is implicit and requires us to solve a tridiagonal linear system at each grid point, in the radial direction and a cyclic tridiagonal system in the direction tangential to the surface. The free parameter α_f is set to 0.1. With such a small value for α_f, filtering the solution once per Cartesian time step is found sufficient to get a stable and accurate solution.

Alternatively, a sixth-order hyperdiffusion operator, which is already implemented in the Pencil Code (see e.g. Haugen and Brandenburg 2006), could have been used to filter the solution. The benefits of this approach are that the hyperdiffusion operator is explicit and fast, and does not require extra communication between processors. It is, however, expected to be less sharp than the 10th-order Padé filter, as Padé filters are known to outperform explicit filtering schemes (Visbal and Gaitonde 1999, 2002).

When IBM is used rather than overset grids, some dissipation can be turned on by using fifth-order upwinding for the advection operators of the density rather than central-difference stencils (details in Dobler *et al.* 2006). The problem of oscillations in the density field is, however, much less prominent when a uniform Cartesian mesh is used, so while Padé filtering is on by default when overset grids are used, a dissipative solution by upwinding is optional for other simulations.

3. Simple geometry

In previous studies, the order of accuracy of the solid object representations described above has been assessed for steady flow computation. Using a slightly modified handling of Neumann boundary conditions, Luo *et al.* (2016) showed that, regardless of boundary condition, the IBM implementation in the Pencil Code is second-order accurate in the vicinity of a resolved circular boundary. For the same geometry, using overset grids, Aarnes et al. (2018a) showed the order of accuracy in the vicinity of the solid differed for different flow variables. The radial velocity component was computed with order of accuracy between third- and fifth orders, while the accuracy of the tangential velocity and density was between second- and third orders, when second-order Lagrangian interpolation was used for communication between grids. Both the mentioned studies also demonstrated that characteristic flow parameters, such as drag, lift and shedding frequency (for unsteady flow), could be reproduced to good agreement with previous studies, with the respective boundary representation in use.

We will not repeat an assessment of accuracy for steady flow computations here. Rather, we investigate the boundary representations by a direct comparison for the case of flow past a circular cylinder in different shedding regimes. Two-dimensional flow past a circular cylinder in the vortex shedding regime is a classical benchmarking case for fluid dynamic simulations with a solid object present in the flow domain. We simulate such a flow with Reynolds number 100, a Reynolds number where the von Kármán vortex street can be observed in the cylinder's wake. The Reynolds number is defined as $Re = U_\infty D/\nu$, where U_∞ is the incoming flow velocity and D is the diameter of the cylinder obstructing the flow. In addition, we test each boundary representation for a steady flow ($Re = 20$) and an unsteady flow with more chaotic tendencies ($Re = 400$). Note that three-dimensional effects in the latter flow are suppressed as we restrict ourselves to a two-dimensional domain. A rectangular domain with domain size $L_x \times L_y = 10D \times 20D$ is used, with a body-fitted grid with diameter $3D$ used in the overset grid simulations. For overset grids,

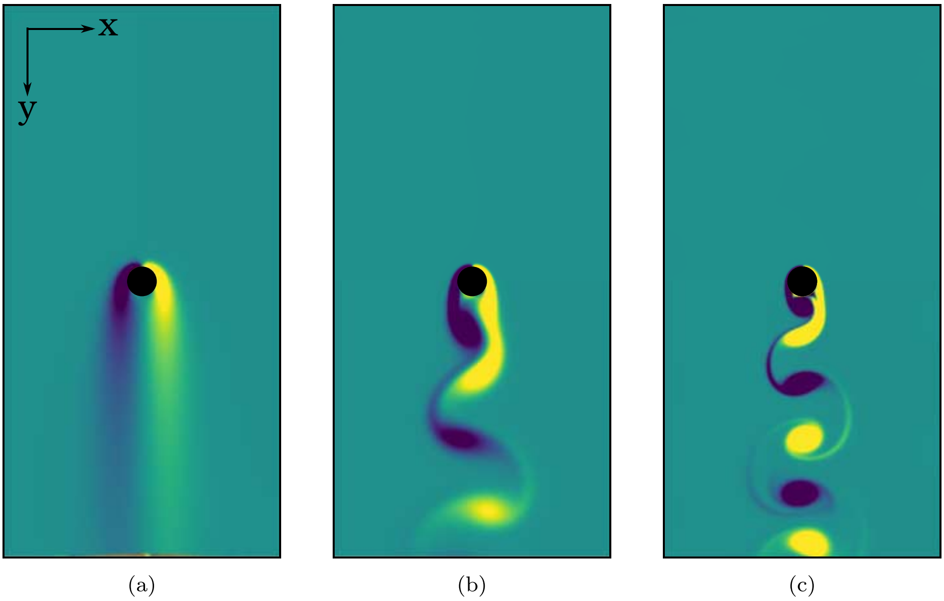


Figure 4. Flow visualisation. Contours of instantaneous vorticity $\omega_z = [\nabla \times \boldsymbol{u}]_z$ (normal to the view plane) plotted for three different Reynolds numbers. Inflow at the top of plane. (a) $Re = 20$, (b) $Re = 100$ and (c) $Re = 400$. (Colour online).

grid stretching is used in the radial direction to obtain approximately quadratic cells at the cylinder surface and in the region where the interpolation between the grids is performed. In this region, the cells of both grids are similar in size. The inflow Mach number ($Ma = U_\infty/c_s$) is set to 0.1, and the Reynolds number is varied by adjusting the value of the kinematic viscosity. The vorticity component normal to the xy-plane for the three different Reynolds number flows with mean flow in the y-direction can be seen in figure 4.

