Project-oriented Software Tutorial

Project Report Flettner Rotor

Jérôme Kühn and Momme Rickmers

25.05.2020

Tutor: Marc Haussmann and Stephan Simonis

PD Dr. Gudrun Thäter, Dr. Mathias J. Krause

Institute of Applied and Numerical Mathematics (IANM)

Institute of Mechanical Process Engineering and Mechanics (MVM)

Karlsruhe Institute of Technology
## Contents

1 Introduction 3

2 Background 3
   2.1 Principles of OpenLB ............................ 3
   2.2 The Flettner Rotor ............................. 5

3 The Test Case 8
   3.1 Changing the geometry .......................... 8
   3.2 Free Flow Domain ............................... 9
   3.3 Rotating the cylinder ........................... 10

4 Simulation Results 14
   4.1 Validating the Free Flow Domain ................ 14
   4.2 Rotation of the rotor ............................ 16
   4.3 Ship calculation ............................... 20

5 Conclusion 23

6 Appendix 24
1 Introduction

Ocean transportation is an important and continuously growing sector, dependent on fossil fuels. It emits around 940 million tonnes of CO₂ annually and represents 13% of the EU greenhouse gas emissions from the transportation sector. [1] Therefore, new sustainable renewable energy technologies are emerging to help fight global climate change. A promising technology could be the use of Flettner rotors, given its simplicity, flexibility, and relatively low implementation cost. The technology can enable the ships to use an additional energy source and thus reducing its fuel consumption.

This report will first look at the history, significance and mechanism of the Flettner rotor. As well as the background of computational fluid dynamics (CFD) and the Open source Lattice Boltzmann code (OpenLB) used to simulate a representation of the Flettner Rotor. The method to create the Flettner rotor test case will be discussed and finally, the simulation results will be analyzed.

2 Background

This section will describe the principles of OpenLB. Additionally, it will give a short background on the Flettner rotor and the theory behind it.

2.1 Principles of OpenLB

The open source Lattice Boltzmann code (OpenLB) is a project providing a C++ package used for fluid flow simulation and optimization based on the Lattice Boltzmann Methods. When considering the kinetic theory and the conservation of mass of momentum, a fluid is described by using the continuity equation 2.1. This characterizes the fluid as a continuum. It shows that in a given volume the mass of a fluid can only change due to the flow into or out of the fluid.[2]

\[ \frac{\partial (\rho u)}{\partial t} + \nabla (\rho u) = 0 \] (2.1)

The second law to describe the fluid as a continuous medium is the conservation of momentum. The Euler equation 2.2 describes this phenomenon for an ideal fluid.[2]

\[ \frac{\partial (\rho u)}{\partial t} + \nabla (\rho uu) = -\nabla p + F \] (2.2)

The conservation of momentum for a real fluid can additionally be expressed by the Navier-Stokes equation. If it can be assumed that the fluid has a constant density and is, therefore, incompressible, the Navier-Stokes can be simplified to 2.3.[2]

\[ \frac{\partial u}{\partial t} + (u \nabla) u - \nu \Delta u + \nabla p = F \] (2.3)

The continuity 2.1 can also be simplified to 2.4.[2]
2 Background

\[ \nabla u = 0 \quad (2.4) \]

The Lattice Boltzmann method operates on a mesoscopic scale. It takes into account the molecular interactions and it assumes the above equations. The equation above is however completed in the macroscopic scale, where properties such as velocity, density, and pressure are of importance. Since it is not feasible to calculate the movement of every single molecule, a kinetic theory was established and takes in account the distribution of particles. The particle distribution function \( f(x, \xi, t) \) represents a generalization of the density of particles with the particle velocity \( \xi \). It also links to the macroscopic values of \( \rho \) and \( u \). The Boltzmann equation is derived by taking the total differential of the particle distribution function \( 2.5 \). [2]

\[
\left( \frac{\partial}{\partial t} + \xi \cdot \nabla + \frac{F}{m} \cdot \nabla \xi \right) f = J(f) \quad (2.5)
\]

\[
J(f) \approx -\frac{1}{\omega}(f - f^{eq}) \quad (2.6)
\]

Where \( J(f) \), is known as the collision operator. This operator can be simplified for the Lattice Boltzmann method by the BGK collision operator \( 2.6 \).[2]

With the Discrete Maxwell distribution given by equation \( 2.7 \).[2]

\[
f^{eq}(\rho, u) = \frac{\rho h^d}{(\frac{2}{3}\pi)^{\frac{d}{2}}} \exp \left( -\frac{2}{3} (\xi h - uh)^2 \right) \quad (2.7)
\]

The Boltzmann equation \( 2.5 \) can be discretized in a velocity space, physical space, and time giving the Lattice Boltzmann equation. This equation can be decomposed into two distinct parts that are performed in succession. The first part being the collision part represented by \( 2.8 \).[2]

\[
\tilde{f}_i(r, t) = f_i(r, t) - \frac{1}{3\nu + 1/2} (f_i(r, t) - f^{eq}_i(r, t)) \quad (2.8)
\]

The second part is streaming.

\[
f_i(r + h^2c_i, t + h^2) = \tilde{f}_i(r, t) \quad (2.9)
\]

In summary, the Lattice Boltzmann method begins with calculating the density \( \rho \) and the macroscopic velocity \( u \). This is used to find the discrete Maxwells distribution \( f^{eq} \) and the post-collision distribution \( \tilde{f}_i(r, t) \). After we stream the distribution function \( \tilde{f}_i(r, t) \) to the neighboring nodes by \( 2.9 \). When these two operations are complete, one-time step has completed.[2]
2 Background

2.2 The Flettner Rotor

The Flettner rotor can be defined as a device which uses the Magnus effect for an alternative type of propulsion. It was discovered in the 1920s to reduce fuel use and increase stable commercial blue water shipping.

