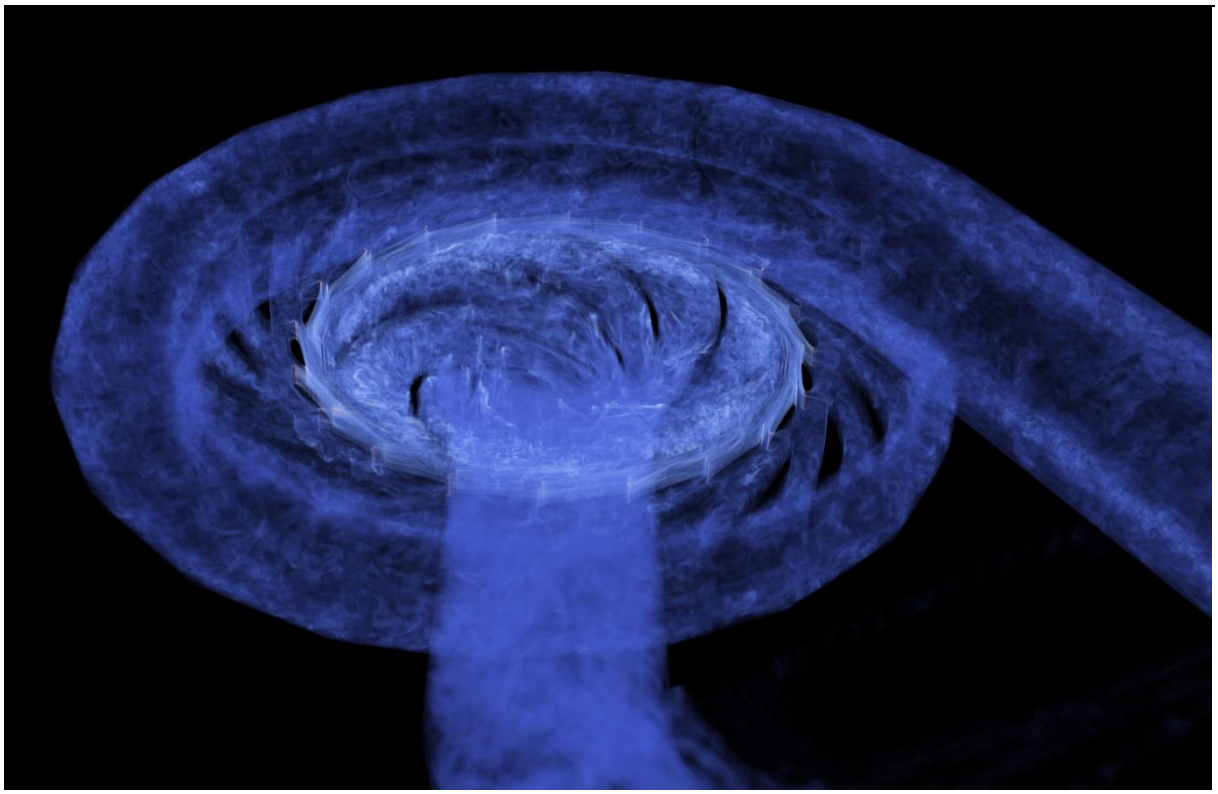
Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Federal Department of the Environment, Transport, Energy and Communications DETEC

Swiss Federal Office of Energy SFOE
Energy Research and Cleantech Division

Final report dated 17. November 2023

Lattice Boltzmann for Hydraulic Turbomachines



Source: ©David Roos Launchbury, HSLU 2023

Lucerne University of
Applied Sciences and Arts

HOCHSCHULE
LUZERN

FH Zentralschweiz

Date: 17. November 2023

Location: Bern

Publisher:
Swiss Federal Office of Energy SFOE
Energy Research and Cleantech
CH-3003 Bern
www.bfe.admin.ch

Subsidy recipients:
Hochschule Luzern, Technik & Architektur (HSLU)
Technikumstrasse 21, 6048 Horw
www.hslu.ch

Authors:
David Roos Launchbury, HSLU, david.roos@hslu.ch
Prof. Dr. Ernesto Casartelli, HSLU, ernesto.casartelli@hslu.ch

SFOE project coordinators:
Dr. Michael Moser, michael.moser@bfe.admin.ch
Dr.-Ing. Klaus Jorde, klaus.jorde@kjconsult.net

SFOE contract number: SI/501903-01

Zusammenfassung

Der Schwerpunkt dieses Projekts lag auf der Entwicklung eines neuen Lattice-Boltzmann-basierten Lösers für die genaue und effiziente Simulation turbulenter Strömungen in hydraulischen Turbomaschinen. Es zielt vor allem auf die Simulation hochgradig instationärer Betriebspunkte ab, bei denen die derzeitigen Simulationsmethoden entweder zu teuer oder nicht genau genug sind, um die zugrunde liegende Physik korrekt darzustellen. Die Anwendung dieser Methode wird zu einem besseren Verständnis der Off-Design-Betriebsbedingungen und damit zu einer verbesserten Leistung und größeren Flexibilität der hydraulischen Maschinen führen.

Es wurde gezeigt, dass der Code in der Lage ist, komplexe Fälle selbst bei sehr hohen Reynoldszahlen ($O(10^7)$) zu berechnen. Dazu gehören Simulationen, die aus rotierenden und stationären Komponenten in mehreren Bezugssystemen bestehen.

Die Simulationen können mit mehreren Ebenen der Gitterverfeinerung durchgeführt werden, wobei lokale CPU- oder GPU-Hardwarebeschleunigung zum Einsatz kommt. Die Parallelität zwischen Rechenknoten ist über das MPI-Protokoll implementiert, so dass der Code mehrere GPUs für beliebig komplexe Geometrien und Gitter nutzen kann, wobei jede manuelle oder automatische Zerlegungsmethode verwendet werden kann.

Besonderes Augenmerk wurde auf die Benutzerfreundlichkeit der Software gelegt, so dass Simulationsingenieure ohne Kenntnisse der Lattice-Boltzmann-Methode oder der Code-Entwicklung solche Simulationen durchführen können.

Der Code wurde anhand einer Reihe von Verifikations- und Validierungsfällen evaluiert, die einfache laminare 2D-Cases, interne turbulente Strömungen wie Diffusoren und Rohrbögen sowie vollständige Pumpen- und Pumpturbinenfälle abdecken. Alle vorgestellten Ergebnisse konnten mit höchstens 4 GPUs und in wenigen Stunden erzielt werden.

Die Arbeit am Solver wird in Zukunft fortgesetzt, um die Effizienz weiter zu steigern und die Hardwareanforderungen zu reduzieren. Außerdem müssen weitere Erfahrungen gesammelt und Best-Practices entwickelt werden, um die erzielten Ergebnisse weiter zu verbessern.

Résumé

Ce projet s'est concentré sur le développement d'un nouveau solveur basé sur Lattice-Boltzmann pour la simulation précise et efficace des flux turbulents dans les turbomachines hydrauliques. Il vise principalement la simulation de points de fonctionnement hors concepte et très instables, pour lesquels les méthodes de simulation actuelles sont soit trop coûteuses, soit pas assez précises pour représenter correctement la physique sous-jacente. L'application de cette méthode permettra de mieux comprendre les conditions de fonctionnement hors paramètres de conception et donc d'améliorer les performances et la flexibilité des machines hydrauliques.

Le code s'est révélé capable de calculer des cas complexes, même à des nombres de Reynolds très élevés ($O(10^7)$). Il s'agit notamment de simulations comprenant des composants rotatifs et stationnaires dans plusieurs cadres de référence.

Les simulations peuvent être effectuées à l'aide de plusieurs niveaux de raffinement de la grille, en utilisant l'accélération matérielle du CPU ou du GPU au niveau du nœud local. Le parallélisme inter-nœuds est mis en œuvre via le protocole MPI, ce qui permet au code d'utiliser plusieurs GPU pour

des géométries et des grilles arbitrairement complexes, en utilisant n'importe quelle méthode de décomposition manuelle ou automatique.

Une attention particulière a été accordée à la facilité d'utilisation du logiciel, permettant aux ingénieurs en simulation sans aucune connaissance de la méthode Lattice Boltzmann ou du développement de code d'effectuer de telles simulations.

Le code a été évalué à l'aide d'une série de cas de vérification et de validation, couvrant des configurations 2D laminaires simples, des flux turbulents internes tels qu'un diffuseur et un coude de tuyau, ainsi que des cas de pompe et de pompe-turbine complets. Tous les résultats présentés ont pu être obtenus en utilisant au maximum 4 GPU et en quelques heures.

Le travail sur le solveur se poursuivra à l'avenir, afin d'améliorer encore les performances et de réduire les besoins en matériel. En outre, il faudra acquérir plus d'expérience et établir les meilleures pratiques, ce qui permettra d'améliorer encore les résultats obtenus.

Summary

The focus of this project was the development of a new Lattice-Boltzmann-based solver for the accurate and efficient simulation of turbulent flows in hydraulic turbomachines. It mainly targets the simulation of highly unsteady off-design operating points, where current simulation methods are either too expensive or not accurate enough to properly represent the underlying physics. Applying this method will lead to better understanding of off-design operating conditions and thus improved performance and more flexibility of hydraulic machines.

The code was shown to be capable of computing complex cases even at very high Reynolds numbers ($O(10^7)$). This includes simulations consisting of rotating and stationary components in multiple reference frames.

Simulations can be performed using multiple levels of grid refinement, utilizing node-local CPU or GPU hardware acceleration. Inter-node parallelism is implemented via the MPI protocol, allowing the code to utilize multiple GPUs for arbitrarily complex geometries and grids, using any manual or automatic decomposition method.

Special attention was paid to the usability of the software, allowing simulation engineers without any knowledge of the Lattice Boltzmann Method or code development to perform such simulations.

The code was evaluated using a series of verification and validation cases, covering simple laminar 2D setups, internal turbulent flows such as a diffusers and a pipe bends as well as fully-featured pump and pump-turbine cases. All results presented could be obtained using at most 4 GPUs and in a matter of hours.

Work on the solver will continue in the future, further increasing performance and reducing hardware requirements. Additionally, more experience will have to be gained and best practices will need to be established, further improving the obtained results.

Main findings

- In this project a turbulent flow solver was developed, based on the entropic MRT Lattice Boltzmann Method.
- The solver allows simulation engineers to compute complex, time-dependent turbulent flows in the context of hydraulic turbomachines in a matter of hours, using GPU hardware accelerators.
- The code was verified using numerical experiments and validated on a number of industrially relevant cases. Results obtained are very good in all but the most complex cases, where further improvement is expected when more experience with the simulation method is gained. Future work is required to optimize the software with respect to memory requirements and performance.
- The entropic MRT Lattice Boltzmann Method was found to be a viable alternative to traditional CFD simulation methods currently used in industry.

Contents

Abbreviations

BGK	Bhatnagar–Gross–Krook collision model
CPU	Central processing unit
DNS	Direct numerical simulation
GPU	Graphics processing unit
FURIES	Future Swiss electrical infrastructure
HSLU	Lucerne University of Applied Sciences and Arts
HVAC	Heating, ventilation and air conditioning
KBC	Karlin/Bösch/Chikatamarla collision model
LBM	Lattice Boltzmann Method
LES	Large eddy simulation
MS	Milestone
MRT	Multi-relaxation time
RANS	Reynolds-averaged Navier-Stokes
SCCER	Swiss competence center for energy research
SoE	Supply of electricity
URANS	Unsteady Reynolds-averaged Navier-Stokes
VLES	Very large eddy simulation
WP	Work package

1 Introduction

1.1 Background information and current situation

According to the SFOE report on existing hydro storage power plants from the 1st of January 2018 the expected electrical power generation is estimated at 17'220.63 [GWh] over the entire year. Approximately 30% of this amount (5166 [GWh]) is produced in Francis-type plants or pump turbines running in turbine mode. These machines are at times run in part-load operating conditions and undergo several starts and stops daily. The influence of deep part-load operation points is not fully understood and accordingly not investigated during the design process of such machines, mainly due to the fact that numerical simulations of such conditions are demanding and very time-consuming. Being able to predict the behaviour in off-design situations more accurately will help to improve the performance and flexibility of these machines, leading to savings in electrical energy.

Assuming the turbines are running in part load at 10% of the operation time and the power generation in these operating points is 30% of the nominal turbine power, a relative increase of 10% in part-load efficiency would lead to additional outputs in the order of 15 [GWh].

In addition, part load conditions will become more and more important in the context of ancillary services which maintain the stability of voltage and frequency in the Swiss power grid. Currently grid stability in Europe is guaranteed by overproduction and excess power is simply dissipated in railroad tracks and power lines. Running hydraulic power plants in part load conditions can deliver a stable grid using renewable energy instead of wasting the excess production of e.g. coal and nuclear power plants. This requires, however, that turbines are adequately designed to be stable over their entire operating range.

The changes in the electricity production market towards more dynamic operation of the machines lead to complex unsteady flow conditions that need to be investigated and understood in order to guarantee safe operation of the equipment. Example publications by companies such as Andritz Hydro [1], Alstom [2], Voith Hydro [3] or Litostroj Power [4] show the relevance of unsteady simulations for the industry. The need for such simulations has been growing in recent years, but the limitations of current methods still require strong simplifications. From the turbine manufacturer point of view, two main arguments toward a largely increased need of unsteady computations can be identified:

- A large number of Francis turbines are coming closer to their end of life and refurbishment is needed. Since every machine has a different behaviour, it is important to have a computational technique/tool able to perform analysis on many machines within reasonable time.
- Rotor-stator interaction of high-head machines is no more limited to the interaction of the single passages but extends to wave propagation in the whole casing and in the hydraulic system. This phenomenon has been observed recently and is beginning to be investigated experimentally and in part with simplified computations. With the present CFD methods it is impossible to perform comprehensive investigations at this level.

From the power plant owner side, the phase of refurbishment often leads to the replacement of Francis turbines with pump-turbines or even the installation of new pump-turbines. This means that the pipes, often connected into networks, will now experience flow in both directions. As an example, every original cross-section reduction will become a “diffuser”, but most likely have a non-optimal

expansion ratio. Additionally, every bifurcation will receive water from independently operated machines.

All these phenomena, in conjunction with unsteady operation, will lead to pulsations and additional flow distortion in the components (valves, bends, etc.) and at the machine inlet and consequently to increased dynamic loads compared to the present situation.

At present the analysis of specific, unsteady phenomena for one single specific machine is mostly performed with one of the following approaches:

- academic: PhD thesis with extensive IT infrastructure and LES resolution
- industrial: simplified model using URANS and industrial equipment, thus neglecting much of the underlying physics (much lower spatial and temporal resolution, and with a series of assumptions)

In the future these investigations need to be performed much more regularly and in more detail using industrial equipment.

Starting from the situation described above the goal of the project can be formulated as follows: to develop a new hybrid CPU/GPU solver based on the Lattice Boltzmann Method that allows the efficient simulation of the turbulent flows in large-scale hydraulic turbomachines.

The main research question is to be able to investigate and increase the understanding of unsteady flow phenomena in off-design conditions. This requires a fast and stable solver that can deliver significantly more accurate results in a much shorter timeframe than is currently possible, thus allowing to account for these particular operating conditions already during the design process of hydraulic machines and components. With the present computational power at industrial level this is only possible with the introduction of a new computational method, like the Lattice Boltzmann Method.

1.2 Purpose of the project

During this project a new solver based on the Lattice Boltzmann is developed. This solver allows the efficient simulation of the turbulent flows in hydraulic turbomachines, delivering more accurate results in a much shorter timeframe than is currently possible with RANS and LES methods. This leads to the possibility to account for off-design operating conditions already during the design process of hydraulic machines. Applying this method will lead to better understanding of off-design operating conditions and thus improved performance and more flexibility of hydraulic machines

1.3 Objectives

The main objective of this project is the creation of the TurboLaB simulation software and the evaluation of the Lattice Boltzmann Method as a simulation technique in the context of hydraulic turbomachines. Results obtained with the software will be compared to analytical solutions, where available, reference simulations and measurements, in order to evaluate their quality.

2 Procedures and methodology

The group of Prof. Dr. Casartelli at the HSLU has performed numerous investigations on the subject of hydraulic machines running in off-design conditions in the context of the SCCER SoE and FURIES, resulting in several publications. One major finding of the investigations was that turbulence modelling has a major impact on the reliability of the result and that conventional unsteady RANS (URANS) simulations using two-equation models are not enough to capture the effects that cause instability in such conditions. In addition, time-dependent URANS simulations of full machines are expensive (i.e. calculations can take several days to weeks to complete) and only deliver satisfactory results when more complex, anisotropic turbulence models are used. A reduced amount of modelling could be provided by large eddy simulations, where most of the turbulent spectrum is resolved instead of modelled. Large eddy simulations are, however, even more expensive, as the mesh resolution requirements increase and time-accurate time steps are employed. This is specifically problematic for incompressible flows due to the need to solve an implicit equation for the pressure. There is currently no method that allows such simulations of hydraulic turbomachines already in the design phase with a reasonable time requirement. One simulation procedure that has recently risen to prominence is the Lattice Boltzmann Method (LBM). Some of its distinct features are the explicit formulation, also for incompressible flows, as well as the suitability for massively parallel applications. This makes it an interesting candidate for performing large-scale time accurate simulations.

The basic Lattice Boltzmann Method, also known as the BGK (Bhatnagar–Gross–Krook) model [5], is, however, unsuitable for high Reynolds flows as it becomes unstable with increasing Reynolds number. A handful of commercial products have recently appeared, allowing the widespread use of Lattice Boltzmann for simulations. They are mostly built on the BGK model, slight variations thereof, or basic MRT methods, and are using the viscosity-increasing effect of turbulence models to artificially stabilize the method. The results of such programs to wall-bounded internal flows are not satisfactory, especially when near-wall effects (boundary layers) are important. In recent years, a new class of LB methods was developed, pioneered by the group of Prof. Dr. I. Karlin (ETHZ), which modifies the BGK model by taking the physical constraints of entropy into account (original model see [6]). This method is extremely stable, also at high Reynolds numbers, and has excellent properties concerning the accurate representation of turbulent flows. In combination with appropriate boundary conditions, the method was shown to be able to produce DNS-like results for wall-bounded flows at a fraction of the computational effort. Various extensions to the method exist for multiphase and compressible flows.

2.1 Turbulent flow modelling

The simulation of turbulent flows can be categorised into three main fields: Reynolds-averaged Navier Stokes (RANS), large eddy simulation (LES) and direct numerical simulation (DNS). The first one is the most widely used method in industry, due to its low computational requirements and ability to directly solve for steady-state solutions. Here, the entire energy spectrum of turbulence is modelled by solving additional equations for the approximation of turbulent quantities. The third and most expensive method is direct numerical simulation, which fully resolves all turbulent scales, ranging from the order of the domain size down to the Kolmogorov length scale $\eta = \left(\frac{\nu^3}{\epsilon}\right)^{\frac{1}{4}}$. Here, ν is the kinematic viscosity of the fluid and ϵ is the dissipation rate of turbulent kinetic energy. Due to the resolution requirements, both in space and time, the computational effort for DNS calculations scales approximately with Re^3, where $Re = lu/\nu$ is the Reynolds number based on a characteristic length l

and velocity u. This restricts the use of this method to a small range of Reynolds numbers, certainly far away from engineering applications.

The second method mentioned above, large eddy simulation, is a compromise between the two extremes: it fully resolves the larger turbulent scales and models the smaller ones with a turbulence model. The filter for distinguishing resolved and modelled scales is usually given by the mesh size. A commonly used practice for large eddy simulations is to fully resolve about 80% of the turbulent spectrum, with the remaining 20% approximated by a turbulence model (see [7], [8]). The accurate, time-dependent solution of the resolved structures requires both small time steps and methods of low numerical diffusion.

2.2 Lattice Boltzmann Method

One method that exhibits these properties is the Lattice Boltzmann Method (LBM). In contrast to traditional fluid computations, LBM does not solve the Navier-Stokes equations but is based on a discretisation of the Boltzmann kinetic equation

$$\frac{\partial f}{\partial t} + c_\alpha \frac{\partial f}{\partial x_\alpha} + \frac{F_\alpha}{\rho}\frac{\partial f}{\partial c_\alpha} = \Omega(f) \tag{1}$$

Here $f(x, c, t)$ is a particle distribution function in phase space, i.e. it is a function of space x, particle velocity c and time t. The first term on the left hand side is simply the change of f in time, while the second term denotes the advection of f and the third includes changes due to external force fields. The right hand side term in Equation (1) is called the collision or relaxation operator and can be approximated by the BGK formulation [5] given by

$$\Omega(f) = \frac{1}{\tau}(f^{eq} - f) \tag{2}$$

where $f^{eq}(x, c, t) = f^{eq}(c, \rho(x, t), u(x, t), T(x, t))$ is an equilibrium distribution, given by the Maxwellian evaluated using the local macroscopic quantities and the particle velocities.