Consider figure 5, depicting normalised deviations of mean drag coefficient and root-mean-square lift coefficient computed at different resolutions with the two methods for $Re = 100$. For both cases, the results from the finest grid are used for normalisation. Strouhal number (dimensionless shedding frequency) is not included in the figure, since it is barely affected by the grid spacing and is therefore not a good measure of grid independence. The number of grid points per diameter, on the Cartesian grid, is given on the horizontal axis. Note that this might be somewhat misleading, as the overset grid uses two grids to cover the flow domain. For the flow domain and grid sizes used in this study, an overset grid simulation uses 10% more grid points than a corresponding IBM simulation when $D/\Delta x$ of the two simulations is the same. No matter this difference, it is clear that the overset grid method greatly outperforms the IBM with respect to the necessary grid required to reach grid independency under the conditions of these simulations. Using a background grid with $D/\Delta x \geq 24$, the deviation from results on a $D/\Delta x = 64$ grid is less than 0.16% for the overset grid. The results from the IBM calculations converge much slower. To reach a comparable level of grid independence to that of the overset grid with $D/\Delta x = 24$, a grid using IBM requires $D/\Delta x \geq 112$. Such a fine grid yields a deviation of less than 0.1% in drag and 0.16% in lift, from the results at the finest grid level ($D/\Delta x = 128$). If these grids ($D/\Delta x = 24$ for overset grids, $D/\Delta x = 112$ for IBM)

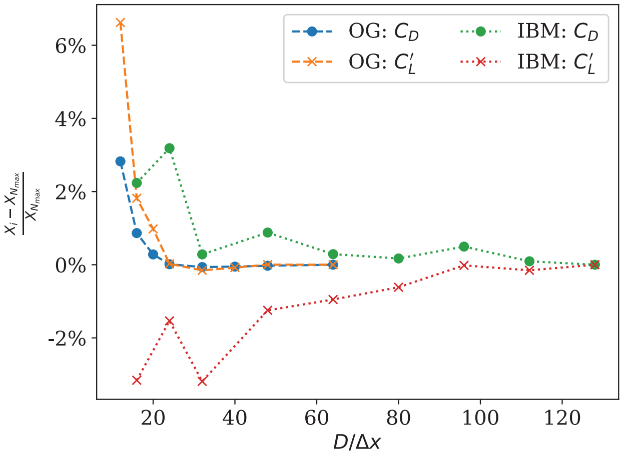


Figure 5. Normalised values for mean drag coefficient (C_D) and root-mean-square lift coefficient (C_L') for flow with $Re = 100$ computed at grids with varying coarseness. The cylinder in the flow is represented either by the immersed boundary method (IBM) or with overset grids (OG). (colour online).

are deemed sufficiently accurate resolutions for grid independent solutions for the different solid object representations, the IBM requires 18.1 times as many grid points as the overset grid method on the two-dimensional domain used in these simulations. In these simulations, the advective restriction is more strict than the viscous restriction, hence, the time step is proportional to the grid spacing. This means that there is a factor 4.7 difference in time step between $D/\Delta x = 24$ and $D/\Delta x = 112$, that has an additional large impact on the (in)efficiency of the IBM as compared to the overset grids.

For practical application, it may perhaps be excessive to require $\approx 0.1\%$ deviation for results to be deemed grid independent. With a resolution $D/\Delta x \geq 64$ in the simulations performed with IBM, there is less than 0.5% deviation in drag and less than 1% deviation in lift. Choosing such a resolution will, in many cases, be an acceptable trade-off between accuracy and efficiency. This reduces the difference between the overset grid and IBM somewhat, although it warrants the use of a somewhat coarser grid for overset grid computations as well. Note that in computing drag and lift forces on the cylinder represented by the IBM, so-called force-points are used. The force-points are distributed uniformly around the cylinder, and viscous and pressure forces are approximated at these points, using data from surrounding grid points. Some of the oscillations in the computed mean drag and root-mean-square lift coefficients seen in figure 5 may be due to a change in the position of force-points when the grid is refined.

A relevant consideration when the different costs associated with overset grids and IBM are compared is the computational cost of interpolation in the two different methods. With equally spaced Cartesian grids, more fringe-points are interpolated with overset grids than mirror-points interpolated with the ghost-point IBM method. This is due to overset grids having two zones of interpolation (one for interpolation from Cartesian to cylindrical and one for interpolation back to Cartesian), a larger circumference of the interpolation regions (interpolation farther from the cylinder) and the need for a deeper ghost-zone on the Cartesian grid in overset grids since no special handling for fluid-points close to the fringe-point region is used. For the $D/\Delta x = 24$ grid, the total number of fringe-points for

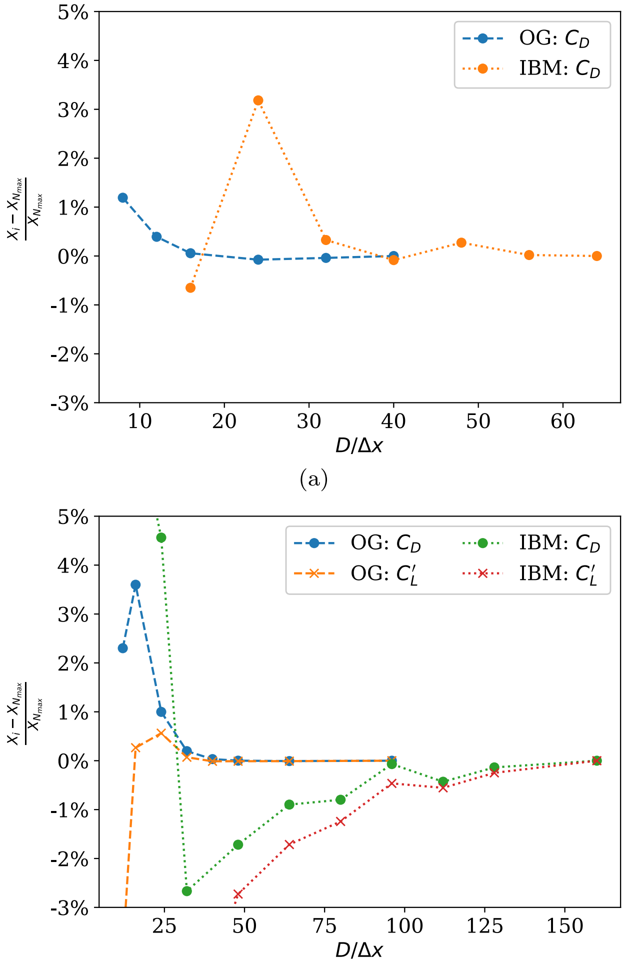
Table 1. Comparison with data sets from previous studies for $Re = 100$.