Shipping is known to be a fossil fuel-powered and therefore unsustainable. Internationally, it is a significant contributor to airborne emissions and it shows a direct impact on global climate change and public and environmental health, including ocean warming, sea-level rise, and acidification of the world’s oceans. Due to current high oil prices and an increasing fuel cost, researchers are driving to search for energy efficiency and replacements for fossil fuel technology across this transportation sector. A technology that employs multiple propulsion technologies known as a hybrid application, either fuel with alternative energy or a combination of alternative energy, has shown to be the most appropriate for future applications. The most compelling alternative energies used for propulsion are in forms of wind, solar, bio-gas, and bio-fuel. [3]

Although the energy is not the lone measure to improve shipping efficiency. Shipping efficiency can be categorized into four approaches:

1. Operational changes for example; slow steaming, port efficiencies, weather, ...

2. Technological for example; hull design, propeller upgrade, waste heat recovery, ..

3. Alternative fuel for example; LNG, H₂, methane, biofuel, ..

4. Renewable energy for example; wind, wave, solar and various biofuels

Wind especially has been regularly discussed in the context of the renewable energy solution. Harnessing the wind power currently is split into four primary categories: soft sails, fixed-wing sails, kite sails, and rotor technology. The rotor technology is the focus of this paper and therefore the prior three categories will not be further discussed. The Flettner Rotor invented by the trained mathematician and self-taught engineer, Anton Flettner, and first patented it in 1922. The basic principle of his invention is a rotating cylinder mounted on a deck providing a propulsive energy perpendicular to the wind’s direction. This force harnessed by the rotating cylinder was discovered by the German physicist Heinrich Gustav Magnus in 1851 and it is also known as the Magnus effect. [3]

The Magnus effect is a well-known force in ball sports such as tennis, football, and golf. Its effect is seen when the ball follows a curved trajectory due to the applied spin. Due to the fact that the side of the cylinder rotating with the incoming flow, has a higher velocity than the other side, according to Bernoulli, it results in a lower pressure. Therefore, the cylinder exerts a force at a right angle to the direction of the incoming flow. [3]

Flettner described the wind fuel as “Blue Coal” and noted that billions of horsepower were cheaply available. In 1924 the 2000 ton sailing vessel, Buckau as seen in the figure 2, was retrofitted with two rotors 15m high and 3m diameter driven by a 37kW electric system to power its first voyage in 1925 across the North Sea. To provide initial rotation at around 100-400 rpm, the rotors must have external energy provided. The motors that
2 Background

Figure 1: A visualisation of the Magnus force on a rotating cylinder. [4]

spun the two towers at 120 rpm required approximately 20 horsepower and Flettner calculated that they took more than 1000 horsepower out of the wind. The vessel achieved up to 8 knots (4.12 m/s) with the rotors compared to 6.5 knots (3.34 m/s) when it was under sail. A factor that could be of concern is the additional weight of the rotors brought to the ship. However, astonishingly the weight of all additional parts (rotor towers, engine, and motors) was just one-fifth of the weight of the discarded sails and rigging. The fact that there were two rotors on board also gave an additional feature. Firstly, the ship could be propelled backward by changing the direction of the rotation, and more impressively the ship could be stopped and turned within its own length. The same vessel renamed to Baden Baden sailed to New York via South America. In this Atlantic crossing, it said that the ship used 12 tons of fuel comparing it with 45 tons for a motor ship of the same size without rotors. [3]

Another big advantage of the rotor comparing it with the sail is the stability. The rotors show no concern even in the stormiest weather. Conventional sailing ships tend to lean leeward (direction downwind) and thus at an increasing force of the wind at a certain point the ship will likely capsize. The Flettner rotor on the other hand leans into the wind providing it with this unique feature of a virtual storm proof sail system. The cylinder is said to have an ideal structural shape with barely any bending stresses thus making fatigue, not a problem. Unlike with a sail, the rotor also doesn’t need to be adjusted with the angle of the wind. No additional crew is needed to operate the rotor because there are no configurational changes in operating it. The captain can easily control the start, stop, select the direction of rotation, and the rotations per minute. [3]

Due to the global economic crisis in 1929, the fuel price became readily and cheaply available. As a consequence, many renewable energy transport technologies and innovations including the Flettner rotor were abolished and displaced by the petroleum-based approach. Anton Flettner continued his research and turned to invent the modern helicopter. However, in the 1970s, Lloyd Bergeson continued to research the potential of the rotor and the ability in fuel savings. Using the yacht Tracker, he tested it in strong winds. The results are shown in table 1. [3]

The astounding results still did not manage to establish the technology. Then as in
1929, in 1986 the oil prices fell and the rotor technology stalled until the last decade. Due to the rising fuel costs and concern on emissions now the interest has returned. The University of Flensburg is researching a small catamaran, UniKat, driven by a rotor. A simple solar cell on the UniKat is used to rotate the rotor. It has been shown that the vessel can sail as efficiently forwards as backward. E-Ship 1 is another example of a recent rotor-ship built. It was manufactured in 2009 by a Germany wind-turbine company Enercon. The ship features an innovative hull and propeller and four modern Flettner rotors with automatic control systems. The rotors are powered by the exhaust fumes of the diesel engine using a steam turbine. In a 170,000 mile journey, the ship showed to average 25% overall fuel savings. [3] [8]