Equation (1) can be discretised in different ways, but the most common approach (see e.g. [9]), ignoring the force term, leads to a system of N discrete kinetic equations, where N is the cardinality of the velocity set used. An example for a velocity set, or population set, corresponding to one possible discretisation of the velocity space, is shown in Figure 1, here using nine populations in two dimensions.

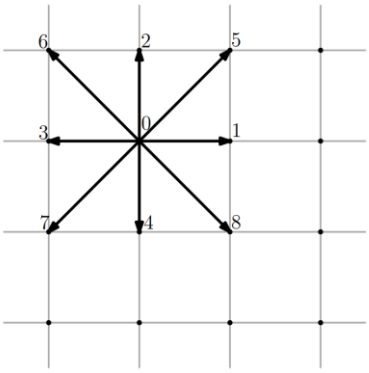


Figure 1 - Velocity set D2Q9 on a Cartesian grid

The discretised equations presented below follow the formulation in [10].

$$f(x + c_i\Delta t, t + \Delta t) = (1 - \beta)f_i(x,t) + \beta f_i^{mirr}(x,t) \quad (3)$$

where $f_i^{mirr}(x,t) = 2f_i^{eq}(x,t) - f_i(x,t)$, $\beta = \Delta t/(2\tau + \Delta t)$, $\tau = \nu/c_s^2$ and $c_s = 1/\sqrt{3}$.

Again, ν is the kinematic viscosity, Δt is the time step and c_s is the speed of sound. In most cases, the equation system is normalised to so-called lattice units, where both Δt and the grid spacing is unity. This naturally leads to a spatial discretisation using uniform Cartesian grids at a speed of sound of $1/\sqrt{3}$.

This system of equations can be solved explicitly in time and can easily be parallelised. This is due to the fact that non-local parts (i.e. advection) are linear and correspond to simple array shifts, while all nonlinearities contained in the collision term on the right hand side, are local. This method is known as LBGK (Lattice BGK) and is able to reproduce solutions of incompressible, or weakly compressible, Navier Stokes equations (see e.g. [11]).

2.3 LBM at high Reynolds numbers

Due to the discretisation procedure, the discrete populations f_i can become negative, which causes the solution to become unphysical and fail. This situation occurs in the presence of unresolved, high gradients, which are especially prevalent at higher Reynolds numbers where the damping, smoothing viscous forces are small compared to the inertial ones. Two main directions have evolved to deal with this issue. One can be found in the cumulant LB method [12], the other is represented by the so-called entropic LB method followed in this work. Karlin et al. have linked the instabilities mentioned above to the violation of the entropy condition in the mirror state (see e.g. [13]). They have then proposed a modification to the original LBGK formulation which scales the components of the mirror state, such that the discrete entropy condition is obeyed:

$$f_i^{mirr} = \alpha f_i^{eq} + (1 - \alpha)f_i \quad (4)$$

The scalar α must be calculated to ensure the following condition is satisfied

$$H(f^{mirr}) = H(f) \quad (5)$$

with H being the discrete entropy function $H(f) = \sum_i f_i \ln\left(\frac{f_i}{w_i}\right)$ where w_i are lattice weights. The scaling of the mirror state therefore ensures that the entropy condition is never violated. This has led to an implementation of the Lattice Boltzmann Method, termed the entropic LBM, which is stable also at high Reynolds numbers. An analysis of the method by Malaspinas et al. [14] has shown that the entropic stabilisation can be understood as a subgrid-turbulence model. The drawback of this method is, that Equation (5) has to be iteratively solved to high precision at every lattice node, making the method significantly slower than the original LBGK.

Recently, the KBC model was developed by Karlin, Bösch and Chikatamarla [15], [16] which also ensures that the entropy condition is not violated, but does so by using different relaxation factors for different moment groups (the multi-relaxation time (MRT) method, introduced by d'Humières et al. [17]). This way, the expensive iterative solution of the H-constraint can be avoided and replaced by a much simpler computation. According to the authors, simulation times are expected to increase by a factor of only two to four, compared to the original LBGK. The method was extended to multiphase and multi-component flows, as well as thermal and compressible flows [18], which remain challenging for Lattice Boltzmann Methods. It has shown excellent stability and accuracy when applied to

academic and semi-industrial cases. It therefore represented an ideal candidate for further investigation and the application to the simulation of hydraulic turbomachines.

2.4 Boundary conditions & rotating domains

The Lattice Boltzmann Method is still very actively researched, also on basic topics such as boundary condition implementations. A very wide range of formulations exist for various boundary conditions, with large differences in stability and accuracy properties (see e.g. [19]). The difficulties in the formulation of boundary conditions for Lattice Boltzmann simulations are two-fold. First, LBM solves transport equations for populations, not macroscopic quantities directly. Therefore boundary conditions have to provide values for missing boundary populations. Secondly, the Cartesian grid structure does not follow complex boundary surfaces directly, such as in finite element or finite volume methods.

Discrete populations can easily be transformed into macroscopic moments by a simple matrix-vector multiplication and the same is true for the opposite direction. If the complete set of macroscopic boundary moments are known, the boundary populations can be accurately prescribed. However, specifying values for higher order moments is difficult as their physical interpretation is less clear.

The approach followed in this work will be to provide enough macroscopic target values in a general way such that for all types of boundary conditions accurate boundary populations can be reconstructed.

One approach to approximate populations is the application of Grad's moment system [20]. It was previously used to construct population data from macroscopic quantities in [21] and this was applied to formulate an accurate and stable boundary condition for stationary and moving walls in [22]. We try to extend this formulation to create a boundary condition formulation which is applicable for any boundary condition on any boundary geometry.

The principal ideas of the present boundary method are put forward in the following discussion surrounding Figure 2. Similar methodologies have been applied by other authors in different contexts, such as wall boundaries for immersed solids in finite volume methods (e.g. [23]) or for additional transported scalars in [24]. Recently in [25] a boundary condition method similar to the one presented here was published, although with a focus on compressible, external aerodynamics and limited to wall and open boundaries.

In order to calculate the missing populations, target macroscopic moments for the density ρ, momentum j and pressure tensor P are required at the boundary node locations. Figure 2 shows a sketch of the situation close to a boundary.

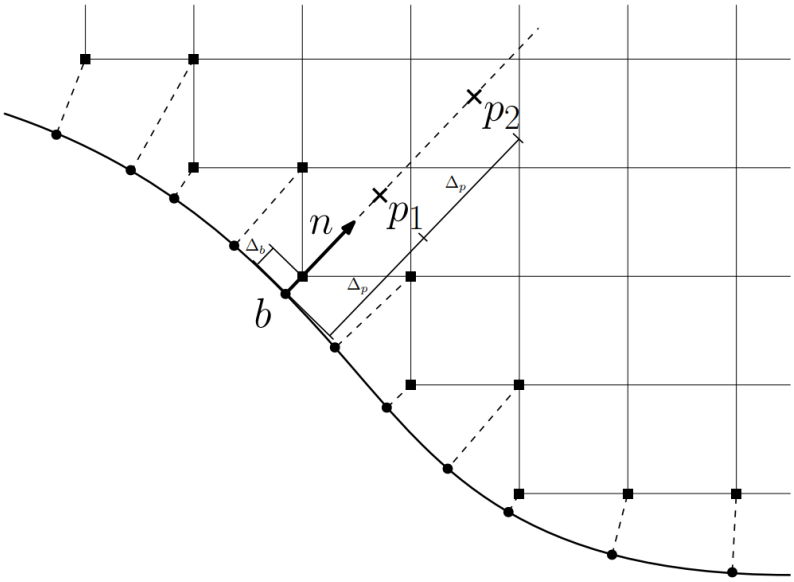


Figure 2 - Boundary interpolation points

The squares mark the boundary nodes where populations need to be provided and therefore target moments are needed. The circles are the closest points on the underlying geometry and n represents a local boundary normal. Using these points and normals, the interpolation nodes, shown as crosses, are constructed Δp and $2\Delta p$ away from the boundary. Note that we here show a version with two interpolation points, but the method can also be applied using only one point.

In order to obtain the target moments at the boundary nodes (squares), we first evaluate the moments at the interpolation points. Then the boundary moments are constructed, either directly in case of Dirichlet boundary conditions, or using the interpolation point values to evaluate Neumann values or, where needed, the surface normal gradients. For example in the case of a wall, this surface normal gradient is used to construct the non-equilibrium part of the pressure tensor in a local coordinate system defined by the flow velocity and the boundary normal. When the boundary moments are complete, the target moments are interpolated to the boundary node using the interpolation point values and the newly created boundary values. The last step is then to reconstruct the missing populations using the Grad approximation

$$f_i(\rho, j, P) = w_i\left(\rho + \frac{j_\alpha c_{i\alpha}}{c_s^2} + \frac{1}{2c_s^4}\left(P_{\alpha\beta} - \rho c_s^2 \delta_{\alpha\beta}\right)\left(c_{i\alpha}c_{i\beta} - c_s^2\delta_{\alpha\beta}\right)\right) \tag{6}$$

Using this method, the following boundary conditions are currently implemented:

- Inlet, using fixed velocity, uniform or correlation-based, with density extrapolation
- Periodic inlet (recycling)
- Non-reflective outlet (based on [26])
- No-slip wall, with and without wall tangential velocities
- Turbulent wall function

- Slip wall/symmetry planes
- Grid interface
- Rotor-stator interface

All listed boundary conditions are applicable to arbitrarily shaped geometries and across grid level changes, with further variants easily added.

The new boundary condition formulation can provide boundary data on the underlying geometry, also for derived quantities such as wall shear stresses. In addition, a unique surface area can be found for each boundary data point by a re-triangulation of the surface geometry, directly allowing area- and mass-flow-averaging of boundary data. This also allows force integration based on local pressure and stress values for arbitrary geometries.

As the name suggests, turbomachines contain rotating machine components with two approaches to their simulation. The first physically rotates the geometry of the rotating part over the background mesh and then recomputes all boundary intersections. Grid nodes which are now hidden by the geometry need to be switched off, and newly uncovered nodes must be switched on and their populations approximately reconstructed. In addition to a significant number of geometry intersections requiring calculation at every time step, the activation and deactivation of grid nodes leads to density fluctuations which can affect the flow field. Dorschner et al. [27] provides examples for such moving geometries.

The second approach is to split the computational domain into subdomains, one containing the rotating parts, the other the remaining stationary components. Here the grid rotates together with the geometry and therefore does not require remeshing or refilling. However, data needs to be exchanged between the rotating and the stationary subdomains at the interface. In [28] this is achieved using overlapping grids and interpolation. Another author uses a volumetric approach in [29], but there the entire implementation of the LB method is formulated in a finite-volume-like manner.

We follow a different approach, similar to the method used in the boundary conditions mentioned above. We generate interpolation points on both sides normal to the interface for each node with missing populations. Using these source points, we interpolate the target values for density, momentum and pressure tensor in cylindrical coordinates, and again reconstruct the missing populations using the Grad approximation.

To simulate the domain rotation, the rotating part is solved in the relative system, which requires the addition of source terms for the centrifugal, Coriolis and Euler forces. All these source components, along with every other force, are incorporated via the exact difference forcing method from [30]. In the present method, the grid of the rotating domain remains stationary and only the interpolation points move in their respective directions. This simple approach has so far proven to be effective, accurate, stable even at very high Reynolds numbers and well suited for parallelization, since every boundary node can update its own interpolation points on the opposite side of the interface independently. Figure 3 shows a sketch of two interface-adjacent boundary nodes: the light gray boundary node belongs to the rotating domain, the dark gray to the stationary one.

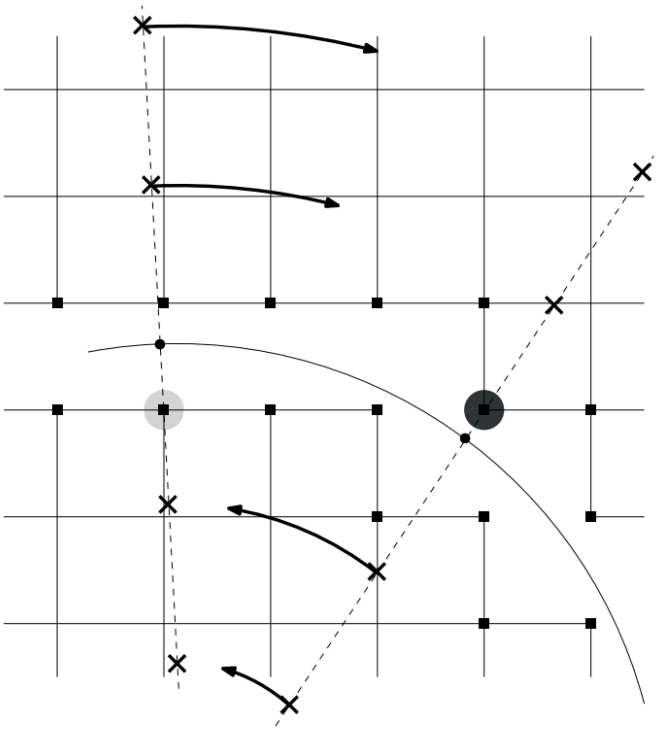


Figure 3 - Interpolation points around interface for a clockwise-rotating inner domain

2.5 Grid generator

In the conventional Lattice Boltzmann Method the time step and spatial resolution are coupled. It therefore relies on uniform Cartesian grids for the space discretisation. While the generation of grids for simple cases is straightforward, additional effort is required when dealing with general geometries to ensure a robust creation of the computational domains. Due to the special requirements to the grid in LBM codes, existing mesh generators for CFD codes based on other methods, such as the finite volume method, could not be directly used. Grid generators from other open-source LBM packages do not provide all the features required for the simulation of complex geometries, especially rotating machines. It was therefore necessary to develop a separate program responsible for grid creation. The octree grid structure and level set method provided in the bitpit library [31] were used as a basis to build the grid generator used in this project.

The grid generator is able to distinguish between points inside and outside of a provided geometry, therefore providing the initial grid resolution on a uniform point distribution. Below we provide an example case using arbitrary geometries. Assume that the green object in Figure 4 is the flow obstacle in the flow and the blue object is a refinement region.

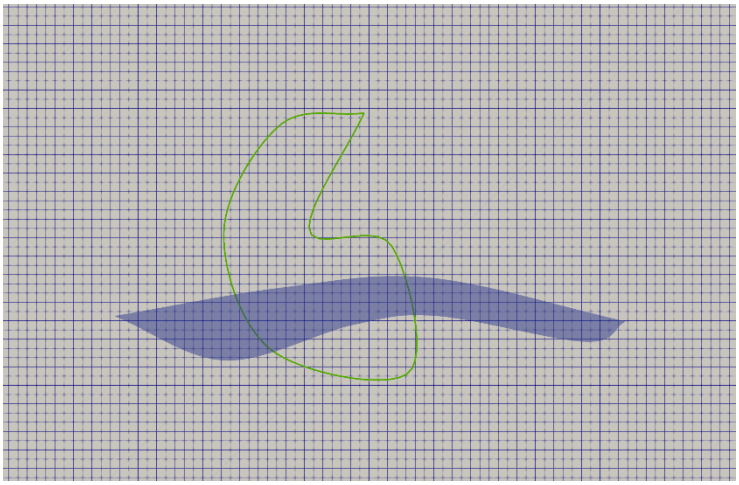


Figure 4–- Grid generator, starting point

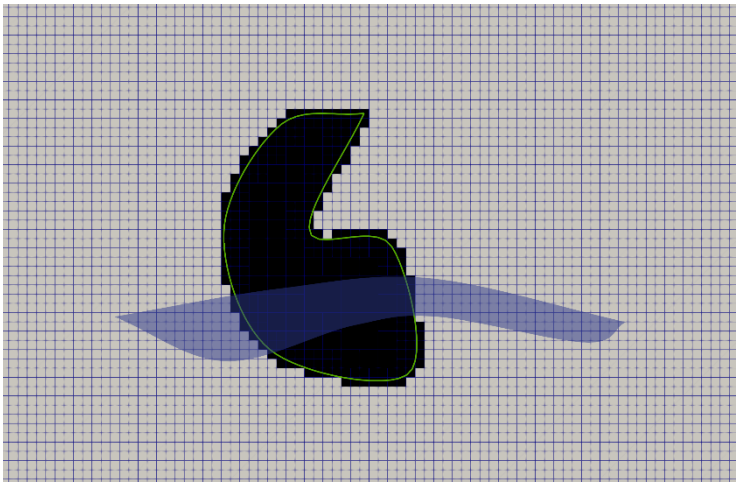


Figure 5–- Grid generator, geometry cutout

The obstacle geometry, provided as a triangulated STL file, can consist of multiple, separate patches. These patches can be used to specify refinement levels. The grid is then locally refined according to these definitions, as seen in Figure 6 below.

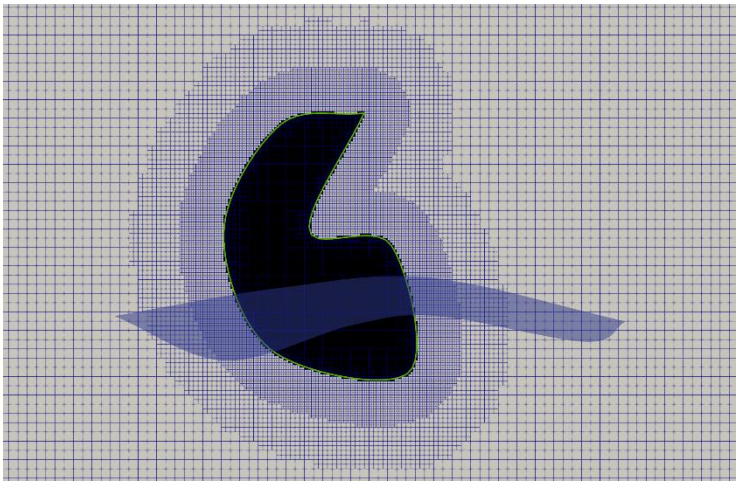


Figure 6–- Grid generator, patch refinement

In addition, a local refinement feature was implemented, which allows the user to specify additional geometries that describe refinement regions. These can be used to refine the grid in arbitrary regions, according to the requirements of the simulation. As an example, the user could run a simulation and find that a certain vortex region is not adequately refined. Using vortex tracking methods, an iso-surface of this region can be extracted from the simulation results, which can then again be fed into the grid generator as a refinement region. This procedure is shown in Figure 7, where, in addition to patch refinement, the grid is further refined in and around the blue refinement geometry.

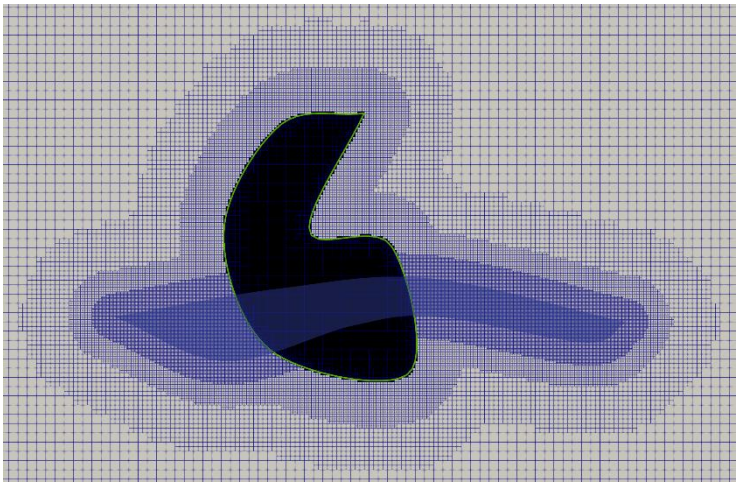


Figure 7–- Grid generator, arbitrary refinement region

When all of the above criteria have been fulfilled, the grid generator marks special nodes, such as boundary nodes, and calculates boundary points and their normals and the local STL geometry, based on a level set field. This data is then stored for the simulation.

A special requirement for the implemented refinement algorithm (see below) is that the grid levels require a certain amount of overlap to allow data exchange. The basic octree data structure does not directly allow this, which is why this is done as a post-processing step.

The grid generator implemented in this project was parallelised using the MPI library, therefore providing the ability to quickly generate grids on multicore machines or, for large cases, even on compute clusters. It allows for flexible and robust generation of refined meshes on arbitrarily complex geometries.

2.6 Grid conversion and domain decomposition

After the octree grid is generated, the mesh converter module translates it into the solver-internal format. During this step, grid levels are overlapped and refinement information as well as additional auxiliary grid data is generated.