	$[(L_{x_u} + L_{x_d}) \times L_y]/D^2$	C_D	C_L'	St
Kim *et al.* (2001)	70×100	1.33	$0.22^{(*)}$	0.165
Pan (2006)	60×60	1.32	$0.226^{(*)}$	0.16
Haugen and Kragset (2010)	70×35	1.328	–	0.166
Park *et al.* (1998)	$(50 + 20) \times 100$, ◖	1.33	$0.235^{(*)}$	0.165
Shi *et al.* (2004)	300 ○	1.318	–	0.164
Mittal (2005)	100×100	1.322	0.226	0.164
Stålberg *et al.* (2006)	160, ○	1.32	$0.233^{(*)}$	–
Li *et al.* (2009)	100×100	1.336	–	0.164
Posdziech and Grundmann (2007)	$(20 + 50) \times 40$ ◖	1.350	$0.234^{(*)}$	0.167
Posdziech and Grundmann (2007)	$(4000 + 50) \times 8000$ ◖	1.312	$0.224^{(*)}$	0.163
Qu *et al.* (2013)	60×60	1.326	0.2191	0.166
Qu *et al.* (2013)	200×200	1.310	0.2151	0.165
Present, IBM	50×50	1.351	0.232	0.166
Present, overset grid	50×50	1.347	0.234	0.166

Asterisk denotes scaled values of C_L. The non-rectangular grids are marked as circular inlet/C-type (◖) or circular/O-grid (○). Domains in which the cylinder is not centred have both upstream and downstream lengths given.

the overset grids is approximately 1600 ($<$ 1.3% of total number of grid points). Note that the fraction of fringe-points to grid points decreases as the grid spacing decreases ($<$0.65% grid points are fringe-points when $D/\Delta x = 48$). The number of mirror-points in an IBM simulation with the same grid spacing is approximately 200. A fringe-point is updated only once every Runge–Kutta time step, while mirror-points are updated every sub-time step. Hence, approximately 2.5 times as much interpolation is performed when the solid is represented by overset grids rather than by IBM, if the same grid spacing is used in the different solid object representations. As much finer grids are required with IBM than with overset grids, the advantage of a smaller interpolation cost with IBM is lost.

To verify that the flow is computed accurately, the resolutions from the discussion above ($D/\Delta x = 24$ for overset grids, $D/\Delta x = 64$ for IBM) are used in a simulation on a large domain for each of the solid body representations. The domain size is set to $L_x = L_y = 50D$. The resulting mean drag, root-mean-square lift and Strouhal frequency are compared to results reported from other studies in table 1. Domain sizes and types are listed in the table, along with the most relevant flow coefficients. Some of the listed values for root-mean-square lift coefficients are scaled values, as only amplitude of the lift coefficient was reported from these particular studies. A scaling factor of $1/\sqrt{2}$ has been used (since the lift coefficient is a smooth sinusoidal-like function with zero mean value), and the scaled results are marked with a superscript (*). The studies in table 1 use a wide range of numerical methods to compute the flow, including finite-volume, finite-difference, finite-element, spectral element and lattice-Boltzmann methods. The top three studies in table 1 use immersed boundary methods to represent the solid cylinder, while the remaining studies (present IBM simulations excluded) use body-fitted methods. Only Haugen and Kragset (2010) and Li *et al.* (2009) simulate compressible flows (where the former of these uses the Pencil Code with the IBM described here, but with different domain size and resolution). Table 1 includes two results from each of the studies by Posdziech and Grundmann (2007) and Qu *et al.* (2013) to include results from both comparable domain sizes to the present study, and highly accurate results from very large domains. The present results, both those computed with the IBM and the results from overset grid simulations, agree well with the results found in the literature.



(a)

(b)

Figure 6. Normalised values for mean drag coefficient (C_D) for flow with $Re = 20$, mean drag coefficient and root-mean-square lift coefficient (C_L') for flow with $Re = 400$. Results are computed at grids with varying coarseness. The cylinder in the flow is represented either by the immersed boundary method (IBM) or with overset grids (OG). (a) $Re = 20$ and (b) $Re = 400$. (Colour online).

Grid independence results for $Re = 20$ and $Re = 400$ are depicted in figure 6. The results are similar to those obtained for $Re = 100$: there is a much more rapid convergence to grid independent solutions with overset grids than with IBM. With overset grids, a background grid with $D/\Delta x \geq 32$ yields less than 0.2% deviation in the drag and lift coefficients, at $Re = 400$. This means using overset grids $N_x \times N_y + N_r \times N_\theta = 320 \times 640 + 64 \times 320$. With IBM, a grid with $D/\Delta x \geq 128$ is necessary to get comparable grid independence (deviation less than 0.25% in drag and lift coefficients). That means using a 1280×2560 grid for this specific domain size, and a factor 4 larger time step than on the background grid in the overset grid simulation. For the steady flow, the lift coefficient is not defined, and for this reason we include only the drag coefficient in the grid independence comparison. The results in figure 6(a) show that $D/\Delta x = 12$ is sufficient to obtain drag with less than

0.4% deviation from the finest grid result for the overset grid computation. With IBM, $D/\Delta x = 32$ is needed to get the deviation down to the same level.

