Three dimensional simulations of a swimmer's hand using an immersed boundary method

van Houwelingen, J.

Award date:
2013

Three-dimensional simulations of a swimmers's hand using an Immersed Boundary Method

J. van Houwelingen

August, 2013

R-1823-A

Master Thesis in Applied Physics

Supervisors:

Dr. ir. W. van de Water
Prof. dr. ir. G.J.F. van Heijst
Prof. dr. H.J.H. Clercx

Abstract

The force coefficients of the hand (and forearm) of a human swimmer have been determined in several experimental studies and direct numerical simulations (DNS) before. In the present study the flow around a swimmer's hand and forearm is simulated at different velocities and angles of attack using an immersed boundary code. The force coefficients were calculated and compared with previous studies. Before undertaking the study of the swimmer's hand, simulations of the flow around a sphere were performed to validate the code. The drag coefficient fitted the expected curve quite well, except for data around the critical Reynolds number where the boundary layer should have become turbulent. Also the distinct flow regimes in the wake of the sphere were consistent with results of previous studies. Furthermore, the additional force on an accelerating sphere seems to agree quite well with the added mass theory. It can be concluded from the study of the hand that under steady flow conditions the drag- and lift coefficients were constant throughout the range of swimming speeds tested, which is consistent with previous research. The drag force for the particular hand model in this study peaked at an angle of attack of $\theta = 0^\circ$, which corresponds with the orientation of the hand palm exactly facing the flow. Throughout the whole range of angles tested, the drag force was by far dominant. The trend of both the drag- and lift coefficients as a function of angle of attack was consistent with the expectations. The resultant force of drag and lift (a vector addition) peaked at an angle of attack of $\alpha = 0^\circ$. It was found that pulling the arms $\sim 3^\circ$ diagonally backward, with an angle of attack of 0° could gain 0.08N more propulsive force (propulsive force 54.76N). This is not a significant result. For a hand under accelerating conditions it was found that the added mass was approximately $\sim \frac{7}{10}$ of the total mass of the hand.

Contents

Introduction

Swimming is one of the major athletic sports and many efforts are being made to set new records. Swimming is a special sport, since the medium from which a swimmer can get its propulsion is also the medium which accounts for a fair amount of resistance. To succeed in swimming faster, thrust should be increased even further and the resistance should be decreased. It is not just about having the biggest muscular power, it is about applying the largest propulsive force to the water and having the smallest drag forces. This kind of studies concern minor improvements, which make the difference between a golden or silver medal on the Olympics or no medal at all.

The explanation of swimming requires the measurement of the fluid forces acting on the body of the swimmer. Especially the body parts responsible for the propulsion or body parts causing the largest resistance will be of interest. Research on swimmers is often difficult since they are constantly moving and rolling in all directions. And especially the measurement of the forces, power and pressure distributions are difficult due to restrictions of existing measurement devices. Nowadays, an increased number of studies is done numerically as an alternative to complex and costly experiments. Simulations of the flow around a human swimmer also pose numerical challenges, in particular because of the complex body shapes and movements.

This study is part of a swim project started by the Turbulence and Vortex Dynamics group (department of Applied Physics from the University of Technology in Eindhoven) in collaboration with InnoSportLab. InnoSportLab is located at the national training center for swimming in Eindhoven, where professional swimmers and a new generation of future elite swimmers train to improve their performances. At InnoSportLab research is performed on professional swimmers. With this swim project the human swimmer will be studied from a more physical point of view to gain more fundamental insight. Hopefully this leads to better understanding of the propulsive and drag forces of a swimmer in the end.

Swimming propulsion is a phenomenon not fully understood. The hand and forearm mainly provide drag-based propulsion, but for decades the discussion is going on whether lift-based propulsion must be included in the arm stroke to obtain optimal

propulsion. Several studies, both experimental and numerical, have been done to understand the hydrodynamics of the swimmer's hand (+ forearm) and hence elucidate the optimal way to propel. A selection of some studies is briefly mentioned below. Experimental results in a towing tank on the hydrodynamic drag and lift forces as a function of angle of attack and sweep back angle of a human hand/arm models were obtained by Berger et al. [1]. Numerical studies on the drag- and lift forces on a swimmer's hand/arm as a function of angle of attack have previously been performed by Bixler et al. [2] and Sato et al. [3]. Bixler performed simulations on a 3D hand and forearm. Sato performed a series of simulations at a range of angles of attack, did a number of simulations on the drag coefficient of the swimmer's hand under accelerating conditions and furthermore simulated a complete crawl stroke of the hand. Rouboa et al. [4] concentrated on the effect of acceleration on propulsive forces by doing simulations on different 2D hand models. It was reported that under accelerating flow conditions the propulsive force was approximately 22.5% higher. Steady state results by Rouboa in the same study were consistent with previous experimental studies. Other studies on the effect of finger spreading on swimming propulsion are performed numerically by Marinho et al. [5] and Minetti et al. [6]. An experimental study was performed by Sidelnik and Young [7]. It can be concluded from all these studies that a small finger spread ($10^\circ - 12^\circ$) creates significantly more propulsion. In addition, Marinho et al. [8] performed a numerical study on the effect of the thumb position on swimming propulsion. Adduction or either abduction influences the drag- and lift forces differently at different angles of attack.

The aim of the present project is to carry out three-dimensional simulations on the hand plus forearm of a human swimmer using an Immersed Boundary Method (IBM). An advantage of an immersed boundary method is that grid generation is much easier, because the body does not necessarily have to conform a Cartesian grid. Also, an immersed boundary method can handle moving boundaries more easily. As a result, the immersed boundary method uses less memory and CPU compared to simulations using a body fitted grid. Since the body of a swimmer is very complex (and moving) the immersed boundary method should be an appropriate choice for this project. A disadvantage is that imposing of the boundary conditions is not straightforward compared to the conventional methods. The immersed boundary code used in this study was made available by Prof. R. Verzicco from the University of Rome 'Tor Vergata' and Dr. M.D. de Tullio from the University of Bari. With a similar code Verzicco and Tullio achieved good results in a research about heart valves [9]. The simulations were performed with an axisymmetric code. This code was converted to a Cartesian version later on, which is used in this study. Another major difference is the active fluid-structure interaction, which was added in the code at that time.

To validate the code first a series of simulations at a broad range of Reynolds numbers ($Re = \rho UD/\mu$) will be performed for the flow around a sphere. The drag coefficient is

calculated. The drag coefficient of a sphere is well documented and therefore a good reference object. This is also a test case to validate in which range of Reynolds numbers the DNS solver or either the Large Eddy Simulator (LES) of the code can be used. Furthermore it will be briefly examined whether flow regimes in the wake of the sphere can be distinguished as in previous studies.

After the simulations with a sphere the flow around a rigid hand and forearm will be simulated, the force coefficients are calculated and compared with previous research. The drag- and lift coefficient as function of velocity (Reynolds number) and angle of attack under steady flow conditions will be studied. An attempt is made to study the effects of accelerating conditions on the drag- and lift coefficient. In addition, in view of experimental purposes, it is attempted to calculate the total force acting on the hand from the surrounding flow using an alternative form of conservation of momentum, where the pressure terms are eliminated.

After this introduction some theoretical concepts relevant for this study will be reviewed in Chapter 1. Governing equations, some basic theory about propulsion and drag in swimming and the flow past a sphere and cylinder will be cited. In chapter 2 the most important numerical methodologies regarding this immersed boundary code are explained, followed by the general approach and numerical set-up in chapter 3. Chapter 4 is concerned with the results and discussions of this study. In the end, this study is concluded and summarized in chapter 4.4, where also a brief outlook to possible future studies is made.

Chapter 1

Theory

This chapter presents some theoretical concepts that are relevant for the present study. In the first section some of the governing equations derived from fluid dynamics will be presented, starting from the Navier-Stokes equations and ending with an alternative way to formulate conservation of momentum. Some basic theory about propulsion and resistance in swimming will be reviewed in the second section, and reference is made to some previous research in swimming. In the subsequent sections the flow past a cylinder and sphere will be discussed, including the concepts of added mass.

1.1 Governing equations

Within the numerical scheme of the simulation, the incompressible Navier-Stokes equations and continuity equation will be solved. Assuming that the density ρ is constant as in the case of an incompressible flow, the (mass) continuity equation reads:

$$\nabla \cdot \mathbf{v} = 0, \tag{1.1}$$

where $\mathbf{v}$ is the velocity vector field. This equations expresses that the divergence of the velocity field is zero everywhere, implying that the local volume dilation rate is zero. Conservation of momentum states:

$$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla)\mathbf{v} = \mathbf{g} + \frac{1}{\rho}\nabla \cdot \bar{\bar{\sigma}}, \tag{1.2}$$

where $\mathbf{g}$ the gravitational acceleration. For a Newtonian flow $\bar{\bar{\sigma}} = -p\mathbb{I} + \bar{\bar{\tau}}$, where $\mathbb{I}$ is the identity matrix. The shear stress tensor $\bar{\bar{\tau}} = 2\mu\bar{\bar{D}}$, with D the strain rate tensor and μ the dynamic viscosity. Combining these relation with eq.(1.2) results in the Navier-Stokes equation:

$$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla)\mathbf{v} = -\frac{1}{\rho}\nabla p + \mathbf{g} + \nu\nabla^2\mathbf{v}, \tag{1.3}$$

where p is the pressure and ν the kinematic viscosity. The physical meaning of each term (from left to right) is as follows: unsteady acceleration, convective acceleration, pressure gradient, gravitational forces and viscous forces [10].

By adopting a typical length scale (L), a typical velocity (U) and a convective time scale (T) the Navier-Stokes equation can be written in non-dimensional form:

$$Sr\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v}\cdot\nabla)\mathbf{v} = -\nabla p + \frac{1}{Fr}\mathbf{g} + \frac{1}{Re}\nabla^2\mathbf{v}, \tag{1.4}$$

where Re is the Reynolds number, Sr is the Strouhal number, Fr is the Froude number (not taken into account in this study) and all other variables have to be considered dimensionless now. The Reynolds number is defined as

$$Re = \rho\frac{UL}{\mu}. \tag{1.5}$$

For large values of the Reynolds numbers the inertia forces are dominating the flow, for small Re values the viscous forces dominate.

In case of a sphere the typical length scale will be the diameter of the sphere. Due to the complexity of the shape of the human body and moving body parts with different relative speeds, it is less trivial to assign a certain length scale to a swimmer. Or as will be studied in this project, the hand of a swimmer. In this study the typical length scale of the hand and forearm is chosen to be the width of the frontal surface. That is approximately the distance from thumb side to little finger side. Since the density, viscosity and length scale in the picture of a swimmer in water (and simulations) are constant, the Reynolds number in this study is just measure of the typical velocity.
The Strouhal number defined as

$$Sr = \frac{fL}{U} = \frac{L}{TU}, \tag{1.6}$$

where f is a frequency. When Strouhal is of the order 1, viscosity is dominating the flow and results in a collective oscillating movement of the fluid. At Strouhal numbers in between, the oscillation is characterized by a buildup of vortices and rapidly followed by vortex shedding. Generally the Strouhal number is relevant in problems of oscillating fluid mechanics, where f is a measure of the vortex shedding. In studies of a human swimmer, the Strouhal number is often used as a measure of the stroke frequency as well.

Taking the divergence of the Navier-Stokes equations while considering the continuity equation, will finally result in the following Poisson equation for the pressure:

$$\nabla^2 p = \nabla\cdot(\frac{1}{Re}\nabla^2\mathbf{v} - \mathbf{v}\cdot\nabla\mathbf{v}) \tag{1.7}$$

This equation presents an explicit form of the pressure, instead of an implicit notation like in the Navier-Stokes equation. This equation is again written in a non-dimensionalized notation.

In integral formulation the momentum conservation equation is given by:

$$\frac{\partial}{\partial t}\iiint_V \rho \mathbf{v} dV + \iint_S \rho \mathbf{v}(\mathbf{v}\cdot\mathbf{n})dS = -\iint_S p\mathbf{n}dS + \iint_S \bar{\bar{\tau}}\cdot\mathbf{n}dS + \mathbf{F}, \tag{1.8}$$

where $\mathbf{n}$ is the normal vector of a control volume V enclosed by surface S and $\mathbf{F}$ represents additional forces exerted by a body in the domain. [11] [10] With this integral momentum equation it is possible to calculate the total forces on a body within the control volume V by using the velocity and pressure data of the surrounding fluid. Following eq.(1.8) it is tried to evaluate a method to calculate the forces on the hand of a swimmer. In the numerical scheme of the simulations the forces on a body are calculated in a different way using the stress and pressure data at the surface of the body. The total pressure force F_p on a surface S bounded by a closed curve is determined by:

$$\mathbf{F_p} = -\iint_S \Delta p\mathbf{n}dS. \tag{1.9}$$

The total viscous force F_v on the surface of the immersed body is given by:

$$\mathbf{F_v} = \iint_S \bar{\bar{\tau}}\cdot\mathbf{n}dS, \tag{1.10}$$

A sum of these two forces gives the total force experienced by the immersed body.

Taking into consideration experimental studies, it is impossible, unlike simulations, to obtain a complete data set of the pressure everywhere in the flow domain and on the surface of the immersed body. It is not convenient to calculate the forces on the immersed body out of the surrounding fluid using a conventional method of momentum conservation with the pressure term present. Graziani et al. [12] developed a method to determine the forces considering the surrounding flow using momentum conservation without using a pressure term. Below most important steps in the derivation will be discussed.

Assume that the flow is irrotational, parallel far from the body and the body surface is impermeable. The force exerted on the body is obtained from the incompressible Navier-Stokes equations:

$$\mathbf{F} = -\frac{d}{dt}\int_{V(t)} \mathbf{v}dV - \int_{S(t)} \mathbf{n}\cdot(\mathbf{v}-\mathbf{U})\mathbf{v}dS + \int_{S(t)} (-p\mathbf{n} + 2\mu\mathbf{n}\cdot\mathbb{E})dS, \tag{1.11}$$

where $V(t)$ is an arbitrary, time-dependent, body-fixed control volume, bounded externally by a smooth compact connected surface $S(t)$ (internally by the surface $\partial B(t)$),

n is the normal vector oriented outwards on $S(t)$, $U(t)$ the body rigid motion and $\mathbb{E} := \frac{1}{2}(\nabla \mathbf{v} + \nabla \mathbf{v}^T)$ is a symmetric tensor with components E_{ij}. Roughly speaking, the pressure is eliminated and the expression for the force is written in terms of vorticity moments using several vector relations. To eliminate the pressure in the expression for **F**, first the momentum of the fluid within $V(t)$ is written in terms of the vorticity. Finally the pressure cancels out by the definition of the Bernoulli function $h := p + \frac{1}{2}|\mathbf{v}|^2$ and the following expression for the force remains:

$$\mathbf{F} = -\frac{1}{N-1}\frac{d}{dt}\left[\int_{V(t)} \mathbf{x} \times \boldsymbol{\omega} dV + \int_{\partial B(t)} \mathbf{x} \times (\mathbf{n} \times \mathbf{v}) dS\right] + \int_{S(t)} \mathbf{n} \cdot \mathbb{G} dS, \qquad (1.12)$$

where **x** is the position vector and $\mathbb{G}$ is a tensor defined as

$$\begin{aligned} \mathbb{G} &:= 2\mu\mathbb{E} - \mathbf{v}\mathbf{v} + \frac{1}{2}|\mathbf{v}|^2 \mathbb{I} \\ &+ \frac{1}{N-1}\left[\mathbf{x} \times \mathbf{v}\boldsymbol{\omega} - (\mathbf{v} - \mathbf{U})\mathbf{x} \times \boldsymbol{\omega} - \mu(\mathbf{x} \cdot \nabla \times \boldsymbol{\omega}\mathbb{I} - \mathbf{x}\nabla \times \boldsymbol{\omega})\right]. \end{aligned}$$

If there is no $\times$ or $\cdot$ present, a dyadic vector product is considered [12].

1.2 Propulsion and drag in swimming

The propulsion in swimming is obtained from the arms (hand and forearm) and the legs, to which extent depends on the kind of crawl. A combination of curvilinear movements

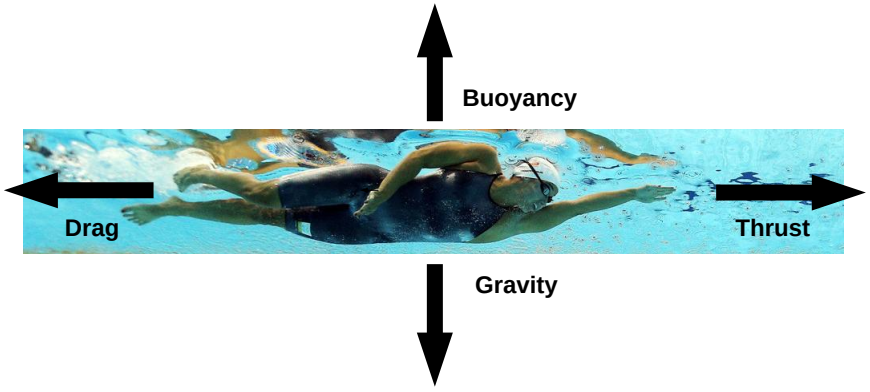


Figure 1.1: The force balance on a swimmer. Gravity and buoyancy work in the vertical plane, drag and thrust work in the horizontal plane.

(up-down, left-right and backward) causes the forward motion of the body. In figure 1.1

the force balance on a swimmer is shown. In the vertical plane gravity and buoyancy work in downward and upward direction, respectively. The weight of the swimmer is offset by the buoyancy. Since the natural ability to float differs per person, possibly an additional force is needed to overcome the net weight. This is accomplished by the arm strokes and kicks, by pressing down on the water. Generally speaking the drag and thrust of a swimmer work in opposite direction in the horizontal plane. When the thrust of the swimmer is greater than the drag, the swimmer will accelerate. When the drag is larger than propulsive forces the swimmer will decelerate. When the thrust and drag are equal, the swimmer will move with a constant velocity. The term drag might be a bit confusing. In the discussion above, drag referred to be resistance induced by the ambient water (and air), the force that is decelerating. But the term drag may also be used in relation to propulsion: drag is then needed to propel.

1.2.1 Drag

Drag force is always exerted in a direction opposite to the direction of motion. The drag on a body can be divided into three major components, the pressure drag (form drag), viscous drag (skin friction) and wave resistance (just near the air-water interface). Minor drag effects can occur from lift-induced drag. However, the lift-induced drag is so small compared to the lift force generated, that no further explanation and reference will be made. The general expression in hydrodynamics for the drag force is given by:

$$F_d = \frac{1}{2}\rho U^2 C_d A, \tag{1.13}$$

where ρ is the fluid density, U is the relative velocity with respect to the fluid and A the frontal surface. C_d is the drag coefficient which depends on the body shape (and on the Reynolds number). The drag coefficient can roughly be divided into a part derived from the pressure drag $C_{d,p}$ and a part originating from the viscous drag $C_{d,v}$, hence

$$C_d = C_{d,p} + C_{d,v}. \tag{1.14}$$

Figure 1.2 shows the ratio between pressure and viscous drag on several basic objects in a uniform flow. It should be mentioned that these flow configurations depend on the Reynolds number. If $Re \ll 1$ all forces are viscous forces. The pressure drag arises because of the form of the body. The larger the frontal surface pushing against the water (in the direction opposite of forward motion) and the larger the flow separation[1] that occurs behind the body, the higher the pressure drag will be. [13] The pressure drag follows the drag equation, meaning that it increases with the square of speed, and

[1] The flow becomes detached from the surface of the body, because the boundary layer travels far enough against an adverse pressure gradient that the speed of the boundary layer relative to the object falls almost to zero. The separated flow is characterized by eddies and vortices.

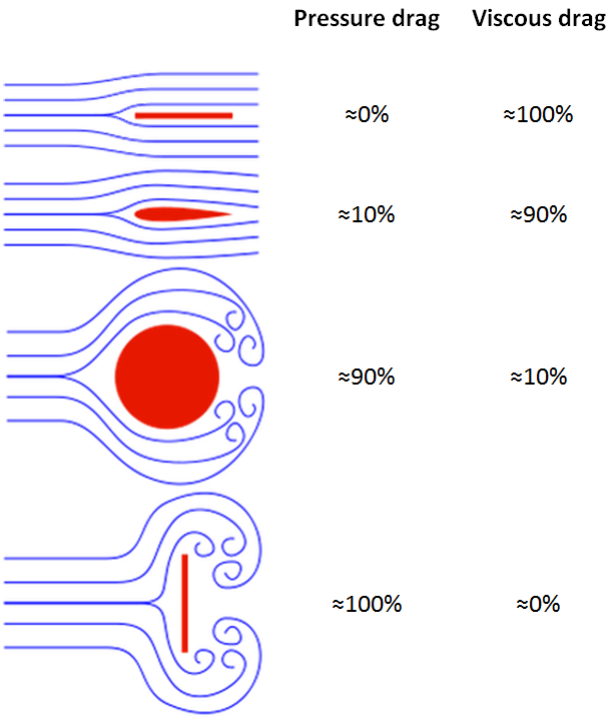


Figure 1.2: Percentages of pressure drag and viscous drag on several objects exposed to an uniform flow. Actually these flow configurations depend on the Reynolds number.

thus becomes more important for higher velocities. The viscous drag arises from the interaction of the flow (in the boundary layer) and the ”skin” of the body. The viscous drag is directly related to the surface of the body that is in contact with the flow and also increases with the velocity squared. Wave resistance just occurs on bodies moving in the interface between two fluids with different densities and increases with increasing velocities. Since the hand in the simulations in this study is totally embedded in the fluid, the wave resistance must not be taken into account. However, in real life a swimmer will undergo a fair amount of wave resistance. In order to reduce the drag, a swimmer must streamline his body to reduce the amount of separation. As a result the viscous drag increases, because more surface area is exposed to shear stresses of the flow. But since the pressure drag is still dominant, the overall drag decreases. To reduce the viscous drag[2] (skin friction) as well, special suits were developed. Manufacturers of these suits claim that almost all surface friction is reduced using specially treated fabrics and seams. However, most of such high-tech suits are not allowed during competitions [14] [15].

[2]Generally the flow around a swimmer is already more or less turbulent. Turbulence increases the surface friction.

1.2.2 Propulsion

The major principles responsible for propulsion in swimming are the lift force and Newton's third law that states that for every action there is an equal opposite reaction. The most obvious production of thrust comes from pushing back on the water like paddling (Newton's third law). Although drag can be the greatest downfall for a swimmer, it also is the greatest ally. Without drag the action-reaction principle of propulsion would be impossible.