Also included in this conversion step is the decomposition into multiple partitions for MPI-based inter-node parallelism. LBM grid refinement algorithms compute every level independently and then exchange information between the levels. Because of this, every level is decomposed separately into equally sized partitions. In simulations of complex geometries with local grid refinement and the non-local boundary treatment, this can lead to complicated communication dependencies. The present code is able to resolve all these dependencies and allows for arbitrary domain decompositions. Some manual decomposition methods are implemented, such as slicing in principal directions, splitting by index and a more involved angular decomposition used in one of the turbo-cases. The code also supports an automatic decomposition method, for which the METIS [32] library was employed. Figure 8 shows the decomposition using the METIS method of the pump-turbine case described in the later sections.

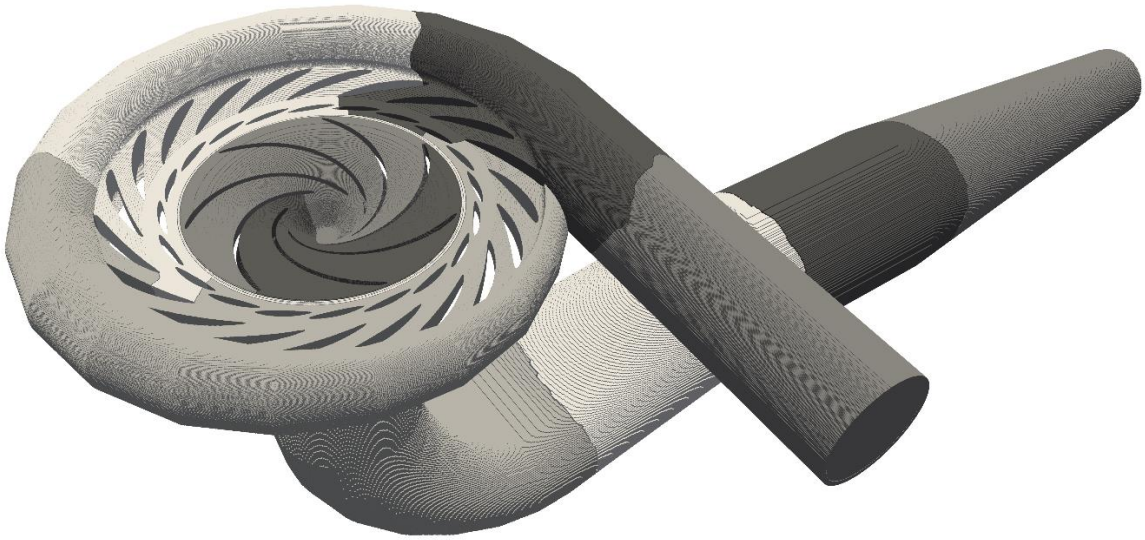


Figure 8 – METIS decomposition into 4 partitions of pump-turbine case

2.7 Refinement algorithm

As mentioned previously, the Lattice Boltzmann Method is based on coupled space and time intervals. Grid refinement, i.e. locally changing the resolution in space, directly leads to a change of the resolution in time if we want to preserve the low-dissipative properties of the LB method. Grid refinement is therefore usually implemented in such a way that the refinement factors are integers (e.g. quadtree/octree refinement) and finer grid levels advance multiple steps in time while the coarsest level only performs a single step.

The refinement algorithm implemented in this project is based on the ideas explained in [18], which provides a method that avoids the low-order temporal interpolation of other methods. The original algorithm can be summarised as follows (for two refinement levels):

1) Advection and collision on fine grid
2) Advection on coarse and fine grids
3) Rescaling of populations on twin nodes
4) Reconstruction (interpolation) of missing information
5) Collision on coarse and fine grids

The algorithm was adapted in multiple ways, however, which allowed the present algorithm to be written in a simple, recursive way for an arbitrary number of grid levels.

Figure 9 shows an example of a timestep using three levels. The computational overhead for the application of the refinement algorithm is contained in the “exchange data” blocks. The populations of overlapping nodes can simply be rescaled, while others need to be reconstructed using interpolation. Compared to the original implementation, the interpolation stencil was modified to allow for better coupling between grid levels in case of local disturbances, therefore making it more stable. This interpolation is currently a series of three sequential steps. It was attempted to condense this into a single step, but the alternative interpolation method required for the single-step procedure was susceptible to checkerboard solutions due to a reduced stencil.

The order of computations in Figure 9 suggests that multiple levels run at the same time. While the solver is free to do this, in most cases the individual levels are large enough to saturate the GPU, therefore leading to a sequential scheduling.

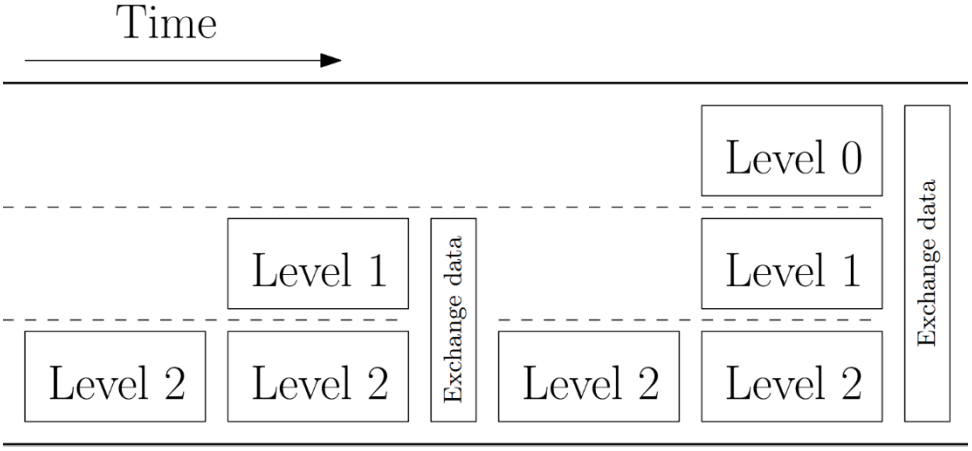


Figure 9 – Single time step on a grid with three levels

2.8 Single- and multi-node parallelisation

The present code is implemented using the Kokkos C++ library [33], [34] which allows the code to be compiled for a variety of different hardware setups. Kokkos is used for node-local parallelism, such as OpenMP [35] or CUDA [36], while the MPI library [37] is used for inter-node communications, allowing computation on large GPU clusters.

Running simulations on GPUs is the preferred method, as they outperform present CPUs in terms of raw parallelism. However, GPU cards are often equipped with less memory than the host device, therefore limiting the size of simulation cases which can be run. The solution to this issue is to take advantage of multiple GPU cards to share the workload. The overall problem is decomposed into multiple chunks, one per device used. Each device can then perform work on its own local data, independently of the other partitions. At some point, however, data between partitions needs to be exchanged to ensure that the simulation as a whole is still solved correctly.

Most MPI libraries nowadays offer direct support for device memory pointers, often labelled CUDA-aware MPI. This has the advantage that in order to communicate data between devices, the MPI library can directly access the device pointers and optimize the communication paths, without the data first having to be explicitly copied to a host memory buffer. Some hardware setups provide direct physical connections between devices, which then allows MPI to use these very high bandwidth connections.

In the present implementation of the code, solution computation and inter-node data communication are overlapped. This means that the code solves priority grid nodes first, then sets up data for transfer. These priority nodes are only a small subset of the overall solution domain. While the computational kernel solves the remaining grid nodes, MPI data is exchanged between devices, therefore avoiding the need to wait for communications before advancing the solution.

2.9 Single precision capability

Special attention was given to the ability of the solver to run computations equally well in both double or single precision. The advantages of using single precision can be found e.g. in lower memory requirements, allowing larger cases to be run on smaller, cheaper, or fewer GPUs. Additionally, since the LBM algorithms are mostly memory-bandwidth-bound, loading and storing less data immediately leads to performance gains.

For some cards, especially ones primarily used for gaming, using single precision provides an additional, significant boost in performance, as the number of available double precision cores is generally much lower for such models.

All results presented in this report, with the exception of the verification cases, were run using the single precision solver.

2.10 Turbulence modelling

While the current implementation using entropic MRT collision models remains stable even at very high Reynolds numbers, the collision model does not provide an eddy-viscosity-like additional diffusion in the classical sense. LES turbulence models can help model the unresolved turbulent stresses. The main difference here is that these turbulence models are not used to stabilize the solver, but to improve an already stable solution. The following pure LES models were implemented up to now:

- Static Smagorinsky model with adjustable coefficient (see [38])
- Wall-adaptive local eddy viscosity model (WALE) [39]
- Sigma model [40]

In addition, the solver was enhanced with an explicit finite-volume solver which runs simultaneously on the same grid as the LBM solution process. Using this method, additional scalar transport equations can easily be added, allowing for the solution of the BSL k-omega turbulence model [41]. This way the solver can be run essentially in full RANS mode, or the VLES (very-large eddy simulation) blending functions can be applied. Currently, the blending from [42] was implemented, but since the focus of this work lies on LES simulations, the evaluation of this method will be performed as future work.

Using the method described above, a turbulent wall function boundary condition was implemented, which can be used with any of the above turbulence models. The boundary condition is based on the iterative solution of Musker's wall profile equation [43]. In this solution process, values for all relevant near-boundary quantities can be derived and used to reconstruct missing boundary populations.

3 Results and discussion

The BFE project “Lattice Boltzmann for Hydraulic Turbomachines” was officially started in July of 2019. In the following, we present an overview of the individual work packages and discuss the completion of their respective milestones.

WP2:

The first milestone MS2.1 was concluded by implementing the basic code requirements for periodic 2D and 3D simulations, including the entropic MRT collision operator [16], which allows stable high Reynolds number simulations. In addition, the code was extended to include a dictionary-based user interface, such that various simulations can easily be set up.

The following milestone MS2.2 included the extension to arbitrary geometries, general boundary conditions and rotating lattices as well as rotor-stator interfaces. With the inclusion of rotor-stator interfaces and the possibility for multiple rotating frames, this milestone could be concluded.

Grid refinement in MS2.3 was completed by implementing a solution algorithm capable of dealing with refined grids. In addition, a grid generator software for such refined grids was developed from scratch.

The semi-Lagrangian method (MS2.4) had some work done at the beginning of the project period but was eventually abandoned due to its complexity.

The first version of an automatic wall function boundary condition was added.

In addition to the points mentioned above, features were included for the evaluation of transient statistics, custom post-processing fields (e.g. vorticity, absolute/relative velocities, etc.), boundary average calculations and force/torque evaluation. All of the above are selectable at run-time: the user can define any number of arbitrary groups of boundary patches for boundary averaging and force/torque evaluation (e.g. rotor blades + hub + shroud).

As such, WP2 could be completed.

WP3:

Node level parallelism is implemented for both CPU and GPU architectures. Inter-node parallelism was implemented via the MPI standard, which allows the code to run in a fully parallel environment and can be employed on large GPU clusters. All the simulation cases shown in this work were run on at most 4 GPUs. The grid generator was extended to tackle domain decomposition for such applications. As such, this work package is complete, with potential for future optimizations.

WP4:

Validation and verification is was completed for this project, but will see further work in the future. The completion date for MS4.1 was moved to coincide with the finalization of the project documentation, which is now finalized.

WP5:

Multiple multiphase models, including a new type of combination of a color-gradient method with the entropic MRT collision operator, were implemented. The new model allowed for simulations of very high Reynolds number two-component flows, but was limited by the density ratio. Ideas for cavitation simulation were developed, but not implemented. The multiphase capability of the solver was left at that state and not completed. This will be revisited in the future.

WP6:

An HVAC radial fan was investigated on the topic of suspected rotating stall effects at low flow rates. The occurrence of these phenomena could be confirmed using the present solver. In addition, a fully detailed simulation of the unsteady rotor-stator interaction of the ERCOFTAC pump impeller was performed and compared to measurements. The solver delivered accurate results in a very short timeframe. A model-scale pump turbine case was investigated in the S-shaped region of the machine characteristic. The instability was captured, but results will require further investigation. Best-practice guidelines for hydraulic machines will have to be developed.

3.1 Solver verification

3.1.1 Plane Poiseuille flow

In this two-dimensional configuration the laminar flow between two parallel plates is investigated. The case is translationally periodic and the flow is driven by a momentum source to counterbalance the viscous friction induced by the top and bottom walls. This flow situation has an exact solution [44], derived from the Navier-Stokes equations, and is used to verify the order of accuracy of the newly implemented boundary conditions. In addition, the case will only allow this verification, if the basic functionalities of the solver (streaming, collision, external forcing, domain integration etc.) are working as intended. Figure 10 shows a sketch of the situation.

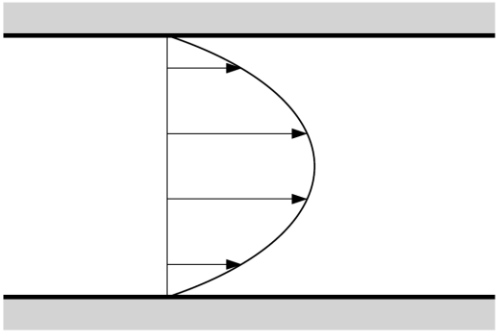


Figure 10 - Poiseuille flow sketch

The case was run in three configurations, in order to investigate the sensitivity of the interpolation point locations, as well as the location of the boundary nodes with respect to the physical wall (for reference, see Figure 2):

1. Δp = 1.5 Δx, $\Delta b \approx 0$, boundary node very close to the physical location of the wall, but still in the fluid domain.
2. Δp = 1.5 Δx, $\Delta b \approx 1$, boundary node just under 1 Δx away from the wall.
3. Δp = 1.3 Δx, $\Delta b \approx 0$, boundary node again very close to the wall.

The grid was refined from 16 to 128 grid nodes over the channel height H and the resulting velocity profile compared to the analytical solution. Figure 11 shows the evolution of the 2-norm $\|e\|_2$ of the velocity magnitude error for all three cases.

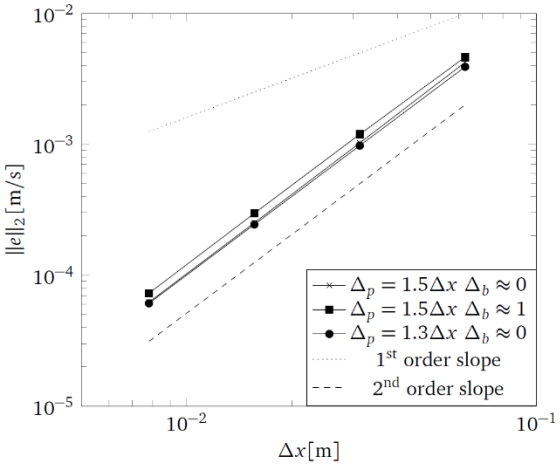


Figure 11 - Velocity error convergence for plane Poiseuille flow

3.1.2 Pipe Poiseuille flow

The second verification case presented here is also a forced, periodic and laminar flow but now applied to a three-dimensional pipe geometry. In contrast to the previous simulation, the solver now has to provide boundary conditions for a curved surface, while still ensuring that the basic functionality performs well in three dimensions. See Figure 12 for an example setup using 64 grid nodes across the diameter.

The solution to the flow field can again be derived analytically and was used to evaluate the simulation errors. The code shows second order convergence for all three cases in the plane Poiseuille flow as well as for both single and double interpolation point versions in the three dimensional pipe flow (Figure 13). For the coarsest cases using two probes has a beneficial effect on the error levels, but as the grid is refined, these differences reduce significantly.

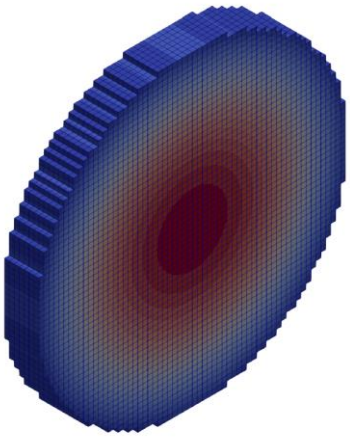


Figure 12 - Poiseuille pipe flow at a resolution of 64

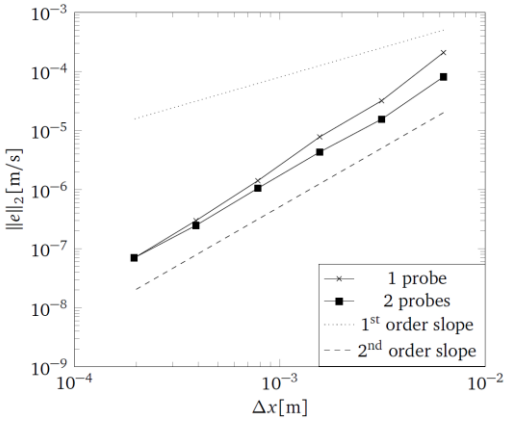


Figure 13 - Velocity error convergence for pipe Poiseuille flow

3.1.3 Taylor-Couette flow with rotor-stator interface

Taylor-Couette flow is a flow configuration between two concentric cylinders, rotating either alone or simultaneously. This represents a minimal setup for rotor-stator systems as they occur in turbomachines. It would be possible to solve this problem in the absolute reference only, using no-slip walls with circumferential velocities. The intent here, however, was to test the rotor-stator interface boundary condition, as well as the forcing terms in the rotating inner domain.

At low Reynolds numbers, i.e. laminar flow, exact solutions for the radial velocity, pressure and shear stress distribution exist (see e.g. [44]). Figure 14 shows a sketch of the simulation setup including the location of the rotor-stator interface, which is half-way between the inner and outer cylinders and marked by the dashed line. The domain below the interface is therefore solved in the relative system.

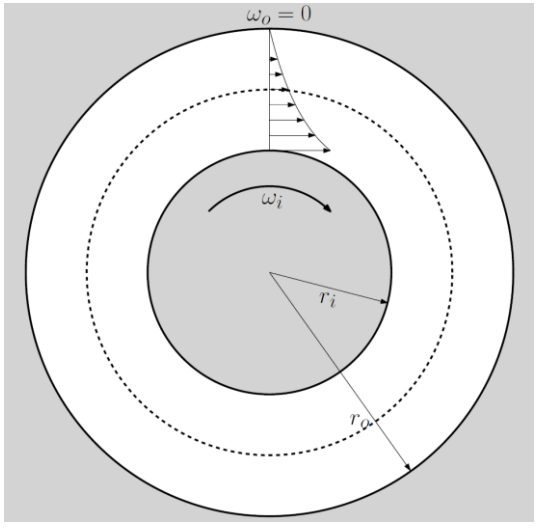


Figure 14 - Taylor-Couette flow sketch

Figure 15 shows a close-up of the refined grid used here. It has to be noted that this level of grid refinement is not necessary for this case and is simply used here to show that the solver can handle such setups.

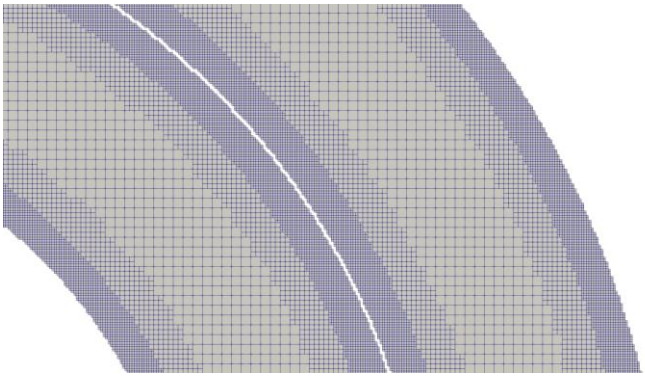


Figure 15 - Taylor-Couette refined grid

Figure 16 to Figure 18 show the comparison of the simulation data with the exact solutions of pressure, velocity and shear stress, respectively. The gaps in the plots indicate the position of the interface and changes in grid levels. The exact pressure was chosen such that the value at r_o is zero and the simulation pressure, given by $p = \rho\, c_s^2$, used the resulting boundary point pressure as the zero-reference. The agreement with the exact solution for both the circumferential velocity and the shear stress is very good. Minimal deviations can be observed in the pressure in the rotating domain. This will be investigated further.