From these tests, it is clear that in representing the simple geometry of a circular cylinder with the Pencil Code, the method of overset grids is far superior to the IBM in terms of efficiency and accuracy. Not only is far less grid points required (and, consequently a larger time step allowed) to reach a grid independent solution, there is also far less variation in the solution before grid independence is reached. With the IBM, we may have to accept a deviation of, say, 1.0% in mean drag and root-mean-square lift coefficients from one grid to a finer one. In the case of overset grids, on the other hand, a deviation one order of magnitude smaller than this can be achieved at reasonable computational costs. As mentioned, however, some of the variation seen in the IBM results may be attributed to the way the coefficients themselves, and not the flow, are computed. The positioning of force-points where drag and lift are computed is affected by the choice of resolution, but has no influence on the solution of the flow equations.

That being said, we should now address the limitations of the overset grid method. In short, the problem of adaptability to different geometries is a major drawback of this method. Even an extension from a 3D cylinder to a sphere would require a completely new grid handling, with updates needed all the way down to the level of finite-differences in the code. For more complex shapes, where an analytic transformation from Cartesian space to the fitted grid coordinates is not available, this will become increasingly difficult, if not impossible. It is in this respect that the full potential of the IBM can be achieved. The simple handling of boundaries and lack of any modification needed in the treatment of the governing equations make the ghost-zone IBM ideal for complex geometries of all kinds. How this is done in the Pencil Code is the topic of the remainder of this paper.

4. Complex geometries

One of the difficulties in extension to irregular geometries of a ghost-cell immersed boundary method's lies in how to track the boundaries correctly. To the best of our knowledge, two ways to overcome this exist: the unstructured triangle surface mesh (Gilmanov *et al.* 2003; Mittal *et al.* 2008; Nagendra *et al.* 2014) and the combination with level-set signed distance functions (Liu and Hu 2014; Uddin *et al.* 2014). The first method can be used to represent arbitrary geometries and has gained its popularity in biological fluid mechanics. For example, interactions between a very complex body, such as a bluegill sunfish pectoral fin or a false vocal fold, and its surrounding flows have been studied in a two-way coupled manner via the first method (Zheng *et al.* 2009; Dong *et al.* 2010). This method, with an unstructured surface mesh for the complex boundary, was introduced and implemented in the Pencil Code by Luo *et al.* (2016). Arbitrary two-dimensional immersed boundaries are represented by many small line-segments. Each line-segment is identified by two vertices as shown in figure 7. The general procedure is still the same as illustrated in section 2.2.1, other than some special handling of fluid-points and mirror-points around the solid object.

The first difference lies in the identification of a given grid point to be a fluid-point or a solid-point. For a circular object, the distance from a given grid point to the centre of the circle is calculated and compared with the radius of the circular object to identify if this grid point is a solid grid point or not, i.e. if it is inside the object or not. This becomes an ineffective method for a complex geometry, where no single radius can be found for

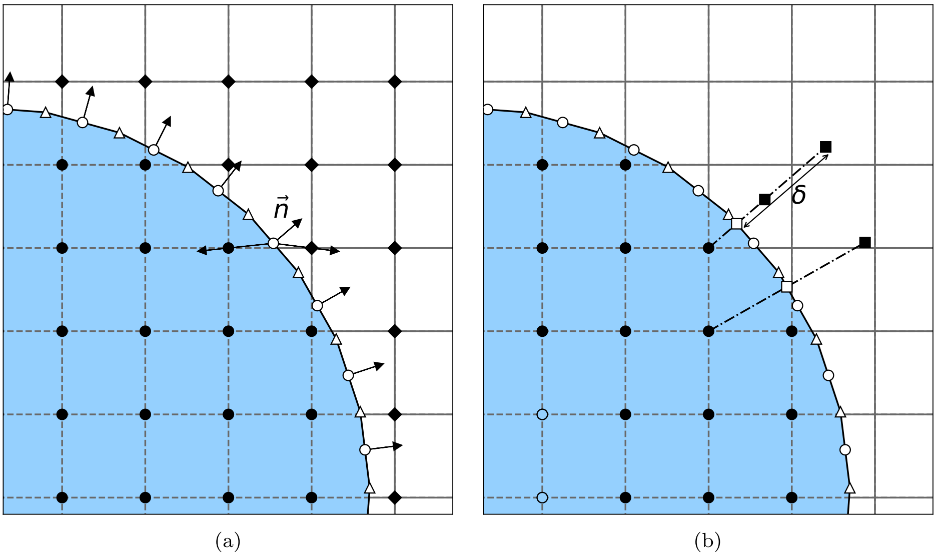


Figure 7. Immersed boundary method. (a) Identifying grid points as solid-points or fluid-points for a two-dimensional geometry by dot product of line-segment normal and directional vector grid points. (b) Schematic diagram for ghost points and the method to assign mirror/image-points along lines normal to line-segments between vertex points. (●) solid-point, (◆) fluid-point, (○) centroid of line-segment, (△) vertex of line-segment, ($\vec{n}$) line-segment's normal vector, (□) boundary intersect point, (■) mirror-point. (a) Identifying fluid/solid-points and (b) mirror-point computation.

the object. In this case, for a given grid point, the closest surface element is detected first. Second, the dot product between the closest line-segment's normal vector and the direction vector pointing from the centroid of the closest facet to the given grid point is calculated. The sign of the dot product determines the identification of the grid point. Generally, a negative result indicates a solid-point for the convex boundary shown in figure 7(a). The treatment of some special cases that may occur during this process is described in Luo *et al.* (2017).