Swimmers also generate a lift type of force, which is always exerted perpendicular to the direction of motion. The lift principle is based on the law of Bernoulli, which states that for an inviscid flow an increase in the velocity of a fluid occurs simultaneously with a decrease in the pressure or a decrease in the fluid's potential energy. A common form of the Bernoulli equation, valid at any point along a streamline is:

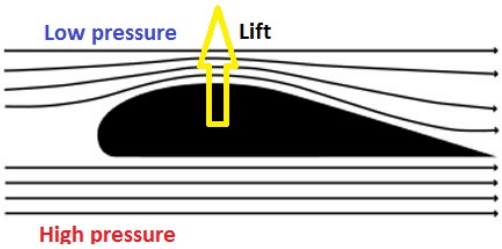


Figure 1.3: The difference in pressure between the low-pressure and high-pressure streams creates a lift force.

$$\frac{1}{2}\rho v^2 + \rho gh + p = C, \tag{1.15}$$

where ρ is the density of the fluid, v the velocity, g the acceleration due to gravity, h the elevation of the point above a reference plane, p the pressure and C is constant along one streamline. For example, when a stream of fluid passes around an airfoil/wing-shaped body (fig. 1.3), the flow over the convex upper surface has a greater velocity and, following Bernoulli, a lower pressure than the surface underneath. The difference in pressure between the two streams creates a force called lift, perpendicular to the direction of motion. That is the reason why planes can fly through the air. Like the drag force, the lift force is proportional to the velocity squared. The general expression for the lift force is similar to the expression of the drag force:

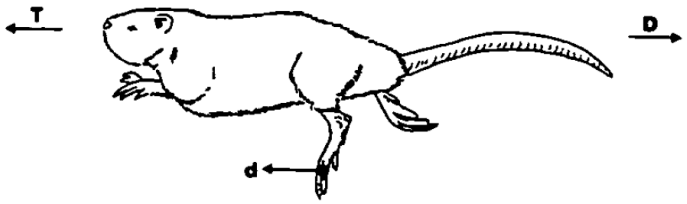
$$F_l = \frac{1}{2}\rho U^2 C_l A_l, \tag{1.16}$$

where C_l is the lift coefficient, which depends on the shape of the body, and A_l is the frontal surface. [4] [2]

That lift forces could be important in swimming propulsion is shown by developed aquatic mammals like the dolphin (fig. 1.4(a)). Dolphins move their caudal fin up and down, causing thrust in forward direction. Their caudal fin works as some kind of oscillating hydrofoil, and lift is continuously produced during their strokes. Lift is the

dominant force and a combination of drag and lift provides constantly a very effective propulsion throughout the stroke. However, a less developed aquatic animal, like the muskrat (fig. 1.4(b)), has drag-based propulsion. The legs are paddling through the water. The stroke is divided into two phases, a power phase and recovery phase. Drag-based propulsion is just generated in the power phase. There are more situations to think of, where lift-based propulsion is more effective than drag-based propulsion, like sailing. [16]

(a)



(b)

Figure 1.4: (a) Lift-based propulsion by a dolphin. (b) Drag-based propulsion from a muskrat. [16]

How to apply this to human swimming? The dolphin kick (fig. 1.5) in human swimming turns out to be a very effective way to propel through the water, for the same reasons it works for dolphins. Lift is dominating and the combination of drag and lift force results in a resultant propulsive force. The main question is whether the propulsion by the arm stroke in competitive swimming could be optimized by using lift-based propulsion as well. To do so the hand probably should not always be pulled directly backward in a straight line. Maybe a curvilinear path of the arm stroke could produce a fair amount of forward lift at times. Or maybe the angle of attack of the hand relative to the forward motion of the body should be constantly adjusted to achieve maximum propulsion. [17] [15]

New views on swimming suggest that vortices shed in the wake of a swimmer can also be used for extra effective propulsion. This idea is again copied from locomotion

Figure 1.5: Drag and lift forces working on the legs during a kick at some stage.

observed from fishes. Vortices generated around the head, hand, arm, feet and legs of a swimmer create extra resistance in the water. A shed vortex is a form of kinetic energy that occurs during an increase in velocity or a dramatic change in direction. Actually this is a loss of energy for the system so that translational velocity is reduced, especially when the vortices are shed out of phase compared to the stroke of the swimmer. The vortices created in the wake of a swimmer carry a fairly high momentum, which can transfer a strong propulsive impulse for the body. It should be a strategy of the swimmer to reuse the energy of these vortices through so-called 'vortex-recapturing'. A principle some animals use in flying or swimming. A swimmer can possibly use the rotating energy of the vortices created near flexing joints to enhance propulsion [14] [18] [13] [19] [20].

1.3 Flow past a cylinder

Analytical solutions for the flow past a cylinder can be found for viscous flows ($Re \ll 1$), when inertia forces are negligible. This situation does not apply to the present study, however. Many experimental and numerical studies have been performed about the flow past a cylinder, providing clear results of the drag coefficient (fig. 1.6) and the wake configurations (fig. 1.7). At low Reynolds number ($Re < 1$) the drag is mainly due to friction, with increasing Re the contribution of inertia forces is growing. At high Reynolds number the skin friction is just a few percent of the total drag, as was shown in figure 1.2. At the critical Reynolds number $Re \approx 3 \times 10^5$ the drag coefficient decreases sharply. First the laminar boundary layer separates at the front side of the cylinder. At the critical Reynolds number the boundary layer becomes turbulent, this is associated with a delayed separation.
Different flow configurations in the wake can be distinguished, depending on the Reynolds number:

i ($5 < Re$), figure 1.7(a)

No separation occurs.

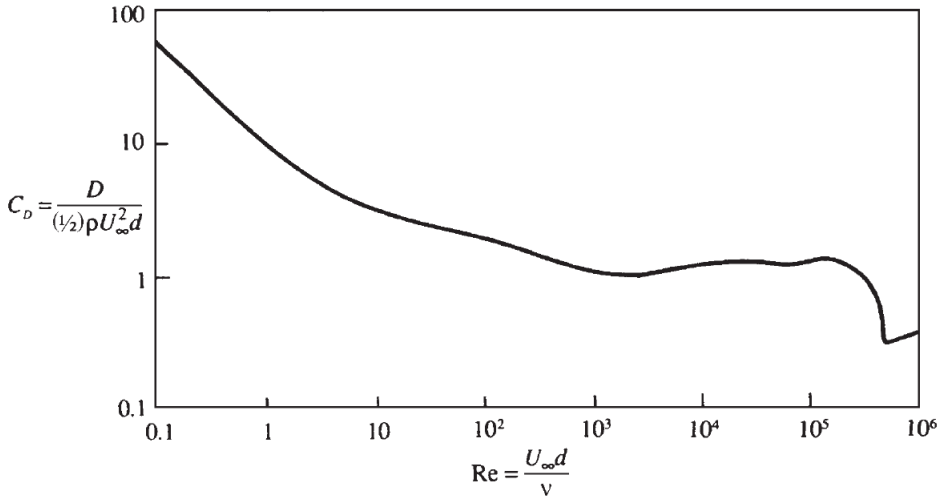


Figure 1.6: The drag coefficient of a cylinder [10].

ii $(5 - 15 < Re < 40)$, figure 1.7(b)

The flow separates and a symmetric pair of vortices is formed in the wake of the cylinder. The size of the vortices increases linearly with increasing Reynolds number.

iii $(40 < Re < 150)$, figure 1.7(c)

The wake becomes unstable. First slow oscillations are visible in the wake, increasing the Reynolds number even more and vortex shedding will be initiated. The shed vortices form a laminar periodic wake of staggered vortices of opposite sign, also called the von Karman vortex street.

iv $(150 < Re < 3 \times 10^5)$, figure 1.7(d)

First periodic irregular disturbances are found, the vortex street becomes gradually turbulent. The boundary layer is still laminar.

v $(3 \times 10^5 < Re < 3.5 \times 10^6)$, figure 1.7(e)

The boundary layer becomes turbulent and the separation point is shifted to the rear side of the cylinder again. Three-dimensional effects disrupt the shedding process, the wake is disorganized.

vi $(3.5 \times 10^6 < Re)$, figure 1.7(f)

This is called the supercritical regime, a regular vortex street is reestablished with a turbulent boundary layer at the surface.

[10] [21]

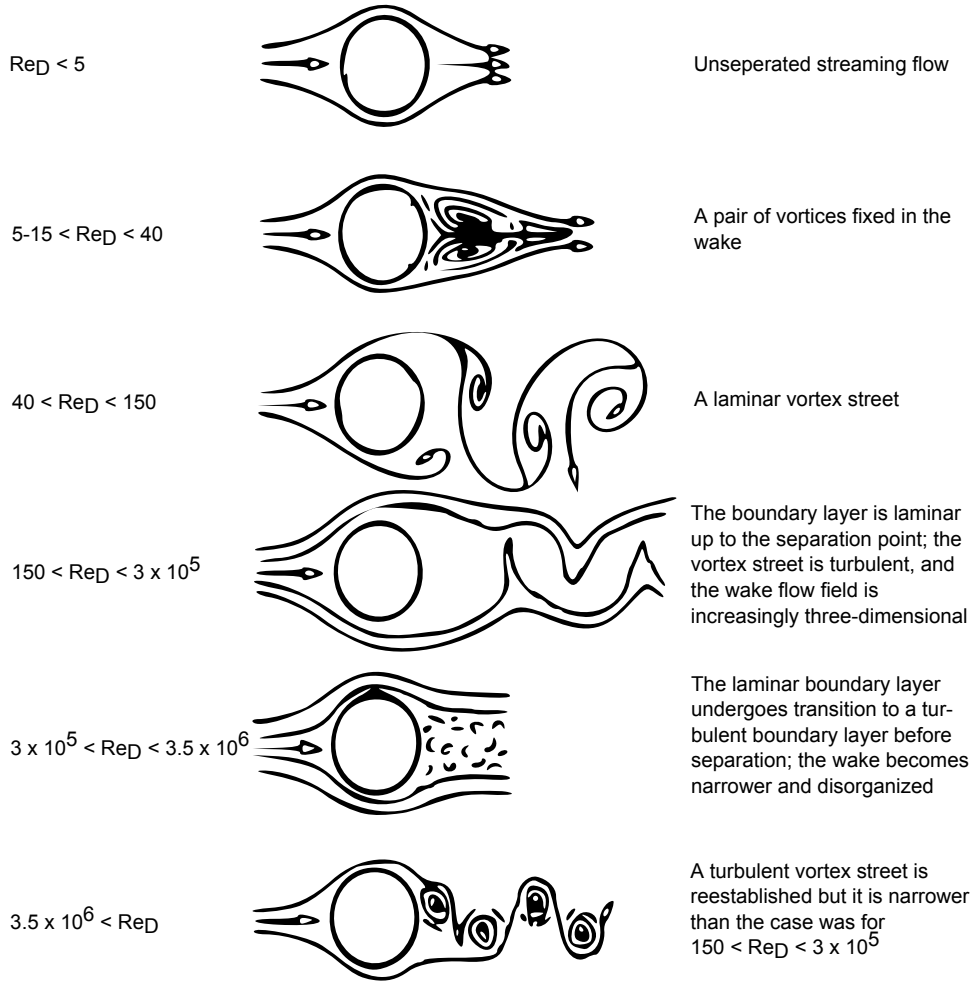


Figure 1.7: Different wake configurations of the flow past a cylinder, distinguished by Reynolds number [22].

1.4 Flow past a sphere

The flow around a (3D) cylinder can be considered as a two-dimensional flow if the cylinder is long enough. The flow around a sphere can not be considered two-dimensional, although certain cross sections look similar to the flow around a cylinder. The flow past a three-dimensional object (like a sphere) shows some fundamental differences. For example no stretching and tilting of vortex tubes occurs in 2D, but they do in 3D and take along some additional effects. At low Reynolds numbers no vortex pair is visible, but instead there is a doughnut-shaped vortex ring. At higher Reynolds numbers no oscillating vortex street is visible, but the ring-eddy oscillates and possibly some of it breaks off periodically in the form of distorted vortex loops.

The drag coefficient of a sphere is well documented. Figure 1.8 shows the drag coefficient as a function of the Reynolds number. The boundary layer around a sphere undergoes a transition to turbulence at a critical Reynolds number of $Re \sim 3 \times 10^5$, which corresponds to the dip of the drag coefficient shown in the graph of figure 1.8. This dip arises due to similar reasons as for the cylinder. Due to the turbulence the boundary layer remains attached to the sphere farther on the downstream side, therefore the wake remains smaller and drag decreases. In fact the turbulence in the boundary layer increases the surface friction as a result of the larger velocity gradient at the surface of the body. The drag rises again for post-critical Reynolds numbers towards the old level, since the separation point of the boundary layer is slowly moving upstream [10].

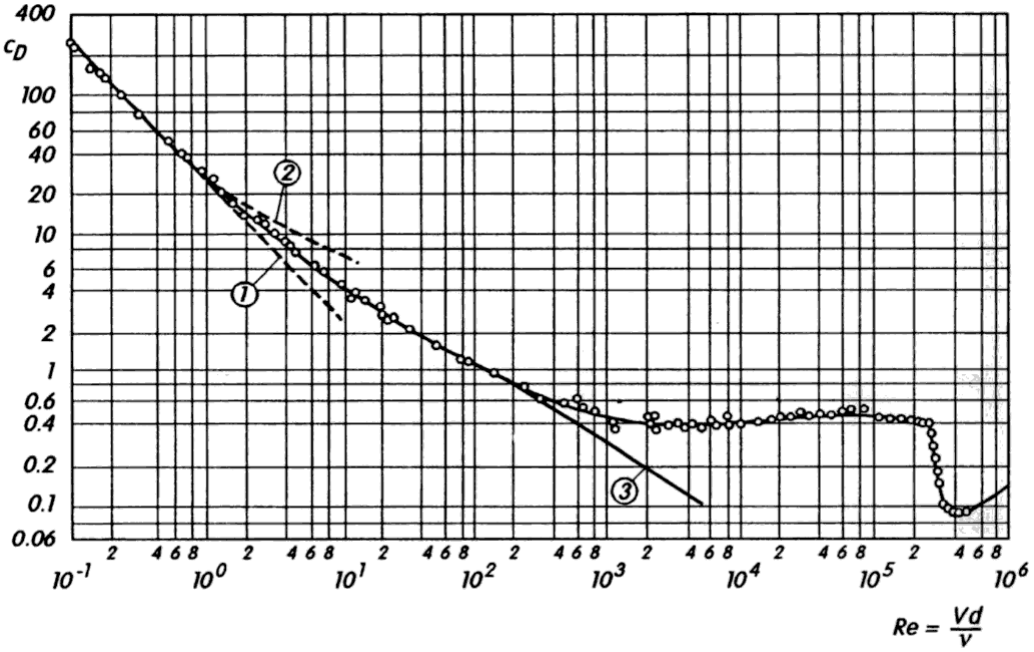


Figure 1.8: The measured drag coefficient of a smooth sphere as a function of the Reynolds number. Curve 1 is the Stokes (1856) solution, $C_D = 24/Re$. Curve 2 is the Oseen (1911) solution, $C_D = (24/Re)(1 + 3Re/16)$. Curve 3 shows the numerical results after B. Fornberg (1988) [23].

Following roughly the results of Sakamoto et al. [24] and other previous studies, a rough distinction can be made regarding the relationship between the Reynolds number and corresponding wake configurations. For $Re < 130$ no separation occurs, the flow is considered to be in the Stokes regime.

i ($130 < Re < 300$), figure 1.9(a)

Faint periodic pulsative motion at the rear of the vortex-ring formed behind the sphere could occur with a very long period.

ii ($300 < Re < 420$), figure 1.9(b)

The wave-like wake turns into a hairpin-shaped vortex.

iii ($420 < Re < 800$), figure 1.9(c)

The waveform of the fluctuating velocity based on the shedding of hairpin-shaped vortices becomes irregular.

iv ($800 < Re < 6 \times 10^3$), figure 1.9(d)

The vortex tubes formed by the vortex sheet separating from the sphere surface flow into the vortex formation region at its rear end, while others are shed in small vortex loops.

v ($6 \times 10^3 < Re < 3.7 \times 10^5$)

The vortex sheet separating from the surface of the sphere becomes completely turbulent.

1.5 Added mass

A body and its surrounding fluid can never occupy the same physical space simultaneously. When a body moves through a fluid, some volume of surrounding fluid must move around the body. Fluid in front of a body must move rapidly to the side. Behind the body fluid must fill in the released space. Therefore a body moving through a fluid has a larger effective inertia than one moving through vacuum. Most studies about forces on a body moving through a fluid treat motion at constant velocity. When the velocity of a body is not constant, additional forces will be present. The additional inertia added to the system of an accelerating (or decelerating) body moving through a fluid, causing the surrounding fluid to accelerate, is the so called "added" mass of the body.

The added mass of an object can be derived by considering the hydrodynamic force acting on it as it accelerates. The additional force on a sphere can be calculated exactly, assuming a potential flow, and is given by

$$-F_a = \frac{2}{3}\rho\pi R^3 a, \tag{1.17}$$

where ρ is the density of the fluid, R the radius of the sphere, a the acceleration and the negative sign indicates that the force is in opposite direction of the acceleration of the sphere [25]. Since $F = m\frac{dv}{dt}$, the added mass of the sphere will be

$$m_a = \frac{2}{3}\rho\pi R^3. \tag{1.18}$$

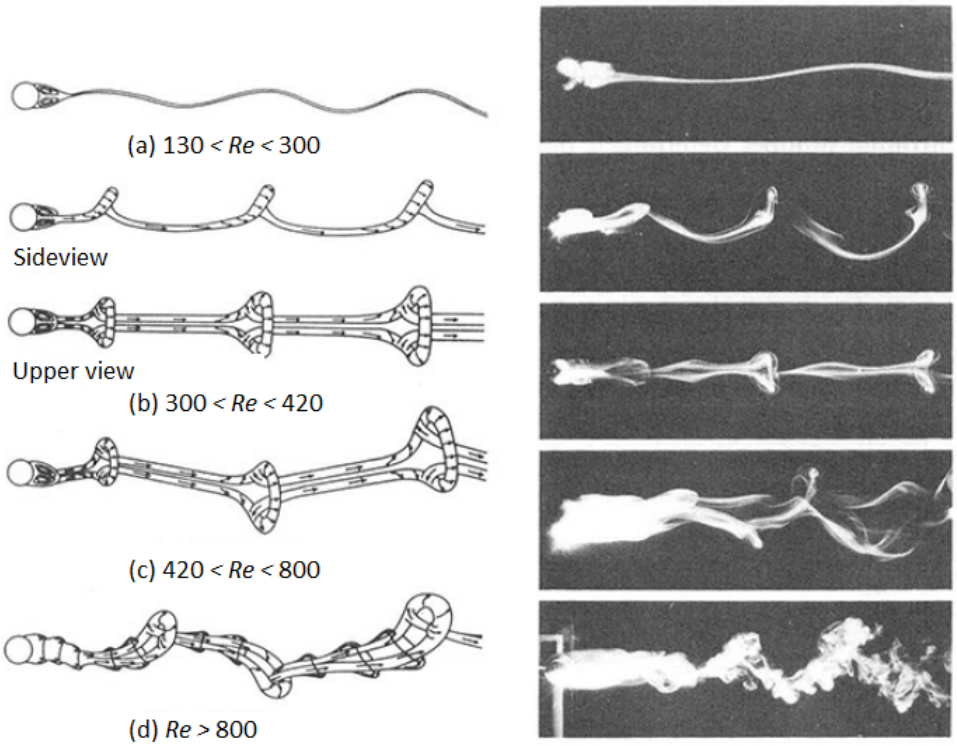


Figure 1.9: Patterns of a wake behind a sphere in several Reynolds number ranges. [24]

Apparently a submerged sphere will behave as if its inertia is larger by one-half of the mass of the fluid it displaces [26].

Also in non-stationary (human) swimming, some of the water around the swimmer is set in motion, which can be thought of as an added mass of water. Previous research on added mass in human swimmers was performed by Caspersen et al. [27]. In that study it was concluded that the added mass in human swimmers, in an extended gliding position, is approximately $1/4$ of the fluid mass displaced by the body. Furthermore, there were minor differences, but significant, in the added mass and relative added mass for women compared to men. This indicates that the possible body shape differences may be an important factor in determining the added mass.

Chapter 2

Numerical Method

In this study simulations of a three-dimensional flow in Cartesian coordinates are performed by a numerical method called Immersed Boundary, IB in short. This chapter is concerned with the numerical techniques involved in this particular immersed boundary code. First the methodologies used will be shortly denoted, then the most important ones will be highlighted in subsequent sections. This IBM code has been developed over the years by Prof. R. Verzicco from the Università di Rome "Tor Vergate", Dr. M. D. de Tullio from the Politecnico de Bari and others.

2.1 Methodologies

The IB technique is based on a Moving-Least-Square formulation (MLS). The MLS formulation is used since for moving bodies it results in smoother forces on the body than the classical direct-forcing method. The Navier-Stokes equations for an incompressible flow are solved by a fractional step method with the pressure in the first step following the ideas of Kim and Moin. [28] The Poisson equation (eq.(1.7)) for the pressure is solved directly by introducing a Fast Fourier Transform (FFT). The IB code has two different solvers, a DNS solver (Direct Numerical Simulation) and a LES solver (Large Eddy Simulation). The LES model in the numerical scheme was based on the Smagorinsky method The time advancement of the solution is obtained by a second-order Adams-Bashforth scheme.The nonlinear terms in this code are computed explicitly while the viscous terms are computed implicitly. The stability limit for the time integration is given by the CFL (Courant-Friedrichs-Lewy) condition. The calculation stops if the velocities are diverging for numerical stability conditions (Courant number restrictions). In this particular study the Courant number restriction was chosen to be $C = U\Delta t/h > 0.25$. However, when a Large Eddy Simulation (LES) is carried out, the additional LES viscous terms are computed explicitly. This introduces an additional stability constraint. This new constraint is generally less restrictive than the CFL con-

dition. All variables are calculated in a staggered grid with the velocities on the faces of the computational cell and all the scalars at the centre. This is important when terms belonging to different equations are evaluated.