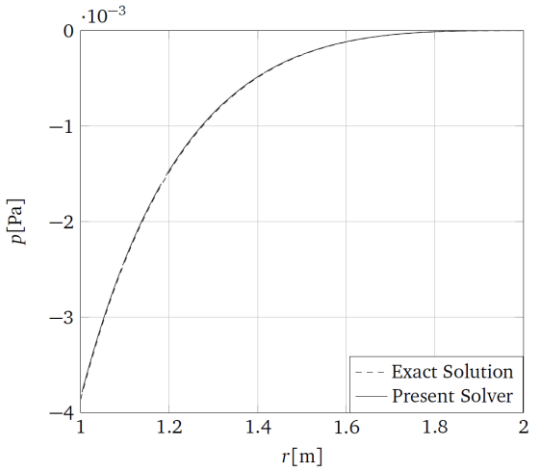


Figure 16 - Pressure

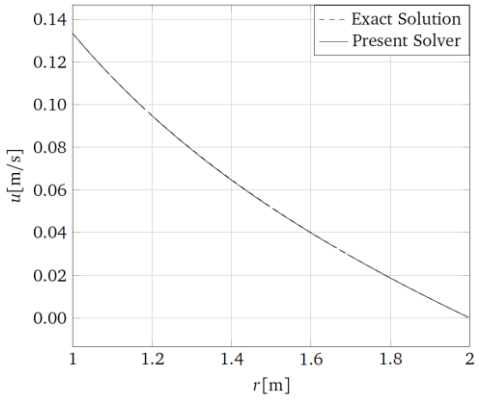


Figure 17 - Velocity

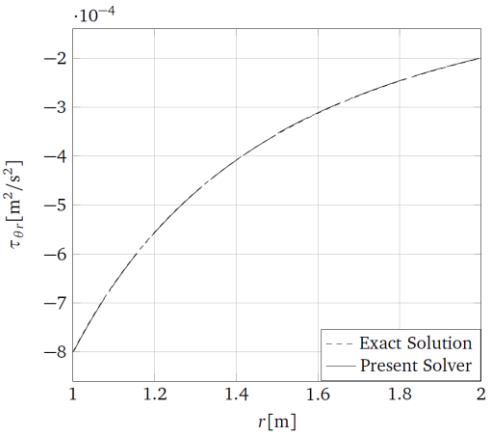


Figure 18 - Shear stress

In addition, the analytical solution provides an expression for the torque on the inner cylinder. For the present case, where ω_i = 0.13333 [s^{-1}], r_i/r_o = 0.5 and ν = 0.0045 [m^2/s], the exact torque value and the value obtained in the simulation, using the stress integration provided by the new boundary conditions, are:

Exact	-0.010053 [Nm]
Present	-0.010075 [Nm]

3.2 Validation

3.2.1 Lid-driven flow in a cavity

As part of the validation of the basic code features, the two-dimensional laminar flow in a square cavity was simulated. The case was studied by Ghia et. al [45] during the development of multigrid methods for incompressible Navier Stokes simulations. It has served as a reference solution for code validation in many publications.

Here, fluid is located in a square domain bounded by walls. The top wall has a tangential velocity component, while all other walls are at rest. The following sketch shows the overall setup of the simulation domain:

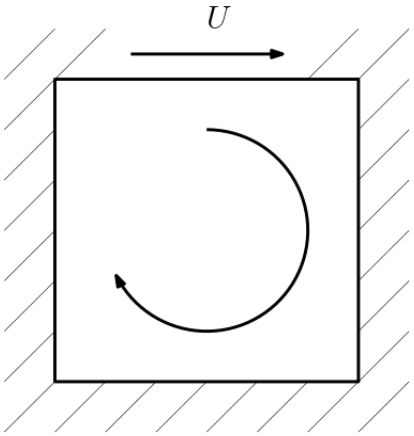


Figure 19 - Lid-driven cavity

The Reynolds number for this case is defined as

$$Re = \frac{LU}{\nu} \tag{7}$$

where the length scale L is taken as the edge length of the square domain and the velocity scale is given by the velocity of the top wall. Here we show the results of a single Reynolds number of 100 on a uniform grid with a resolution of 128 x 128 nodes. Figure 20 below shows the resulting flow field of the asymmetric vortex, along with counter-rotating corner vortices in the lower half of the domain.

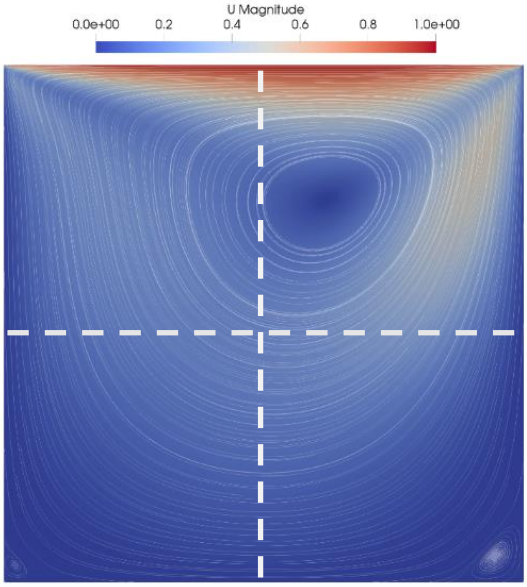


Figure 20 – Cavity flow field

The horizontal and vertical velocity components were evaluated along the dashed lines indicated in Figure 20 above. In addition to the results obtained on a uniform grid, a variant with a coarsened grid was added to this evaluation. The grid can be seen in Figure 21 below. Note that the area close to the walls has the same resolution as the uniform grid, while the core resolution is reduced by a factor of two. This reduces the overall mesh size, therefore reducing computational effort, but still producing good results.

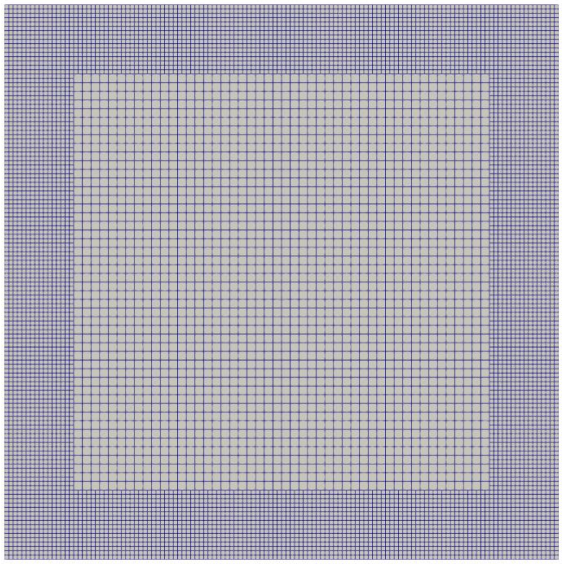


Figure 21 - Coarsened grid of lid-driven cavity

Figure 22 and Figure 23 below show these velocity components in comparison with the original data from [45]. The thick vertical lines represent the position of the grid level transition.

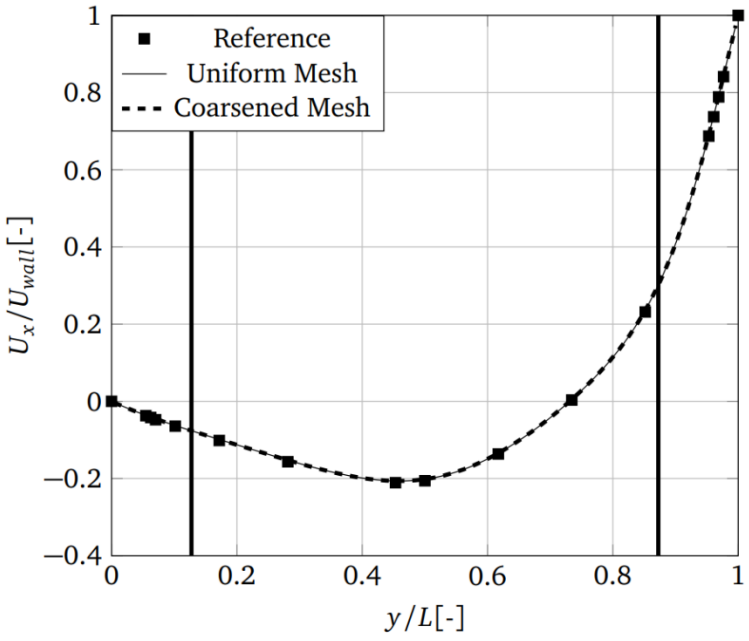


Figure 22 - Velocity in x-direction along y-slice

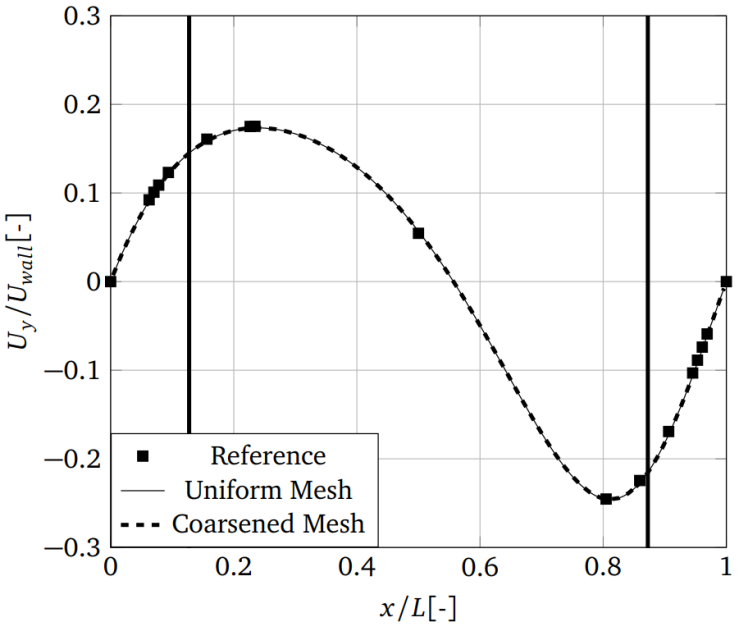


Figure 23 - Velocity in y-direction along x-slice

The results agree very well with the reference solution. The solutions on the uniform fine grid and the one obtained with a coarser core domain are almost indistinguishable, and no artifacts are visible at the grid transition regions.

3.2.2 Flow around cylinder

This simulation shows the flow around a cylinder in free stream. Figure 24 below shows a sketch of the simulation domain:

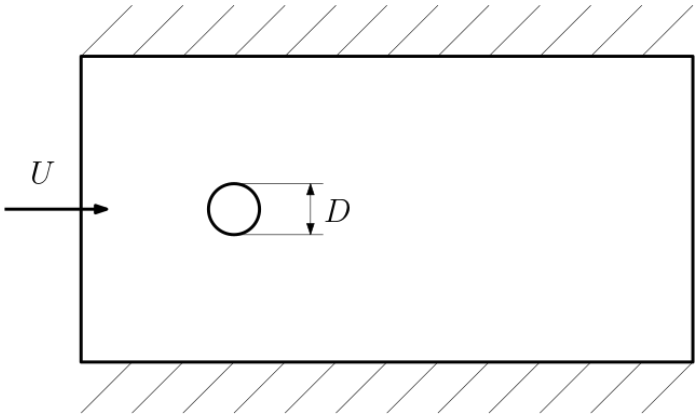


Figure 24 - Cylinder case setup

The results are shown here for three different Reynolds numbers of 20, 100 and 500. The Reynolds number is defined using the free-stream velocity and the diameter of the cylinder, which in this case is 0.1 [m]. The lowest Reynolds number results in steady, laminar flow, while in the medium case, a vortex street will start to form. In case of the highest Reynolds number, turbulent structures will start to appear in the wake of the cylinder. For all cases, the drag coefficient and Strouhal numbers are compared to simulations and simulations of other authors. The drag coefficient relates the drag force on the cylinder to the dynamic free stream pressure and is defined as

$$\mathrm{C_d} = \frac{F_d}{1/2\rho U^2 A} \tag{8}$$

The Strouhal number characterizes the dimensionless frequency of the intermittent flow separation from the cylinder surface. It is defined as

$$St = \frac{f\,D}{U} \tag{9}$$

where f is the separation frequency, D is the cylinder diameter and U is the free-stream velocity.

This simulation was run using two types of grids: a uniform grid, and a coarsened grid. In both cases the diameter of the cylinder was resolved with only 10 grid nodes. Figure 25 below shows a snapshot of the instantaneous vorticity distribution in the uniform grid, Re = 500 case as a volume plot.

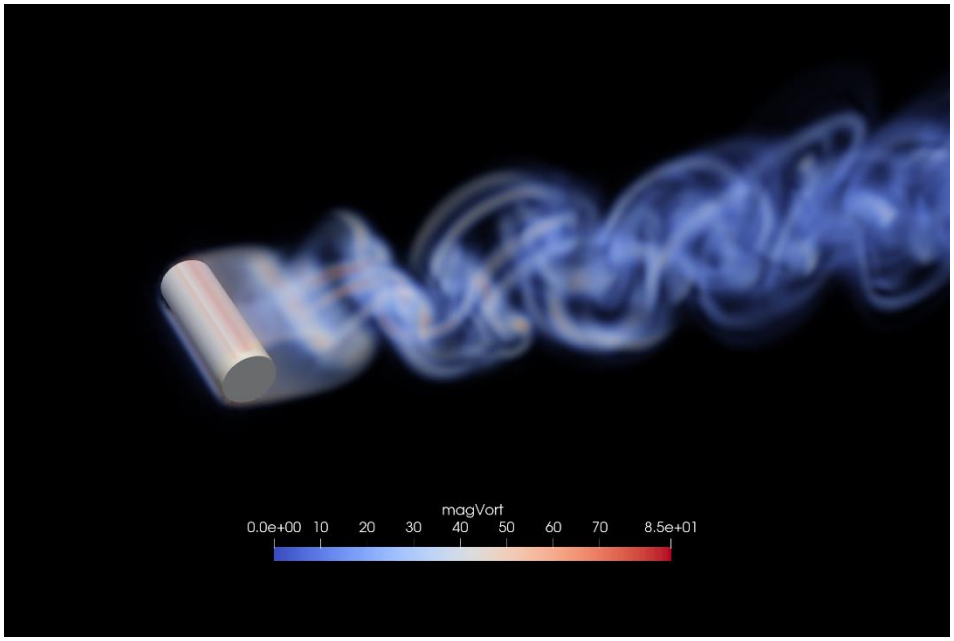


Figure 25 - Cylinder wake vorticity

Figure 27 and Figure 28 below show the respective comparisons to reference data. The reference simulations were performed in [15] and the references to the measurements are also contained therein. These data points serve as the greyed-out background of the diagrams which are overlaid with the results of the present solver. The Strouhal number for $Re = 20$ is 0, as there is no periodic vortex formation.

The uniform-grid results agree well with the reference simulations and the measurement data for both the drag coefficient and the Strouhal number. This shows that the implemented boundary conditions are able to deal with grid-aligned boundaries as well as such general “off-grid” boundaries. They deliver accurate force results for the investigated Reynolds number range, even at a very low grid resolution of the cylinder. As mentioned above, in addition to the uniform-grid solutions, a second grid was run using the refinement procedure. The resolution of the cylinder was kept the same as in the uniform case, but the surrounding domain was coarsened by one level. A closeup of the grid can be seen in Figure 26. For that case the drag coefficient was again in good agreement with reference data, while the results of the coarsened grid are not visibly worse than the uniform grid. However the number of nodes were lowered by >85%, making the simulation much faster to compute.

The uniform grid case can easily be run on a normal workstation and the results over 50 seconds of physical simulation time can be obtained within five minutes of calculation time.

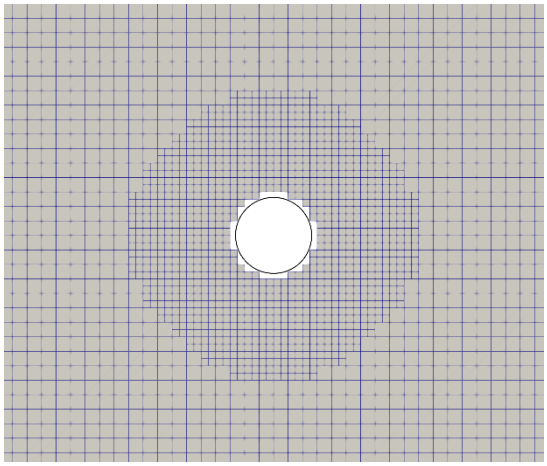


Figure 26 - Close-up of coarsened mesh in cylinder case

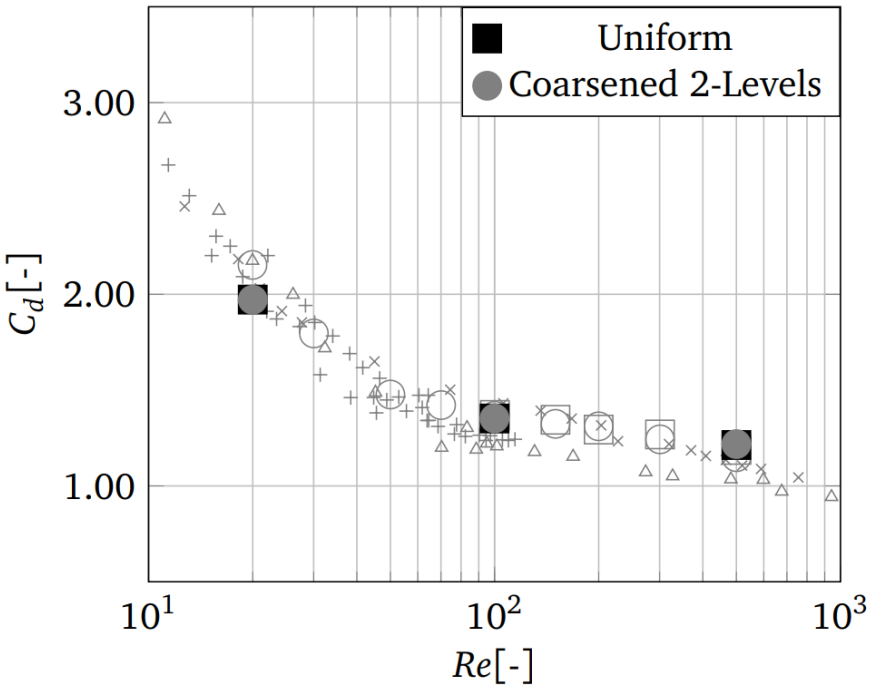


Figure 27 - Cylinder drag coefficient at Re = 20, 100 and 500

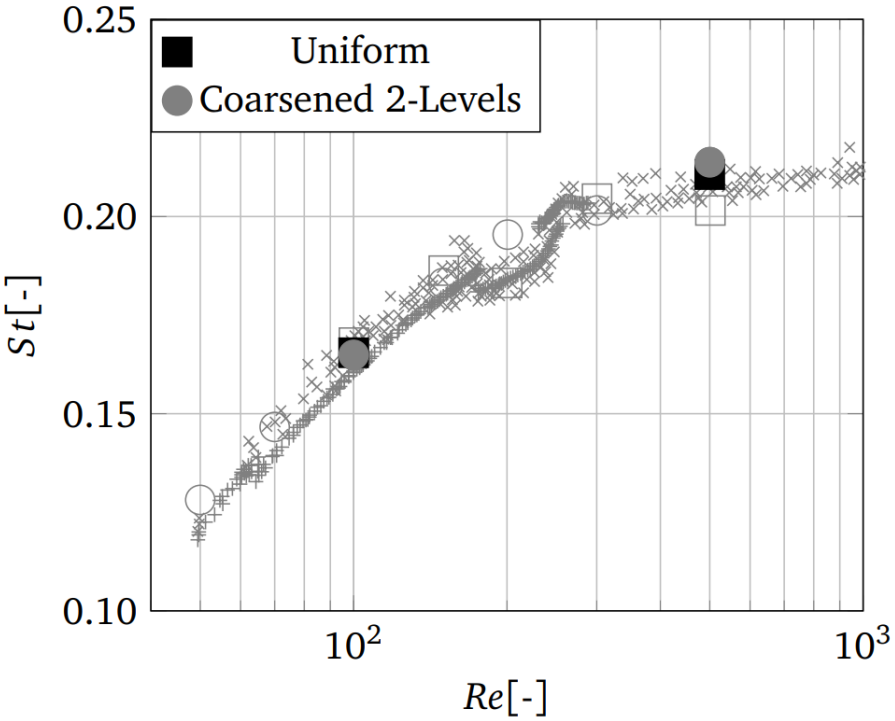


Figure 28 - Cylinder Strouhal number at Re = 100 and 500

3.2.3 Flow between parallel plates

One of the most important cases for evaluating a code's ability to accurately represent wall-bounded flows is the flow between parallel plates. The reference DNS solution by Moser et al. [46] is publicly available and has therefore widely been used for code validation in the fields of LES solvers as well as wall functions in RANS turbulence modelling.

The case mimics flow between two infinitely large parallel plates, as shown in Figure 29 below. The two plates are stationary wall boundaries and the flow is driven by an external acceleration to compensate for the friction losses of the walls and maintain a constant average velocity. The infinitely large domain size is modelled by using periodic boundaries in streamwise and lateral directions.