After the identification of solid/fluid-points, three layers of ghost points are assigned to construct a six-order central finite-difference stencil as shown in figure 7(b). Following this, a corresponding boundary intercept point is determined for each one of them. The method to detect the boundary intercept points is different from that of the simple circular geometry. First, the vertex closest to a given ghost point is determined. Then, the set of line/surface elements sharing that vertex can be identified and a search is carried out among these elements to find the boundary intercept point (which should lie within the line/surface elements) as shown in figure 7(b). While conceptually simple, the implementation can be very complicated and special attention is needed to find the correct intercept point. Here, we adopt a method based on the robust procedure proposed in Mittal *et al.* (2008). For details, see Luo *et al.* (2017).

Once boundary intercept points are determined for every ghost point, a corresponding mirror-point can be obtained. The mirror-points are set either by symmetry over the solid's line element (corresponding to the way mirror-points are set with a simple geometry, see figure 2), or at a constant distance away from the boundary intercept point. This distance, δ

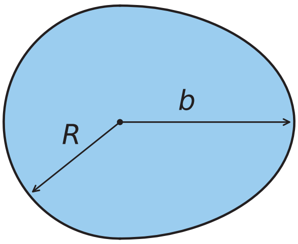


Figure 8. Complex geometry. A combination of a semi-circular cylinder with radius *R*, and semi-elliptical cylinder with major axis *b* and minor axis *R*. The circular side of the geometry faces the inlet.

in figure 7(b), is typically set to $\sqrt{2}\Delta x$ for 2D geometries and $\sqrt{3}\Delta x$ for 3D geometries, to ensure that every mirror-point is surrounded by fluid-points only. The same interpolation procedure as for the simple circular geometry (i.e. the bilinear interpolation method) can be adopted for the calculations of the parameters of the mirror-points. An optional way is the inverse distance weight interpolation method (Chaudhuri *et al.* 2011). Finally, the flow variables at the ghost-points can be calculated with the aid of the mirror-points and the given boundary conditions at the boundary intercept point by linear interpolation. Three types of boundary conditions, the Dirichlet, Neumann or Robin boundary condition, have been implemented, similarly as for the simple circular object discussed in section 2.2.1. For more details and test of the boundary conditions, the reader is referred to Luo *et al.* (2016).

This new method can be straightforwardly extended to complex three-dimensional geometries, where triangular surface elements can be adopted to represent the surfaces. Details about the method and its implementation can be found in Luo *et al.* (2017) and will not be repeated here. In Luo *et al.* (2017), a spatial convergence test indicates that only the bilinear interpolation procedure can obtain a local second-order accuracy. Systematic validations have also been conducted through calculations of flow past an elliptical cylinder, square cylinder, semi-cylinder, as well as an NACA0012 airfoil. Quantitative comparisons with reported results in the literature show that the present method can accurately reproduce the main features of the fluid flow past solid objects with complex geometry, quantified by coefficients such as drag and lift coefficients, Nusselt number and Strouhal number.

To demonstrate the IBM capabilities for a two-dimensional flow, we have simulated flow past geometries constructed by combining a semi-circle and semi-elliptical cylinder. The geometry is seen in figure 8. The radius of the semi-circle (R) and the major axis of the semi-ellipse (b) can be varied to construct different geometries. Three cases, with $R/b =$ 2.0, 1.0 (circle), and 0.5, respectively, are considered. For each case, 360 line segments are used to resolve the immersed geometry. Other parameters related to the computational domain are kept consistent with the $Re = 100$ case in the grid refinement part of section 3, except that the solid body is placed at a distance $5D$ from the inlet, in the streamwise direction, rather than in the centre of the flow domain ($10D$ from the inlet).

The vorticity component ω_z normal to the view plane for flow past the three different geometries can be seen in figure 9. It can be seen that as the length b of the semi-ellipse is increased, the length of the bound vortex increases accordingly. This results in different patterns of von Kármán vortex streets for each of the three cases. The corresponding mean

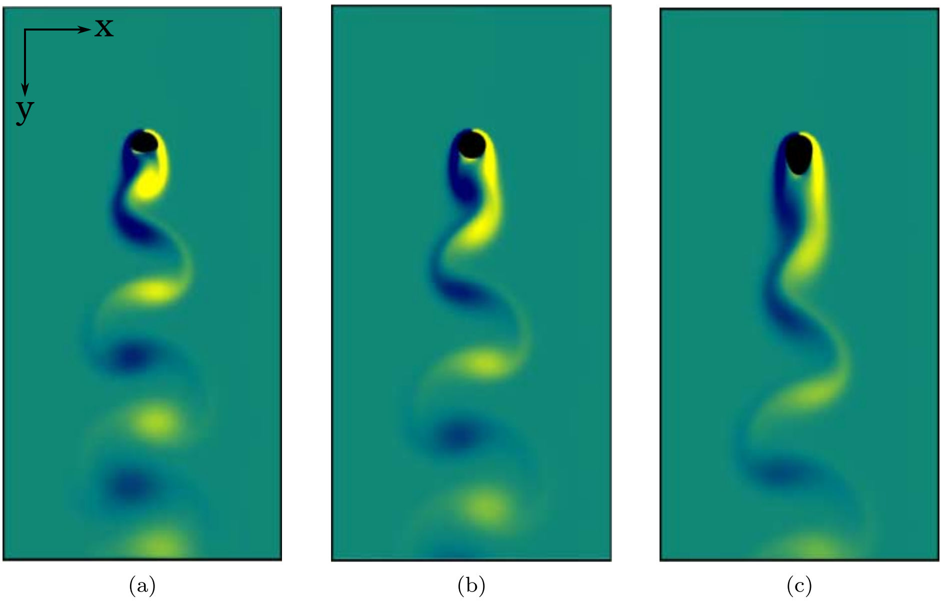


Figure 9. Flow visualisation. Contours of instantaneous vorticity $\omega_z = [\nabla \times \boldsymbol{u}]_z$ (normal to the view plane) plotted for three different geometries at $Re = 100$. Inflow at the top of plane. (a) $R/b = 2.0$, (b) $R/b = 1.0$ and (c) $R/b = 0.5$. (Colour online).