2.2 Immersed boundary method

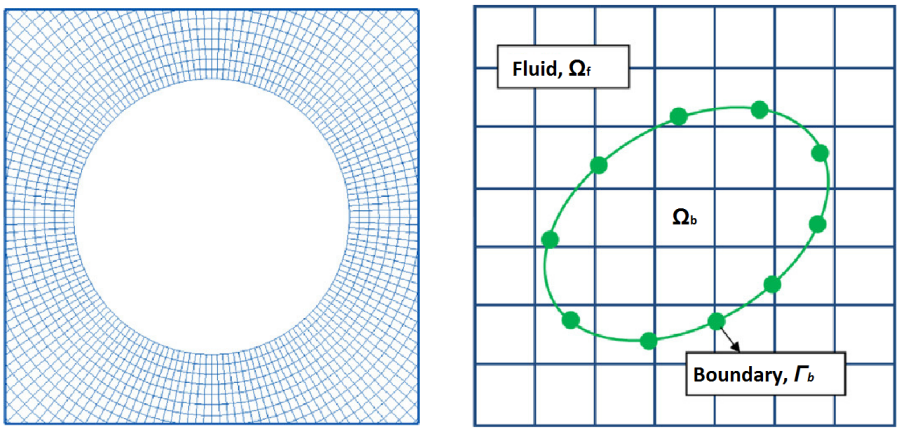
The IB method is a numerical scheme, which involves both Eulerian and Lagrangian variables. The IB method is a methodology for dealing with complex (moving) body shapes and boundary conditions, which do not necessarily have to conform a Cartesian grid. Despite the complexity of a body the equations of motion are usually solved on a fixed structured grid. The IB code used in the present study is based on a direct-forcing scheme that utilizes a versatile moving-least-square (MLS) approximation to build the transfer functions between the Eulerian and Lagrangian grids, and can be applied to arbitrary moving/deforming bodies. Treating the coupling of the structure deformations and the fluid flow poses also a number of challenging problems for numerical simulations.

Before reviewing the numerical scheme, first a comparison is made between IB and conventional methods, based on the study by Mittal et al. [29].

2.2.1 IB method vs conventional methods

The conventional approach to consider the simulation of a flow past a solid body is to employ structured and unstructured grids that conform to the body (fig. 2.1(a)). First a surface grid covering the boundaries of the body (Γ_b) is generated, which is then used as a boundary condition to generate a grid in the volume (Ω_f) occupied by the fluid. Within the IB method a non-body conformal Cartesian grid is used for simulations of the flow (fig. 2.1(b)). The immersed body is still described by a surface grid, but the Cartesian volume grid is generated with no regards to the surface grid. The boundary of the immersed body will cut through the Cartesian volume grid. Since the grid does not conform the surface grid of the body, to implement the boundary conditions it is required to modify the governing equations in the vicinity of the boundary. Imposition of the boundary conditions on the immersed boundary is the key factor in developing an IB algorithm.

Within conventional methods a discretized version of the Navier-Stokes equation is solved on a body-conformal grid where the boundary condition on the immersed boundary is enforced directly. In an IB method, the Navier-Stokes equation is discretized on a nonbody conformal Cartesian grid and the boundary conditions are imposed indirectly through a forcing-function in the governing equations that reproduces the effect of a boundary.



(a) (b)

Figure 2.1: (a) Sketch of a body-conformal grid. (b) Sketch of a grid used in an immersed boundary method. [30]

The advantage of the IB method compared to conventional methods is that grid generation is simplified. Grid complexity and quality are not significantly affected by the complexity of the geometry for the simulation carried out on a non-body conformal Cartesian grid. In conventional methods it is usually an expensive task to generate the body-conformal structured or unstructured grids. The requirements are to construct a grid that provides an adequate local resolution with the minimum number of total grid points. These conflicting requirements can lead to a poor grid quality and have negative impact on the accuracy and convergence properties of the solver. Especially with more complex geometries an acceptable grid becomes increasingly difficult. Furthermore, simulations for flows with moving boundaries can be handled more easily with IB methods including body motion. For a body-conformal grid it is required to generate a complete new grid at each time-step, with again negative impact on the accuracy, costs and simplicity. Since IB methods use a stationary, non-deforming Cartesian grid it is relatively simple to include moving boundaries.

A major disadvantage of the IB method is that imposing the boundary conditions is not straightforward. The accuracy and conservation properties of the numerical scheme are not clear because of the treatment of the boundary in IB. Moreover, body-conformal grids give better control of the grid resolution in surroundings of the body surface. This could be useful with increasing Reynolds numbers and solving the boundary layer. [29]

2.3 Direct-forcing scheme and MLS approximation

The MLS formulation for the immersed boundary technique used in this particular code is similar to the technique used by Vanella and Balaras. This technical description is

derived from their paper. [31] This paper also includes some accuracy studies of this method.

For the sake of clarity, the body is immersed in an Eulerian grid, but is itself described by a Lagrangian grid. The current and next time step in this description are denoted by n and $n+1$, respectively, the three vector components (directions) are denoted with i. The uppercase symbols denote variables at Lagrangian points on the immersed body, normal symbols denote the same variables but then on Eulerian points.

In this code the direct-forcing function is computed on the Lagrangian markers instead of Eulerian grid nodes as in other IB codes. This results in smoother hydrodynamic forces. The direct-forcing function is computed on each Lagrangian marker by a rewritten form of the time-discretized momentum equation:

$$F_i^{n+1/2} = \frac{U_i^{n+1} - U_i^n}{\Delta t} - RHS^{n+1/2}, \tag{2.1}$$

where F is the direct-forcing function which is different form zero only at the grid points in the vicinity of the immersed body, U^n and U^{n+1} are discrete approximations of the velocity field, Δt the time step and RHS contains all advective and diffuse terms. For every point where $F_i \neq 0$ the desired velocity U^b, to be considered as a boundary condition, can be used instead of U^{n+1}:

$$F_i^{n+1/2} = \frac{U_i^b - U_i^n}{\Delta t} - RHS^{n+1/2}. \tag{2.2}$$

Eq. (2.2) is actually the essence of IB. This force is not a force with any physical meaning, but is used to impose the proper boundary conditions. Replacing $U^n + \Delta t RHS^{n+1/2}$ by $\tilde{U}$, the Lagrangian counterpart of the predicted velocity, yields:

$$F_i^{n+1/2} = \frac{U_i^b - \tilde{U}_i}{\Delta t}, \tag{2.3}$$

see [32] [31]. The predicited velocity on Eulerian grid nodes is computed as follows:

$$\tilde{u}_i = u_i^n + \frac{\Delta t}{2}(3H(u_i^n) - H(u_i^{n-1})) - \Delta t \frac{\partial p^n}{\partial x_i}, \tag{2.4}$$

where H is a discrete operator representing the spatially discretized convective and viscous terms, p the pressure term and x the position vector. Note that the density is supposed to be 1 and that eq. (2.4) is actually a rewritten form of the discrete Navier-Stokes equation. The predicted velocities do not satisfy the incompressibility constraint and the boundary conditions but this is corrected during the computations. A direct-forcing function will enforce these proper boundary conditions on all the Eulerian grid nodes influenced by the immersed body [31].

2.3.1 MLS reconstruction

As mentioned in the previous section, the direct-forcing function is computed on the Lagrangian markers. This force must then be transferred to the Eulerian grid nodes. The transfer operators are constructed by using MLS shape functions. For clarity, fig.

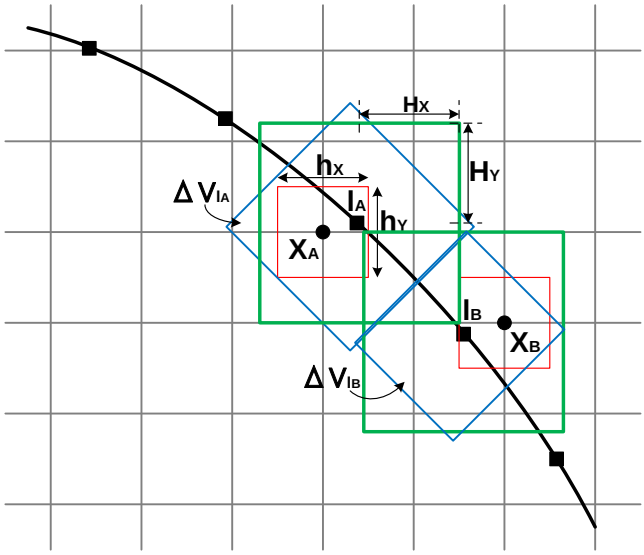


Figure 2.2: Definition of the associated volumes ΔV_{l_A} and ΔV_{l_A} (blue) for the markers l_A and l_B respectively. The marker l_A and l_B are associated to the closest grid nodes X_A and X_B respectively, which is in the centre of a cell with dimensions h_x and h_y (red). The support domain (green) is a rectangular box of size $2H_x \times 2H_y \times 2H_z$ centred at the location of the marker.

2.2 gives a schematic representation of the following description about how the transfer operators are created.

(i) First the closest Eulerian grid node for each Lagrangian marker must be identified. Thus Lagrangian marker l_A is associated to the Eulerian grid node x_A, which is in the centre of a cell with dimensions h_x and h_y. Even so the marker l_B is associated to grid node x_B. It is possible and allowed that more than one Lagrangian marker from the same or different immersed bodies is associated with the same Eulerian grid node.

(ii) Then a support-domain, in which the shape functions are constructed, is defined around the Lagrangian markers centred at the location of the markers. The support-domain is a rectangular box of size $2H_x \times 2H_y \times 2H_z$. The lengths H_x, H_y and H_z are dependent on the Eulerian grid and can be different for each marker.

(iii) Thereafter a volume $\Delta V^l = A^l h^l$ is associated to each marker point. Where A^l is the area of the body surface associated to marker l and h^l the local thickness that

depends on the local grid size. The volumes do not overlap and the sum of all local A^l is equal to the total area of the immersed body surface.

With this in mind, the transfer operators that relate $\tilde{U}_i$ to the corresponding velocities $\tilde{u}_i$ can be defined. Where $\tilde{U}_i$ is approximated in its support domain by a linear interpolation of the velocity. A weighted MLS approximation is used to tune these two velocities as good as possible. Cubic splines were used for the weight functions.
In the end using this MLS method, $\tilde{U}_i$ for each Lagrangian marker can be approximated in its support domain as follows:

$$\tilde{U}_i = \sum_{k=1}^{ne} \phi_k^l(\mathbf{x}) \tilde{u}_i^k = \kappa^T \tilde{\mathbf{u}}_i^k, \tag{2.5}$$

where k denotes a Eulerian point in the interpolation stencil, ne is the total number of grid points in the interpolation stencil, ϕ_k^l the shape function between grid point k and marker l and $\kappa(\mathbf{x})$ is a column vector of length ne, containing the shape function values for marker point l.

Eq. (2.5) gives $\tilde{U}_i$, which can be substituted into eq. (2.3), giving the volume force F_i on all the Lagrangian markers. The same shape functions used in the interpolation procedure can then be used to transfer F_i to the Eulerian points associated with that Lagrangian marker l. The final forces on the Eulerian grid become:

$$f_i^k = \sum_{l=1}^{nl} c_l \phi_k^l F_i^l, \tag{2.6}$$

where f_i^k is the volume force in the Eulerian point k in the direction i, nl the number of Lagrangian markers related to grid point k, F_i^l the force in Lagrangian marker l and ϕ_k^l again the shape function relating the variables between the grid point k and marker l. The factor c_l is used to properly scale the shape functions. To rescale the shape functions the following conditions must be satisfied:

$$\sum_{k=1}^{nte} f_i^k \Delta V^k = \sum_{l=1}^{ntl} F_i^l \Delta V^l, \tag{2.7}$$

where nte and ntl are the total number of forced grid points, $\Delta V^k = (h_x \times h_y \times h_z)$ is the volume associated with the Eulerian grid point l and $\Delta V^l = A^l h^l$ the volume associated with the marker l. The total force acting on the fluid must not be changed by the transfer from Eulerian to Lagrangian grids and the other way around. The transfer operators conserve momentum as well. In case of uniform grids, the torque will be also conserved.

The force form eq. (2.6) can be used to correct the predicted velocity $\tilde{u}_i$ with respect to the boundary conditions on the immersed body, leading to an approximate velocity u_i^*, which is not divergence-free:

$$u_i^* = \tilde{u}_i + \Delta f_i. \tag{2.8}$$

When a correction term is applied, this approximate velocity can be applied to a divergence free space:

$$u_i^{n+1} = u_i^* - \Delta t \frac{\partial}{\partial x_i}(\delta p), \tag{2.9}$$

where $\delta p = p^{n+1} - p^n$ is the pressure correction satisfying the Poisson equation, eq.(1.7). The velocity field u_i^{n+1} satisfies the boundary conditions to the order of $O(\Delta t^2)$. By using these corrections the incompressibility constraint is also satisfied. [31]

So, in summary, the predicted velocity on the Eulerian grid is transferred to the Lagrangian grid. The Lagrangian velocity is substituted in the direct-forcing equation to compute the Lagrangian volume force. This force is again transferred back to the Eulerian grid and can be used to correct the predicted velocity to respect the boundary conditions.

2.3.2 Calculation of forces

As mentioned previously, the direct-forcing function is not a force with any physical meaning. In this section the computation of the hydrodynamic forces will be shortly explained. However, extra complications are introduced in the computation of those forces generated by the surrounding fluid, since the computational grid and the body are almost never aligned. For rigid bodies eq. (2.6) is sufficient to compute the hydrodynamic forces, provided that all interior points are properly treated. In the general case with moving and deforming bodies, the approach is not trivial. In this study the hydrodynamic forces per unit area on a surface element is directly computed from the flow field around the body by:

$$f_i^H = \sigma_{ij} n_j = \left[-p\delta_{ij} + \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right] n_j, \tag{2.10}$$

where f_i^H is the hydrodynamic surface force in direction x_i, σ_{ij} the stress tensor, n_j the direction cosine of the normal unit vector in x_j direction, δ_{ij} an alternative way to formulate the identity tensor, and μ the viscosity term. Note that this equation is an addition of eq. (1.9) and eq. (1.10). In order to use eq. (2.10), p and $\partial u_i/\partial x_j$ on the body surface should be known. Using the same transfer functions to estimate p and $\partial u_i/\partial x_j$ at the Lagrangian makers will probably underestimate the actual traction forces, since the boundary is defined in a sharp manner, while the pressure and velocity are forced to vary smoothly through the surface of the body. To solve this problem a normal probe is created on the body surface for each Lagrangian marker by locating an external point e at a distance h_n from the surface (see fig. 2.3). The distance h_n is proportional to the local grid spacing and is given by $h_n = (h_x + h_y + h_z)/3$. The pressure p at point e is computed using a similar MLS formulation as in the previous section for

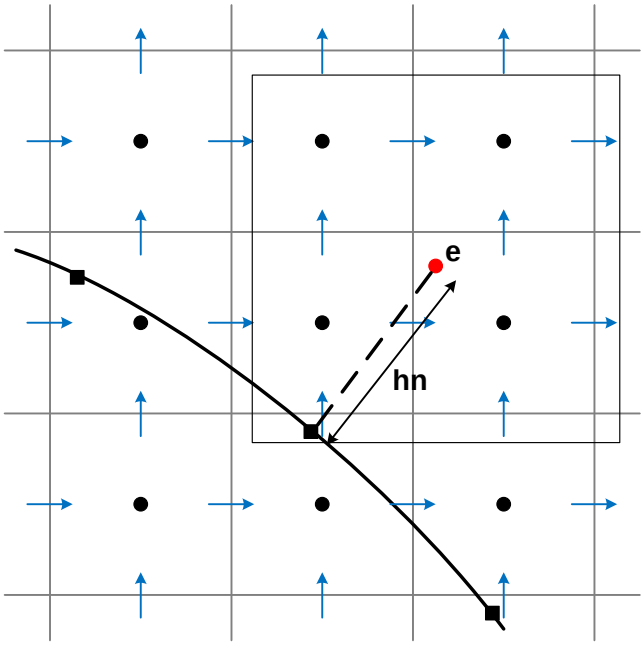


Figure 2.3: Schematic representation of the normal probe defined between a Lagrangian marker l and point e within the support domain, which is used for the MLS approximation.

the velocity, but now the support domain is centred around point e. The pressure at the surface (needed for eq. (2.10)) is obtained from:

$$p^l = p^e - \frac{\partial p}{\partial n} h_n, \tag{2.11}$$

where $\partial p/\partial n$ is obtained from the momentum equation normal to the boundary. The velocity gradients at location e for each Lagrangian marker l are derived from eq. (2.5) by differentiating:

$$\frac{\partial U_i}{\partial x_j} = \sum_{k=1}^{ne} \frac{\partial \phi_k}{\partial x_j} u_i. \tag{2.12}$$

Because h_n is in the order of the local grid size, and because it is assumed that the variation in the velocity is linear close to the body, the derivatives $\partial U_i/\partial x_j$ are a good approximation of the derivatives $\partial u_i/\partial x_j$.

This method of computing the forces on the surface of an immersed body is very accurate compared to boundary-conforming methods at the same grid resolution [31].

2.4 DNS and LES solver

Two different solvers can be chosen within this code, a Direct Numerical Simulation (DNS) and a Large Eddy Simulation (LES). DNS is restricted to low Reynolds number values so that LES is preferred in more practical situations.

2.4.1 DNS

In a DNS the Navier-Stokes equations must numerically solved on all spatial and temporal scales without using a model for turbulence. DNS is the most accurate approach to solve turbulence, provided that the grid is sufficiently fine to resolve the motion at the smallest scales of importance. This means that turbulence must be resolved from the smallest dissipative scales up to the scales with motions containing most kinetic energy. This smallest scale, also called the Kolmogorov scale η, is given by:

$$\eta = (\nu^3/\varepsilon)^{1/4}, \tag{2.13}$$

where ν is the kinematic viscosity and ε the rate of kinetic (viscous) dissipation. In order to resolve turbulence at the smallest scales, the mesh size h must be smaller then the Kolmogorov length scale $h \leq \eta$. The estimated energy dissipation per unit mass is $\varepsilon \propto U^3/L$, where U is the macroscopic velocity scale and L the macroscopic length scale. The number of points N in a given mesh must satisfy $Nh > L$, so the large scales are contained within the computational domain. Combining all previous relations gives the following relation for the number of mesh points:

$$N \sim \frac{L}{\eta} \sim \left(\frac{L^4\varepsilon}{\nu^3}\right)^{1/4} = \left(\frac{L^3U^3}{\nu^3}\right)^{1/4} = Re^{3/4}. \tag{2.14}$$

Thus in a three-dimensional DNS the number of mesh points satisfies:

$$N^3 \geq Re^{9/4}. \tag{2.15}$$

It can be concluded that the memory storage requirement in DNS grows very fast with the Reynolds number and that the computational costs of DNS are very high. The number of floating point operations grows even as Re^3, since the number of time steps grows also as a power law of the Reynolds number. In order to be accurate, the integration time step Δt must be small enough such that the fluid particles move only a fraction of the mesh size in each time step. That is,

$$C = U\Delta t/h < 1, \tag{2.16}$$

with C the Courant number. The time interval simulated will be proportional to the turbulence time scale $\tau = L/U$ and h must be in the order of η. Combining these relations results in

$$L/\eta \sim Re^{3/4} \tag{2.17}$$

and consequently the number of time steps has got a similar kind of dependence on *Re* like one spatial dimension [33] [34].

2.4.2 LES

Within LES the range of length scales to solve Navier-Stokes is reduced to suppress the computational costs. Instead of resolving the turbulence on small scales like DNS, a model is used to mimic the turbulence. The difficulty of modeling this is to model the interaction between the large and small scales, preventing separation of scales. The main operation of LES is to apply a filter on a spatial and temporal field. Scales smaller than a certain cutoff length scale Δ and cutoff time scale τ_c will be eliminated from the actual field. Thus the field ϕ is split up into a filtered $\bar{\phi}$ and sub-filtered part ϕ' as $\phi = \bar{\phi} + \phi'$. The filtering is usually done by a convolution with a filter function G, $\bar{\phi} = G \star \phi$. Assuming that the flow is incompressible, the filtered continuity equation and Navier-Stokes equation (1.4) become:

$$\frac{\partial \bar{u}_i}{\partial x_i} = 0 \tag{2.18}$$

$$\frac{\partial \bar{u}_i}{\partial t} + \frac{\partial \overline{u_i u_j}}{\partial x_j} = -\frac{1}{\rho}\frac{\partial \bar{p}}{\partial x_i} + \nu\left(\frac{\partial \bar{u}_i}{\partial x_i} + \frac{\partial \bar{u}_j}{\partial x_i}\right) = -\frac{1}{\rho}\frac{\partial \bar{p}}{\partial x_i} + 2\nu\frac{\partial \bar{S}_{ij}}{\partial x_i}, \tag{2.19}$$

where $\bar{S}_{ij}$ is the rate-of-strain (or deformation) tensor. These equations are now written in index notation, with i and j denoting the different vector components. The only term causing difficulties within LES modeling is the nonlinear filtered advection term $\overline{u_i u_j}$. To solve this term, knowledge of the unfiltered velocity field is required. This field is unknown and includes the interaction between the large and small scales. So it must be modeled under appropriate assumptions to prevent separation of scales. The filtered advection term can be split up as $\overline{u_i u_j} = \tau^r_{ij} + \bar{u}_i \bar{u}_j$, where τ^r_{ij} is the residual stress tensor (also known as subgrid-scale tensor) grouping all unclosed terms. This stress tensor can be decomposed as $\tau^r_{ij} = L_{ij} + C_{ij} + R_{ij}$, where L_{ij} is the Leonard tensor representing interactions among large scales, R_{ij} the Reynolds stress-like term representing the interactions among the sub-filter scales, and C_{ij} the Clark tensor representing cross-scale interactions between large and small scales. Actually the challenge in this is to model the particular stress tensor with a subgrid-scale model (SGS) [35]. Most SGS models are based on the eddy-viscosity assumption (the Reynolds stresses could be linked to the mean rate of deformation similarly as viscous stresses) to model the subgrid-scale tensor:

$$\tau^r_{ij} - \frac{1}{3}\tau^r_{kk}\delta_{ij} = 2\nu_t \bar{S}_{ij}, \tag{2.20}$$

where $\bar{S}_{ij}$ is again the rate-of-strain tensor of the resolved field and ν_t the turbulent viscosity. The SGS model used in this code is the so-called Smagorinsky method. In Smagorinsky's model, the eddy-viscosity is assumed to be proportional to the subgrid characteristic length scale Δ (named as the cut off length scale before) and to a characteristic turbulent velocity taken as the magnitude of the local strain $|\bar{S}|$:

$$\nu_t = (C_s\Delta)^2 |\bar{S}|. \tag{2.21}$$

Above, $|\bar{S}| = \sqrt{2\bar{S_{ij}}\bar{S_{ij}}}$ and the typical length scale Δ is computed as $(\Delta x_1 \Delta x_2 \Delta x_3)^{1/3}$, where Δx_i is the grid spacing in direction i. The coefficient C_s is an externally provided coefficient that has been tuned from the decay of homogeneous isotropic turbulence. Actually the value of the Smagorinsky constant for isotropic turbulence is given by

$$C_s = \frac{1}{\pi}\left(\frac{2}{3C_k}\right)^{3/4}, \tag{2.22}$$

assuming that the energy production and dissipation on small scales is in equilibrium, and where C_k is the Kolmogorov constant, with for the energy spectrum $E(k) = C_k \varepsilon^{\frac{2}{3}} k^{\frac{-5}{3}}$. Practically, the value of C_s depends on the type of flow and mesh resolution. The coefficient is not a universal constant and therefore this coefficient is the most serious shortcoming of this model [36] [37] [38].