It is clear now that Flettner’s design is a promising technology and it is being researched by multiple universities and the leading naval architects. The technology is shown to have its greatest application when combined as a hybrid. By combining it with other propulsion methods, either the conventional fossil fuel power or in new generation motors powered by biofuel or solar energy. [3]

<table>
<thead>
<tr>
<th>Power Mode</th>
<th>Av. Wind Speed (Knots)</th>
<th>Av. Boat Speed (Knots)</th>
<th>Av. Fuel Savings (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rotor Assist</td>
<td>16.1</td>
<td>7.0</td>
<td>44</td>
</tr>
<tr>
<td>Rotor Assist</td>
<td>12.9</td>
<td>6.0</td>
<td>27</td>
</tr>
<tr>
<td>Rotor Sailing</td>
<td>17.7</td>
<td>5.3</td>
<td>100</td>
</tr>
</tbody>
</table>

Table 1: Summary of the Trail Results of Tracker. [3]

Figure 2: One of the first Flettner Rotor ships built, the Buckau in 1920. [3]
3 The Test Case

This section describes the process of changing and developing a test case. The problem will be covered as a two-dimensional case. As the rotor can be portrayed by a cylinder in a free flow domain. Therefore, the case is quite similar to the cylinder2D case of the common exercises and thus this test case is taken as a framework and will be changed at the necessary places. This chapter shows the process of changing the case and creating the Flettner rotor as needed.

3.1 Changing the geometry

As a little reminder, the cylinder2D case is showcasing the laminar flow around a cylinder in a channel flow. The predefined channel with the height of $h = 0.41$ m and length of $l = 2.2$ m has a cylinder with radius $r = 0.05$ m positioned at $x = 0.2$ m, $y = 0.2$ m. The laminar steady state flow is characterised with a Reynolds number of $Re = 20$ and a velocity of $u = 0.2$ m/s.

The dimensions for our test case shall be the same as in the paper of [5], thus allowing comparing the results more accurately. The rotor has a radius of $r = 0.33$ m and the channel has a height of $h = 4$ m. The length is set as a combination of distances to inlet and outlet. The distance from inlet to the rotor is $w_i = 10 \cdot r$, the distance from rotor to outlet is $w_o = 20 \cdot r$ resulting in a width of $w = (10 + 2 + 20) \cdot r = 32 \cdot r = 10.56$ m. To get this geometry, changes in the parameters for the simulations are made as follows.

```cpp
const T radiusCylinder = 0.33;
const T lengthX = 32. * radiusCylinder;
const T lengthY = 4. + 2. * L;
const T centerCylinderX = 11. * radiusCylinder;
const T centerCylinderY = lengthY / 2.;
```

![Figure 3: Geometry of the test case](image)

To achieve a convergence of the much higher velocity, the resolution of the geometry has to be changed. This leads to higher computation time, but more importantly, the simulation is stable. After the geometry is changed accordingly the first validation, with
the cylinder standing still, is carried out. Therefore, a characteristic value to describe the
flow is calculated. The drag coefficient is calculated as in 3.1 and represents the force in
x-direction on the cylinder related to the inlet velocity and dimensions of the cylinder.
Even though it is a two-dimensional simulation, the voxels must have an extension in
the z-direction. This extension is also defined by the resolution and equals the length of
a voxel in x and y direction. Therefore, the overflowed area calculates as \(d \cdot \Delta x\) in the
equations.

\[
c_{\text{drag}} = \frac{F_x}{1/2 \cdot \rho \cdot U_{\infty} \cdot d \cdot \Delta x} \tag{3.1}
\]

The lift coefficient is defined analogously.

\[
c_{\text{lift}} = \frac{F_y}{1/2 \cdot \rho \cdot U_{\infty} \cdot d \cdot \Delta x} \tag{3.2}
\]

To get the force on the cylinder, the functor
\textit{SuperLatticePhysBoundaryForce2D} is used, which gets the physical force, acting on a
boundary with a given material on the local lattice. As the functor returns the force only
pointwise, a summation over the cylinder boundary is required. The class \textit{SuperSum2D}
is used to get the sum of the previously defined force.

```java
// Added BoundaryForce
SuperLatticePhysBoundaryForce2D<T, DESCRIPTOR> boundaryForce(
    sLattice, superGeometry, 5, converter);
// Sum over boundaryForce
SuperSum2D<T> sumForce( boundaryForce, superGeometry, 5 );

int input[3] = { };
T_sum[sumForce.getTargetDim()];
sumForce( _sum, input );
```

The variable _sum stores the sum of the boundary force and with the implementation
of the equation 3.1 and 3.2 the coefficients can be calculated. To get a plot of these
values, the gnu plot interface is used. Therefore, the development of the coefficients can
be watched over the start-up time and potential fluctuations afterward. (e.g. figure 15e)

### 3.2 Free Flow Domain

Including the geometry, the boundary conditions, and the inlet velocity needs to be
adjusted to achieve a more realistic domain for the Flettner rotor test case. As we want
to simulate the rotor in a free air stream, the boundary conditions were changed from
the example cylinder case. In the cylinder test case, the upper and lower walls aimed
to function as a solid wall using a bounce-back boundary condition. The bounce-back
boundary condition demonstrates a no-slip condition and is not the condition needed
for the test case. The simulation shall represent a free flow of the fluid. Therefore, a slip
boundary condition replaces is set to the upper and lower boundary former the walls.
3 The Test Case

The slip boundary allows a velocity in the x-direction at the boundary, but no flow outwards of the geometry. Like the bounce-back boundary, velocities pointing outwards of the geometry will be reflected.