Table 2. Comparisons of mean drag coefficient, root-mean-square lift coefficient, and Strouhal number for different geometries.

R/b	C_D	C_L'	St
2.0	1.55	0.34	0.190
1.0	1.35	0.26	0.175
0.5	1.17	0.11	0.168

drag coefficient, root-mean-square lift coefficient and Strouhal number are listed in table 2. It is obvious that even though the immersed boundary of the third geometry is the longest, the drag force it experiences is the least. This is perhaps not surprising, as the shape of the object is closer to a streamlined body for the largest value of b, a shape that is known for low drag.

5. Concluding remarks

In this study, we have described and compared the two solid body representations available in the high-order finite-difference code known as the Pencil Code. The two methods, the immersed boundary method and overset grids, are fundamentally different in many aspects. These differences can be summed up as

(a) The ghost-point IBM can be implemented straightforwardly in an existing flow solver, by extending the code without requiring major modifications to the existing solver. Using overset grids requires a more generalised flow solver, able to handle all grids that are overset one another (Cartesian, cylindrical, etc.). This may require a modification of the flow solver itself, when overset grids are first implemented in an existing fluid dynamics code.

(b) Neither the IBM nor overset grids are mass conserving, as they both rely on interpolation of flow quantities to either mirror-points or non-conforming grid points, respectively. As the interpolation is moved away from the solid surface when overset grids are used, the accuracy loss following from interpolation is expected to have a reduced impact on flow properties directly related to the solid object (boundary layer properties, etc.).

(c) For a circular cylinder, using the overset grid method in the Pencil Code is far superior to using IBM. Reaching grid independent solutions with IBM required 4.7 and 4 times as many grid points in each direction for $Re = 100$ and 400, respectively, as compared to the background grid used in the overset grids method. In total, this meant using 18.1 and 14.5 times as many grid points in our two-dimensional simulations with IBM as compared to that with overset grids at these Reynolds numbers. In addition, there comes a much stricter limitation on the time step for the fine grid used in the IBM. Such a limitation is only present on the curvilinear grid in the overset grid method, while a 4–5 times as large time step can be used on the coarse background grid.

(d) The IBM is highly flexible. The implementation in the Pencil Code can handle complex geometries, that is, surfaces where an analytic surface representation is not available. This opens up a large area of research that cannot be studied with overset grids.

The development of both IBM and overset grids is far from over, and the evolution of these methods in the Pencil Code is destined to continue as long as researchers use the code and implement their own improvements into this open-source software. Perhaps, in time, overset grids can become more flexible, and the high-order accuracy achieved for the simple geometry can be available for more complex shapes (e.g. by using many grids overset one another). Alternatively (or, perhaps, in addition), the accuracy of the IBM implementation may be improved through the implementation of stable, high-order interpolation of mirror-points in the vicinity of the solid object. We hope that with this paper, more researchers will be attracted to use the Pencil Code for simulations of flow past solid objects. Only in this way can the advancements of the solid object representations in the software continue, in the spirit of open-source software development.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

We would like to acknowledge that this research is funded by the Research Council of Norway (Norges Forskingsråd) under the FRINATEK Grant [grant number 231444] and by the National Natural Science Foundation of China [No. 51576176]. The research is supported in part by the GrateCFD project [grant 267957/E20], which is funded by LOGE AB, Statkraft Varme AS, EGE Oslo, Vattenfall AB, Hitachi Zosen Inova AG and Returkraft AS together with the Research Council of Norway through the ENERGIX program. Computational resources were provided by UNINETT Sigma2 AS [project numbers NN9405K, NN2649K].

ORCID

Jørgen R. Aarnes http://orcid.org/0000-0002-5899-2597

References

Aarnes, J.R., Andersson, H.I. and Haugen, N.E.L., High-order overset grid method for detecting particle impaction on a cylinder in a cross flow (Manuscript submitted for publication), 2018a.

Aarnes, J.R., Andersson, H.I. and Haugen, N.E.L., Numerical investigation of free-stream turbulence effects on the transition-in-wake state of flow past a circular cylinder. *J. Turbul.* 2018b, **19**, 252–273.

Benek, J.A., Buningt, P.G. and Steger, J.L., A 3-D chimera grid embedding technique, in *AIAA 7th Computational Fluid Dynamics Conference*, 1985.

Berthelsen, P.A. and Faltinsen, O.M., A local directional ghost cell approach for incompressible viscous flow problems with irregular boundaries. *J. Comput. Phys.* 2008, **227**, 4354–4397.

Causon, D.M., Ingram, D.M., Mingham, C.G., Yang, G. and Pearson, R.V., Calculation of shallow water flows using a Cartesian cut cell approach. *Adv. Water Resour.* 2000, **23**, 545–562.

Chaudhuri, A., Hadjadj, A. and Chinnayya, A., On the use of immersed boundary methods for shock/obstacle interactions. *J. Comput. Phys.* 2011, **230**, 1731–1748.

Chesshire, G. and Henshaw, W.D., Composite overlapping meshes for the solution of partial differential equations. *J. Comput. Phys.* 1990, **90**, 1–64.

Chicheportiche, J. and Gloerfelt, X., Study of interpolation methods for high-accuracy computations on overlapping grids. *Comput. Fluids* 2012, **68**, 112–133.