2.4.3 LES applied in the code

The filter operations in LES can be implicit or explicit. Within this code the additional viscous LES terms, when a LES simulation is carried out, are computed explicitly inside the nonlinear terms routines. The filtering is performed in all directions using a box filter in the physical space. The contributions will not be weighted according to the volume of their corresponding cell, because it turns out to be a quite heavy computation. But a uniform Cartesian grid is used and furthermore using an LES model approximations are made anyway. The turbulent viscosity is calculated at the cell centre. Therefore, after computation of the strain tensor, all elements will be averaged at the cell centre. The turbulent viscosity (2.22) is calculated using an C_s coefficient of $\sqrt{0.025}$. This coefficient probably gained the best results in previous studies by Verzicco and was chosen through empirical testing.

2.5 Time advancement

The time advancement of the solution is obtained by a Adams method. This method is popular to solve numerically non-stiff ordinary differential equations (ODEs). Adams

methods are useful to reduce the number of functions calls, but require more CPU time than Runge-Kutta methods. For the simulations of this study the Adams-Bashfort method is used.

2.5.1 Adams-Bashforth

The Adams-Bashforth methods are linear multi-step methods and commonly used in non-stiff initial value-problems. It is an explicit method with very small regions of absolute stability. The description of the Adams-Bashforth method in this section is obtained from the derivation by Ascher [39]. In general a k-step linear multi-step method is given by

$$\sum_{j=0}^{k} \alpha_j \mathbf{y}_{n-j} = h \sum_{j=0}^{k} \beta_j \mathbf{f}_{n-j}, \tag{2.23}$$

where α_j, β_j are the method's coefficients and h is the step size. The discretized terms $\mathbf{y}$ and $\mathbf{f}$ originate form the ordinary differential equation system $\mathbf{y}' = \mathbf{f}(t, \mathbf{y})$. For all Adams methods it is considered that $\alpha_0 = 1, \alpha_1 = -1$ and $\alpha_j = 0, j > 1$. In a k-step Adams method $\mathbf{f}$ is interpolated through the previous point $t = t_{n-1}, t_{n-2}, ..., t_{n-k}$. Thus the general formulas for the Adams-Bashforth method become

$$y_n = y_{n-1} + h \sum_{j=1}^{k} \beta_j f_{n-j}, \tag{2.24}$$

where the coefficients β_j are found using a polynomial interpolation for the function $\mathbf{f}$ to be solved. The coefficients turn out to be

$$\beta_j = (-1)^{j-1} \sum_{i=j-1}^{k-1} \binom{i}{j-1} \gamma_i, \tag{2.25}$$

$$\gamma_i = (-1)^i \int_0^1 \binom{-s}{i} ds. \tag{2.26}$$

Notice that the first-order Adams-Bashforth method is known as the forward Euler method. The second-order Adams-Bashforth with $k = 2$ and $\gamma_0 = 1, \gamma_1 = 1/2$ yields

$$y_n = y_{n-1} = +h \left(\frac{3}{2} f_{n-1} - \frac{1}{2} f_{n-2} \right), \tag{2.27}$$

or equivalently

$$y_n = y_{n-1} = +h \left(f_{n-1} + \frac{1}{2} \nabla^1 f_{n-1} \right). \tag{2.28}$$

where ∇^1 is part of the series $\nabla^i = \frac{\nabla^{i-1} f_{n-1} - \nabla^{i-1} f_{n-2}}{f_{n-1}}$ with $\nabla_0 = 1$. The first equation with coefficients $3/2$ and $-1/2$ is used in the Adams-Bashforth calculations in this code. Compared to Euler's method, the second-order Adams-Bashforth is more accurate [39]. In this derivation a constant step size h is assumed, whilst in the code the step size is varying to satisfy the Courant number restrictions. From a mathematical point of view only third order Runge-Kutta is capable of handling variable time steps without reducing the order of accuracy. From a practical point of view it is shown that also the Adams-Bashforth method performs well concerning "small" CFL. Although there exist Adams-Bashforth methods with variable step size, a standard Adams-Bashforth scheme with a variable time step is used in present study [38].

2.6 Integration scheme

The integration scheme used to solve the three-dimensional, time-dependent incompressible Navier-Stokes equations is a so-called fractional step method described by Kim and Moin [28]. This section continues with a review of the ideas of Kim and Moin, following their paper. The method herein works in conjunction with the approximate-factorization technique. Using this method, the velocities solved from Navier-Stokes satisfy the continuity equation up to machine accuracy at every time-step. The three-dimensional Poisson equations are solved directly by a transform method. The pressure is solved in the first step of the fractional step method, wherein the pressure correction is solved by a direct method based on trigonometric expansions.

The fractional step method is a method of approximation of the evolution equations based on the decomposition of the operators they contain. The pressure in the Navier-Stokes equations is interpret as a projection operator. This operator projects an arbitrary vector field into a divergence-free vector field.

The following two-step time-advancement equations are solved in the code:

$$\frac{\hat{u_i} - u_i^n}{\Delta t} = -\alpha \nabla p^n + \gamma H_i^n + \rho H_i^{n-1} + \frac{\alpha}{2Re} \nabla^2 \left(\hat{u}_i + u_i^n \right), \tag{2.29}$$

where the superscript n indicates the time step level, $H_i = -(\partial / \partial x_j u_i u_j)$ contains the nonlinear terms (ie. the convective terms and viscous terms with a single velocity derivative), p is the pressure, u the velocities, $\hat{u}$ the provisional non-solenoidal velocity field. α, γ and ρ are the coefficients derived from the time advancement scheme. Thus in this case, $\alpha = 1$, $\gamma = 3/2$ and $\rho = -1/2$ (see eq.(2.27)). The non-solenoidal velocity field is projected onto the solenoidal field (incompressible vector field) by a scalar Φ using the following equation:

$$\nabla^2 \Phi = \frac{1}{\alpha \Delta t} \nabla \cdot \hat{u}_i. \tag{2.30}$$

Then the solenoidal velocity field at the new time step is computed through

$$u^{n+1} = \hat{u}_i - \alpha \Delta t \nabla \Phi. \tag{2.31}$$

The pressure field is updated using

$$p^{n+1} = p^n + \Phi - \frac{\alpha \Delta t}{2Re} \nabla^2 \Phi. \tag{2.32}$$

All of these variables are located on a staggered grid (figure 2.4) with the velocities on the faces of the computational cell and all the scalars at the centre. The method is second-order accurate in both space and time [40] [28].

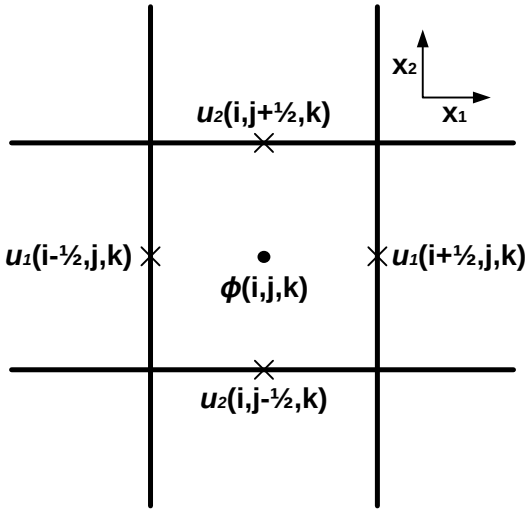


Figure 2.4: The staggered mesh in two dimensions.

Chapter 3

Method

This chapter is concerned with the numerical set up and approach used in this study. The code is written in Fortran and an OpenMP version can be used to run the code parallel. The simulations are performed on a server with two Intel Xeon E5405 processors, both with four parallel cores, using an Ifort compiler for the parallel simulations. Serial runs used the GNU compiler or either the same Ifort compiler.

3.1 Numerical setup

In figure 3.1 the XZ- and YZ-sections (through $(0,0,2.5)$) of both the simulations with the sphere and hand are shown. In all simulations a cubic domain is used of $5 \times 5 \times 5$ non-dimensional length units consisting of 201 nodes in each directions. The 200 grid cells are of uniform size $0.025 \times 0.025 \times 0.025$. The boundaries at $z = 0$ and $z = 5$ are the in- and the outlet, respectively. The inlet velocity is set uniform across the entire inlet, $U = 1$ in non-dimensional units. At the outflow a radiative boundary conditions (for open boundaries) for the velocity applies. The boundaries at $x = -2.5$ and $x = 2.5$ have periodic boundary conditions, thus data at $x = -2.5$ and $x = 2.5$ are exactly the same. Thus actually it can be assumed that the immersed body is within one of a linear array of bodies in the X-direction. The boundaries at $y = -2.5$ and $y = 2.5$ have a free-slip conditions. The free-slip condition consists of a non-penetration condition $\mathbf{v} \cdot \mathbf{n} = 0$ and a stress condition on the wall. The stress condition implies that the slip velocity at the wall is dependent on the shear stress, where the shear stress on its turn is assumed to be zero. Thus no vorticity can be produced at the walls and no fluid can leak through the walls.

This code can handle a single (rigid moving) immersed body in the domain. The typical width of the immersed bodies is 1 in non-dimensional units, so one fifth of the domain. In the case of the sphere this typical width is the diameter, while in the case of the hand it is the distance between thumb and little finger side. The relative

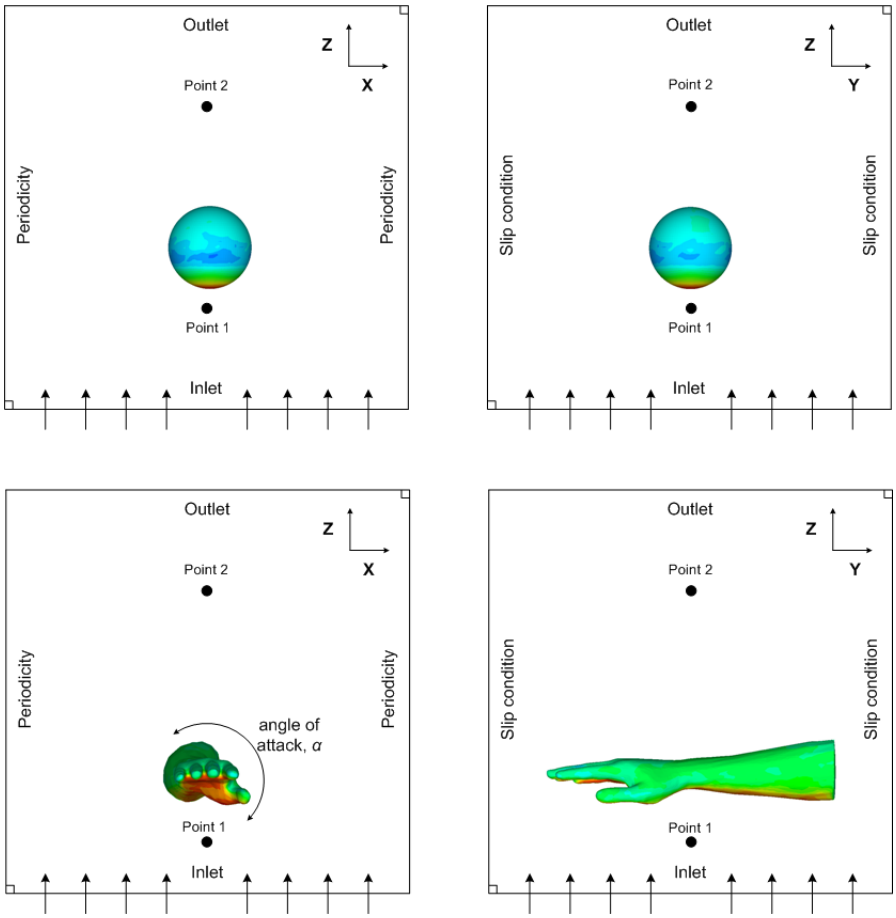


Figure 3.1: XZ- and YZ-sections through (0,0,2.5) for both the simulations of the sphere and hand. Boundary conditions are shown within the figure.

density of the body was set to 1, thus it is assumed that gravity forces are turned off in this study. The boundary condition on the immersed body is a no-slip condition, which states that the velocity is zero at the boundary. Input files for the immersed body considered in a certain simulation must be in STL (ASCII) format, where STL means STereoLithography. STL files describe a raw unstructured triangulated surface by the unit normal and vertices of the triangles using a three-dimensional Cartesian coordinate system. The files do not contain any scaling information, so the units are arbitrary. The triangles of the triangulation have to be evenly spread on the surface of the body in order to run the code properly. Furthermore the size of the triangles should not deviate too much from the grid cell size. Basically, the bodies can be placed everywhere in the domain under every configuration as long as they are entirely within the domain. In case of simulations with a stationary body, the body was placed a little towards the inlet, in order to survey a bigger part of the wake. Just in front of the body and further upstream

numerical ”probes” are placed to measure the velocities across time (denoted by point 1 and point 2 in figure 3.1). Movements of the rigid bodies can be prescribed in the code by functions for position, velocity and acceleration for both the translational movements and rotational movements.

3.2 Output of simulations

The output of the simulation is given in several files and formats. There is a file containing the viscous-, pressure- and gravity forces, momenta and the velocities in point 1 and 2, on the body at each time step of the simulation. At times indicated two files representing all velocity, pressure, vorticity and velocity gradient data on all nodes of the XZ-plane at $y = 0$ and the YZ-plane at $x = 0$ will be generated. Another file containing all pressure and stress data on all nodes at the surface of the object will be written at these times as well. These files are designed to open with Tecplot360 software. Furthermore there are two kinds of binary files containing the three-dimensional data. One of these files contains all the velocity and pressure data on each node, one of them all turbulent viscosity data which will be just generated when applying a LES simulation. The times on which these binary files are generated can be controlled separately from the other files. Storage limitations prohibit saving all data at each time step.

3.3 Simulations with a sphere

In order to test whether the code produces reliable results and to build up some experience with the simulations, first some well-documented results will be reproduced. Since the drag coefficient of a sphere embedded in a uniform flow is well documented, the sphere is a good test case. The flow around a fixed sphere will be simulated at different Reynolds numbers for 100 (simulation) time units. To see which solver gives the most reliable results in particular Reynolds regimes, a part of the simulations will be performed with the LES (Large Eddy Simulation) solver turned on and part with the DNS solver turned on. The drag coefficient will be calculated using the averaged drag force (the sum of the viscous and pressure forces on the body), parallel to the mean stream direction. Just results of the second half of the total simulation time will be used to avoid starting up errors. Since the code is totally dimensionless, so is the drag force resulting from the output. Calculating the drag coefficient from this drag force is a matter of multiplying the force with a factor to normalize with the frontal surface. Following eq. (1.13) and considering that in the simulations, the velocity $U = 1$, density $\rho = 1$ and the radius of the sphere $R = 0.5$, the drag coefficient following from the

simulations is:

$$C_d = \frac{F_d}{\frac{1}{2}\pi 0.5^2} = \frac{8F_d}{\pi}. \tag{3.1}$$

The drag coefficient will be plotted as a function of the Reynolds number, just like figure 1.8, together with the documented values. Furthermore an attempt is made to distinguish the different regimes in the wake qualitatively, by comparing the behaviour of the force in time with the regimes reported by Sakamoto (section 1.4) [24]. It will also be examined whether useful information can be extracted from the fluctuations of the wake. Therefore the Strouhal number (Sr) of the fluctuating velocity in point 2 (fig. 3.1) is computed using a Fourier transform, following the ideas of Sakamoto.

3.3.1 Accelerating sphere

The flow around an accelerating sphere has also been simulated. An attempt is made to validate the added mass principle (1.5) of a sphere and to uncover the effects of the accelerating. The left figure 3.2 shows a sketch of the numerical set up. The sphere will

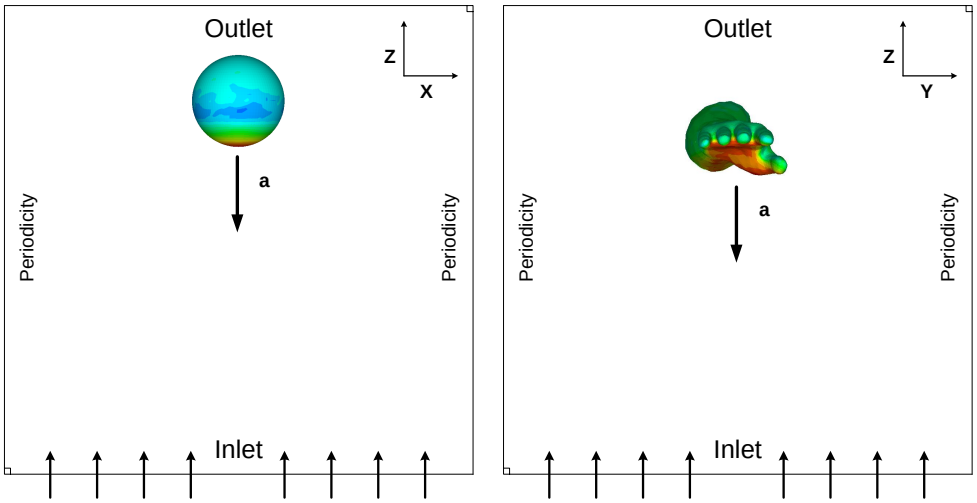


Figure 3.2: Numerical setup for the simulation of the flow around an accelerating sphere and hand. The sphere is accelerated from initial position $z = 4$, the hand is accelerated from initial position $z = 3.5$. The acceleration is directed towards the inlet.

be accelerated from initial position $z = 4$ towards the inlet (approximately three length

units). The simulations diverge when the immersed body is too close to the boundary of the domain (inlet) and stop concerning out of bound reasons. First a short reference run at constant flow speed will be done with the initial conditions. Then six different (non-dimensional) accelerations will be simulated, namely $a_s = 0.1, 0.2, 0.5, 1, 2$ and 5. The drag force will be plotted together with an analytical result of the force on an accelerating sphere. If the results from the simulation agree with the analytical result, added mass will be present. Note that the velocity is actually $v = v_0 + at$, where $v_0 = 1$ is the initial velocity of the flow and a_s the acceleration of the sphere. It is interesting to examine how much the drag coefficient is increased by the accelerations and whether it agrees with the theoretical added mass principle.

3.4 Simulations with a hand

Subsequently, simulations of the flow around a rigid (right) hand and forearm will be performed. Again the simulation will run off 100 time units and results will be averaged among the second half of the total simulation time. For the first set of simulations, the hand will have a fixed position and orientation throughout all simulations. To see whether the drag coefficient is dependent on velocity, the flow will be simulated at a range of different Reynolds numbers. The hand was positioned a little towards the inlet at $z = 1.5$, with the hand palm orientated towards the inlet, in order to study a larger part of the wake. Actually the problem of a hand moving with constant velocity through quiescent water was reversed to a stationary hand in a steady flow. This is a matter of the choice of the reference frame. The frontal surface of the hand will be determined using a Matlab script. A cross-section (image) of the hand is imported, all pixels within the hand were blackened, all pixels outside the hand whitened. The result is a matrix of 0's and 1's. The surface of black pixels was calculated, by summing the amount of black pixels in each row. This is converted to a non-dimensional surface by using the ratio between the number of black pixels from the little finger to the thumb side and the non-dimensional size of this typical width. Again this frontal surface will be used to calculate the drag coefficient, as in the simulations with the sphere.

In the second set of simulations, the hand will have a fixed position during a simulation. But throughout all simulations the angle of attack is increased. What is meant by the angle of attack, is shown in figure 3.1 and highlighted in figure 3.3. The angle of attack is 0° when the hand palm is oriented towards the inlet, 90° when the thumb side is oriented towards the inlet, etcetera. Possibly different from the expectations, the drag and lift coefficients will not be calculated using the frontal surface in each case. The maximum projected area, thus the frontal surface at 0°, is used for each angle of attack, in order to compare the results with previous studies. It would be interesting to study how the drag and lift coefficients change with the angle of attack and whether an addition of drag and lift forces could obtain an even higher "propulsive" force at certain

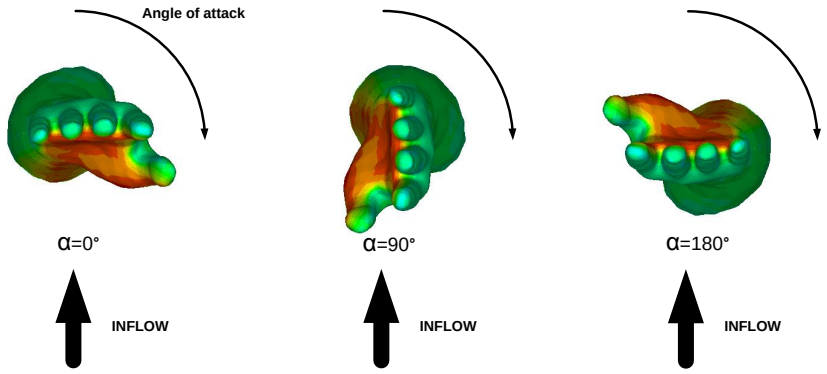


Figure 3.3: The hand at different angles of attack, shown from the XZ-section.

angles. Therefore the maximum resultant force $\sqrt{F_d^2 + F_l^2}$ is plotted as a function of angle of attack. The resultant will always be bigger than the drag force with presence of a lift force. The resultant force will be at an certain angle θ with the drag force, as shown in figure 3.4. This angle is of interest and must be calculated. In the end the arm stroke

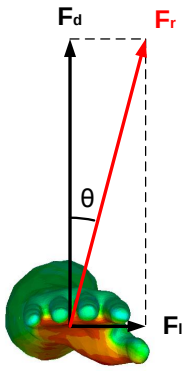


Figure 3.4: The resultant force of drag and lift. The angle between the resultant force and drag force is indicated by θ.

should be adjusted to the maximum resultant force. The results obtained with these simulations of the hand will be discussed using previous results found in literature.