Furthermore, the wind flowing against the rotor is described by the constant inlet velocity, this differs from the poisseuille profile of the cylinder2D example case. The inlet must be set accordingly to the reference. An inflow velocity of $U_\infty = 15 \text{ m/s}$ has to be set. As a default, the inflow is set as poisseuille profile. This is as well not necessary anymore, as a constant velocity over the inflow is needed. To achieve this type of inflow, a functor with a constant velocity in the x-direction is set for the inflow and then applied to the superLattice. The functor is represented by the listing below as well as the boundary condition of the walls. The smooth start-up shall be kept as it is in the cylinder example case.

```cpp
// Set boundary of upper and lower boundary to slip boundary
sBoundaryCondition.addSlipBoundary( superGeometry, 2 );
...
// set constant velocity on inlet
AnalyticalConst2D<T,T> inletVelocityStartUp( velocity );
sLattice.defineU( superGeometry, 3, inletVelocityStartUp );
```

3.3 Rotating the cylinder

The key element of the simulation and to get a fully functional Flettner rotor, it is necessary to rotate the cylinder. Therefore, it is needed to set a velocity tangential to the surface on the surface of the cylinder. The direction and value of the velocity can be calculated for any given point on the surface of the cylinder, using vector algebra. This leads to the following equation 3.3, which returns for any $x, y$ position the equivalent velocity. It is only necessary to have the center of the cylinder $(c_x, c_y)$, radius $r$ and the rotation velocity $u_{rot}$. The velocity $u_{rot}$ is the surface velocity of the cylinder.

In order to implement the rotation, a new class is set up. As mentioned, it needs a few parameters to set the velocity. These are stored as private variables. These include the rotation speed, the center of the cylinder, and the radius. The class inherits from `AnalyticalF2D < T, T >` and is also set up as a functor. To set the speed the '()' operator is overwritten and implements the equation 3.3. The actual call to set the velocity is over the `defineU` method of the `superLattice`. Giving the `superGeometry`, the material number of the cylinder, and the functor for the rotation, it sets the velocity on the surface. The function needs lattice velocities, hence the `unitConverter` is given to the `rotatingCylinder` functor to get access to the conversion function. For easier use, the velocity of the cylinder can be set as rotations per minute or meters per second. To change between the input units, a boolean variable has to be switched. The conversion between rpm and m/s can be made with the following equation 3.4.

$$u_{rot} = \frac{rpm}{60} \cdot 2 \cdot \pi \cdot r \tag{3.4}$$

It was tested to have the boundary of the cylinder as an `onLatticeBoundary` as well as an `offLatticeBoundary`. To choose between the bouzidi- and the on lattice- boundary
3 The Test Case

Figure 4: Visualisation of the tangential velocity

\[ \vec{u}(x, y) = \left( \frac{(y - c_y) \cdot u_{rot}/r}{r}, \frac{(x - c_x) \cdot u_{rot}/r}{r} \right) \] (3.3)

A boolean can be set. Theoretically, all necessary differences are implemented, though the off lattice variation does not work and leads not to a rotation of the cylinder. Therefore, all tests are made with an on lattice boundary.

```cpp
// functor for rotating the cylinder
template <typename T>
class RotatingCylinder2D : public AnalyticalF2D<T>{
    ...
    // set velocity tangential to the cylinder according to the set velocity
    bool operator()(T output[], const T input[]) override
    {
        // implementation of equations to calculate the x and y component of tangential velocity u
        output[0] = -(input[1] - center[1]) * uPhysVelocityTangential/radius;
        output[1] = (input[0] - center[0]) * uPhysVelocityTangential/radius;

        // converting the velocity from physical to lattice velocity
        output[0] = (*converter * getLatticeVelocity)(output[0]);
        output[1] = (*converter * getLatticeVelocity)(output[1]);

        return true;
    }
    ...
```

In order to have a smooth start-up with the rotation of the cylinder for stability reasons, the class has a member function to change the rotation speed. Therefore, at startup time the velocity can be increased slowly as well as the inlet velocity. The first tests of the rotation showed an artifact at the top of the cylinder, see figure 5a. This was the case because a low resolution was set. Therefore, higher resolutions are needed and a variety of different resolutions is carried out. Obviously, the higher resolution leads to increased computation time, but the artifact vanishes (figure 5b). When looking at the velocity split in x and y directions, the rotation can be seen better in figure 5c and 5d.
3 The Test Case

Since the inside of the cylinder is set to material number zero, no velocity can be seen within the cylinder.