Dobler, W., Stix, M. and Brandenburg, A., Magnetic field generation in fully convective rotating spheres. *Astrophys. J.* 2006, **638**, 336–347.

Dong, H., Bozkurttas, M., Mittal, R., Madden, P. and Lauder, G., Computational modelling and analysis of the hydrodynamics of a highly deformable fish pectoral fin. *J. Fluid Mech.* 2010, **645**, 345–373.

Gaitonde, D.V. and Visbal, M.R., Padé-type higher-order boundary filters for the Navier–Stokes equations. *AIAA J.* 2000, **38**, 2103–2112.

Gilmanov, A., Sotiropoulos, F. and Balaras, E., A general reconstruction algorithm for simulating flows with complex 3D immersed boundaries on Cartesian grids. *J. Comput. Phys.* 2003, **191**, 660–669.

Haugen, N.E.L. and Brandenburg, A., Hydrodynamic and hydromagnetic energy spectra from large eddy simulations. *Phys. Fluids* 2006, **18**, 075106.

Haugen, N.E.L. and Kragset, S., Particle impaction on a cylinder in a crossflow as function of Stokes and Reynolds numbers. *J. Fluid Mech.* 2010, **661**, 239–261.

Ingram, D.M., Causon, D.M. and Mingham, C.G., Developments in Cartesian cut cell methods. *Math. Comput. Simul.* 2003, **61**, 561–572.

Kageyama, A. and Sato, T., 'Yin-Yang grid': an overset grid in spherical geometry. *Geochem. Geophys. Geosyst.* 2004, **5**.

Kim, J., Kim, D. and Choi, H., An immersed-boundary finite-volume method for simulations of flow in complex geometries. *J. Comput. Phys.* 2001, **171**, 132–150.

Li, Y., Zhang, R., Shock, R. and Chen, H., Prediction of vortex shedding from a circular cylinder using a volumetric Lattice-Boltzmann boundary approach. *Eur. Phys. J. Spec. Top.* 2009, **171**, 91–97.

Liu, C. and Hu, C., An efficient immersed boundary treatment for complex moving object. *J. Comput. Phys.* 2014, **274**, 654–680.

Luo, K., Zhuang, Z., Fan, J. and Haugen, N.E.L., A ghost-cell immersed boundary method for simulations of heat transfer in compressible flows under different boundary conditions. *Int. J. Heat Mass Transfer* 2016, **92**, 708–717.

Luo, K., Mao, C., Zhuang, Z., Fan, J. and Haugen, N.E.L., A ghost-cell immersed boundary method for the simulations of heat transfer in compressible flows under different boundary conditions part-II: complex geometries. *Int. J. Heat Mass Transfer* 2017, **104**, 98–111.

Mattsson, K. and Nordström, J., Summation by parts operators for finite difference approximations of second derivatives. *J. Comput. Phys.* 2004, **199**, 503–540.

Mavriplis, D., Unstructured grid techniques. *Annu. Rev. Fluid Mech.* 1997, **29**, 473–514.

Meakin, R., The chimera method of simulations for unsteady three-dimensional viscous flow. In *Computational Fluid Dynamics Review*, edited by M. Hafez and K. Oshima, pp. 70–86, 1995 (Wiley: New York, NY).

Mittal, S., Excitation of shear layer instability in flow past a cylinder at low Reynolds number. *Int. J. Numer. Methods Fluids* 2005, **49**, 1147–1167.

Mittal, R. and Iaccarino, G., Immersed boundary methods. *Annu. Rev. Fluid Mech.* 2005, **37**, 239–261.

Mittal, R., Dong, H., Bozkurttas, M., Najjar, F., Vargas, A. and von Loebbecke, A., A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries. *J. Comput. Phys.* 2008, **227**, 4825–4852.

Nagendra, K., Tafti, D.K. and Viswanath, K., A new approach for conjugate heat transfer problems using immersed boundary method for curvilinear grid based solvers. *J. Comput. Phys.* 2014, **267**, 225–246.

Noack, R.W., SUGGAR: a general capability for moving body overset grid assembly, in *17th AIAA Computational Fluid Dynamics Conference*, 2005.

Owen, S.J., A survey of unstructured mesh generation technology, in *Proceedings of 7th International Meshing Roundtable*, 1998, pp. 239–267.

Pan, D., An immersed boundary method on unstructured Cartesian meshes for incompressible flows with heat transfer. *Numer. Heat. Tr. B.-Fund.* 2006, **49**, 277–297.

Park, J., Kwon, K. and Choi, H., Numerical solutions of flow past a circular cylinder at Reynolds numbers up to 160. *KSME Int. J.* 1998, **12**, 1200–1205.

Pärt-Enander, E. and Sjögreen, B., Conservative and non-conservative interpolation between overlapping grids for finite volume solutions of hyperbolic problems. *Comput. Fluids* 1994, **23**, 551–574.

Peskin, C.S., Flow patterns around heart valves: a numerical method. *J. Comput. Phys.* 1972, **10**, 252–271.

Peskin, C.S., The immersed boundary method. *Acta Numer.* 2002, **11**, 479–517.

Poinsot, T.J. and Lele, S., Boundary conditions for direct simulations of compressible viscous flows. *J. Comput. Phys.* 1992, **101**, 104–129.

Posdziech, O. and Grundmann, R., A systematic approach to the numerical calculation of fundamental quantities of the two-dimensional flow over a circular cylinder. *J. Fluids Struct.* 2007, **23**, 479–499.

Qu, L., Norberg, C., Davidson, L., Peng, S.H. and Wang, F., Quantitative numerical analysis of flow past a circular cylinder at Reynolds number between 50 and 200. *J. Fluids Struct.* 2013, **39**, 347–370.

Quirk, J.J., An alternative to unstructured grids for computing gas dynamic flows around arbitrarily complex two-dimensional bodies. *Comput. Fluids* 1994, **23**, 125–142.