3.4.1 Simulations with an accelerating hand

In reality the arms of a swimmer will not move at a constant speed all the time. The next step in this study will be the simulation of an accelerating hand, which is less trivial than a hand in a steady flow. It will be interesting to investigate how drag is affected by accelerated conditions and to see what is for example the added mass of an accelerating hand. To avoid starting up errors within the short time period available for accelerating, first a reference run at constant flow speed will be done, allowing the flow to "settle". The hand is again positioned with the hand palm oriented towards the inlet, but now the hand is placed more towards the outlet of the domain at $z = 3.5$, see the right figure in 3.2. The hand will be accelerated over approximately 3 length units (0.36 m in real). Accelerations of $a_r = 0.5, 1, 2, 3, 4, 5$ m/s^2 towards the inlet will be imposed. Note that there is an initial velocity v_0 due to the imposed velocity of the flow.

In order to calculate the corresponding acceleration in non-dimensional simulation units, simulation time units t_s and real time units t_r (seconds) must be related first. Velocity and length could be related using the Reynolds number. To relate t_s and t_r, the expression $s = vt$ is used, with s a distance. In the non-dimensional simulations it takes 1 time unit to pass 1 length unit with (initial flow) velocity of 1. In real it takes $t_r = s_r/v_r$ to pass the distance s_r (corresponding with $s_s = 1$) with (initial flow) velocity v_r, following from the Reynolds number. This gives a relation between the two time units. These corresponding variables together with the expression $s = \frac{1}{2}at^2 + v_0t$, should gave a relation between the accelerations.

3.5 Calculation of forces from surrounding flow

In the simulations the net forces on the hand are computed from integration of the viscous and pressure forces at the surface of the immersed body (equations (1.9), (1.10) and (2.10)). In experiments this is not that evident. For research on swimming efficiency it is desirable to compute the forces on the hand from the velocity field at some distance from the hand. Through these simulations an attempt is made to calculate the forces from the surrounding flow. For example, forces could be deduced from the vorticity field, induced by the hand motion. Such a vorticity field could be measured from the video analysis of tracer particle motion. The simplest application would be to compute the forces from a fluid momentum balance using a test volume surrounding the hand (eq.(1.8)).

Forces on the body will be calculated from the three-dimensional flow field, contained in the output files of the simulations. To do so, an arbitrary volume must be chosen around the immersed body, preferably at some distance of the periodic and free-slip walls of the domain. This is to avoid as much of the influences of the imposed boundary conditions

as possible. Figure 3.5 shows the volume chosen. The forces will be calculated just at

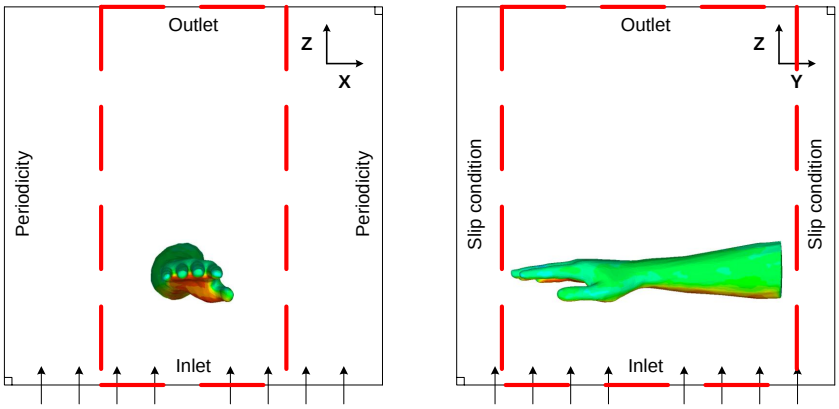


Figure 3.5: The arbitrary volume used to apply conservation of momentum on the flow is indicated by the red striped lines in both the XZ- and YZ-section.

one point of time. Not all time dependent information could be stored due to storage limitations. The results will be compared with the forces resulting from the simulations.

Clearly, eq.(1.8) needs information about the pressure field which is hard to come by experimentally. At most the pressure at certain points is known using a sensor. A reformulation of eq.(1.8), which avoids the pressure field was done by Graziani et al. [12]. Following this method an attempt was made to calculate the forces as well.

Chapter 4

Results

In this chapter we will present and discus the obtained results in this study. Part of the results is focused on reproducing results of previous work to see whether our results show agreement with documented cases. After presenting and discussing the results of the sphere and swimmer's hand there will be continued with a general consideration of domain size, dependency on Z-position and influences of the wall.

4.1 Drag coefficient of a sphere

4.1.1 Drag coefficient of a stationary sphere

Primarily the code was validated by simulations of the flow past a sphere. Simulations were performed in the Reynolds number range 10 to $4 \cdot 10^6$. Results are shown in figure 4.1 in a *log*-*log*-plot and plotted on top of the result reported by Schlichting [23]. The red circles indicate the results obtained with the DNS solver, while the blue triangles denote the results from the LES solver. The black dotted curve in the back shows the digitized data from Schlichting. Any inaccuracies in the data from Schlichting came from digitizing the graph with Origin9 software. The drag force (and coefficient) is defined as a time average, because even for a "steady" flow the drag force can fluctuate. From a certain Reynolds number, a sphere will asymmetrically shed its wake. Especially this vortex shedding induces large fluctuations in the force along the direction of motion which affects the drag force. Inside the data points small black error bars are visible, indicating the deviation from the averaged drag force. The error calculation is based on the standard deviation of the non-dimensional force in time. The forces on the immersed body fluctuate in time due to fluctuations in the wake of the passing flow (vortex shedding for example). The computational errors are small and do not influence the results considerably. One error check is the maximum divergence within the computational domain. When the simulations are stable, the maximum divergence was of

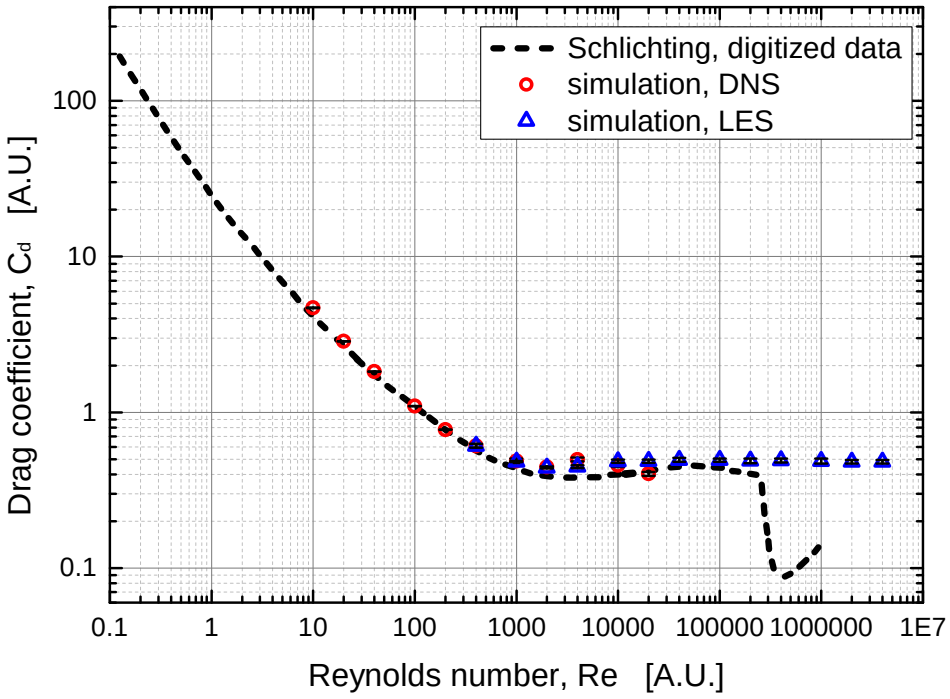


Figure 4.1: *log-log*-plot of the drag coefficient of a sphere versus the Reynolds number. Red circles indicate results obtained with the DNS solver, the blue triangles the results from the LES solver. Small error bars are visible within the data points. The black dotted curve in the back shows the digitized results from Schlichting. [23]

the order $10^{-11} - 10^{-14}$ (machine error: 10^{-16}) and can be considered zero, as it should be. This immediately explains why the errors in the laminar regime up to $Re = 100$ are that small (of the order 10^{-10}) in comparison with the vortex shedding and turbulent regime (order 10^{-2}). As an exception, the simulation at $Re = 10$ showed a larger error in relation to all other runs, which is due to an instability present throughout the whole simulation.

Generally the data with both solvers agree quite well with the documented values. Data from the DNS solver are not reliable anymore for Reynolds number values larger than approximately 1000. Using 201 mesh points allows us to perform calculations up to Reynolds numbers of 1177, following eq.(2.15). Also a coarse estimation of the boundary layer thickness is made, using the Blasius boundary layer thickness of a flat plate. From this it can be concluded that there are less than five nodes within the boundary

layer when the Reynolds number is 1000 or more. In fact the boundary layer can not be solved at this resolution. After $Re = 1000$ it can be noticed that the data from the simulation slightly lose the trend described by Schlichting. Data obtained with the LES solver still follow the trend of Schlichting. Except for the data points around the critical Reynolds number $2 \cdot 10^5$, where the data points keep on following the constant level of the drag coefficient. Actually the boundary layer around a sphere should undergo a transition to turbulence, which corresponds to the dip of the drag coefficient in the data of Schlichting (section 1.4). First the idea was, that the turbulence in the boundary layer had to be initiated by a minor distortion, because a "clean" simulation would be too perfect compared with the real situation. To test whether this was the case, random noise was added on top of the uniform inlet velocity at levels of 1% and 5%. The results

Table 4.1: The effect on the drag coefficient when adding noise on top of the uniform inlet velocity.

Re	**random noise %**	**drag coefficient**
10^5	0	0.49 ± 0.01
	1	0.49 ± 0.02
	5	0.48 ± 0.01
$2 \cdot 10^5$	0	0.49 ± 0.02
	1	0.49 ± 0.02
	5	0.48 ± 0.01
$4 \cdot 10^5$	0	0.49 ± 0.02
	1	0.48 ± 0.01
	5	0.49 ± 0.01
10^6	0	0.49 ± 0.02
	1	0.48 ± 0.01
	5	0.46 ± 0.01

in table 4.1 reveal that the random noise did not effect the data considerably. Another, more obvious, explanation would be in using the LES solver. Since a LES solver is used, the turbulence is modelled and thus not exactly resolved. Furthermore the boundary layer thickness is approximately of the order 10^{-2} (using the turbulent boundary layer thickness of a flat plate) while the grid size is 0.025. It is not possible to solve all fine flow structures in the boundary layer at this resolution with just a maximum of one node inside the boundary layer. In fact the Kolmogorov length scale around the critical Reynolds number is in the order of $10^{-4} - 10^{-6}$.

4.1.2 Wake survey

The different flow regimes, when increasing the Reynolds number were coarsely analysed by considering the development and fluctuations of the force components in time.

Of course the steps in Reynolds number are too large to define sharp regions of different wakes, but at least the data can be grouped and compared with previous studies. In appendix A the associated figures are shown.

i ($Re \leq 100$), figure A.1:
The force exerted on the sphere is constant up to Reynolds numbers of about $Re = 100$, where the Z-force is just of importance, the other components are negligible small. It can be concluded from this that the flow is laminar up to approximately $Re = 100$, which is consistent with the rough distinction made by Sakamoto and Hanui. (section 1.4). There it was stated that the flow is laminar up to $Re = 130$.

ii ($Re = 200$), figure A.2:
The forces in the X- and Y-direction seem to grow gradually in time. Probably due to an asymmetry in the wake of the sphere. However, the X- and Y-forces are still negligible compared to the Z-force. An even longer simulation should have been carried out to see whether this is faint periodic pulsative motion, as Sakamoto and Hanui. described the region of $130 < Re < 300$. The force in Z-direction is still constant.

iii ($Re = 400$), figure A.3:
The X- and Y-forces start to develop clear fluctuations, probably due to vortex shedding. The averaged value of the forces in X- and Y-directions is still around zero, but the fluctuations are definitely not negligible. Besides that, the fluctuations in X- and Y-direction are approximately in phase. Probably regular hair-pin vortices are shed, providing the in phase fluctuations, which is again consistent with the results of Sakamoto and Hanui.

iv ($1000 \leq Re \leq 4000$), figure A.4:
The X- and Y-forces are no longer in phase, which suggest irregular shedding of vortices, as described in the third region stated by Sakamoto and Hanui. in section 1.4. Furthermore the force signal is not everywhere that smooth in time anymore, due to minor fluctuations on top of the clear fluctuations due to vortex shedding. Especially on the Z-force, which seems to be more or less constant over time, but constantly experiences minor fluctuations.

v ($Re \geq 10000$), figure A.5:
After $Re = 10000$ all runs seem to be in the same regime. X- and Y-forces show intens irregular fluctuations. The Z-force is approximately constant over time, with a minor fluctuations on top of the signal. Probably the vortex sheets separating from the surface of the sphere become turbulent. This is consistent with the last regime that was again formulated by Sakamoto and Hanui.

Thus the results of these numerical simulations seem to agree quite well with the different regions distinguished by Sakamoto and Hanui [24].

Furthermore the irregular fluctuations in the wake were examined by investigating the Strouhal numbers of the fluctuations in the three components of the velocity. The velocity was probed at a point in the wake behind the sphere at $z = 3.75$. Figure 4.2 shows

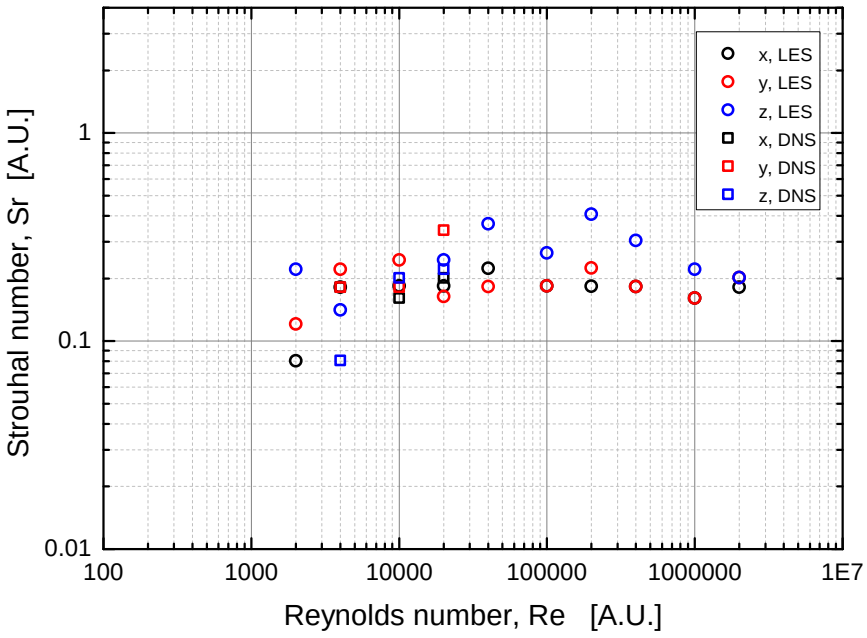


Figure 4.2: The Strouhal numbers of the different components of the velocity in point 2 as a function of the Reynolds number. Circles indicate the results from the LES solver, squares the results from the DNS solver. Black, red and blue respectively indicate the X-, Y- and Z-components of the velocity.

the results. The black, red and blue data points indicate the data obtained from X-, Y- and Z-components of the velocity, respectively. The circles represent the results obtained with the LES solver, the squares the results obtained with the DNS solver. It can be concluded that all Strouhal numbers are approximately in a band around $Sr = 0.2$. These results of this study agree with results obtained in previous studies (like Sakamoto and Hanui [24]). Sakomoto, Hino and others showed that the Strouhal number of the fluctuations in the wake of a sphere is approximately 0.2 at a broad range of Reynolds numbers. A considerable spread is seen in figure 4.2, which probably could be attributed to a limited time span of the simulations. The peaks in the frequency spectrum were not

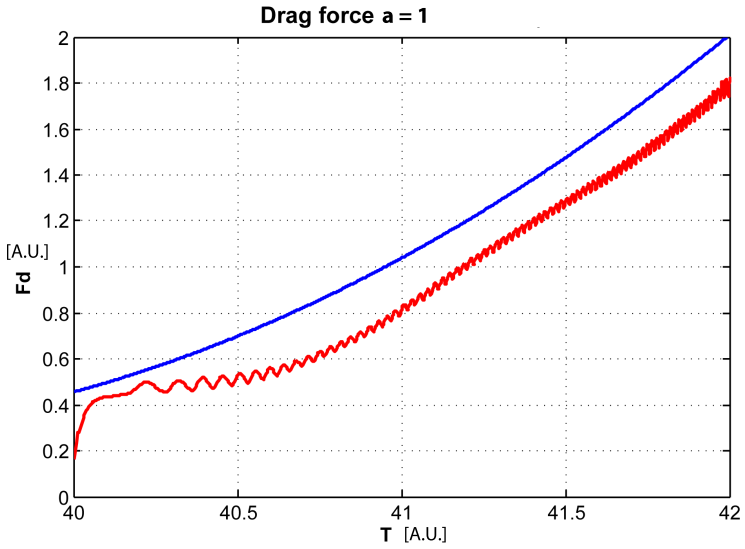
all that clearly, especially not for the Z-component. Maybe the position of the probe in the wake could be chosen different as well to gain better results.

4.1.3 Accelerating sphere

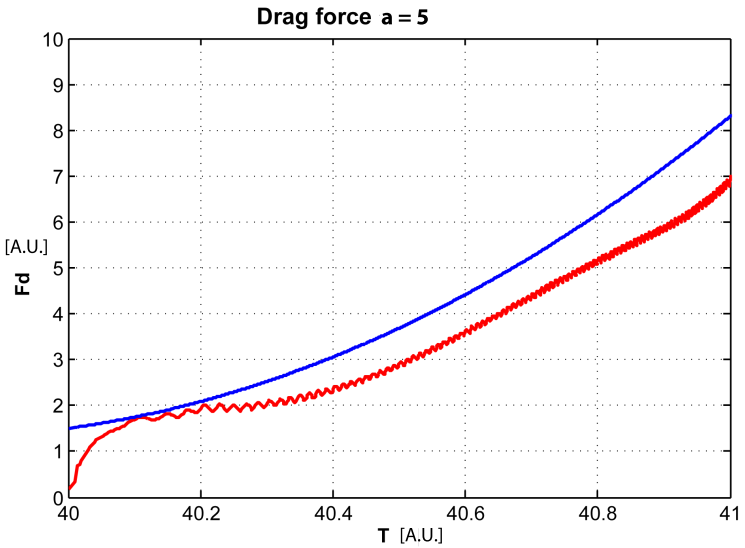
Before accelerating the sphere, first a reference simulation of 40 time units was performed with the initial conditions for the follow-up runs. The Reynolds number of the initial flow was set on $Re = 10^5$, the non-dimensional initial velocity is 1. Accelerations of $a = 0.1, 0.2, 0.5, 1, 2$ and 5 were tested. Figure 4.3(a) and figure 4.3(b) show the results of the (non-dimensional) drag force of an accelerating sphere at $a = 1$ and $a = 5$, respectively. The red graph represents the drag force resulting from the simulations. The blue graph represents an analytical result for the force. In the context of added mass the time-dependent force can be described by:

$$F(t) = \frac{1}{2}am + \frac{1}{2}A\rho v(t)^2 C_{d,stationary}, \tag{4.1}$$

with $v(t) = v_0 + at$, where v_0 is the initial velocity and a is the acceleration imposed at $t = 0$. The first term on the right side is the force due to the added mass of a sphere, where $\frac{1}{2}m = \frac{1}{2}\frac{4}{3}\pi 0.5^3$ is the added mass of a sphere, see section 1.5. The second term is the drag force when the velocity is increased linearly, assuming that the drag coefficient C_d is constant. Figure 4.3(a) and figure 4.3(b) show that eq. (4.1) applies, with a shift of the time axis. For clarity, figure 4.4 shows the additional force due to added mass as a function of acceleration. The blue graph represents an analytical result, the red data points result from the accelerated simulations. This result proves that the jump of the drag force in the beginning (at $t = 0$) of the accelerated simulation is probably originating from added mass. The jump of the drag force was expected to be sharp where the acceleration starts. However, as the acceleration needs to be communicated to the environment of the sphere, some time is required to adjust to the new conditions. Therefore a smooth increase is found. The trend of both graphs is similar, and thus possibly described by the last two terms of eq.(4.1). From the results it was concluded as well, that the increase due to added mass was dependent on the acceleration. When the acceleration was larger, the added mass effect was larger (the jump in the beginning), which is consistent with theory. The deviations from the analytical result at longer times seem to be quite constant. Maybe this is related to the delay at $t = 0$, when the simulation adjusts to the accelerating conditions. If the results could be shifted in time, the graphs will overlap. Other deviations could be explained by some dependence on Z-position as will be reported in section 4.3. Actually these simulations should have been performed in larger computational domains.
Moreover, the drag coefficient is in fact dependent on the Reynolds number. For certain velocities (Reynolds numbers) the drag coefficient will drop due to the drag crisis.



(a)



(b)

Figure 4.3: The (non-dimensional) drag force of an accelerating sphere as a function of time. Panel (a) shows the results with an acceleration of $a = 1$, while panel (b) shows the results with an acceleration of $a = 5$.

However, this dip is not seen in the results of this study, figure 4.1. For similar reasons, this dip does not seem to affect the results of the accelerating sphere.

The agreement of the added mass found in these results with the theoretical value ($\frac{1}{2}m$) is surprising as the latter result follows from potential flow, which does not apply to

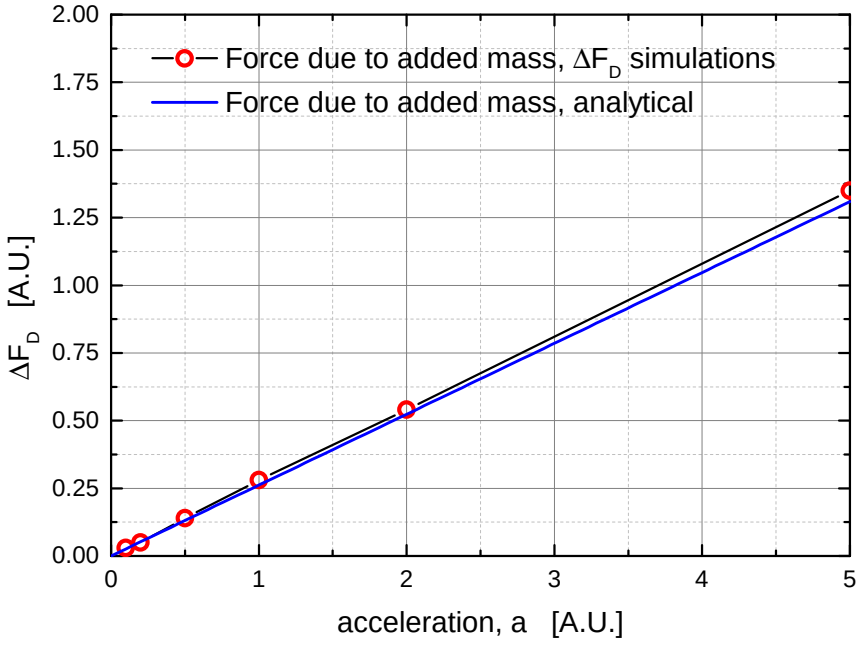


Figure 4.4: The additional force due to added mass as a function of acceleration. The blue graph represents an analytical result, the red data points the results following from the accelerated simulations.

the turbulent flows in our simulations. Interestingly, observations by Pantaleone and Messer [25] also showed that the added mass principle obtains quite good results for flows with high *Re*-values.