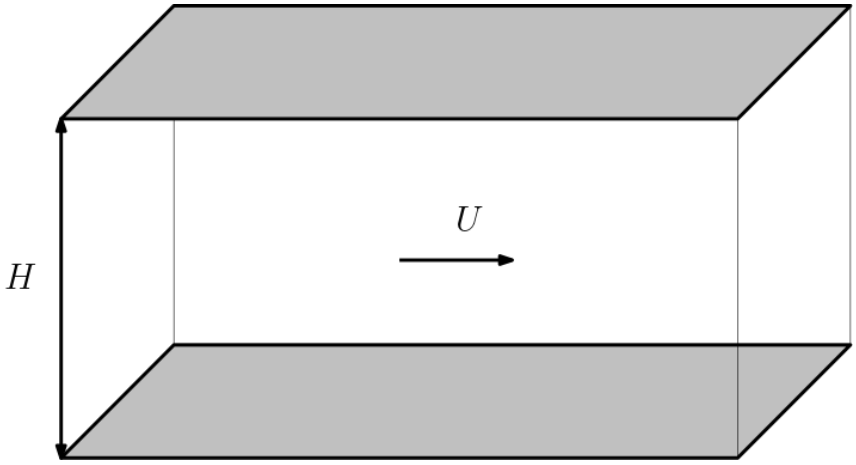


Figure 29 - Flow between parallel plates setup

The flow is characterized by the friction Reynolds number defined as

$$\mathrm{Re}_\tau = \frac{hU_\tau}{\nu} \tag{10}$$

where h is the channel half-height $(H/2)$ and U_τ is the friction velocity defined as

$$U_\tau = \sqrt{\frac{\tau_w}{\rho}} \tag{11}$$

with τ_w being the wall shear stress.

Here we present the results for a simulation at Re_τ = 395, leading to an average velocity of 0.138 [m/s], run on uniform grid using 129 grid points over the total channel height, as well as two coarsened grids consisting of two and three grid levels. Figure 30 shows the two coarsened grids used for the present simulation. All three grids use the same resolution near the walls.

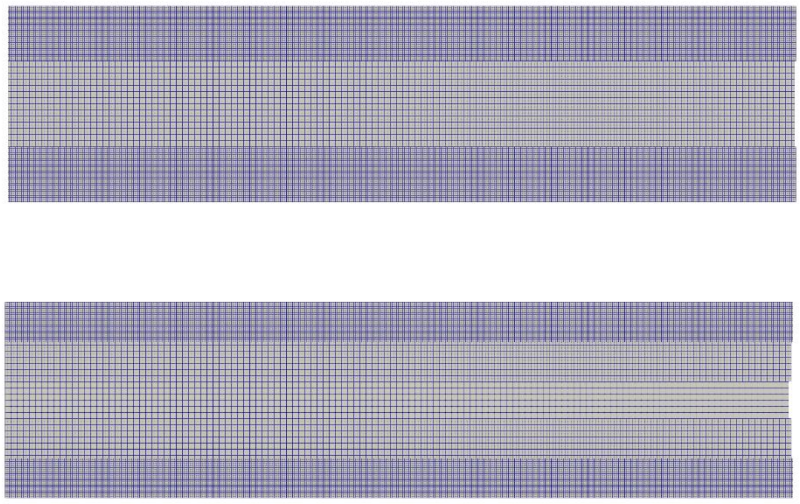


Figure 30 - Grid of 2- and 3-level channel setup

Figure 31 shows a snapshot of the instantaneous flow field along with surface contours of constant vorticity, which highlight the turbulent structures close to the walls. This figure was obtained on a coarser, uniform grid.

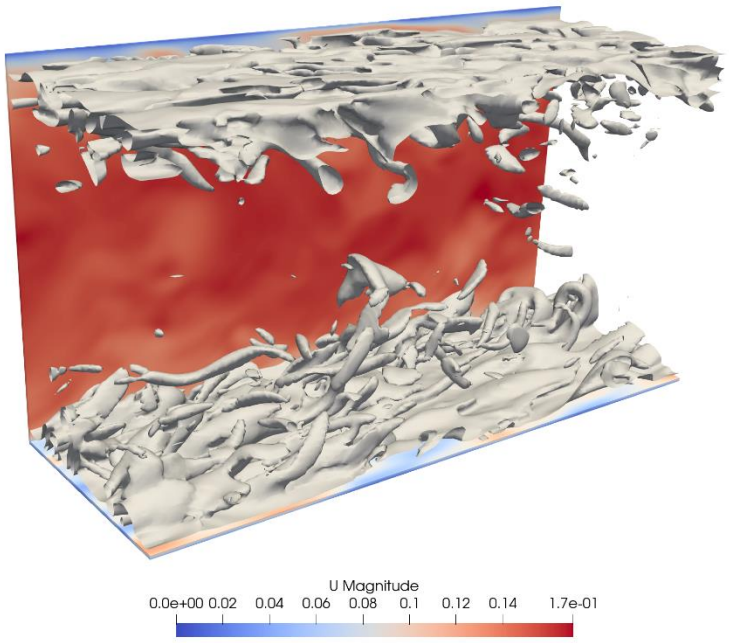


Figure 31 - Instantaneous velocity and vorticity from coarse grid uniform grid solution

Figure 32 to Figure 35 below show the time-averaged velocity profiles for both the uniform and the two coarsened cases, as well as the velocity fluctuations and Reynolds shear. All data is shown in comparison with the DNS results by Moser et al. [46]. The results are presented as dimensionless quantities, whose definitions are as follows:

$$y^+ = \frac{\Delta y\, u_\tau}{\nu} \tag{12}$$

$$u^+ = \frac{u}{u_\tau} \tag{13}$$

$$u'^+ = \frac{\overline{u'}}{u_\tau} \tag{14}$$

$$v'^+ = \frac{\overline{v'}}{u_\tau} \tag{15}$$

$$w'^+ = \frac{\overline{w'}}{u_\tau} \tag{16}$$

$$u'v'^+ = \frac{\overline{u'v'}}{u_\tau^2} \tag{17}$$

Here Δy is the physical distance from the wall, u is the streamwise velocity and u_τ is the friction velocity, calculated from the wall shear stress. All primed quantities represent fluctuations of the respective velocity components. The $\overline{u'v'}$ covariance represents the Reynolds shear stress divided by the viscosity and is normalized with the square of the friction velocity.

The agreement of the results with the reference DNS solution for the velocity profile is good for all cases.

The second order statistics shown in Figure 33 and Figure 34 agree with the DNS results, but the $\overline{u'}$ fluctuations, as well as the $\overline{u'v'}$ covariance is slightly underpredicted, which can be attributed to the relatively large first wall distance.

The three-level case shows some deficiencies in the wake region for the velocity fluctuations, which is simply due to the coarseness of the grid in this region. Artifacts due to grid changes are barely noticeable.

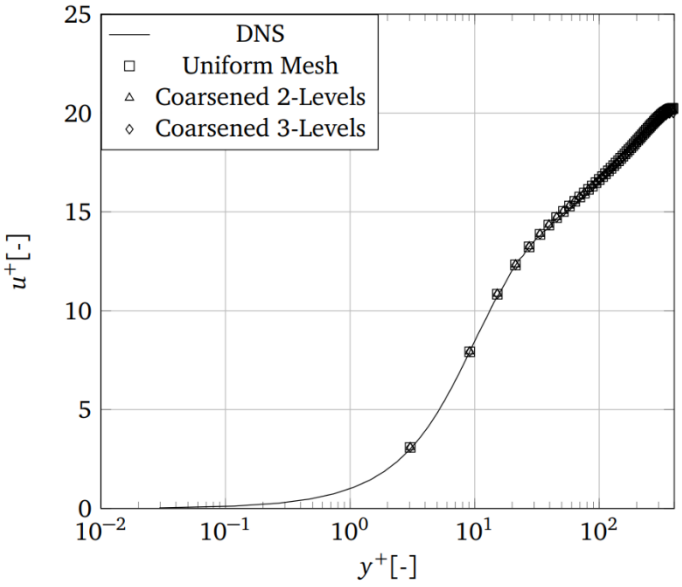


Figure 32 - Dimensionless velocity profile for all 3 cases compared to DNS

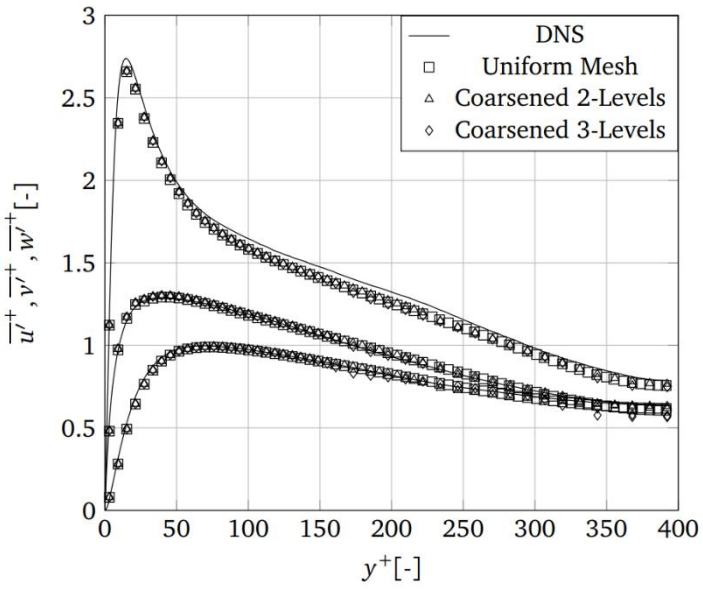


Figure 33 - Dimensionless velocity fluctuation profiles for all 3 cases compared to DNS

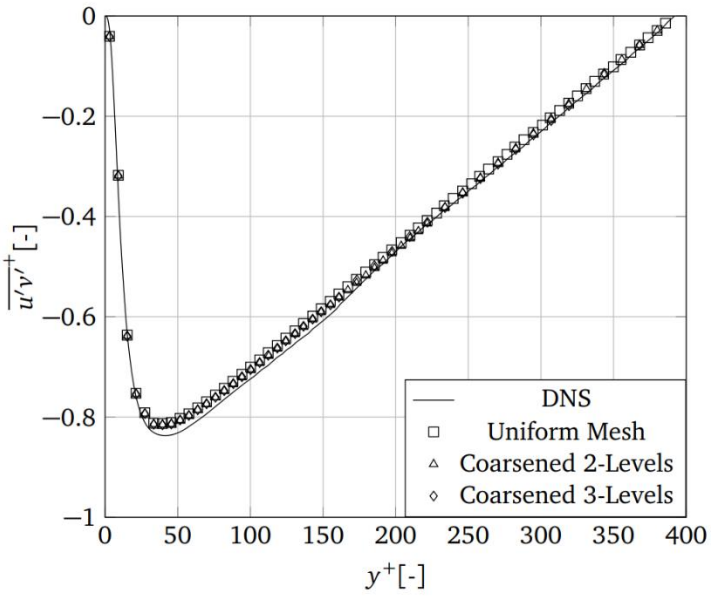


Figure 34 - Dimensionless Reynolds shear component for all 3 cases compared to DNS

3.2.4 Flow between parallel plates using a wall function

In order to test the implemented wall function boundary condition, the channel case was re-run at significantly higher Re_τ values, using a very coarse mesh. All cases presented below were run on a mesh consisting of only 21 cells across the full channel height H. At Re_τ = 5200, the bulk Reynolds number based on H and the average flow velocity becomes Re_b = 125'000. In this case the first grid point is located at a y^+ value of 245. For all wall function cases the static Smagorinsky turbulence model was applied, using a constant of C_s = 0.2. This is a rather high value, but acceptable, considering the large Reynolds numbers and very low mesh resolution. The results of the time-average wall profiles in comparison with DNS results from [47] in Figure 35 show that the wall function delivers good results for the wall friction and its effect on the core flow. Some oscillations in the first 2 grid nodes can be observed, which will be investigated and improved in the future. Deviations are also visible in the wake region of the wall profile, which can be attributed to the coarseness of the grid.

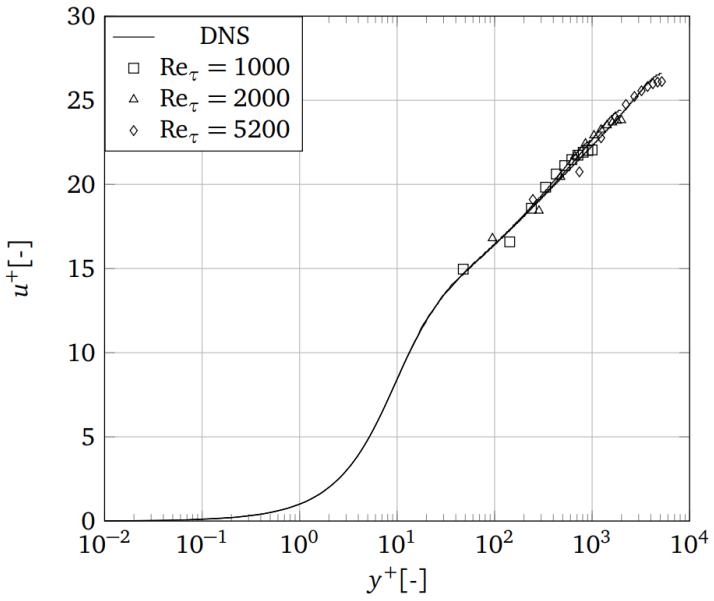


Figure 35 - Velocity profiles of wall function solution compared to DNS

3.2.5 NACA0012 tip vortex

This simulation of the flow around a 3D NACA0012 airfoil in a constrained channel serves as a popular test case for turbulence models in a RANS context (see e.g. [48]). The focal point of the investigations lies on the development of the wing tip vortex and its evolution downstream of the wing profile. Most RANS models are unable to sustain the vortex without modifications to the model formulation.

This case is particularly interesting from a numerical perspective as the Reynolds number is very high for a Lattice Boltzmann simulation. The Reynolds number, defined by the airfoil chord and the inlet velocity, is $4.6 \cdot 10^6$.

The case consists of a 3D wing profile with rounded tip attached to the side wall of a rectangular wind tunnel. Figure 36 below shows a sketch of the computational domain.

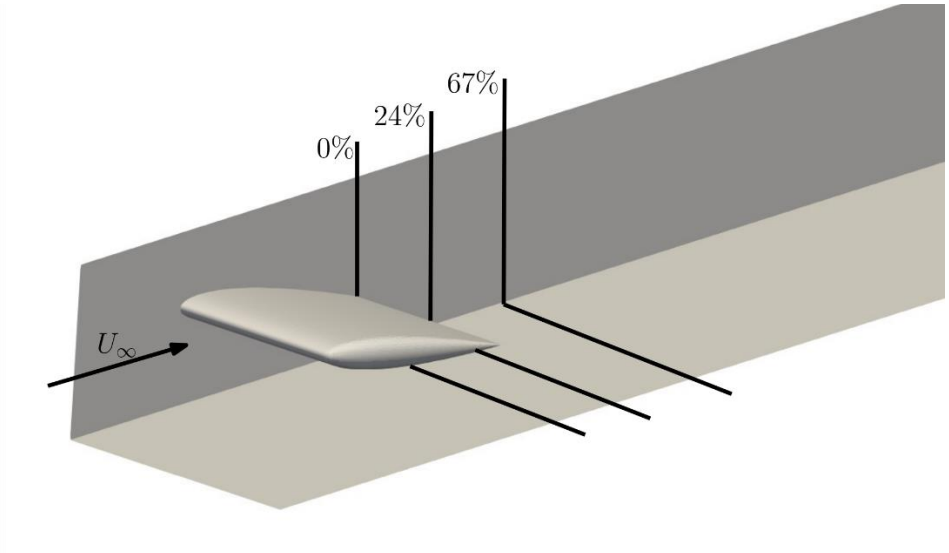


Figure 36 - Sketch NACA0012 case

Figure 37 below shows a snapshot of the vorticity magnitude as a volume plot on a coarse, uniform grid. The tip vortex can very clearly be identified.

The simulation grid consists of approximately $150 \cdot 10^6$ grid nodes

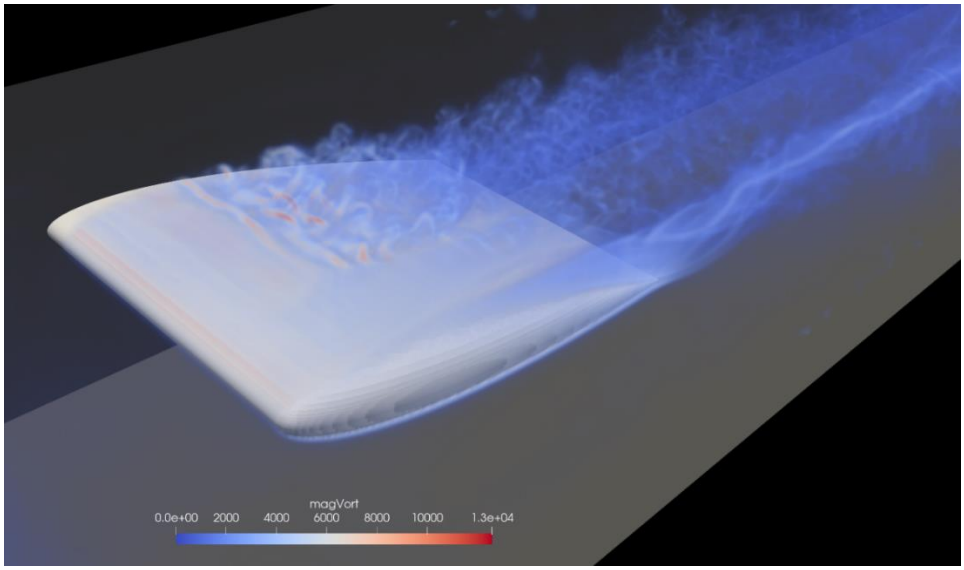


Figure 37 - Vorticity snapshot NACA0012

Figure 38 to Figure 40 below show the comparison of the crossflow velocity

$$U_c = \frac{\sqrt{U_y^2 + U_z^2}}{U_\infty} \tag{18}$$

at different positions downstream of the wing to measurements also used in [48] (references therein). The main flow direction is x and U_∞ = 58.258 [m/s] is the inlet velocity. The distances are measured relative to the airfoil chord and are set at 0%, 24% and 67%. The 0% point corresponds to the position of the trailing edge of the wing.

Overall, the vortex is captured quite accurately, especially close to the blade at 0%, where the triple-peak of the crossflow velocity can be observed. Further downstream, at 24% and 67%, the vortex size and strength remains nicely conserved, but becomes slightly asymmetric. This is most likely due to the way the vortex separates from the wing, which will need to be investigated further. The choice of LES model only has a minimal influence on the solution, which is a desirable property. Results shown here were obtained using the static Smagorinsky model, using a default constant of 0.15.

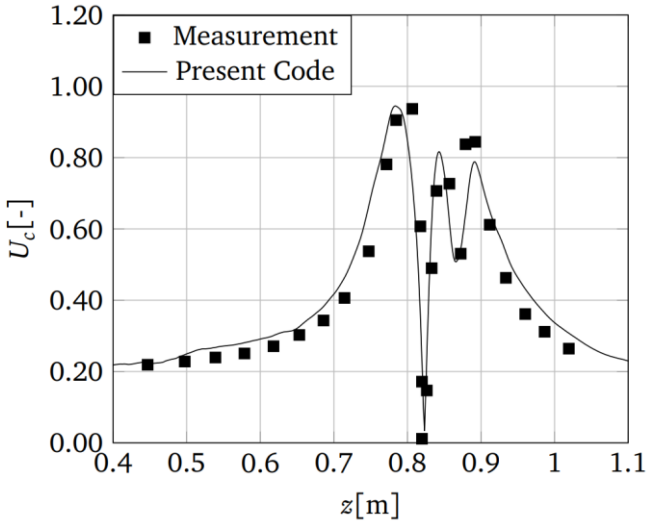


Figure 38 - Crossflow velocity at 0%

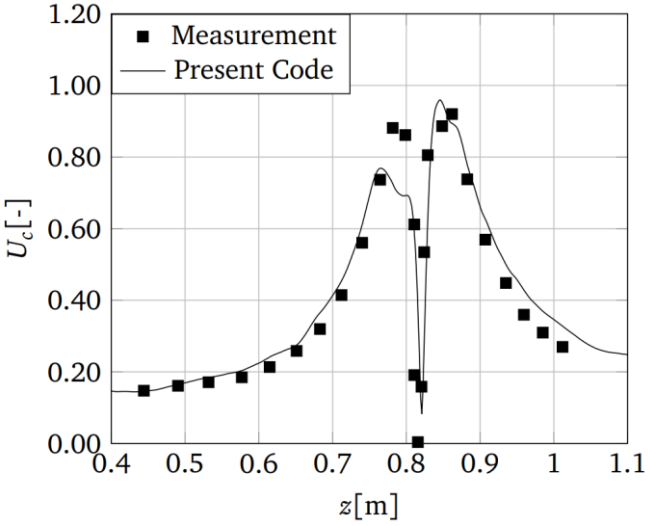


Figure 39 - Crossflow velocity at 24%

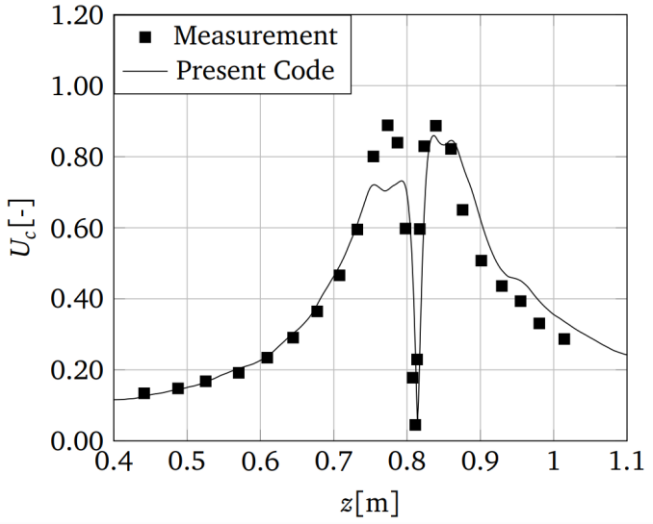


Figure 40 - Crossflow velocity at 67%

3.2.6 Turbulent conical diffuser

The turbulent flow in a conical diffuser is a challenging problem for RANS-based CFD solvers and is presented here to show the advantages of using a scale-resolving simulation method. The case involves a conical diffuser with half-angle of 14°, operated at a Reynolds number of 15'400. A periodic inlet section was one-way coupled to the diffuser domain to provide fully developed turbulent inlet flow quantities. Figure 41 shows the turbulent structures in the diffuser section, visualized by the Q-criterion of the instantaneous flow field, coloured by the velocity magnitude.