(a) Rotation with low resolution N=15  
(b) Rotation with higher resolution N=30  
(c) Velocity in x direction N=30  
(d) Velocity in y direction N=30

Figure 5: The rotating cylinder with no inlet stream

A vast range of rotational velocities is covered within the reference paper [6], leading to rotation velocities higher than the inlet velocity. Therefore, it is necessary to change the numerical parameters to guarantee convergence of the simulation. The approach here is to change the unit converter from one using resolution and relaxation time, to one using resolution and lattice velocity. With this unit converter the lattice velocity will be changed to lower values, if the rotation speed exceeds the inlet velocity. Therefore, the numerical restrictions are always met and the simulation converges. To get the lattice velocity, the ratio of rotation speed and inlet velocity ($\text{speedRatio} = \frac{U_\infty}{u_{rot}}$) is calculated in the beginning. If the ratio is greater than one, it is set to one. The lattice velocity is then set to $\text{LatticeVelocity} = 0.1 \cdot \text{ratio}$.

```cpp
T speedRatio = inRPM ? inletVelocity / (rotationSpeed * 2. * PI * radiusCylinder / 60.) : inletVelocity / rotationSpeed;
speedRatio = speedRatio > 1. ? 1. : speedRatio;
if (!isRotating) speedRatio = 1.;
UnitConverterFromResolutionAndLatticeVelocity<T, DESCRIPTOR> const converter(
    N, // resolution: number of voxels per charPhysL
    0.1 * speedRatio, // lattice velocity has to be smaller than 0.1
    2.0 * radiusCylinder, // charPhysLength: reference length of simulation geometry
    inletVelocity, // charPhysVelocity: maximal/highest expected velocity during simulation in m / s
);```

12
3 The Test Case

\( \text{inletVelocity} \times 2 \times \text{radiusCylinder}/\text{Re}, \quad \text{physViscosity: physical kinematic viscosity in m}^2 / \text{s} \)

\( 1.0 \quad \text{physDensity: physical density in kg / m}^3 \)
4 Simulation Results

In order to get the results of the simulations, different types of methods are used. First, the implemented vtm-writer was expanded, thus outputting the force field of the cylinder as well.