Rogers, S.E., Suhs, N.E. and Dietz, W.E., PEGASUS 5: an automated preprocessor for overset-grid computational fluid dynamics. *AIAA J.* 2003, **41**, 1037–1045.

Schneiders, L., Hartmann, D., Meinke, M. and Schröder, W., An accurate moving boundary formulation in cut-cell methods. *J. Comput. Phys.* 2013, **235**, 786–809.

Seo, J.H. and Mittal, R., A high-order immersed boundary method for acoustic wave scattering and low-Mach number flow-induced sound in complex geometries. *J. Comput. Phys.* 2011, **230**, 1000–1019.

Sherer, S.E. and Scott, J.N., High-order compact finite-difference methods on general overset grids. *J. Comput. Phys.* 2005, **210**, 459–496.

Shi, J.M., Gerlach, D., Breuer, M., Biswas, G. and Durst, F., Heating effect on steady and unsteady horizontal laminar flow of air past a circular cylinder. *Phys. Fluids* 2004, **16**, 4331–4345.

Stålberg, E., Brüger, A., Lötstedt, P., Johansson, A.V. and Henningson, D.S., High order accurate solution of flow past a circular cylinder. *J. Sci. Comput.* 2006, **27**, 431–441.

Steger, J.L. and Benek, J.A., On the use of composite grid schemes in computational aerodynamics. *Comput. Methods. Appl. Mech. Eng.* 1987, **64**, 301–320.

Steger, J.L., Dougherty, F.C. and Benek, J.A., A chimera grid scheme, in *Advances in Grid Generation*, edited by K.N. Ghia and U. Ghia, pp. 59–69, 1983 (American Society of Mechanical Engineers: New York, NY).

Strand, B., Summation by parts for finite difference approximations for d/dx. *J. Comput. Phys.* 1994, **110**, 47–67.

Tannehill, J.C., Anderson, D. and Pletcher, R.H., *Computational Fluid Mechanics and Heat Transfer*, 2 edn, 1997 (Taylor & Francis: Philadelphia, PA).

The Pencil Code, (pencil-code.nordita.org [Internet]), 2018. Stockholm (SE): NORDITA; [updated February 2018]. Available online at: https://github.com/pencil-code.

Tseng, Y.H. and Ferziger, J.H., A ghost-cell immersed boundary method for flow in complex geometry. *J. Comput. Phys.* 2003, **192**, 593–623.

Uddin, H., Kramer, R. and Pantano, C., A Cartesian-based embedded geometry technique with adaptive high-order finite differences for compressible flow around complex geometries. *J. Comput. Phys.* 2014, **262**, 379–407.

Versteeg, H.K. and Malalasekera, W., *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*, 2 edn, 2007 (Pearson Education: Harlow).

Visbal, M.R. and Gaitonde, D.V., High-order-accurate methods for complex unsteady subsonic flows. *AIAA J.* 1999, **37**, 1231–1239.

Visbal, M.R. and Gaitonde, D.V., On the use of higher-order finite-difference schemes on curvilinear and deforming meshes. *J. Comput. Phys.* 2002, **181**, 155–185.

Völkner, S., Brunswig, J. and Rung, T., Analysis of non-conservative interpolation techniques in overset grid finite-volume methods. *Comput. Fluids* 2017, **148**, 39–55.

White, F.M., *Viscous Fluid Flow*, 3 edn, 2006 (McGraw-Hill Education: New York, NY).

Williamson, J., Low-storage runge-kutta schemes. *J. Comput. Phys.* 1980, **35**, 48–56.

Xia, J., Luo, K. and Fan, J., A ghost-cell based high-order immersed boundary method for inter-phase heat transfer simulation. *Int. J. Heat Mass Transfer* 2014, **75**, 302–312.

Zang, Y. and Street, R.L., A composite multigrid method for calculating unsteady incompressible flows in geometrically complex domains. *Int. J. Numer. Methods Fluids* 1995, **20**, 341–361.

Zheng, X., Bielamowicz, S., Luo, H. and Mittal, R., A computational study of the effect of false vocal folds on glottal flow and vocal fold vibration during phonation. *Ann. Biomed. Eng.* 2009, **37**, 625–642.

Appendix. Sample cases

To get started with simulations of flow past a cylindrical geometry using the Pencil Code, sample cases that are available with the download of the code (The Pencil Code 2018) may be useful. From the pencil-code directory, the sample cases can be found in

```
./samples/2d-tests/cylinder_deposition
./samples/2d-tests/cylinder_deposition_ogrid
```

The postfix ogrid denotes the overset grid sample case. To compile and run a sample case, use the commands `pc_build`, `pc_start` and `pc_run`. Both sample cases are simulations of a particle-laden flow past a cylinder at $Re = 100$, in which particles may impact and deposit on the cylinder surface. For documentation on the handling of particles and particle deposition with the cylinder represented by IBM and overset grids, the reader is referred to Haugen and Kragset (2010) and Aarnes et al. (2018a), respectively.

NTNU

GODKJENNING
FRA
OPPNEVNT SENSURKOMITE
av avhandling for graden

PERSONALOPPLYSNINGER

Kandidat: Jørgen Røysland Aarnes

Institutt: Department of Energy and Process Engineering

AVHANDLING

Tittel på avhandlingen:

Simple Geometry – Complex Flow

A computational study of turbulent particle-laden flows impinging on a cylinder

SENSURKOMITEENS VURDERING

Avhandlingen er bedømt og godkjent for graden philosophiae doctor □
Avhandlingen er bedømt og godkjent for graden doctor philosophiae □

Trondheim, den

.... navn .... underskrift

.... navn .... underskrift

.... navn .... underskrift

.... navn .... underskrift