Figure 41 - Turbulent structures in conical diffuser

Figure 42 to Figure 47 show the simulation results compared to measurements performed in [49]. For visibility the u and v velocity components, as well as the turbulent stresses use different scaling factors. The presented results are recorded after an initial start-up phase and subsequently averaged circumferentially.

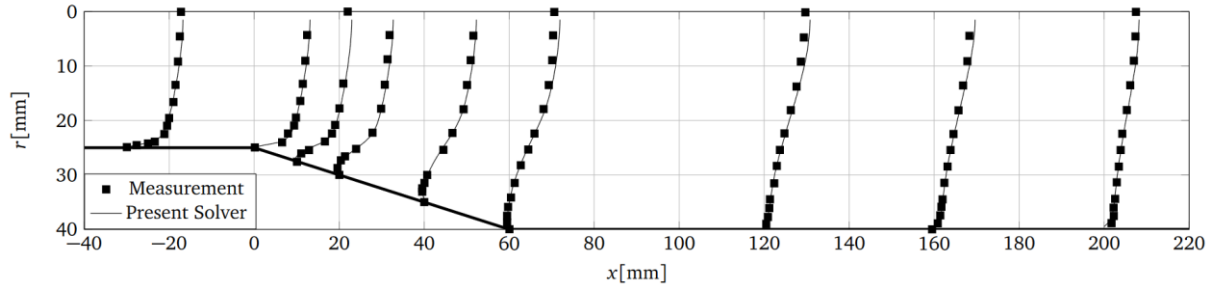


Figure 42 - Velocity x-component

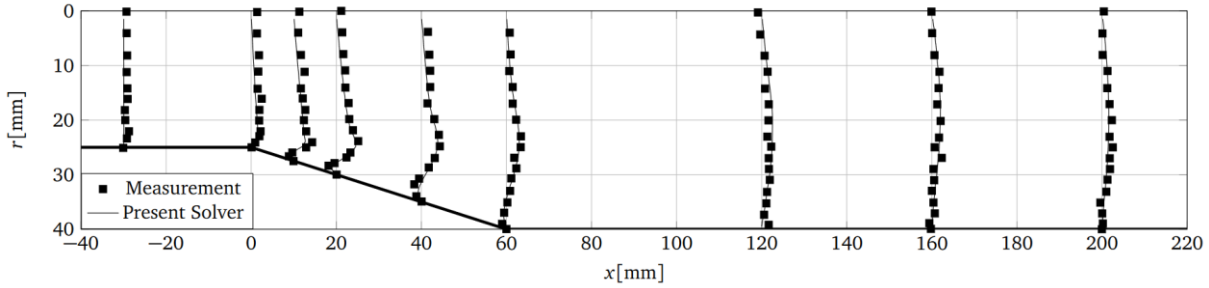


Figure 43 - Velocity y-component

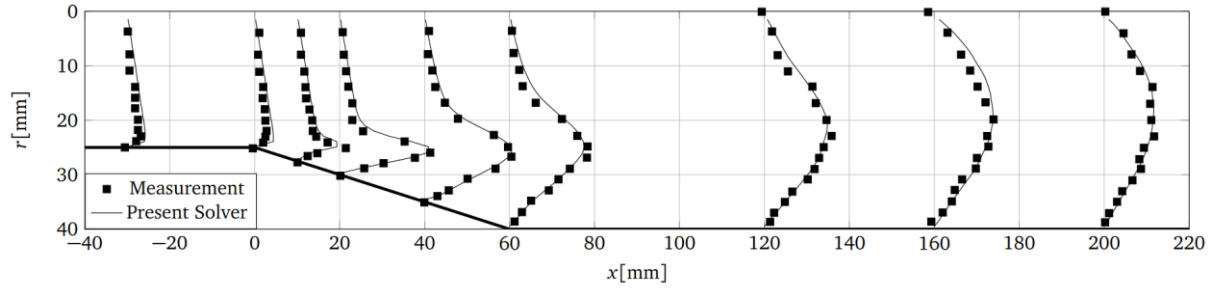


Figure 44 - Reynolds shear stresses $\overline{u'v'}$

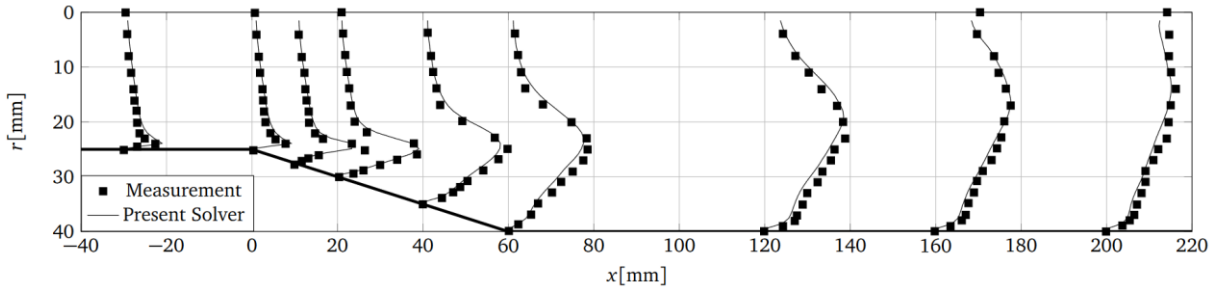


Figure 45 - Reynolds normal stresses $\overline{u'u'}$

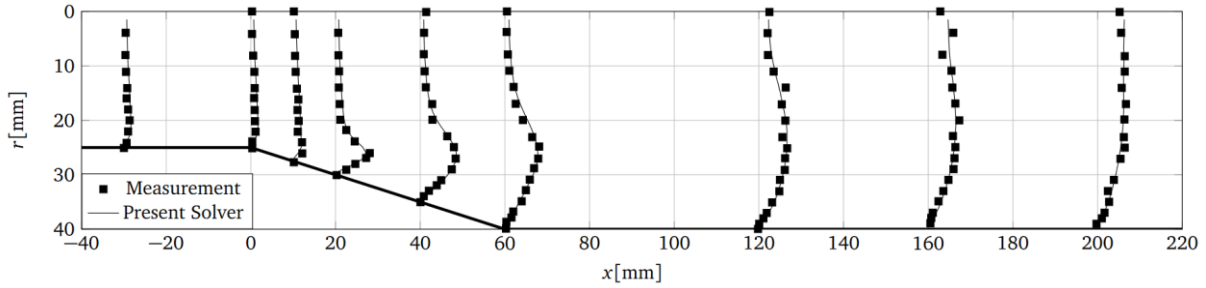


Figure 46 - Reynolds normal stresses $\overline{v'v'}$

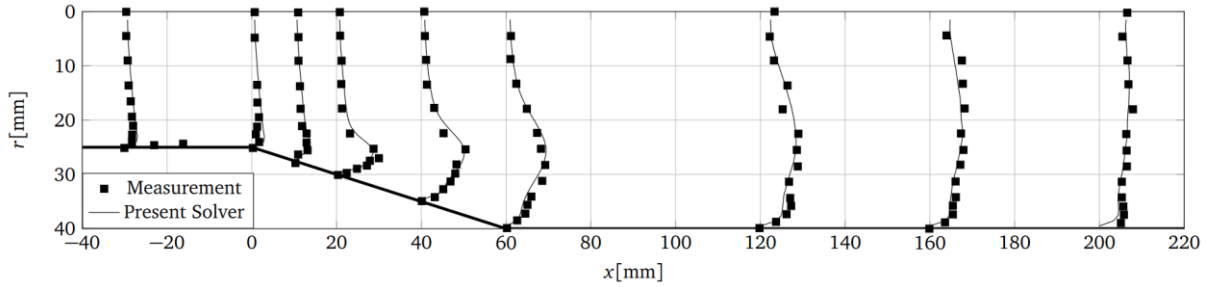


Figure 47 - Reynolds normal stresses $\overline{w'w'}$

3.2.7 Multiphase simulations

The simulation of cavitation phenomena in hydraulic machines will require a way of simulating multiple phases of the involved fluid. For this purpose, the multirange pseudopotential method by [50] was implemented into the current KBC framework. This implementation is based on the constant temperature assumption and allows the simulation of two distinct phases at a fixed density ratio. The density ratio can be selected by adjusting the temperature.

In a first step, the implementation was validated by simulating a single stationary bubble in a periodic domain and recording the resulting density ratios. The simulation was initialized with a circular hyperbolic tangent profile of two density levels (liquid and gas) close to their respective targets. During the run, the domain will evolve until it reaches a stationary equilibrium at a certain density ratio.

The resulting values for the respective densities need to follow the coexistence curve given by Maxwell's equal area rule for the method to be physically meaningful. Figure 48 below shows one example of such a bubble at the final state. The fluid density was evaluated at the center of the bubble and the gas density in the lower left corner (indicated by the two crosses).

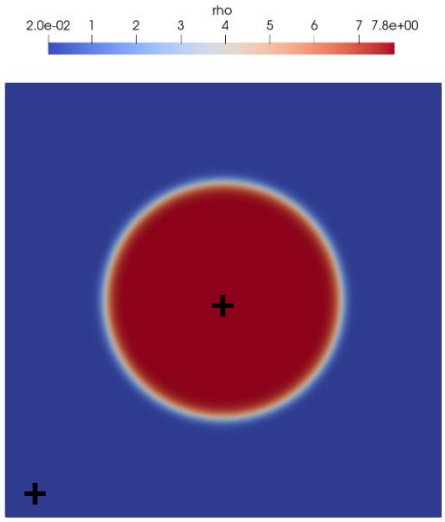


Figure 48 - Two-phase stationary droplet

Figure 49 below shows the current results of this evaluation. The greyed-out background was taken from [50] and serves as a reference for the present results. The results from the present code, up to a density ratio of ~385, are plotted over the reference data and generally show very good agreement.

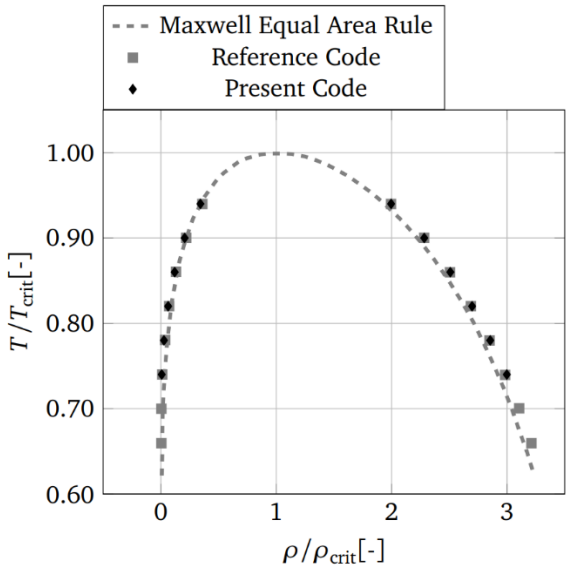


Figure 49 - Liquid-vapour coexistence curve for stationary droplet

In order to further validate the implementation, the splashing case from the same reference paper was reproduced using the present code. Figure 50 shows a sketch of the initial condition.

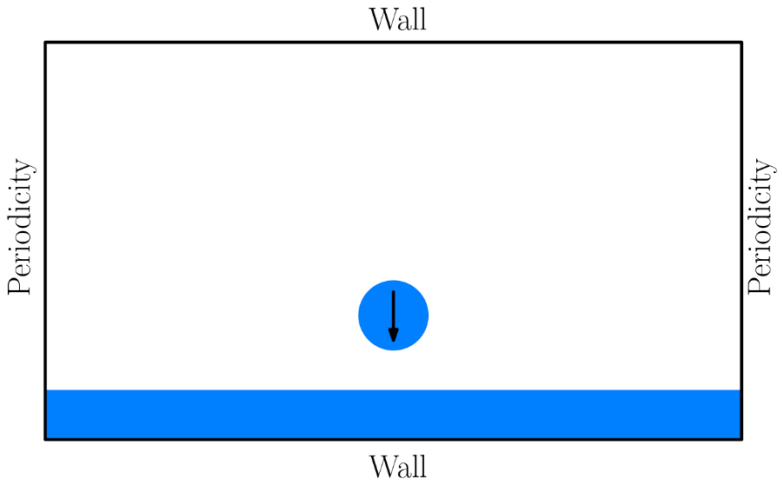


Figure 50 - Droplet splashing setup

Based on the initial droplet velocity, the fluid density, viscosity and surface tension, the case can be classified with a Reynolds number Re = 5000 and a Weber number We = 220, where

$$We = \frac{\mathrm{U}^2 \mathrm{D} \rho}{\sigma} \tag{19}$$

The snapshots of the evolution of the simulation shown below qualitatively agree well with the results presented in [50] but were not analyzed in detail.

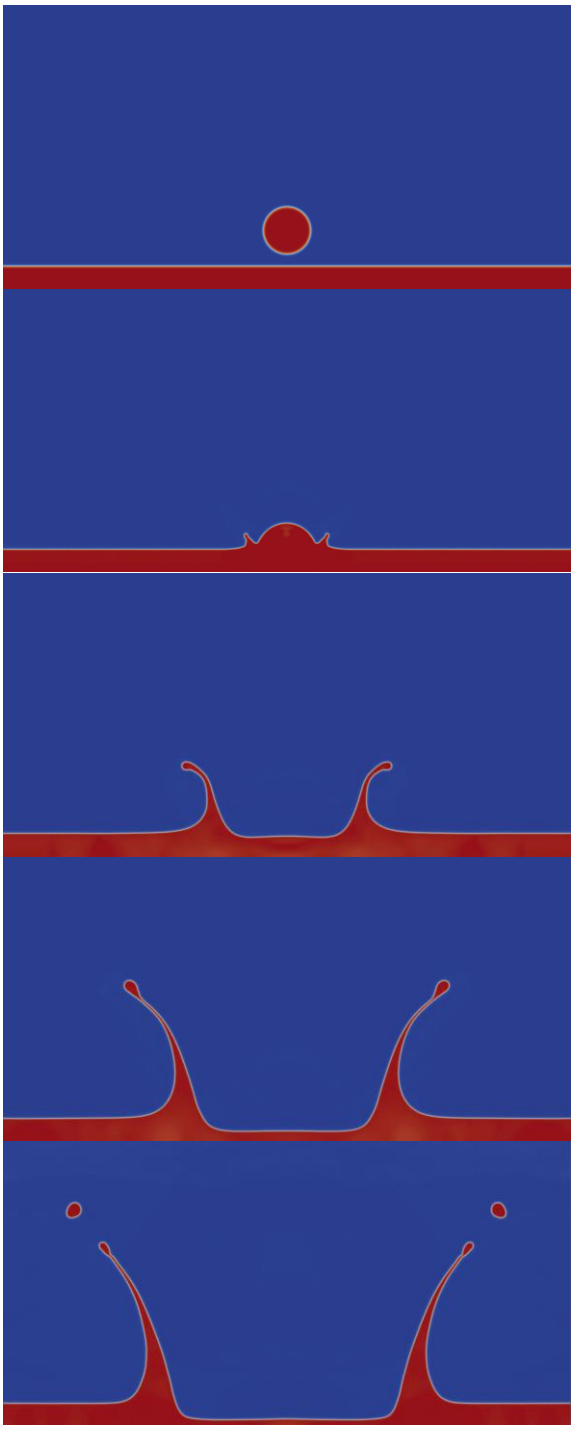


Figure 51 - Evolution of splashing simulation

In addition to the model presented above, the color-gradient method for multi-component flows, shown for example in [51], was implemented into the present code. The method has the advantage that the interface thickness is significantly reduced, leading to lower grid requirements for comparable simulations. Additionally, physical properties, such as the surface tension, can be more directly specified.

The model was embedded into the present entropic MRT method, allowing for multi-component flows at very high Reynolds numbers. Rayleigh-Taylor instability simulations could be carried out at Reynolds numbers of 10^6.

A very early test case from a jet injection at a density ratio of 3 is shown in Figure 52 below.

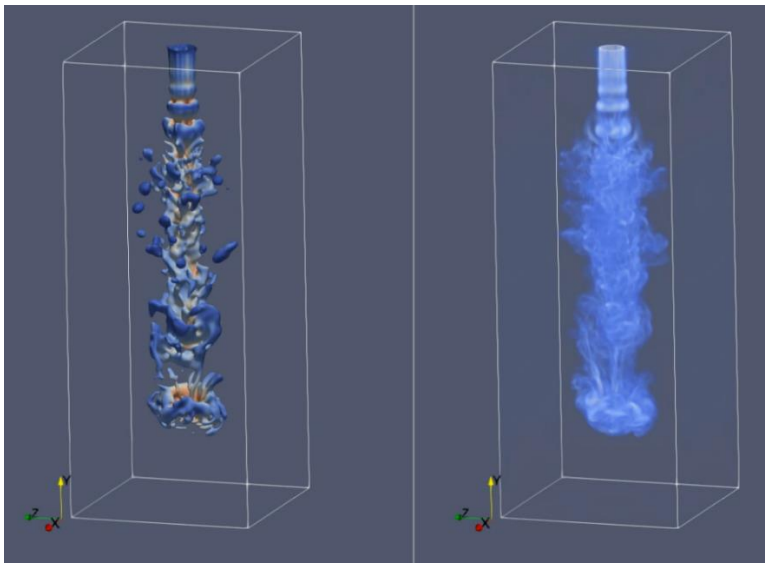


Figure 52 - Multicomponent jet flow, interface iso-surface (left) and vorticity field (right)

The presently implemented model was able to run static droplet cases at very large density ratios (1000 and higher). However, due to error terms which depend on the product of the velocity with the density ratio, simulations become unstable when high density ratio flows are moving. Ideas for a remedy were discussed, but could never be fully developed.

3.2.8 90° pipe bend

The flow through a 90° pipe bend was simulated at a Reynolds number of 5300, based on the bulk velocity and the pipe diameter D. In order to provide a fully developed, turbulent inlet flow, a recycling section was added to the inlet part, which is one-way connected to the rest of the domain. For the outlet, the non-reflective outflow boundary condition was used.

The case was run using the WALE turbulence model, without wall function boundary conditions, as the grid was sufficiently refined.

Using a bulk velocity of 1 [m/s], the case was run for 10 seconds of physical time, with averaging starting after 5 seconds. At a uniform grid size of approximately 10^7 grid points, the simulation can be run on a simple workstation GPU in about 30 minutes.

The time-averaged streamwise velocity was recorded through the pipe center at a distance of one diameter below the bend, and compared to results from a DNS computation in [52]. A sketch of the evaluation location along with a snapshot of the instantaneous velocity flow field can be seen in Figure 53.

Figure 54 shows the comparison between the present solver and the reference data. The velocity profiles of the two cases agree very well, with slight deviations near the inside wall, which may be due to the closeness of the inlet section to the bend.