```c
// part of the getResults function
// Added BoundaryForce to write in vtm Files
SuperLatticePhysBoundaryForce2D< T, DESCRIPTOR> boundaryForce( sLattice, superGeometry,
5, converter );
vtmWriter.addFunctor( boundaryForce );
```

Therefore, the force field can be visualized in post-processing using paraview. An example can be seen in figure 16. Furthermore, as already mentioned the gnuplot interface is used to plot the development of drag and lift coefficient. The actual values for the sum of boundary force, drag-, lift-coefficient, and other values defining the simulation as Reynolds number, resolution, radius, etc. are written during the simulation in a text file, to enable further post-processing of the data.

4.1 Validating the Free Flow Domain

Before beginning to work on the rotation of the cylinder it was vital to validate the code and the free flow domain. The first step of the validation was to conduct a convergence analysis. The analysis was completed by studying the drag and lift coefficient against the resolution with a range of resolution from $N = 5$ to $N = 50$. The results are shown in figure 6.

![Convergence Test](image)

(a) Drag coefficient  
(b) Lift coefficient

Figure 6: Convergence test showing the drag & lift coefficient against the resolution $N$.

It can be noted that the maximum error of the drag coefficient as a result of the resolution is less than $c_{\text{DRAG}} = 0.2$. The greatest lift coefficient is $c_{\text{LIFT}} = 0.55$. The results of the convergence study show that a resolution higher that $N = 15$, is sufficient when comparing the drag and lift coefficients. This is due to the fact that with a resolution greater than 15 the drag and lift value only alters very slightly and thus this error can be neglected. To examine the lift coefficient could be more helpful than the
4 Simulation Results

drag, because the setpoint has to be zero. As the inlet flow is solely in the x-direction and the test case is symmetric there should be no force acting the cylinder in the y-direction, meaning the lift is zero.

Continuing the validation using a resolution of \( N = 15 \) the drag coefficient was compared with the reference paper [7]. The drag coefficient was plotted against the Reynolds number and is shown in figure 7.

![Figure 7: Geometry of the test case](image)

Figure 7: Geometry of the test case

However, the drag coefficient simulated is always greater than that of the reference. This difference could be explained by the smaller total height to the diameter of the cylinder ratio (H/D ratio). Although the boundary conditions and uniform inlet velocity are equivalent, the difference in H/D was very significant. The H/D ratio used in our geometry is \( H/D = 6.1 \) whereas, Placzek [7] provides a H/D of \( H/D = 20 \). Due to the smaller ratio, the computational domain is not large enough to contain the affected region, and thus the artificial boundary conditions disturb the solution and cause an error.

Two examples of the validated results with a Reynolds number of \( Re = 60 \) and \( Re = 120 \) are shown in two post-processing tools of OpenLB. The first shows a Gnuplot of the drag coefficient against the simulation time in figure 8.

From the plot, one can see that there are no fluctuations in the drag coefficient when the Reynolds number is \( Re = 60 \). The Gnuplot also shows the smooth start of the inlet velocity by the curve of the line until time of about 16 seconds, where it reaches a constant value. Interestingly when nearing the end of the simulation of figure 8b one can see a slight fluctuation. This is due to the fact that at higher Reynolds number of \( Re > 60 \) a phenomenon of fluid dynamics occurs. Namely, vortex shedding, seen by the
repeating swirls pattern. This phenomenon occurs when a fluid flows past a bluff at a certain Reynolds number. The velocity profiles of this feature can be seen in figure 9. The profiles are shown using Paraview a different post-processing tool. One can see that the vortex shredding appears on the figure 9b and in comparison one without shredding at a Reynolds number of Re 60 (figure 9a).

4.2 Rotation of the rotor

The reference paper [6] gives some orientation for what values in rotation speed and inlet velocity should be set to compare. First, the inlet velocity is set to $u_\infty = 15 \text{ m/s}$ and the rotation speed is in the range of $u_{rot} \in [25, 2700] \text{ rpm}$ equaling surface velocities of $u_{rot} \in [0.864, 93.305] \text{ m/s}$. The Reynolds number is set to $Re = 20$ for the simulations even though it is not given, what the reference paper uses. Therefore, it is not quite possible to use the values of the paper to validate the results of the simulation. However, the results are plotted over the velocity ratio of $\alpha$ defined as in equation 4.1.

$$\alpha = \frac{u_{rot}}{U_\infty} \quad (4.1)$$

As can be seen in Figure 10 the simulation and the reference are very different. The drag coefficient in 10a drops continuously from 2.70 to -0.51 and then starts rising again. The reference is only in an area $\alpha \in [0.01, 1.08]$, where it drops 0.925 to 0.578. The lift coefficient is dropping steadily with the rising velocity ratio in the simulation. Starting at nearly -0.18 and ending at -34.41, thus being up to 2.5 times higher than the reference.

Because the values are far from the reference, the idea was to change the Reynolds number. As already mentioned, there was none given. And as the lower drag values suggest with low rotation speeds the Reynolds number could be greater, as the drag coefficient drops with rising Reynolds number (see chapter 4.1). Therefore, the Reynolds number is set in the following simulations to $Re = 100$. This leads also to the develop-
4 Simulation Results

(a) Reynolds = 60.

(b) Reynolds = 120.

Figure 9: Velocity profiles of different Reynolds number with a standing cylinder.

...ment of a kármán vortex behind the cylinder. Thus, the drag and lift coefficient starts to oscillate. As no fixed value can develop, minimal, maximal, and mean values are determined in post-processing. Results are shown in figure 11. The drag coefficient is now lower than with low Reynolds number and has a similar development with a rising velocity ratio. At a ratio of $\alpha > 1.8$ the kármán vortex disappears therefore minimal, maximal, and mean value of lift and drag are the same. Before it can oscillate with up to 17% around the mean value. However, the range is still far from the reference. As mentioned, the reference covers only a range of velocity ratios $\alpha \in [0.01, 1.08]$ meaning with the given rotation speed either the inlet velocity is changed or the radius of the cylinder is smaller. First, the inlet velocity will be changed. Afterward, a test with a smaller cylinder will be made.