4.1.4 Summary results sphere

To summarize the results achieved with the sphere:

- The drag coefficient of the sphere follows the trend of the documented values quite well, except for the data points around the critical Reynolds number.

- The different kind of wakes can be distinguished by analysing the behaviour of the force components. These results were consistent with the results obtained by Sakamoto and Hanui [24].

- The additional force of the accelerating sphere due to added mass principles seems to agree quite well with the theory. At $t = 0$, when the acceleration starts, the drag force increases with a jump which equals the size of the added mass of a sphere $\frac{1}{2}m_{sphere}$ times the imposed acceleration.

To finish this paragraph, figure 4.5 shows a typical snapshot of a simulation at $Re = 10^5$. The image was obtained by processing the data with Tecplot. The (non-dimensional) pressure field of both the flow and on the sphere is visualized by the colours, with red representing the high pressure areas and with blue representing the low pressure areas. The little black vectors indicate the velocity field.
The high pressure area in front of the sphere is due to a stagnation point. The low pressure areas behind the sphere originate from vorticity. Two clear (turbulent) vortex patches are shed, but the life time of these vortices is short, they become quickly turbulent. Although the assumption was made, that the boundary layer will not be exactly solved, the separation of the boundary layer at the sphere's surface seems to be visible. It would be interesting for a follow-up study to study the relation between the Reynolds number and location of the separation at the surface of the sphere.

4.2 Drag and lift coefficient of a hand

As a next step, simulations were also performed on the flow around around a rigid hand (+ forearm). The discussion and comparison of the results of the hand are less obvious than the results of the sphere. A sphere is a sphere, but each hand model is unique and thus to be precise, each hand model has its own unique set of force coefficients. Of course, previous results give an idea of how the results should be, but comparing the values up to detail is probably nonsense. It is about the trend of the data and finding the variations in results within a model, which could lead to a better understanding of propulsion.

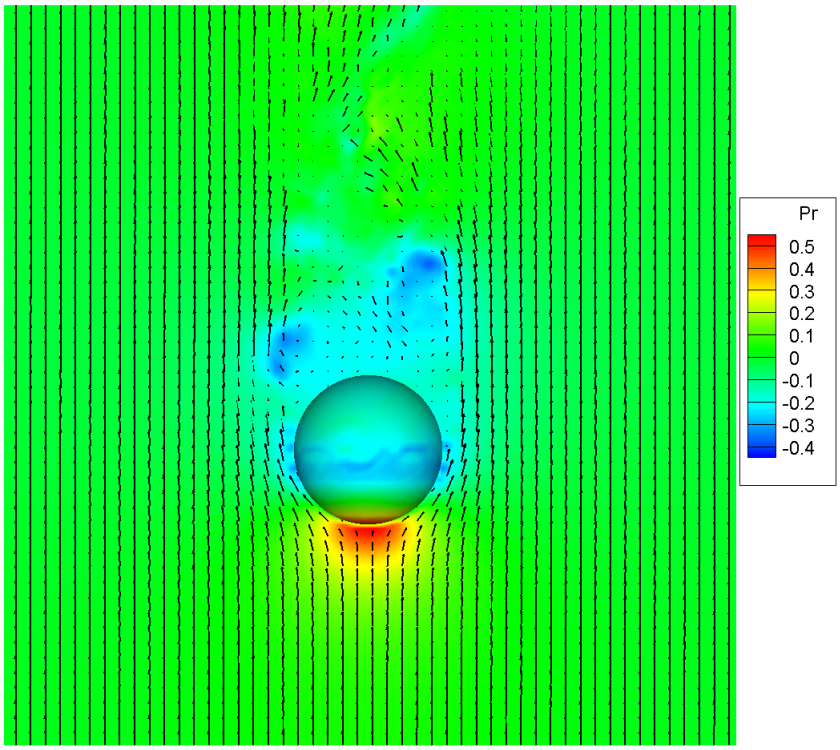


Figure 4.5: A snap shot of the flow passed a sphere at $Re = 10^5$. The colours in the back and on the sphere indicate the (non-dimensional) pressure field. Red represents high pressure areas, blue represents low pressure areas. The black vectors represent the velocity field.

4.2.1 Drag and lift coefficient of a stationary hand

First a set of simulations was performed where the force coefficient were tested as a function of velocity. The flow was simulated at a Reynolds number range of 10^5 up to $2.8 \cdot 10^5$, with increments of $2 \cdot 10^4$ for each simulations. These Reynolds numbers where chosen in order to let the velocities to be within a broad range of swimming speeds. The "swmiming" velocities ranged from approximately 0.83 m/s to 2.33 m/s, assuming a hand width (typical length) of 0.12 m. Figure 4.6 shows the results of the drag- and lift coefficients in this particular range of Reynolds numbers. The results are compared with results obtained in a previous study by Rouboa et al. [4]. The red circles and squares represent the drag and lift coefficients, respectively, of the simulations in the present study. The blue circles and squares represent the the drag and lift coefficients of the previous study by Rouboa et al. The data points from the study by Rouboa et al. were digitized using Origin9 software. Rouboa et al. used CFD code with a k-ε turbulence model and did simulations on three two-dimensional models of a right arm, a frontal model and two lateral models. The results of this study were processed similar

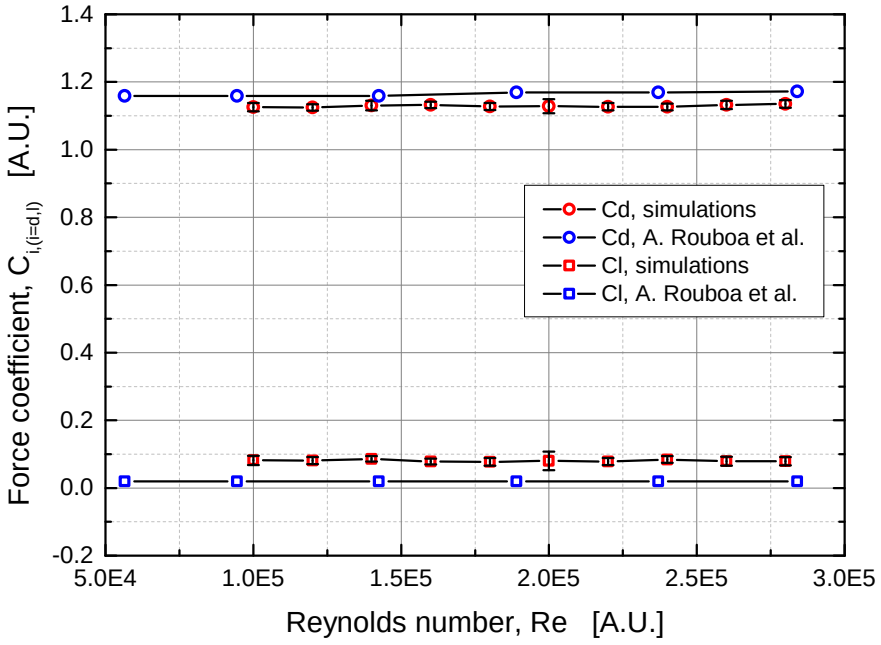


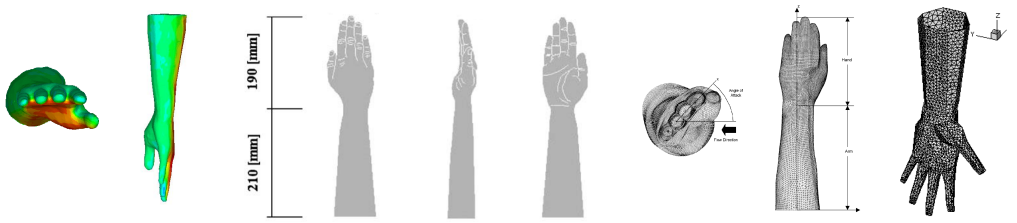
Figure 4.6: Drag and lift coefficients of a rigid stationary hand at a certain range of Reynolds numbers. The red circles and squares represent the drag and lift coefficient, respectively, of the simulations in this study. The blue circles and squares represent the the drag and lift coefficients of a previous study by Rouboa et al. [4]

to the study of Rouboa et al. The drag coefficient was calculated using the projection area of the hand at an angle of attack of 0°. The area was calculated using Matlab and turned out to be 0.029 m^2 (2.0054 in non-dimensional units). According to the results it can be concluded that the drag and lift coefficients are constant throughout this range of *Re*-values (at the particular range of swimming speeds). The drag coefficient, the coefficient accounting for "propulsion", has an averaged value of 1.128 ± 0.001, the lift coefficient an averaged value of 0.081 ± 0.001. Furthermore, the obtained results are show good agreement with the results of Rouboa et al. ($C_d = 1.16$, $C_l = 0.02$), those results were also constant throughout the whole range of velocities. Furthermore the results by Rouboa were already consistent with experimental work. Keep in mind that the values of the results will always be off, since the models used are different. There are differences in shape of the hands and the fact that the simulations by Rouboa et al. were two-dimensional is also of importance. The differences occurring because of projected area size are excluded, because the drag coefficient is in fact normalized on this surface. Effects of drag crisis (expected around approximately $\sim 3 \cdot 10^5$ compared with

a cylinder) due to boundary layers getting turbulent around the hand and fingers are not experienced. But since the data by Rouboa et al. were constant as well and already consistent with previous experimental work, it is assumed that the trend of current results is correct.

Discussion finger spreading

Figure 4.7(a) and 4.7(b) show the hand models used in the simulations of this study and for the simulations by Rouboa et al., respectively. Later on in this study the other hand models in this figure are discussed. We noticed that, the hand in present study has a



(a) (b) (c) (d)

Figure 4.7: The hand models used in different numerical studies. (a) present study (b) study by Rouboa et al. [4] (c) study by Bixler and Riewald [2] (d) study by Sato and Hino [3]

significant finger spreading, while in the hand models used by Rouboa et al. the fingers are closed. Previous studies on finger spreading have revealed that a slight spread could allow the hand to create more propulsive force. This effect will be smaller when the fingers are spread too much. Minetti et al. [6] showed that a 12° finger spreading (8mm inter-digit distance at mid finger) offered 8.8% extra propulsion with respect to a fully closed hand. A numerical study by Marinho et al. [5] showed that a small finger spread of 0.32 cm (fingertip to fingertip) offered more propulsion than the fingers closed together or with the fingers spread 0.64 cm. A study by Sidelnik and Young [7] concluded that more propulsion can be gained with a finger spread of 10°. Although there are reasonable differences between these previous studies, maybe this effect must be taken into account to analyse current results. The finger spreading in current hand model is not the same between all fingers, the inter-digit distance between the index- and mid finger is approximately $7-8$mm. Following Minetti et al. this would be the finger spread for most optimal propulsion, which means that the maximum propulsive force is already achieved concerning finger spread. Following Marinho et al. the effect of finger spread is already offset. The effects of finger spreading of this hand model are worth investigating in a future study.

Discussion thumb position

Another study by Marinho et al. [8] showed that besides fingers spreading also the orientation of the thumb compared to the hand is of importance for both the drag and lift coefficient. Marinho et al. stated that adduction of the thumb presented slightly higher values for the drag coefficient and moreover, that the position with the thumb fully abducted allowed increasing lift coefficients for angles of attack of 45° and 90°. In current model the thumb is partially abducted. Probably the thumb position of current model could be more optimal to allow the drag coefficient to increase. However at other angles of attack, the abducted thumb could be more favorable to gain higher lift coefficients. Moreover, the thumb position could probably explain the differences in ratio of drag and lift forces between different hand models.

Considering the corresponding drag forces from the results in figure 4.6, shows that the (trend of the) drag force is also in good agreement with a previous study by Sato and Hino [3]. Figure 4.8 present the results for the drag force as a function of the velocity. The red data represent the drag force resulting from the simulations. The blue data were

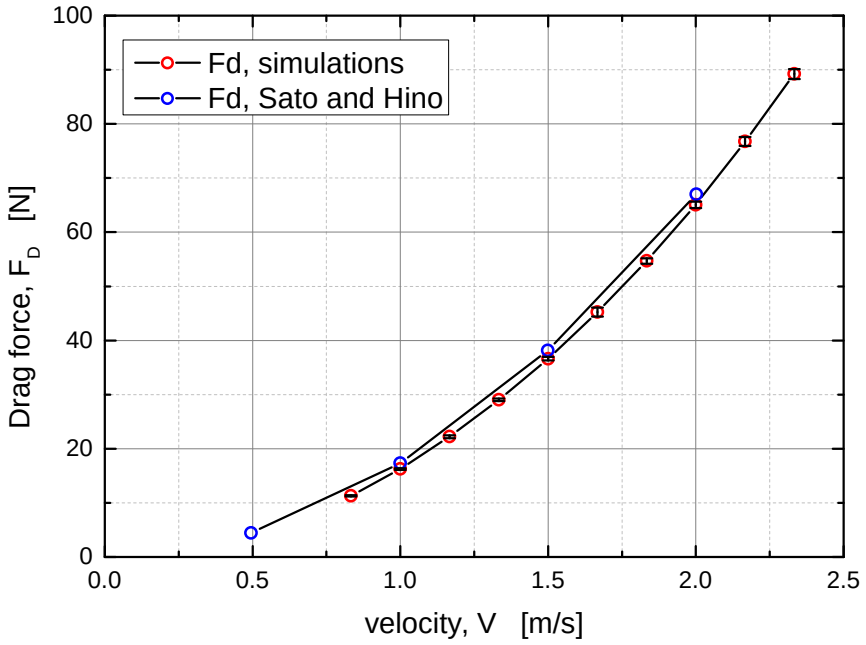


Figure 4.8: The red data points represent the drag force as a function of velocity resulting from the simulations. The blue data points the drag force as a function of velocity obtained from a study by Sato and Hino [3].

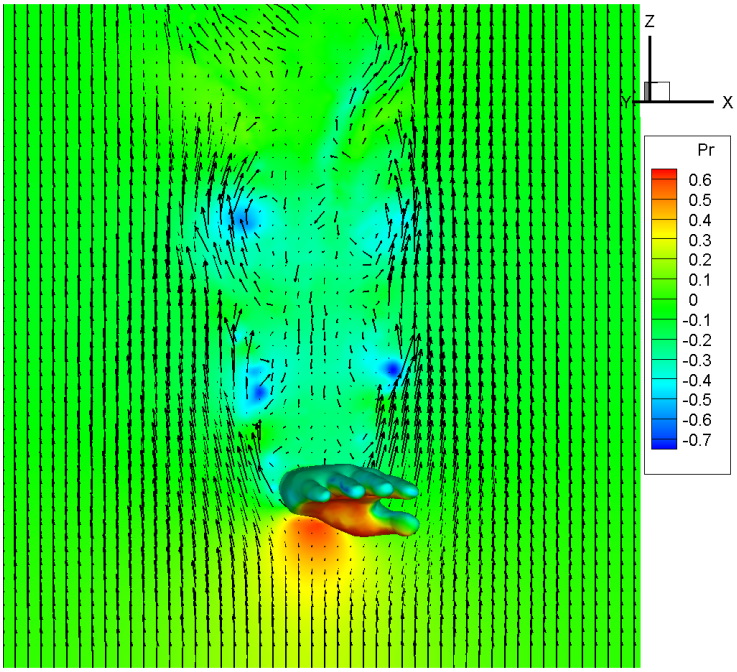
obtained by digitizing the results obtained by Sato and Hino. Of course, these data have a parabolic trend, according to (eq. (1.13)).

Snapshots simulation

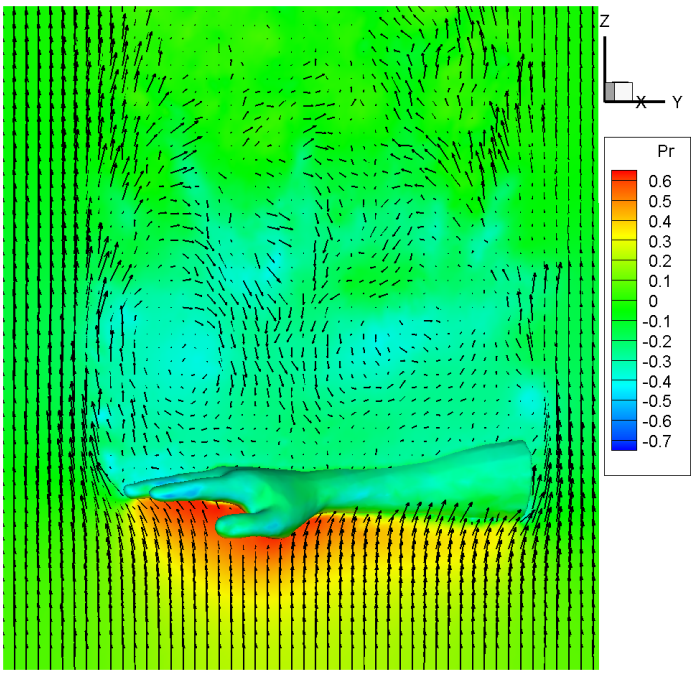
Figures 4.9(a) and 4.9(b) show snapshots of a simulation at $Re = 2.2 \cdot 10^5$ in both the XZ- and YZ-plane. The images were obtained using Tecplot. The colours indicate the (non-dimensional) pressure field on the surface of the hand and at a cross section of the flow. High pressure are coloured red, while the low pressure areas are denoted by blue. The black vectors indicate the velocity field.
The flow in the XZ-plane at the level of the wrist can be roughly approximated by the flow around a cylinder. Behind the hand a sort of Von Karman vortex street is visible. The Reynolds number indeed seems to be fairly consistent with a familiar regime (fully turbulent vortex street: $300 < Re < 3 \cdot 10^5$) for the flow around a cylinder, see Blevins [22]. In the YZ-plane a big vortex structure is visible just behind the fingers, originating when passing the finger tips. But the flow in the YZ-plane seems to become turbulent.

Also a section of the flow through at the level of the fingers is visualized, see figure 4.10. For clarity, the fingers are pointing out of the paper. The vectors indicate again the velocity field, the colours represent the pressure. The visualized flow between the fingers is interesting to observe in relation to the effect of finger spreading. With a small finger spread the boundary layer around the fingers acts as a blockage. The surface of the hand is in fact increased and this is a possibility to increase propulsion. At a certain finger spread the effect is offset and the flow can easily pass through the fingers. Possibly those effects can explain the amount of flow passing between the fingers with different inter-digit distances in figure 4.10.

The vortex generation at the fingers is clearly visible. Behind each finger two vortices are present, like the wake of a flow around a cylinder. Further downstream, the flow seems to be turbulent inside the wake, but vortices shed from the little finger and index finger seem to intact. Actually the flow around the fingers could be approximated as the flow around an array of (small) circular cylinders. And in fact, locally the flow around a finger is characterised by a smaller Reynolds number.



(a)



(b)

Figure 4.9: The flow around a hand ($Re = 2.2 \cdot 10^5$). The image was obtained by processing the data with Tecplot. Colours in the back and on the hand indicate the (non-dimensional) pressure. The black vectors indicate the velocity field. (a) XZ-section at level of the wrist (b) YZ-section

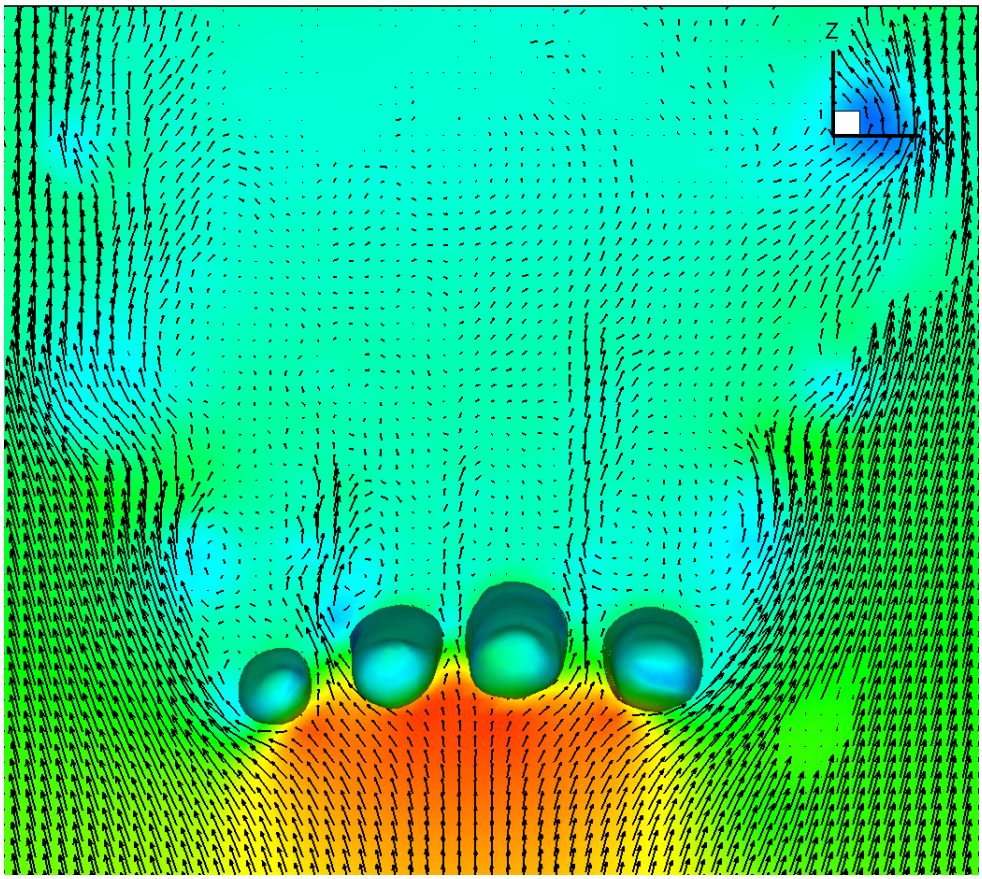


Figure 4.10: The flow at a cross section at the level of the fingers, with the fingers pointing out of the paper. Vortex generation behind the fingers is clearly visible. The vectors indicate the velocity field, colours the pressure field.