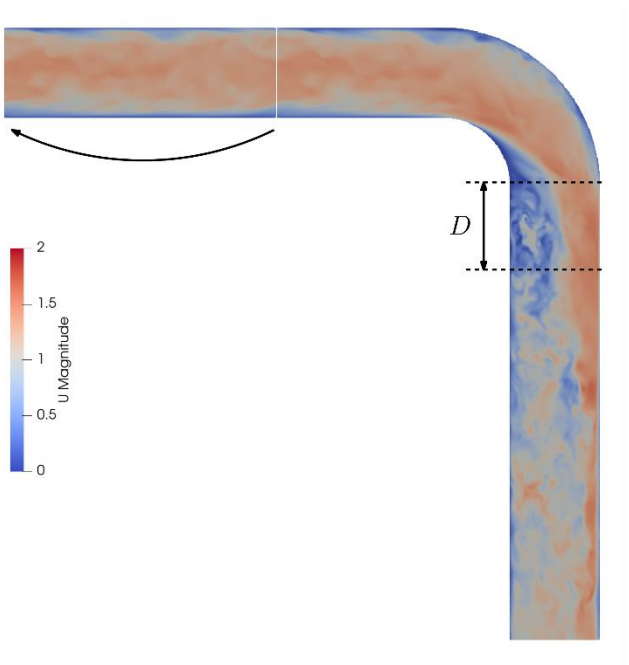


Figure 53 – Snapshot of instantaneous flow field around 90° bend

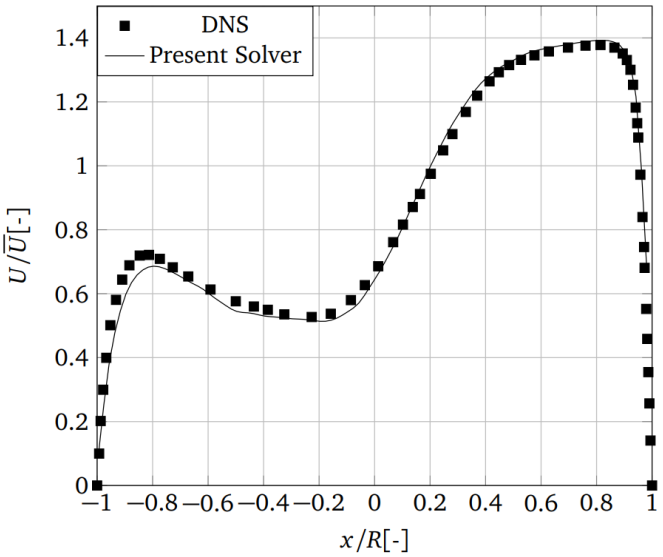


Figure 54 - Mean streamwise velocity component

3.2.9 Radial fan

The case presented in this section was provided by a client, who approached HSLU for analysis of an existing radial fan geometry, suspecting optimization potential in terms of efficiency and noise generation. While the task for the client was carried out using conventional CFD methods, it was additionally investigated using the present LBM solver, specifically at lower flow rates, where hints of rotating stall were noticed.

Figure 55 shows an overview of the investigated geometry, while Figure 56 shows the same geometry as a volumetric plot of the vorticity. The artifacts visible in the geometry overview are due to the domain partitioning.

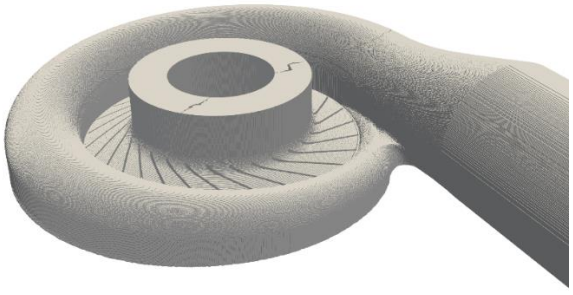


Figure 55 - Multi-level voxel grid of radial fan simulation

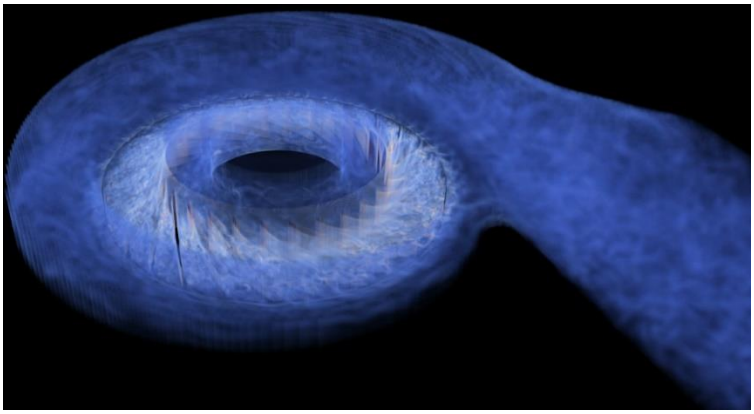


Figure 56 - Volumetric plot of instantaneous vorticity

Because the geometry and the results are proprietary, operating data cannot be described in detail, and only dimensionless results can be shown in the following figures. The Reynolds number of the case, based on the rotational speed of the impeller, was in the order of 10^5.

The simulation was carried out using a grid size of approximately 90 million grid points and was run on a workstation using two GPUs. A simulation of four impeller revolutions, two for initializing, and two for recording, can be completed in about 4 hours. Figure 57 below shows the close-up of the simulation grid.

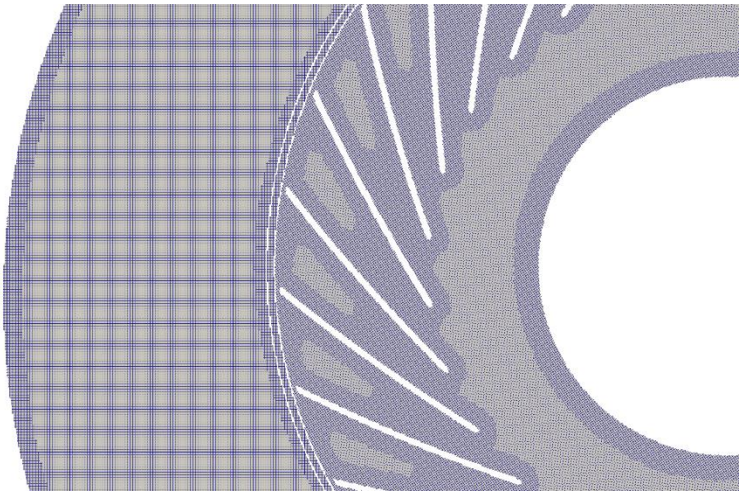


Figure 57 - Radial fan grid

Figure 58 shows the comparison of rotor torque over two impeller revolutions for two different flow rates. Q1 corresponds to approximately 70% of the nominal flow rate used in Q2. The torque level is lower for the Q1 case and low-frequency oscillations can be observed, which are largely absent in Q2. This frequency of these oscillations hints at single-cell rotating stall occurring at the lower Q1 flow rate.

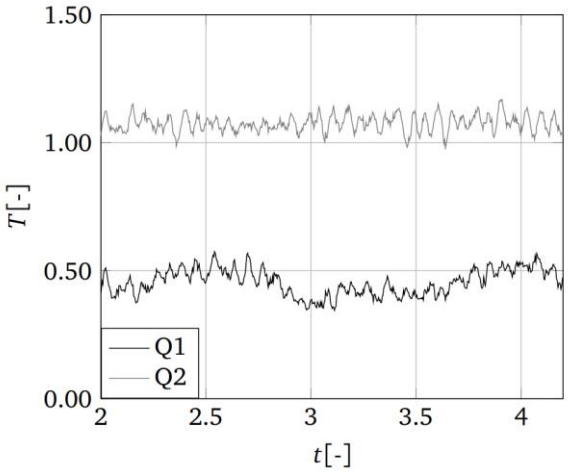


Figure 58 - Normalised torque of fan rotor shown over dimensionless time

Figure 59 and Figure 60 below show the relative-frame velocity normalised by the rotational velocity for case with the lower flow rate Q1 at two different points in time A and B. The relative-frame velocity is defined such that the velocity is taken relative to the local domain, which means that in the stationary spiral casing, the relative and absolute velocities are identical.

The black circles indicate the regions in the rotor, which are completely blocked due to the rotating stall cell. Comparing the two figures and taking into account that the partitioning of the rotor domain remains constant during the simulation, it is evident that the stall cell is moving at a different rotational speed compared to the rotor.

Unfortunately, there were no available measurements for comparison, and the other CFD simulations did not further investigate this topic.

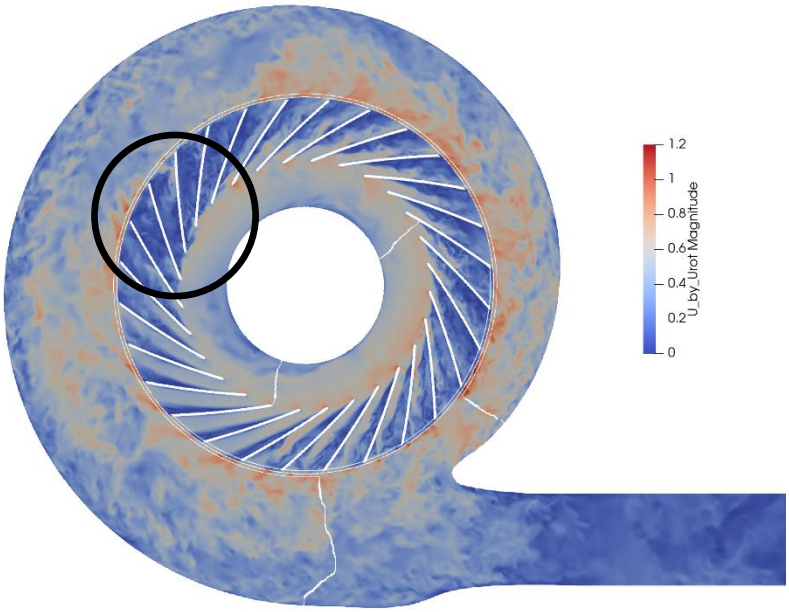


Figure 59 - Rotating stall cell in radial fan, time position A

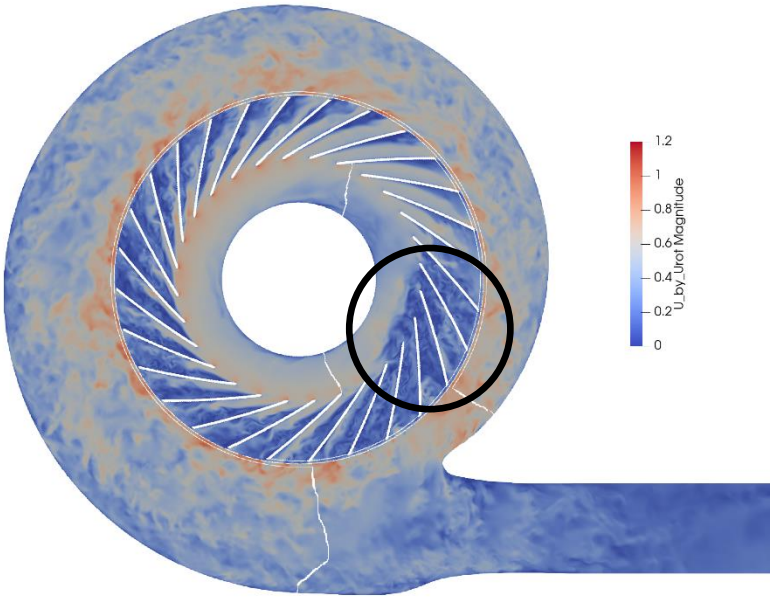


Figure 60 - Rotating stall cell in radial fan, time position B

3.2.10 ERCOFTAC radial pump impeller with bladed stator

The case presented here is a radial pump impeller originally measured by Ubaldi et al. [53]. It was used as a test case for the turbomachinery capabilities of the open-source library OpenFOAM and other commercial CFD codes. It was presented in the context of an ERCOFTAC seminar [54]. The overall pump geometry is shown in Figure 61. Figure 62 shows a detailed depiction of the purely radial impeller and diffuser blades along with the local coordinate system used in the results section below.

The position of the latter could be varied in the original experiment, both in terms of angle and radial position. Figure 63 shows the setup used in the present calculations. The machine runs at a speed of n = 2000 [min^{-1}] and a Reynolds number of 6.5·10^5, based on the impeller blade chord length c = 0.222 [m] and the tip speed at the impeller outlet U_2 = 43.982 [m/s]. For additional parameters, the reader is referred to the original publication [53].

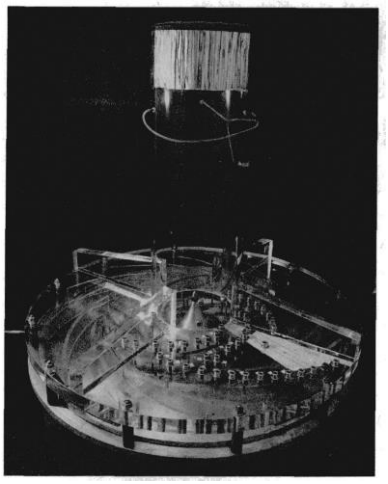


Figure 61 – Experimental setup, picture taken from [53]

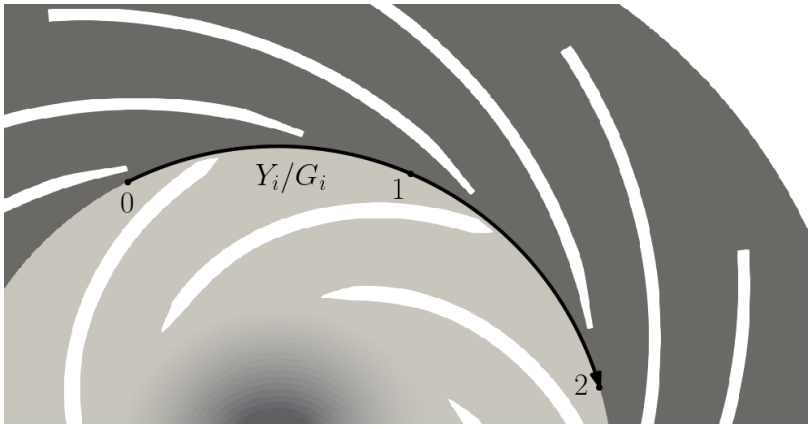


Figure 62 – Local coordinate system

The pump impeller is treated as unshrouded, but tip gaps were not resolved. The rotor domain was solved in the relative (rotating) frame of reference and was connected to the stator domain via a

sliding-grid rotor-stator interface developed in this work. The boundary condition setup for the simulation was as follows:

- Inlet: Fixed value pipe velocity profile based on the Gersten-Herwig correlations from [55].
- Outlet: Fixed-pressure non-reflecting outlet, based on [26].
- Rotor hub and blades: no-slip walls (relative system).
- Rotor shroud: counter-rotating wall (relative system).
- Stator walls: no-slip wall.

The grid resolution on the coarsest grid (level 0) was given by $\Delta x \approx 2$ [mm] and the wall boundaries and the interface were refined by two additional grid levels. This lead to an overall grid size of about 93.5 million nodes. Taking into account that, due to the refinement algorithm, more work needs to be done on finer grid, the effective mesh size (in terms of work done) can be estimated at 360 million nodes. Figure 64 shows a close-up of the computational grid in the rotor-stator region.

The simulation was run for a total of approximately 0.2 [s] of real time, equating to about 6.5 impeller revolutions. Data was gathered after two full impeller revolutions in order to eliminate initial transient effects.

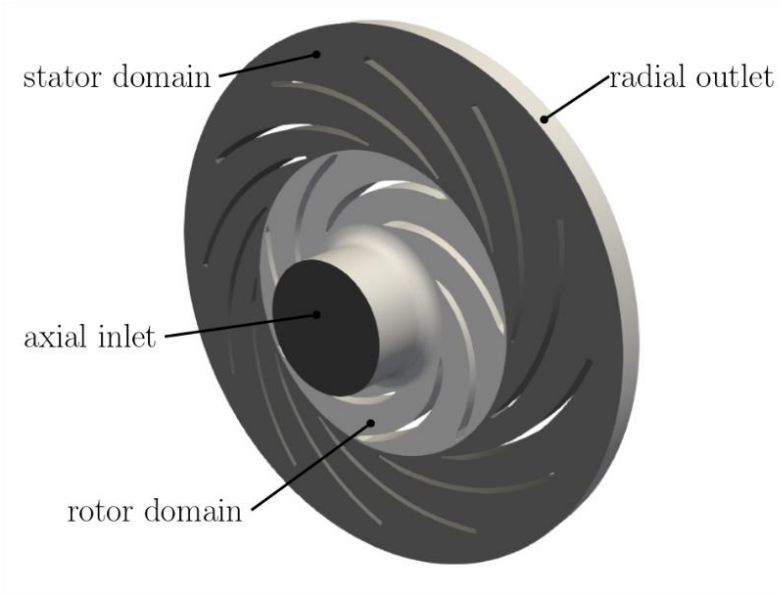


Figure 63 – Simulation setup

Figure 64 – Grid close-up

In the following, velocity profiles at the rotor outlet are investigated. The profiles are taken at mid-span at a radial location of 1.02 R_2 and cover a circumferential region of two impeller blade passages. For clarification, the local coordinate system refers to the one previously shown in Figure 62. The coordinate label Y_i/G_i, consisting of the circumferential coordinate Y_i, normalised with the circumferential pitch G_i, is again taken from the original publication [53], which also provides the measurement values used for the comparison below.

The measurements in [53] were performed on a model centrifugal compressor setup consisting of an impeller with 7 blades and an adjustable 12-blade diffuser component. The velocity distributions were

recorded in the vaneless region between the rotating and the stationary part by using a single hot-wire probe, which recorded signals over two rotor channels once per revolution. These signals were recorded for 700 revolutions and subsequently averaged. The process was then repeated for 30 different relative rotor-stator positions. The axial direction was resolved by traversing the probe using 17 points. The obtained data was then reduced to obtain the once-per-revolution averaged velocity profile over two rotor passages for the 30 relative positions of the rotor with respect to the stator. Averaging over all these positions provides the circumferentially-averaged velocity profiles. Both the radial and circumferential velocity values are normalised with the rotor tip circumferential velocity U_2 = 43.982 [m/s]. The simulation results in these locations were averaged in two different ways. The first one corresponds to the "ensemble-averaged" quantities referred to in the original paper. Figure 65 shows an example for how the measurement results are presented in the original publication, here for the radial velocity at the impeller outlet, both ensemble averaged and its circumferential average.

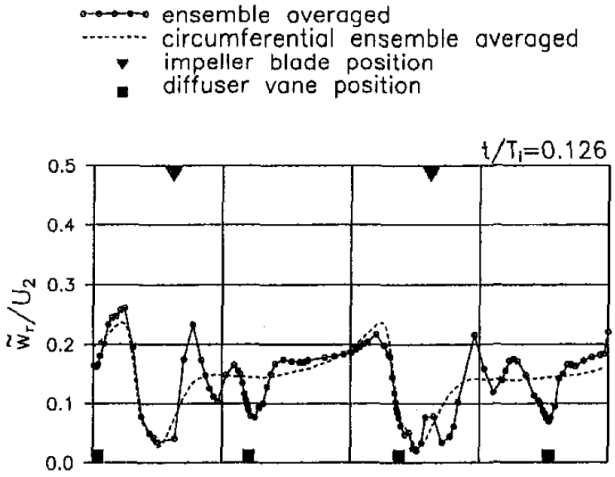


Figure 65 – Measurements at z/b = 0.5 over 2 impeller passages, Ubaldi [53]

In order to keep the amount of required impeller revolutions low, the averaging procedure was adapted for the simulation. The velocity profiles over two rotor passages are recorded with a resolution of 160 points which are stationary in the relative frame. These points are then triggered simultaneously to obtain an instantaneous snapshot. This process is repeated at different steps in time to obtain such profiles for 30 different relative rotor-stator positions over one stator blade passage. Each rotor blade contributes a full set of 30 times 160 data points at different points in time. The "circumferentially ensemble-averaged" profiles can immediately be obtained again by averaging over all 30 relative positions.

Figure 66 shows the evaluation points for a single rotor blade over the two involved passages.