With equation 4.1 a inlet velocity of $U_\infty = 86.6 \, \text{m/s}$ would represent the reference data. The simulations run with a Reynolds number of $Re = 20$ and $Re = 100$, so it is possible to compare with the previous results. In figure 12 results are shown with $Re = 20$. Drag and lift coefficients are the same as with the same Reynolds number but lower inlet velocity. If Reynolds increased to $Re = 100$ the results of the drag coefficient are higher than the one with lower inlet velocity. Also, the amplitude of the oscillation is increased. However, the lift oscillates around the same values as with lower inlet velocity.

That the charts for the same Reynolds numbers but different inlet velocities are quite similar can be seen in figure 14. For low Reynolds numbers, the results are almost the same, whereas for high Reynolds numbers the oscillation results in a bit different values,
4 Simulation Results

![Figure 10](image1.png)

(a) Drag coefficient over velocity ratio  
(b) Lift coefficient over velocity ratio  

**Figure 10:** Comparison of simulation results and reference $Re = 20$

![Figure 11](image2.png)

(a) Drag coefficient over velocity ratio  
(b) Lift coefficient over velocity ratio  

**Figure 11:** Comparison of simulation results and reference $Re = 100$

especially for the drag values. The mean values of the lift coefficient for high Reynolds numbers are oscillating around the same value. Furthermore, the lift shows a linear behavior with low-velocity ratios.

To end the simulation part, some tests are summarized in table 2. A variety of different resolutions is carried out. This is made with only one rotation speed each, as computation time is highly increased. Furthermore, a simulation with a reduced radius is made. This leads also to an increased resolution, as it is defined as voxels per diameter. Therefore, if the diameter is decreased but the amount of voxels per diameter is kept, the overall resolution rises. For comparison, the values of the previous simulations with similar rotation speed are listed as well. Although there are some differences between the different resolutions, the values only differ within a few percent at maximum. The only big difference is with high Reynolds number and inlet velocity. The mean values between the resolutions $N = 30$ and $N = 50$ differ up to 22%. However, these are the mean values in the oscillating stream. Looking also on the min, max, and the standard deviation the values are in the same range. Therefore, the used resolution of $N = 30$ for the majority of the simulations is a good compromise between accuracy and time efficiency. The simulation with smaller radius results in slightly lower values than with
4 Simulation Results

Figure 12: Comparison of simulation results and reference $Re = 20, U_\infty = 86.4 \, m/s$

Figure 13: Comparison of simulation results and reference $Re = 100, U_\infty = 86.4 \, m/s$

the bigger radius with similar stream characteristics. This could be a result of reduced boundary influences, as the walls are further away from the cylinder. Therefore, a second simulation is run, with the same channel height to radius ratio as the one with $r = 0.33$ (marked with *). The drag coefficient is then the same, but the lift coefficient is still lower. However, this proves the influence of the boundaries, as they influence the stream and therefore the drag and lift force on the cylinder. For perfect results, the distance from walls to the cylinder would have to be infinite. For the simulation, the influence should be tested and a compromise between accuracy and time efficiency should be chosen for the distance of the walls. As the reference paper gave a specific ratio for channel height to the radius, it was kept the same for comparing reasons, but the run with higher height to radius ratio shows, that there is still quite an influence on the results.

For some simulation states pictures were added in the appendix. For visualization in figure 15 the rotation start up was shifted to the inlet start up. Therefore, it can be seen how the lift force develops as the rotor starts rotating. As the simulation is run with $Re = 100$ the values are oscillating as result of the kármán vortex. The magnus effect can be easily observed in the pressure and velocity fields of figures 15b and 15d. As the velocity of the stream gets reduced at the top of the cylinder and increased at
4 Simulation Results

(a) Drag coefficient over velocity ratio    (b) Lift coefficient over velocity ratio

Figure 14: Comparison of simulation results and reference

<table>
<thead>
<tr>
<th>Radius</th>
<th>Resolution</th>
<th>Reynolds number</th>
<th>Inlet velocity</th>
<th>velocity ratio</th>
<th>$c_{drag}$</th>
<th>$c_{lift}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.33</td>
<td>15</td>
<td>20</td>
<td>15</td>
<td>0.46</td>
<td>2.66</td>
<td>-1.41</td>
</tr>
<tr>
<td>0.33</td>
<td>30</td>
<td>20</td>
<td>15</td>
<td>0.46</td>
<td>2.66</td>
<td>-1.39</td>
</tr>
<tr>
<td>0.33</td>
<td>50</td>
<td>20</td>
<td>15</td>
<td>0.46</td>
<td>2.68</td>
<td>-1.39</td>
</tr>
<tr>
<td>0.33</td>
<td>100</td>
<td>20</td>
<td>15</td>
<td>0.46</td>
<td>2.69</td>
<td>-1.41</td>
</tr>
<tr>
<td>0.0573</td>
<td>30</td>
<td>20</td>
<td>15</td>
<td>0.46</td>
<td>2.21</td>
<td>-1.23</td>
</tr>
<tr>
<td>0.0573*</td>
<td>30</td>
<td>20</td>
<td>15</td>
<td>0.46</td>
<td>2.65</td>
<td>-1.36</td>
</tr>
<tr>
<td>0.33</td>
<td>30</td>
<td>100</td>
<td>86.4</td>
<td>0.08</td>
<td>1.90</td>
<td>-0.24</td>
</tr>
<tr>
<td>0.33</td>
<td>50</td>
<td>100</td>
<td>86.4</td>
<td>0.08</td>
<td>2.45</td>
<td>-0.18</td>
</tr>
<tr>
<td>0.33</td>
<td>30</td>
<td>100</td>
<td>15</td>
<td>0.46</td>
<td>1.58</td>
<td>-1.26</td>
</tr>
<tr>
<td>0.33</td>
<td>50</td>
<td>100</td>
<td>15</td>
<td>0.46</td>
<td>1.60</td>
<td>-1.20</td>
</tr>
<tr>
<td>0.33</td>
<td>100</td>
<td>100</td>
<td>15</td>
<td>0.46</td>
<td>1.60</td>
<td>-1.22</td>
</tr>
</tbody>
</table>

Table 2: Results of simulations with different resolutions for $Re = 20$ and $Re = 100$.

*Second simulation with $r = 0.0573$ has same height-radius ratio as the simulations with $r = 0.33$.

the bottom, the pressure is at the bottom lower compared to the top resulting in a force downwards on the cylinder.

4.3 Ship calculation