4.2.2 Drag coefficient as a function of angle of attack

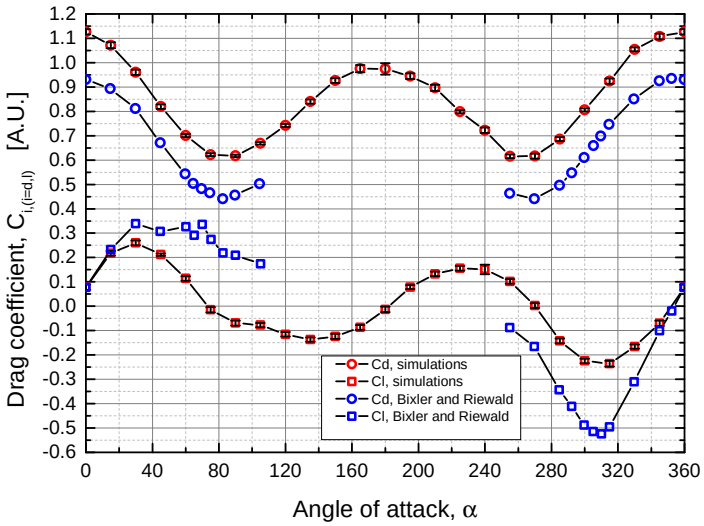
Throughout the second set of simulations the angle of attack was changed from 0° to 360° with increments of 15°. The drag and lift coefficients were calculated using the maximum projected area, which is again 0.029 m^2 (2.0054 in non-dimensional units). In the previous section it was concluded that the drag coefficient practically does not change with velocity, the velocity was now kept fixed: the flows were all simulated with the same Reynolds number $Re = 2.2 \cdot 10^5$. This Reynolds number corresponds with a velocity of 1.83 m/s, assuming the typical length of 0.12 m. Figures 4.11(a) and 4.11(b) show the result obtained in this study in red. In the first figure 4.11(a) the results are compared with previous numerical work by Bixler and Riewald, given in blue [2]. The second figure 4.11(b) shows a comparison with previous numerical work by Sato and

Hino, given in light green coloured data points [3]. The circles indicate again the drag coefficient, while the lift coefficients are denoted by squares.

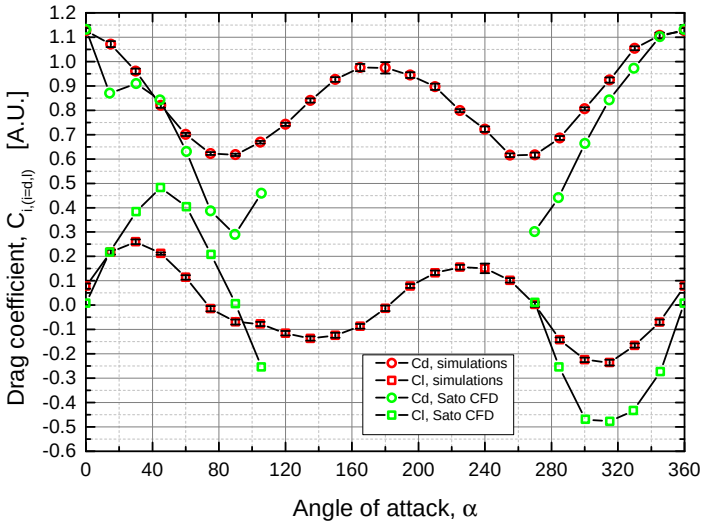
The results of previous studies were again digitized using Origin9 software. The angle of attack $\alpha = 0^\circ$ was defined differently in other studies. To compare with the results obtained in this study, there was accounted for those differences in $\alpha = 0^\circ$. The results by Bixler and Riewald were obtained by using the three-dimensional Fluent CFD code with a standard k-ε model. Sato and Hino also used a CFD code with an unsteady three-dimensional Navier-Stokes solver based on a unstructured grid. The drag coefficient measured in this study as a function of the angle of attack showed predictable trends. The results resemble the data of previous studies concerning the trend, with the peaks and lows approximately at the same angles of attack as in the previous studies shown. However, most values seem to be off a fair amount.

The peak value of the drag coefficient is 1.13 ± 0.01 and is located at $\alpha = 0^\circ$, which is consistent with the result by Sato and Hino, and Bixler and Riewald. For the sake of clarity, with this orientation the hand palm is directed towards the inlet. Another peak is visible when the back of the hand is orientated towards the inlet. The lowest value of the drag coefficient, 0.62 ± 0.01, is located at $\alpha = 255^\circ$, this is somehow with the little finger oriented towards the inlet. Also when the thumb is leading the drag coefficient has a low. The lift coefficient peaks at $\alpha = 30^\circ$ with a value of 0.26 ± 0.01 and has its minimum of -0.24 ± 0.01 on $\alpha = 315^\circ$. The location of the peaks and lows is consistent with the results of Bixler and Riewald, although the peak in the study by Bixler and Riewald is not clear. The peak of Sato and Hino seems to be at an higher angle of attack $\alpha = 45^\circ$. Moreover, the drag coefficient was dominant over the lift coefficient throughout the whole range of angles tested. This was also reported by Loebbecke and Mittal [41]. In general the drag coefficient in this study seems to be consequently higher than previous studies across the entire range and the amplitude of the sinusoidal shape trend of the lift coefficient seems to be smaller.

The discussion of these results is based on the same arguments as in previous section 4.2.1. The differences can not be explained by velocity and frontal surface differences, since the drag coefficient is normalized on these variables. It is not expected that the results will be consistent up to detail, due to the difference in hand models used. The model by Bixler and Riewald is smooth and has closed fingers (figure 4.7(c)), whereas the model of Sato and Hino is pretty coarse with the fingers wide open (figure 4.7(d)). Maybe the small finger spread in this study is more favorable to obtain higher drag forces/coefficients. And probably the thumb position could have a considerable effect on both the drag and lift force and the ratio between those forces. Again only future studies with this hand model could reveal these questions.



(a)



(b)

Figure 4.11: Drag and lift coefficient as a function of the angle of attack. The drag coefficients are indicated with circles, the lift coefficients by squares. Results of the present study are shown in red. Panel (a) shows these results together with those of Bixler and Riewald (in blue) [2]. Panel (b) shows the comparison with the results by Sato and Hino (light green) [3]

.

An addition of drag and lift forces

Studies of this kind have been arisen for the purpose of optimizing the propulsive capabilities of the swimmer's hands (and arms).

Propulsion by the legs is mainly due to lift forces, which turns out to be very effective. However, for the hand, drag forces are by far dominant compared to the lift forces under each angle of attack, as shown in previous section 4.2.2. Biggest drag-based propulsion can be obtained with an angle of attack of 0°, when the hand is perpendicular to the flow. The question is whether a resultant force of the drag and lift can obtain more propulsion than just drag-based propulsion. Non of the previous studies referred to this, although this is a big question within swimming and furthermore the actual reason to do these studies. Translating this question towards the arm movements of a swimmer, it says: Could a diagonal or curvilinear path of the arms obtain more propulsion than just pulling the arms straightly backward? The swimmer must move its hand in such a way that the resultant force (a vector addition) of drag and lift obtains the propulsion. This resultant force will not be exactly parallel to the direction of motion of the hand, like the drag force.

The maximum resultant force from drag and lift force is found by $\sqrt{F_d^2 + F_l^2}$. Figure 4.12 shows the resultant force as a function of the angle of attack. The maximum of the

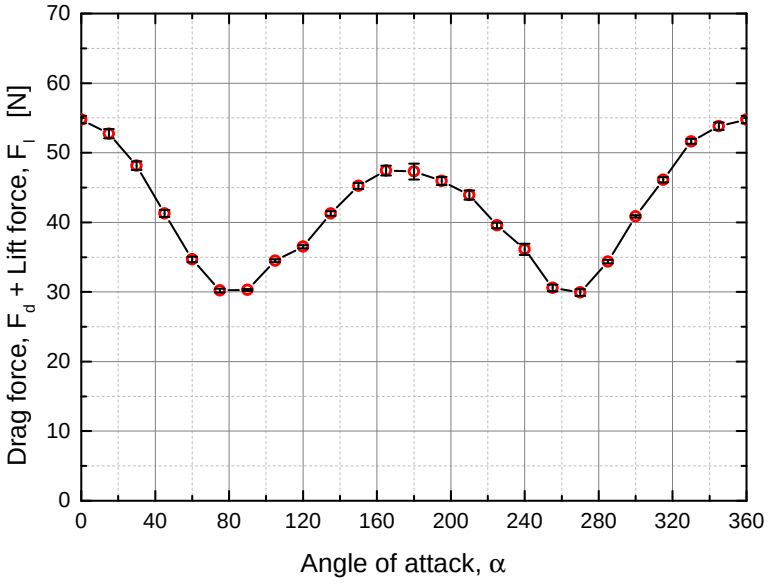


Figure 4.12: The resultant force of drag and lift. The resultant force has its maximum at $\alpha = 0°$. This force turned out to be slightly bigger than the maximum drag force.

resultant force is also found at an angle of attack of $\alpha = 0°$. The maximum resultant force is furthermore slightly higher than the maximum drag force found, 54.8 ± 0.5 N against 54.7 ± 0.5 N. It is a difference of just 0.08 N and this is considerably smaller than the error margins of the forces. Thus it can not be stated that there is a significant difference. But it can be stated that the resultant force is always bigger than the drag force on its own, just looking at vector additions. To use this resultant force as a propulsive force, the arms must not be pulled straightly backward, but slightly diagonally with an angle of $\theta = 3.05°$, as shown in figure 4.13. Still the hand palm must be used to push against the water, the angle of attack (compared with direction of hand movement) must be 0°). Of course it is not doable for a swimmer to adjust for an angle of just 3°, to

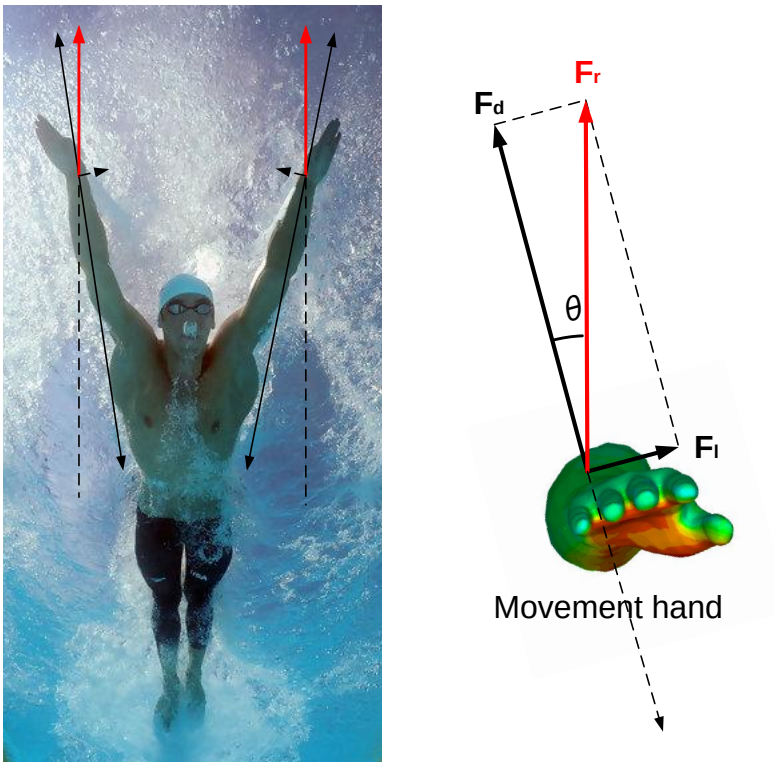


Figure 4.13: The resultant force used for propulsion. The movement of the arms must be slightly diagonal (3°) with the hand palm leading (angle of attack $\alpha = 0°$).

gain 0.08 N extra propulsion. Furthermore it must be noted that these results are valid for this particular hand model. Every hand is unique, drag and lift coefficient could be slightly different for other hands. Therefore the angle θ could be slightly different as well and it is even possible that the maximum resultant force is at a different angle of attack.

4.2.3 An accelerating hand

Like the simulations of the sphere, first a reference simulation was performed with the initial conditions for the follow-up runs. The Reynolds number of the initial flow was set on $2.2 \cdot 10^5$, so the initial velocity is 1.833 m/s. Acceleration of $a = 0.5, 1, 2, 3, 4$ and 5 m/s^2 were imposed. Figures 4.14(a) and 4.14(b) show some results obtained when accelerating the hand. The red graph represents the results for the drag force on the hand, following from the simulations. The blue graph represents an analytical result of the total force without an added mass term, see eq. (4.1). The added mass effect is less obvious than the results of the sphere, because the imposed (non-dimensional) accelerations set in the numerical scheme were much smaller. However, the jump in the drag force at the start of the acceleration originates from added mass effects. Further down the graph the difference is increasing, while it was expected to be constant. However, the drag coefficient of the hand turned out be strongly dependent on the position in the domain, as will be reported in section 4.3. Actually, these simulations had to be performed in larger domains, to obtain better results.

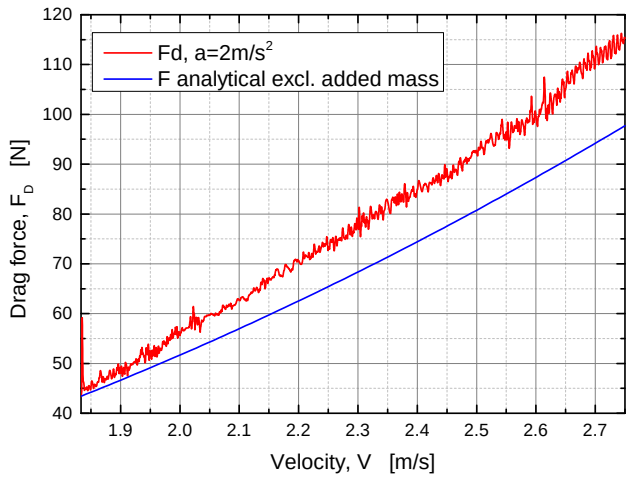
The difference in force was roughly picked by eye and gained surprisingly good results for the added mass by using the relation $F = ma$. The added mass of a hand turned out to be ~ 0.7 kg with accelerations of $a = 2, 3, 4$ and 5 m/s^2. At accelerations of 0.5 and 1 m/s^2 the differences between the graphs were overruled by the fluctuations of the force. The real mass of a similar hand (+forearm) is approximately $\sim 0.9 - 1$ kg (assuming $\rho = 1$), thus the added mass is approximately $7/10$ of the total mass of the hand and forearm. For comparison: The added mass of a sphere is $1/2$ of the total mass of fluid displaced by the sphere. More tests must be performed to validate this result, at least the order of magnitude seems to be quite well.

From these results it appears that the added mass concept also approximately applies to the accelerating hand, with a force dependence as

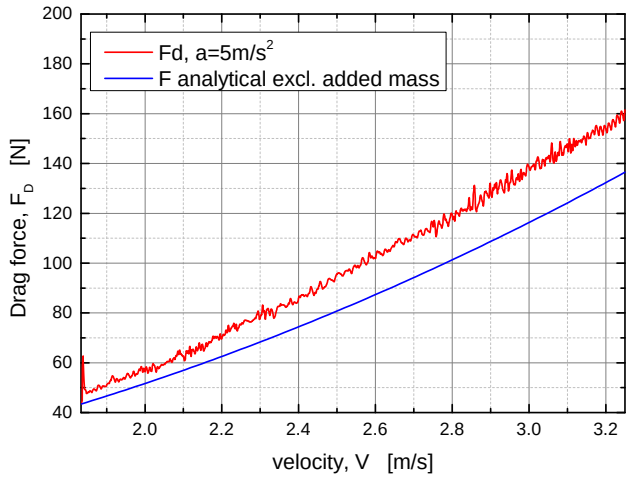
$$F(t) = am_a + \frac{1}{2} A \rho v(t)^2 C_{d,stationary}, \tag{4.2}$$

where $m_a = C_a \rho V_{hand}$ is the added mass of the hand, with C_a a constant indicating the ratio of added mass with respect to the total mass of the hand. An interesting question now is about the optimization of the trajectory of the hand and arms that maximizes trust and minimizes energy.

Despite the fact that higher velocities (and forces) are obtained faster with larger accelerations, the accelerations do not seem to have an additional, significant advantage. Just at the start of an acceleration a significant difference is present, due to the added mass. If the behaviour of the drag force is really as reported in eq. (4.1), it can be concluded that from a certain velocity the effect of added mass is not significant anymore. The added mass is constant during the whole acceleration, while the other contribution of



(a)



(b)

Figure 4.14: The drag force of an accelerating hand as a function of velocity. Panel (a) shows the results with an acceleration of $a = 2$ m/s^2. While panel (b) shows the results at an acceleration of $a = 5$ m/s^2.

the force increases with the velocity squared. Besides that, does added mass really contribute in the propulsive forces anyway or is it just of importance regarding resistance? Previous studies mentioned added mass just with regard to additional resistance. For a future study it would be interesting to study the added mass effect in more detail as well

as the differences between dynamic and static drag.

4.2.4 Summary results hand

To summarize the results achieved with the stationary hand:

- The drag and lift coefficient as a function of Reynolds number (or velocity) seems to be constant in the range of simulations done. Moreover, our results show good agreement with results of previous studies. The values of the drag and lift coefficients are 1.128 ± 0.001 and 0.081 ± 0.001, respectively.
- When changing the angle of attack, the trend of the data follows the expectations. The maximum value of the drag force 1.13 ± 0.01 was obtained at an angle of attack of $\alpha = 0°$. The maximum lift force (0.26 ± 0.01) was obtained at an angle of attack of $\alpha = 30°$. Moreover, the drag was dominant throughout the whole range of angles tested.
- The resultant force of drag and lift has a maximum of 54.8 ± 0.5N at an angle of attack of $0°$. This force is 0.08N higher than the drag force, this is not a significant difference. However, by pulling the arms $3°$ diagonally backward this resultant force can be used for propulsion.
- The added mass of a hand and forearm turned out to be $\sim \frac{7}{10}$ of the total mass of the hand and forearm. More tests must be performed to validate this result.

4.3 General considerations

Some general considerations about domain size, dependency on Z-position and influences of the wall which could have affected the results.

4.3.1 Effect of domain size

The effect of the domain size and the grid size on the drag coefficient was briefly investigated by some additional simulations on a sphere. Runs in domains sized $4 \times 4 \times 4$ and $6 \times 6 \times 6$ with equal grid size were performed. Table 4.2 shows the results. The domain size did not affect the drag coefficient of the sphere, the different results are even within their error margins. The grid size seems to have its effect on the results, but the drag coefficient was even larger than expected. However, to avoid lack of time issues, the bulk of the simulations was not performed with finer meshes and larger domains. A three-dimensional LES simulation of domain size $5 \times 5 \times 5$, grid size 0.025 for 100 simulation seconds took already more than 48 hours on 4 processors.

Table 4.2: The effect of domain size and grid size on the drag coefficient.

Re	nodes	domain size	grid size	drag coefficient
$2 \cdot 10^5$	161x161x161	4x4x4	0.025	0.48 ± 0.02
	201x201x201	5x5x5	0.025	0.49 ± 0.02
	241x241x241	6x6x6	0.025	0.48 ± 0.02
	251x251x251	5x5x5	0.02	0.51 ± 0.01
$4 \cdot 10^5$	161x161x161	4x4x4	0.025	0.49 ± 0.02
	201x201x201	5x5x5	0.025	0.49 ± 0.02
	241x241x241	6x6x6	0.025	0.48 ± 0.01
	251x251x251	5x5x5	0.02	0.51 ± 0.02

4.3.2 Dependency on Z-position

Although the results of a stationary at different velocities fitted the previous research pretty well, the drag coefficient turned out to be dependent on the Z-position. Some additional runs were done to figure this out. Results are shown in table 4.3. It was

Table 4.3: The effect of Z-position on the drag and lift coefficient. $z = 0$ is inlet, $z = 5$ is outlet

Re	Z-position	drag coefficient	lift coefficient
$2 \cdot 10^5$	1	1.46 ± 0.02	0.11 ± 0.02
	1.5	1.13 ± 0.01	0.08 ± 0.01
	2	1.01 ± 0.02	0.07 ± 0.01
	2.5	0.99 ± 0.06	0.07 ± 0.03
	3.5	1.11 ± 0.21	0.07 ± 0.13

expected that the drag coefficient would be much higher when the hand was placed closed to the inlet. Actually this is a non-physical situation. The inlet velocity is forced to be 1 and then rapidly forced to flow around the body. In reality the flow would adjust more gradually. After this initial jump, the mean force shows a slight variation with Z- position. However, this variation is comparable with the size of the temporal fluctuations.

Comparing the reference simulation of the accelerating runs with the stationary runs of a sphere, showed that also the drag of the sphere is dependent on Z-position. Actually these results had to be gained in advance of all simulations. And the simulations had to be performed in bigger computational domains.

4.3.3 Influences of the wall

The numerical configurations of the flow domain and the boundary conditions was completely different for different studies. Most likely the flow around the hand in present study experiences influences of the walls, especially in the Y-direction and from the inlet. Therefore a brief consideration of the influences of the walls is done. The size of this domain was a compromise between the simulation time, available memory and size of the wake visible. The domain could not be chosen much larger and the mesh could not be chosen finer to avoid memory and time issues. Free-slip walls do not add any vorticity in the flow, but a non-penetrating condition applies (at the walls on $y = -2.5$ and $y = 2.5$ $v_y = 0$), forcing the fluid to be inside the boundaries. For the sake of mass conservation, extra momentum is added to the flow, when the y-components of the velocity are forced to be zero, the x- and z-components will increase. This additional momentum causes the drag coefficient to be slightly lower. The periodic walls on $x = -2.5$ and $x = 2.5$ will not affect the results, the wakes of all simulations are well within the boundaries and thus would not be influenced by the wakes of "neighbouring" hands. Like at the free-slip walls the velocity at the inlet is forced to satisfy a specific condition, i.e. $v_z = 1$. When the body is placed too close to the inlet, the solution might be non-physical. The flow should be given some distance to adjust for the presence of the object. Otherwise extreme adjustments occur, leading to non-physical pressure distributions and velocity distributions, causing the drag to be higher.

4.4 Calculation of forces from surrounding flow

The few results obtained for the calculation of forces, using conservation of momentum (eq. (1.8)), from the surrounding flow are presented in table 4.4. The corresponding forces resulting from the simulations are also presented in this table. The differences

Table 4.4: Force calculations from surrounding flow based on conservation of momentum compared with force output of the IB code.