Lastly, the simulations shall be used to calculate which speed an actual ship could travel. As already ships with Flettner rotors existed, this can be used to prove that the simulations generate useful data or not. To calculate the speed of a ship a few things are simplified to have a quick approach. First, some data about a Flettner rotor vessel is needed. In the paper, [3] a ship with a Flettner rotor is described, with some characteristic values and an actual travel speed using the Flettner rotor. The rotor has a diameter of $d = 0.6 \, m$, a height of $h = 7.3 \, m$ and maximum rotation speed of $\omega = 400 \, \text{rpm}$. The 18 tons heavy ship would reach a velocity of $u_{\text{ship}} = 6.1 \, \text{knots} \approx 3.14 \, m/s$ with a wind speed of $u_{\infty} = 18.4 \, \text{knots} \approx 9.47 \, m/s$. Another ship of the Uni Flensburg the UniKat has the
4 Simulation Results

For the calculation of the speed using the simulation data equation 3.1 and 3.2 are used to calculate the force on the rotor. The coefficients are specified by the velocity ratio of \( \alpha \). All results are listed in table 3. To calculate an approximated velocity using the results of the simulation some further equations are needed. As the drag and lift coefficient only gives the possibility to calculate the force on the rotor and not a velocity in the first place. Therefore a force equilibrium is defined, whereas the resistance force equals the acceleration force of the rotor.

\[
F_{\text{rotor}} = F_{\text{resistance}} \tag{4.2}
\]

As the force on the rotor is defined over the speed of the airflow around the cylinder, which would change during the acceleration, a simplification is made. Thus, saying the acceleration force is constant, defined by the velocity ratio of rotor speed and wind velocity. The resistance force of the ship is a function of three sub resistance forces, viscous resistance \( R_V \), wave-making resistance \( R_W \), and air resistance \( R_A \). \[9\] \[10\]

\[
R_S = R_V + R_W + R_A \tag{4.3}
\]

The viscous resistance and force are defined by equation 4.4 over the density of the water, the ship velocity, the wetted surface area \( S_S \), and a resistance coefficient of \( c_V \). This coefficient is defined by equation 4.5, however, the \( K \) has is a regression factor approximated by many parameters, which describes the effect of the form of the ship. This can’t be done in this calculation, so it will be set to zero as no typical values are mentioned for this parameter. Reynolds is defined by the ship velocity, the water viscosity, and the ship’s length, which is in underwater. \[9\] \[10\]

\[
R_V = c_V \cdot \frac{\rho}{2} \cdot u_{\text{ship}}^2 \cdot S_S \tag{4.4}
\]

\[
c_V = (1 + k) \cdot \frac{0.075}{(\log(Re) - 2)^2} \tag{4.5}
\]

As the wave-making resistance is a function of ship length, wavelength, and ship speed, there is no explicit equation. However, it can be a huge part of the resistance force in total it could be another cause for high calculated velocities. Lastly, the air resistance will be ignored, which should not make much of a difference. \[9\] \[10\] This reduces the approximated resistance force to equation 4.6.

\[
R_T \approx \frac{0.075}{(\log(u_{\text{ship}} \cdot L/\nu) - 2)^2} \cdot \frac{\rho_w}{2} \cdot u_{\text{ship}}^2 \cdot S_S = \sqrt{c_{\text{drag}}^2 + c_{\text{lift}}^2 \cdot 2\pi \cdot r_{\text{rotor}} \cdot h_{\text{rotor}}} \tag{4.6}
\]

The wetted surface area of the ships has to be approximated as well. The kinematic viscosity of water are \( \nu = 1.1882E-06 \text{ m}^2/\text{s} \) and the density is \( \rho = 1025 \text{ kg/m}^3 \). \[9\] The final equation cannot be solved for \( u_{\text{ship}} \). It must be solved numerically. Nevertheless, the calculated values are in the same region as the real ones. For the first ship, it is
just about 4% off. The second ship is about 30% off, however, the calculations are full of simplifications and assumptions and the results should be lower, as mainly the resistance force was simplified by neglecting parts of it.

\[
0.075 \frac{\rho_w}{(\log(u_{ship} \cdot L/\nu) - 2)^2} \cdot \frac{u_{ship}^2}{2} \cdot S_S = \sqrt{c_{drag}^2 + c_{lift}^2} \cdot \frac{2\pi \cdot r_{rotor} \cdot h_{rotor}}{2} \cdot \rho_a \cdot U_\infty \quad (4.7)
\]

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Weight [kg]</td>
<td>18 000</td>
<td>500*</td>
</tr>
<tr>
<td>Rotor diameter [m]</td>
<td>0.6</td>
<td>0.8</td>
</tr>
<tr>
<td>Rotor height [m]</td>
<td>7.3</td>
<td>4</td>
</tr>
<tr>
<td>Rotation speed [rpm]</td>
<td>400</td>
<td>380</td>
</tr>
<tr>
<td>Rotation speed peripheral [m/s]</td>
<td>12.56</td>
<td>15.92</td>
</tr>
<tr>
<td>Wind velocity [m/s]</td>
<td>9.47</td>
<td>10</td>
</tr>
<tr>
<td>Velocity ratio [-]</td>
<td>1.33</td>
<td>1.59</td>
</tr>
<tr>
<td>(c_{drag} [-])</td>
<td>2.3</td>
<td>2.16</td>
</tr>
<tr>
<td>(c_{lift} [-])</td>
<td>4.2</td>
<td>4.94</td>
</tr>
<tr>
<td>Ship wetted length [m]</td>
<td>13*</td>
<td>6.1*</td>
</tr>
<tr>
<td>Ship wetted area [m²]</td>
<td>26*</td>
<td>9.76*</td>
</tr>
<tr>
<td>Ship speed [m/s]</td>
<td>3.14</td>
<td>3.5</td>
</tr>
<tr>
<td>Calculated velocity [m/s]</td>
<td>3.02</td>
<td>4.47</td>
</tr>
</tbody>
</table>

Table 3: Ship details and calculation results. *Approximated values over the ships dimensions.
5 Conclusion

Overall, the Flettner rotor could be successfully implemented in a test case of openLB. The rotation of the rotor is set up as functor and provides fast calculations. As the cylinder is approximated with a staircase, it is necessary to have a resolution high enough to avoid artifacts, which could interfere with the velocity field and create false results. All describing parameters and possibilities to run the simulation are easy and fast to change, just by changing constants at the beginning of the cpp-file. All results are outputted in various forms, leaving a great possibility to evaluate and visualize the simulation results. The validation of the test case without rotation of the cylinder already showed the influence of the geometry, however it validated the set up of the free flow. It gave also an orientation for the needed resolution, before starting with the more time consuming simulations of the rotating cylinder. A vast amount of different simulations are run with different settings. These showed the influence of Reynolds number, velocity ratio, and geometry. The implemented tools gave an easy approach to evaluate the simulation data. Furthermore, the simulation results, drag and lift coefficients, could be used to calculate a theoretical velocity of a vessel. These were even with some assumptions in the general area of the real measured velocities of these ships. However, the whole simulations and calculations showed the working principle of the Flettner rotor, using the magnus effect for propulsion.
Figure 15: Shifted start up of the rotation of the cylinder with $Re = 100$, $\omega = 400$ rpm, $U_\infty = 15$ m/s
Figure 16: Force on the cylinder with $Re = 100$, $\omega = 400$ rpm, $U_\infty = 15 \text{ m/s}$
Figure 17: Rotation of the cylinder with $Re = 20$, $\omega = 400$ rpm, $U_\infty = 15$ m/s
Literatur