Re	**method**	F_x	F_y	F_z
$1 \cdot 10^5$	output IB code	0.084	0.101	1.129
	conservation of momentum	0.129	0.091	1.264
$2.2 \cdot 10^5$	output IB code	0.069	0.099	1.108
	conservation of momentum	0.056	0.110	1.352
$2.8 \cdot 10^5$	output IB code	0.218	0.077	1.068
	conservation of momentum	0.217	0.064	1.116

between these forces is due to the neglect of fluid velocities inside the object. Also the neglect of viscous forces (i.e. turbulent viscosity) on the boundary of the domain could

affect the results. However, the trend and order of magnitude of these data do already agree up to a certain extent.

Unfortunately, the attempt to calculate the forces on the immersed body following Graziani's method [12] was unsuccessful. The different terms in eq. (1.12) must be studied in more detail. A future study of this method is still of interest, because it would be very useful in experimental studies of swimming.

Conclusion

Validation of the immersed boundary code was made by performing simulations of the flow around a sphere. The drag coefficient of the sphere followed the trend of documented values quite well, except for data points around the critical Reynolds number. The fluctuations of the force components were analysed and a classification of different types of wake flows was made. The (Reynolds number) regimes of those wakes were consistent with previous research by Sakamoto et al. [24]. Furthermore the results of the accelerated spheres were quite consistent with the added mass principle. These arguments strengthened the confidence in the proper working of the numerical code, so that we could continue with the study of a swimmer's hand.

It can be concluded from this study that the drag and lift coefficients, of the particular hand model that was used, are constant throughout a certain range of velocities ($10^5 \leq Re \leq 2.8 \cdot 10^5$), namely $C_d = 1.128 \pm 0.001$ and $C_l = 0.081 \pm 0.001$. This result is similar to what was found in previous research by Rouboa et al. [4]. By varying the angle of attack it was concluded that the largest drag force is obtained while pushing the hand palm exactly against the water. The drag coefficient had its maximum at angle of attack $\alpha = 0^\circ$, $C_d = 1.128 \pm 0.001$. The lift coefficient had its maximum at 30° with $C_l = 0.260 \pm 0.010$. Although, the trend (sinusoidal graph) of these data was similar to the trend observed in previous research, the actual values show a difference. Moreover, the drag coefficient was dominant over the lift coefficient throughout the whole range of angles of attack tested. This was also reported by Loebbecke and Mittal [41]. From these results we suggest that swimmers to pull their arms backwards in a straight line with the hand palms exactly facing the flow.
A resultant force of the drag and lift force is always larger than just the drag force. By analysing the resultant force it was found that pulling the arms $\sim 3^\circ$ diagonally backward, with an angle of attack of $\alpha = 0^\circ$ could gain 0.08 N more propulsive force (total propulsive force 54.76 N). This result is not significant. For accelerating hands it was found that the added mass is approximately $\sim 7/10$ of the total mass of the hand.
Besides that higher velocities, and thus higher forces, are obtained faster with higher accelerations, the acceleration does not seem to have additional, significant advantages on the drag force.

Throughout this study it became clearly that the comparison with other research (using other hand models) is probably less relevant than analysing the results of one unique hand. Of course, the outcome of other research provides an approximate reference and trends can be compared, but it is probably not of importance to compare the values up to detail. There will always be differences in comparing hand models. Future studies on finger spreading, thumb position etcetera of this particular hand would give a more complete picture to search and understand the highest possible propulsive force. In reality the hands of different swimmers will have a slightly different set of force coefficients as well. Studies like the present one are aimed at better understanding the trend of the data and finding variations of importance, which could lead to more propulsion.

Finding the optimum propulsive force in a physicist point of view, does not have to be the perfect propulsion for a swimmer. There is much more to be taken into account. When swimming at a certain velocity, the swimmer will experience drag (in terms of resistance) and the swimmer will generate propulsive forces. These propulsive forces are dependent on muscular power as well. Furthermore, a swimmer will make conscious and unconscious adjustments with the body, thus affecting the flow. All of this has to be taken into account when searching for optimal propulsion. And there is more, for example: What is energetically the best propulsion for a swimmer? A sprinter would be less worried about this compared to a long-distance swimmer. And what is possible regarding human movements? These are just a few examples out of many aspects to be considered as well. Basically this kind or research helps us in understanding the basics of propulsion in water. Results can be taken into account as background information to adjust swimming techniques.

In retrospect, concerning the approach and methods, I would have taken a different approach. More tests should have been done in advance to figure out the dependency on grid size, domain size and position etc. In addition, I have some doubts concerning the advantage of using an IBM. Although lots of computational efforts are saved by working with fixed grids, the computational efforts add up fast when performing three-dimensional (IB) simulations. The mesh size is constant throughout the entire computational domain. Concerning a flow around a complex body (for example the hand), the mesh size must be chosen small enough, resulting in a fine mesh throughout the whole domain, also in areas where it is not necessary. If more detailed results (finer meshes) or larger computational domains are required, I would suggest to perform the simulations on computers with more computational power. Concerning the IBM, maybe alternative methods of IB are more useful for this study. There exist IBMs were unstructured meshes are used, with the finer grids close to the body. But in my opinion this is a solution in between DNS and IB. It offsets the essence of a purely IBM of the separate fixed (cartesian) grid for the flow domain and the surface grid of the immersed body, as the grids are not be recalculated in each computational step. Applying an “unstructured”

grid in IB probably means that the grid must be recalculated in each computational step when moving the immersed body, despite the fact that the grid is not conformal with the body. In that case, the profit concerning computational times and efforts compared with purely IB will be limited.

Acknowledgement

Since I am an avid swimmer myself, I couldn't wish myself a better graduation project to work on. Other research in this swim project was done by the graduate Gijs Custers, on PIV/Optical flow techniques and towing experiments. I would like to thank him for the nice cooperation during this year. I would like to express my gratitude to Willem van de Water, GertJan van Heijst and Herman Clercx for supervising me during this project.

This project originated from a collaboration between the research group Turbulence and vortex dynamics from the University of Technology in Eindhoven and InnoSportLab[1] also located in Eindhoven. At InnoSportLab in 'De Tongelreep' research is done on professional swimmers. Together with my supervisors I would like to thank InnoSportLab for making this project possible.

Special acknowledge goes out to Roberto Verzicco from the University 'Tor Vergata' in Rome and Marco de Tullio from the University of Bari for the development of the code and giving the support and advice needed. Furthermore thanks for the hospitality to receive me in Rome to work on the code.

Furthermore I would like to thank Jelle van der Horst (aka Salmojella), Paul Cardinaels (aka Priest), Vitor Marques Rosas Fernandes (aka Shotgun), Neehar Moharana (aka Chicken), Ivo van Hooijdonk (aka Rooky), Michel van Hinsberg (aka Mega Michel), Dennis van der Voort (aka Mr. Swag) and again Gijs Custers (aka Robin Hood) for the memorable matches of table soccer and fun at the office in between the hours of serious working.

[1] www.innosportlabdetongelreep.nl

Appendix A

Results: Flow regimes sphere

The associated figures with the flow regimes stated in section 4.1.2. The development in time of the forces in respectively the X-, Y- and Z-directions is shown in the figures below. Both forces and time are non-dimensional, as they were computed in the code. The red coloured graph shows the results of the gravitational forces. Gravitational effects were turned off in the simulations, thus gravitational forces are zero. The (dark) blue graph indicates the viscous forces (eq.(1.10)). The (dark) green graph indicates the pressure force (eq.1.9). The turquoise graph the total forces, an addition of viscous and pressure forces.

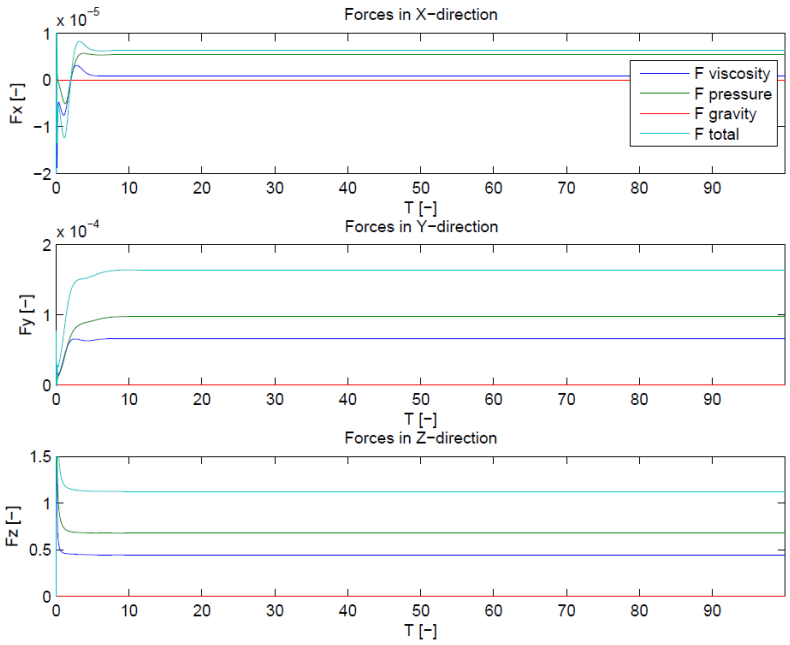


Figure A.1: range: $Re \leq 100$, graph: $Re = 20$

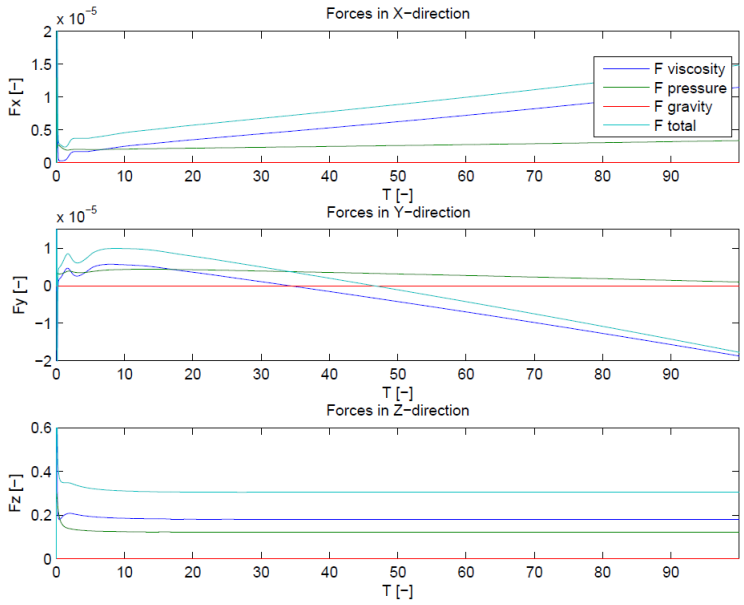


Figure A.2: range: $Re = 200$, graph: $Re = 200$

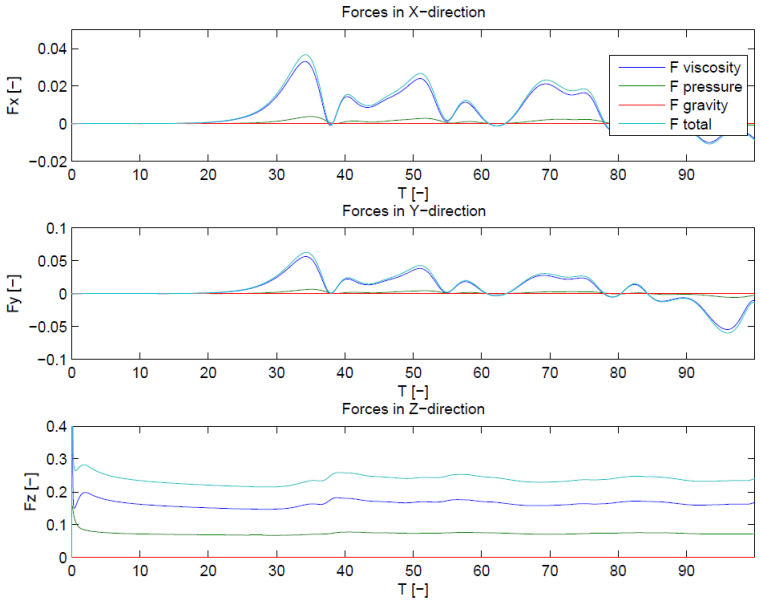


Figure A.3: range: $Re = 400$, graph: $Re = 400$

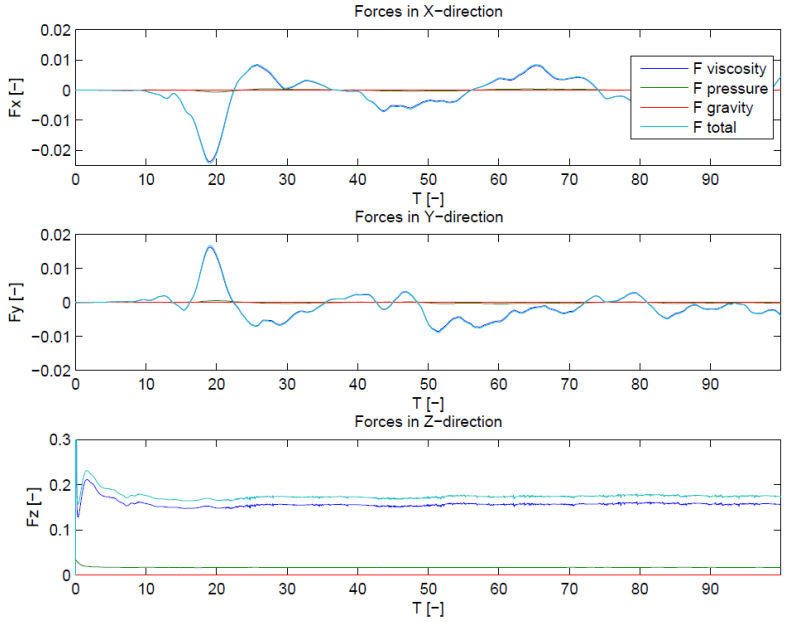


Figure A.4: range: $1000 \leq Re \leq 4000$, graph: $Re = 2000$

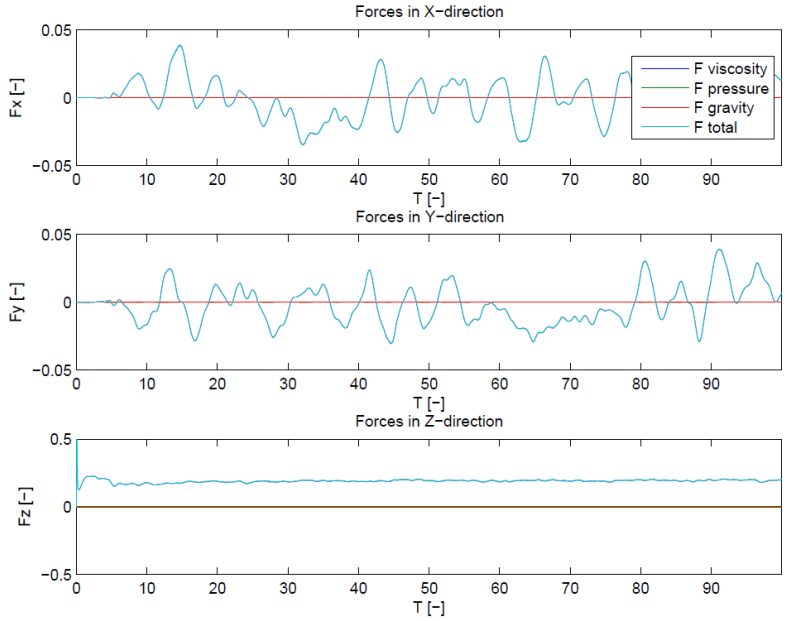


Figure A.5: range: $Re \geq 10000$, graph: $Re = 100000$

Bibliography

[1] M. A. M. Berger, G. de Groot, A. P. Hollander. Hydrodynamic drag and lift forces on human hand arm models. *Journal of Biomechanics*, 28(2):125–133, 1995.

[2] B. Bixler, S. Riewald. Analysis of a swimmer's hand and arm in steady flow conditions using computational fluid dynamics. *Journal of Biomechanics*, 35:713–717, 2002.

[3] Y. Sato, T. Hino. Estimation of thrust of swimmers's hand using CFD. *Proceedings of second international symposium on aqua biomechanisms*, 2003.

[4] A. Rouboa, A. Silva, L. Leal, J. Rocha, F. Alves. The effect of swimmers's hand/-forearm accelerations on propulsive forces generation using computational fluid dynamics. *Journal of Biomechanics*, 39:1239–1248, 2006.

[5] D. A. Marinho, T. M. Barbosa, V. M. Reis, P. L. Kjendlie, F. B. Alves, J. P. Villas-Boas, L. Machado, A. J. Silva, A. I. Rouboa. Swimming propulsion forces are enhanced by a small finger spread. *Journal of Applied Biomechanics*, 26:87–92, 2010.

[6] A. E. Minetti, G. Machtsiras, J. C. Masters. The optimum finger spacing in human swimming. *Journal of Biomechanics*, 42:2188–2190, 2009.

[7] N. O. Sidelink, B. W. Young. Optimising the freestyle swimming stroke: the effect of finger spread. *Sport Engineering*, 9:129–135, 2006.

[8] D. A. Marinho, A. I. Rouboa, F. B. Alves, J. P. Vilas-Boas, L. Machade, V. M. Reis, A. J. Silva. Hydrodynamic analysis of different thumb positions in swimming. *Journal of Sports Science and Medicine*, 8:58–66, 2009.

[9] M. D. de Tullio, A. Cristallo, E. Balaras, R. Verzicco. Direct numerical simulation of the pulsatile flow through an aortic bileaflet mechanical heart valve. *Journal of Computational Physics*, 622(1):259–290, 2009.

[10] P. K. Kundu, I. M. Cohen, D. R. Dowling. *Fluid mechanics*. Elsevier Academic Press, 5th edition, 2012.

[11] G. J. F. van Heijst. *Lecture notes, Fysica van Transportverschijnselen*. Eindhoven, Universtiy of Technology, 2010.

[12] G. Graziani, P. Bassanini. Unsteady viscous flows about bodies: Vorticity release and forces. *Mecannica*, 37:283–303, 2002.

[13] E. W. Maglischo. *Swimming fastest*. Human Kinetics, 2003.

[14] Tina Osness. Swimming Forces. *http://ffden-2.phys.uaf.edu/211.fall2000.web.projects/Tina%20Osness/Physics%20Project%20Index.html*.

[15] Fluid Dynamics of Swimming. *http://www.fi.edu/wright/again/wings.avkids.com/wings.avkids.com/Book/Sports/instructor/swimming-01.html*.

[16] F. E. Fish. Transition from drag-based to lift-based propulsion in mammalian swimming. *American Zoologist*, 36:628–641, 1996.

[17] C. M. Colwin. *Breakthrough swimming*. Human Kinetics, 2002.

[18] S. Hochstein, R. Blickhan. Vortex re-capturing and kinematics in human underwater undulatory swimming. *Human Movement Science*, 30:998–1007, 2011.

[19] R. Arellano. Vortices and propulsion. *Applied proceedings of the XVII International Symposium on Biomechanics in Sports: Swimming*, pages 53–65, 1999.

[20] R. Arellano, J. M. Terrs-Nicoli, J.M. Redondo. Fundamental hydrodynamics of swimming propulsion. *Portugese Journal of Sport Sciences*, 6(2):17–22, 2006.

[21] Tubes, Crossflow over. *http://www.thermopedia.com/content/1216/?tid=104&sn=1410*.

[22] R. D. Blevins. *Flow-Induced Vibrations*. Van Nostrand Reinhold, 1990.

[23] H. Schlichting, K. Gersten. *Boundary-layer Theory*. Springer, 2000.

[24] H. Sakamoto, H. Haniu. A study on vortex shedding from spheres in a uniform flow. *Journal of Fluids Engineering*, 112(387), 1990.

[25] J. Pantaleone, J. Messer. The added mass of a spherical projectile. *American Journal of Physics*, 79(12):1202, 2011.

[26] A. H. Techet, B. P. Epps. Derivation of added mass around a sphere. *http://web.mit.edu/2.016/www/handouts/Added_Mass_Derivation_050916.pdf*.

[27] C. Caspersen, P. A. Berthelsen, M. Eik, C. Pakozdi, P. L. Kjendlie. Added mass in human swimmers: Age and gender differences. *Journal of Biomechanics*, 43:2369–2373, 2010.

[28] J. Kim, P. Moin. Application of a Fractional-Step Method to Incompressible Navier-Stokes Equations. *Journal of Computational Physics*, 59:308–323, 1985.

[29] R. Mittal, G. Iaccarino. Immersed Boundary Methods. *Annual Review Fluid Mechanics*, 37:239–261, 2005.

[30] J. Y. Shao, C. Shu, Y. T. Chew. Development of an immersed boundary-phase field-lattice Boltzmann method for Neumann boundary condition to study contact line dynamics. *Journal of Computational Physics*, 234:8–32, 2013.

[31] M. Vanella, E. Balaras. A moving-least-square reconstruction for embedded-boundary formulation. *Journal of Computational Physics*, 228(18):6617–6628, 2009.

[32] M. Uhlmann. An immersed boundary method with direct forcing for the simulation of particulate flows. *Journal of Computational Physics*, 209(2):448–476, 2005.

[33] H. J. H. Clercx. *Lecture notes, Turbulent flow phenomena*. Eindhoven, University of Technology.

[34] S. A. Orszag. Analytical theories of turbulence. *Journal of Fluid Mechanics*, 41:363–386, 1970.

[35] R. Clark, J. Ferziger, W. Reynolds. Evaluation of subgrid-scale models using an accurately simulated turbulent flow. *Journal of Fluid Mechanics*, 91:1–16, 1979.

[36] P. Majander T. Siikonen. Evaluation of Smagorinsky-based subgrid-scale models in a finite-volume computation. *International Journal for Numerical Methods in Fluids*, 40:735–774, 2002.

[37] F. Nicoud, F. Ducros. Subgrid-scale stress modelling based on the square of the velocity gradient tensor. *Flow, Turbulence and Combustion*, 62(3):183–200, 1999.

[38] R. Verzicco. personal communication.

[39] U. M. Ascher, L. R. Petzold. *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. Society for Industrial and Applied Mathematics, 1998.

[40] R. Verzicco, P. Orlandi. A Finite-Difference Scheme for Three-Dimensional Incompressible Flows in Cylindrical Coordinates. *Journal of Computational Physics*, 123(2):402–414, 1996.

[41] A. von Loebbecke, R. Mittal. Comparative analysis of thrust production for distinct arm-pull styles in competitive swimming. *Journal of Biomechanical Engineering*, 134, 2012.