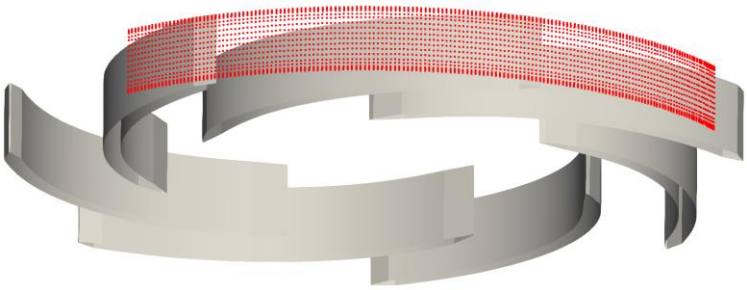


Figure 66 – Evaluation probe locations

As mentioned above, to obtain reasonably converged ensemble-averaged quantities, at least 3 full impeller revolutions are required after the initial startup. Due to the averaging procedure developed here, the circumferentially averaged profiles already converge after about half an impeller rotation. Other quantities, such as global pressure difference or impeller torque, already start to stabilize after a single impeller rotation counted from the start of the simulation. Simulations involving only such information can be completed in significantly shorter time than when local flow profiles are required.

Figure 67 to Figure 70 show the ensemble-averaged radial and relative circumferential velocities over two impeller passages and the full hub-to-shroud region. Each figure shows a different relative rotor-stator position. The solver is able to capture the general flow features and the almost two-dimensional structure of the flow field.

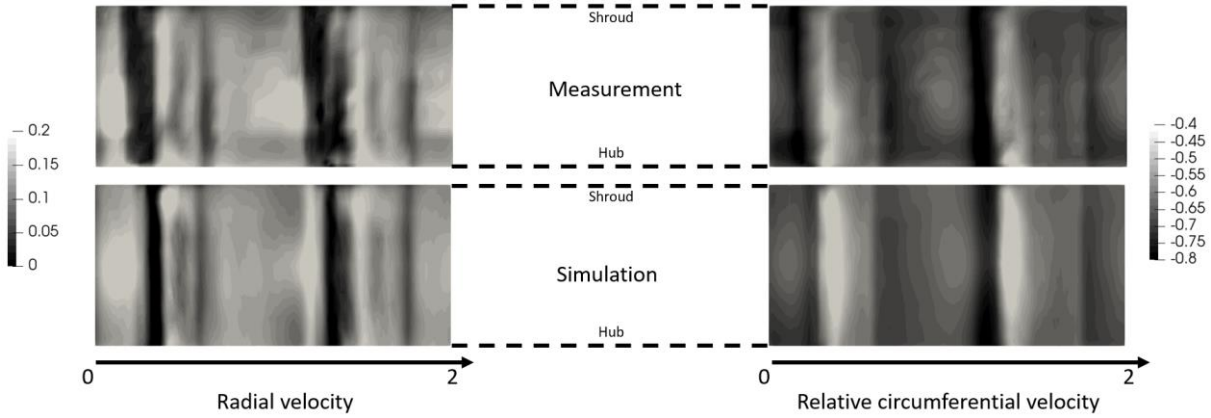


Figure 67 – Comparison of hub-to-shroud measurement data and simulation results over two impeller passages at rotor-stator relative position $t/T_i = 0.126$

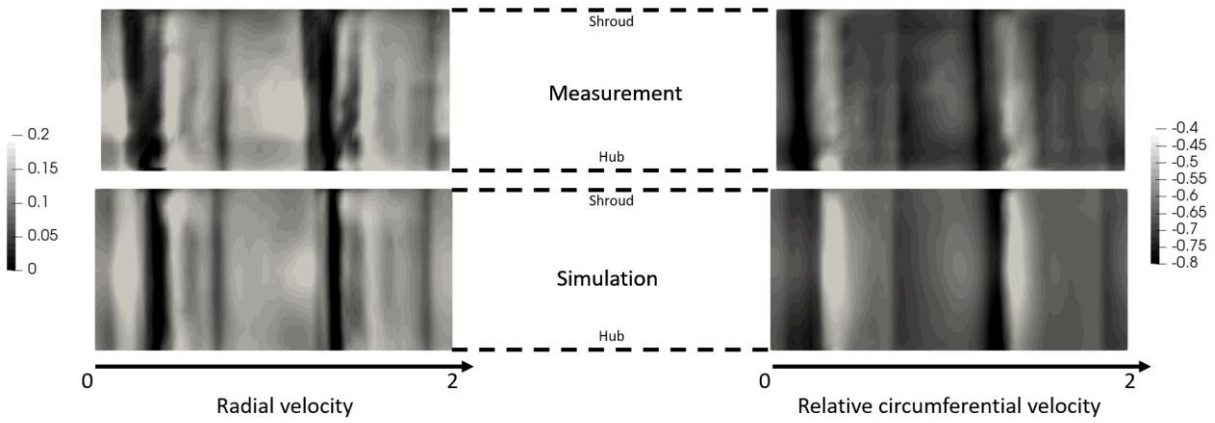


Figure 68 – Comparison of hub-to-shroud measurement data and simulation results over two impeller passages at rotor-stator relative position $t/T_i = 0.226$

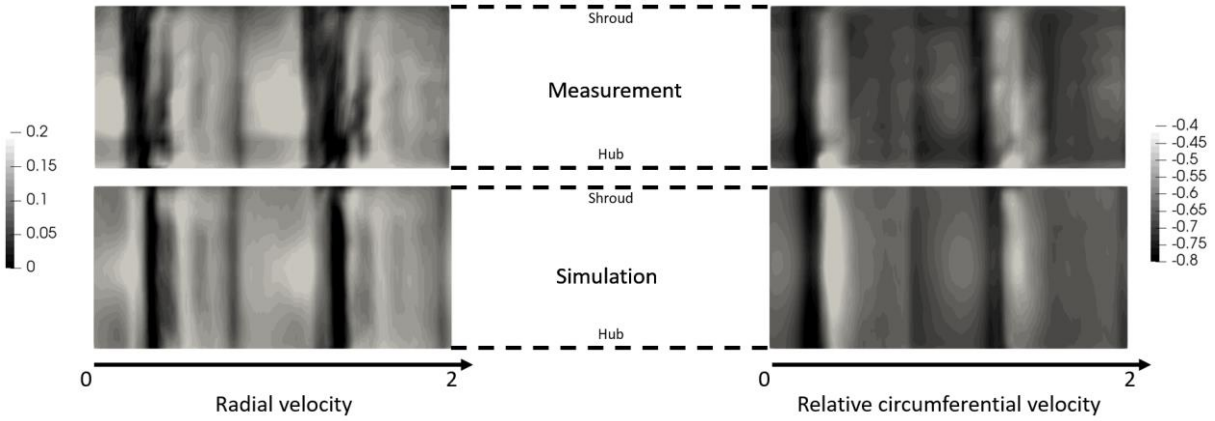


Figure 69 – Comparison of hub-to-shroud measurement data and simulation results over two impeller passages at rotor-stator relative position $t/T_i = 0.326$

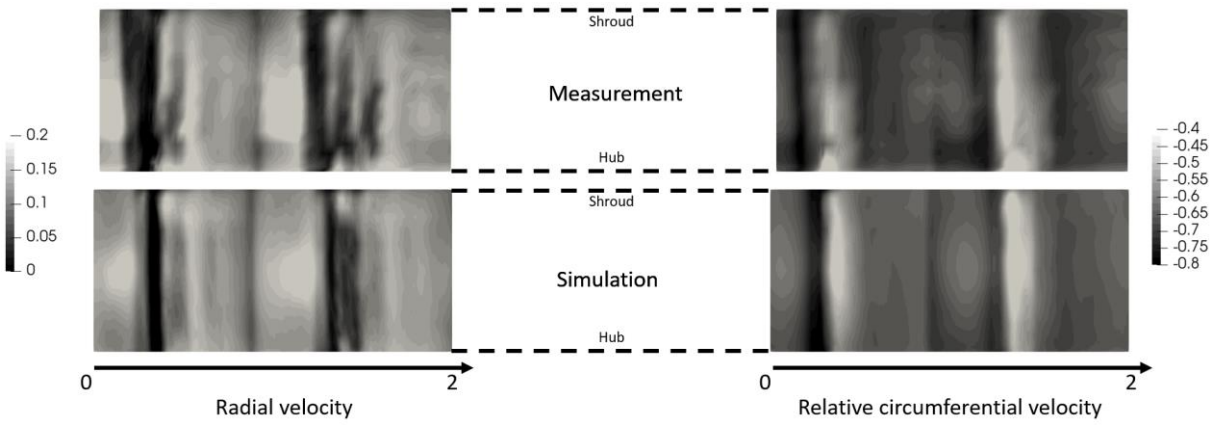


Figure 70 – Comparison of hub-to-shroud measurement data and simulation results over two impeller passages at rotor-stator relative position $t/T_i = 0.426$

Figure 72 to Figure 75 below show the ensemble-averaged velocity profiles at the mid-span section, obtained with the present code, for the radial and the relative circumferential velocity. The square and triangular markers in all figures indicate the position of the stator and rotor blades, respectively. In Figure 71 the circumferentially averaged velocity components are displayed.

The circumferentially averaged radial velocity as well as the relative circumferential velocity in Figure 71 are evenly periodic and follow the trend of the measured data. The circumferential velocity is underpredicted in its magnitude, which could explain the circumferential shift the location of impeller blade wakes, but the overall profile is accurately captured.

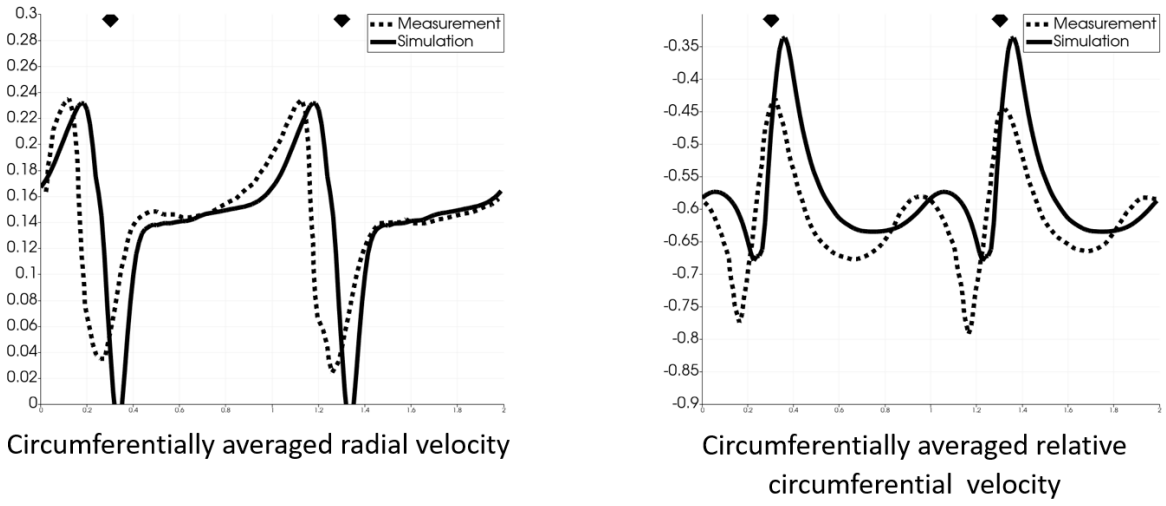


Figure 71 – Comparison of circumferentially averaged data at mid-span z/b = 0.5

The ensemble-averaged velocity profiles in Figure 72 to Figure 75 show the same shift in wake positioning and emphasize the reduced wake thickness in the simulation. This suggests that the wake separation region is generally underpredicted, which could also explain the influence on the circumferential velocity.

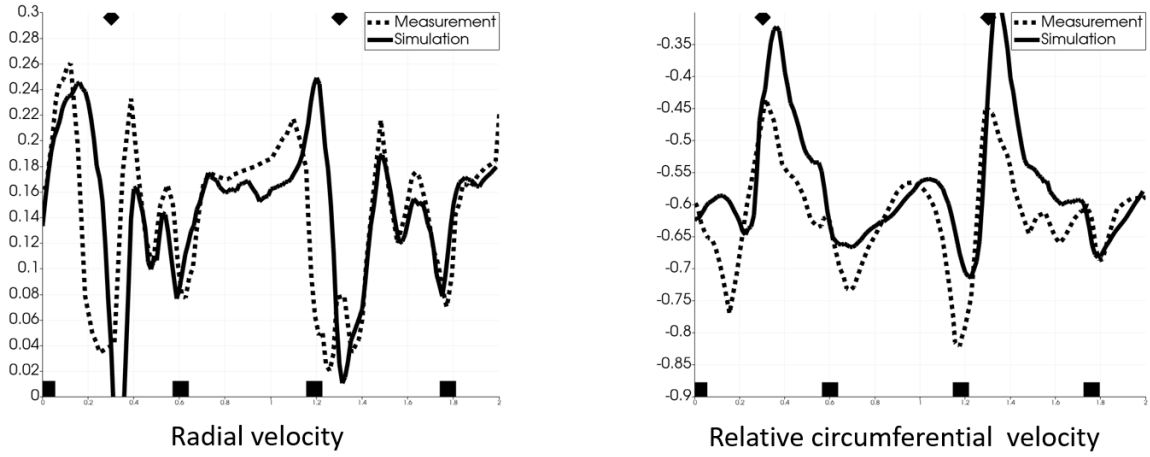


Figure 72 – Comparison of ensemble-averaged data at z/b = 0.5 and rotor-stator relative position t/T_i = 0.126

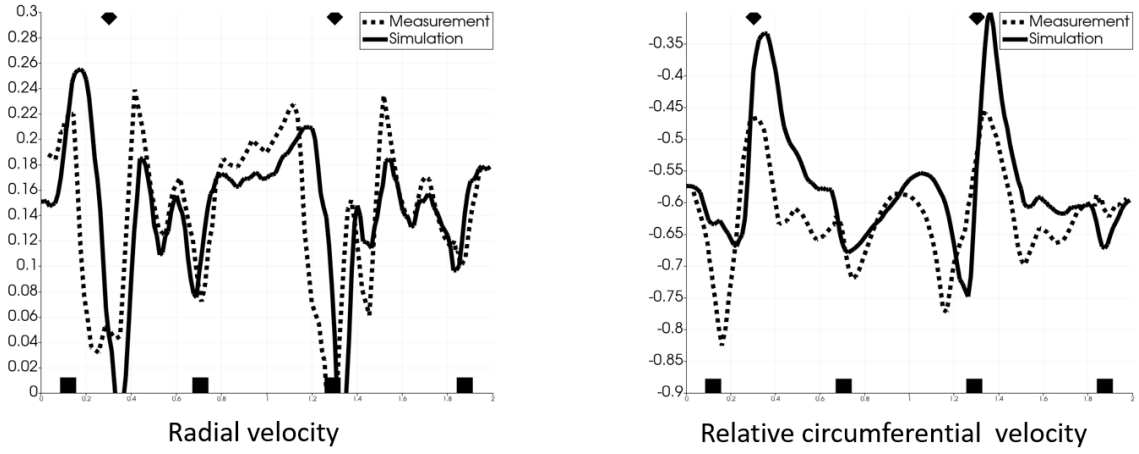


Figure 73 – Comparison of ensemble-averaged data at z/b = 0.5 and rotor-stator relative position t/T_i = 0.226

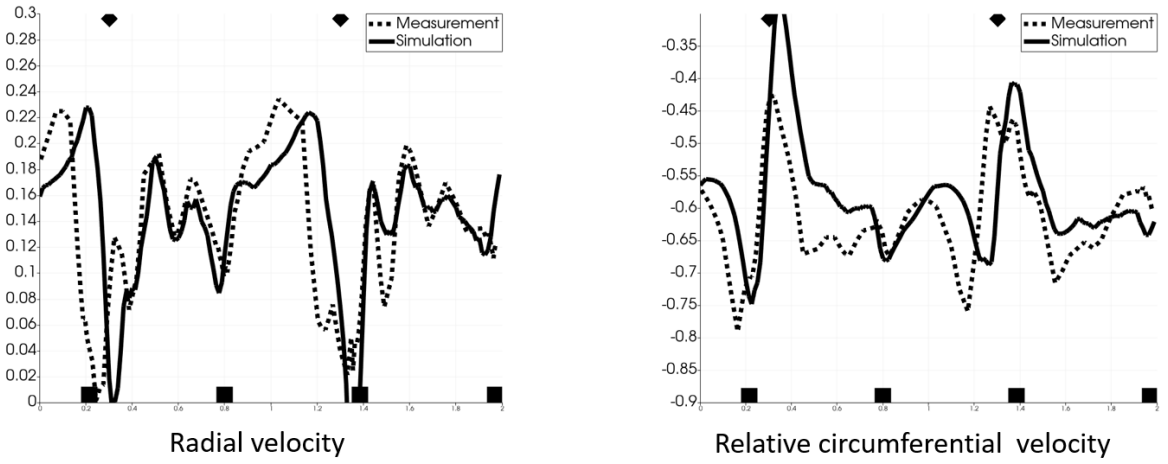


Figure 74 – Comparison of ensemble-averaged data at z/b = 0.5 and rotor-stator relative position t/T_i = 0.326

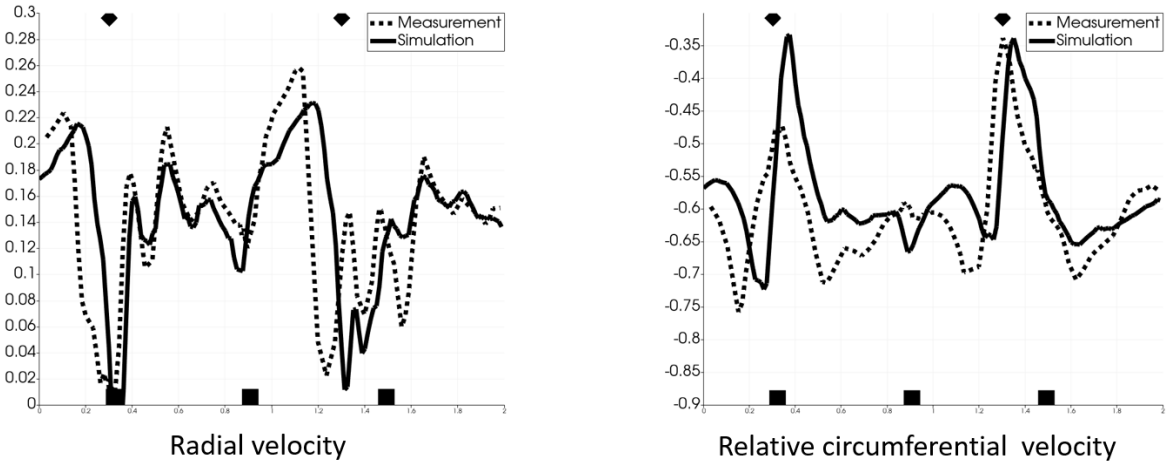


Figure 75 – Comparison of ensemble-averaged data at z/b = 0.5 and rotor-stator relative position t/T_i = 0.426

The radial velocity in Figure 72 to Figure 75 clearly show the blockages due the stator blades at Y_i/G_i = 0.5 and 1.8, which are not present in the circumferentially averaged profile. The stator induced unsteadiness, e.g. in the region from about Y_i/G_i = 0.4 to 1.0 and 1.4 to 2.0, which was the main interest of the investigations in [53], is accurately captured and evolves from a two-peak structure in Figure 72 and Figure 73 into the three-peak shape in Figure 75.

The relative circumferential velocity in Figure 72 to Figure 75 exhibits similar properties as already observed in Figure 71. The impeller suction side regions show reduced circumferential velocities for both instances of the impeller blades compared to measurements, but most of the stator induced unsteadiness is observable.

The present case was investigated in a number of publications and theses, such as [56] [57] [58]. The best agreement with measurements were obtained in [56] using a 3D unsteady RANS simulation, but

it has to be noted that the result quality varies significantly with the chosen turbulence model (for examples see the original paper). The best solution for the radial velocity uses a different turbulence model from the best solution for the circumferential velocity.

The present method has the advantage that this dependency is minimal or non-existent, depending on the grid resolution, as most of the turbulent spectrum is resolved. Model influence investigations can therefore be omitted.

Despite the discrepancies in the impeller blade wakes, the present results shown above are promising. Further improvements can be made by simply increasing the grid resolution, which is currently limited by the amount of GPU memory. In addition, the inlet conditions have a direct influence on the results but are essentially unknown. The present setup, where the inlet is turbulence-free, could be improved by providing more meaningful transient turbulent boundary conditions.

Most notably, with the code in its current unoptimized state, using the single new cluster node mentioned in section 3 above, the present results are obtainable in 5-6 hours. Considering that the code is not optimized and the meshing and postprocessing processes are done fully automatically, this simulation technique promises to be very attractive.

3.2.11 Pump-turbine

For this transient analysis a model-scale low specific speed (n_q = 26) pump-turbine is used, which was produced by Andritz hydro and measured at HSLU. The specific speed is defined as

$$n_q = n\frac{\sqrt{Q}}{H^{\frac{3}{4}}} \tag{20}$$

where n is the runner speed, Q is the volume flow rate and H is the head.

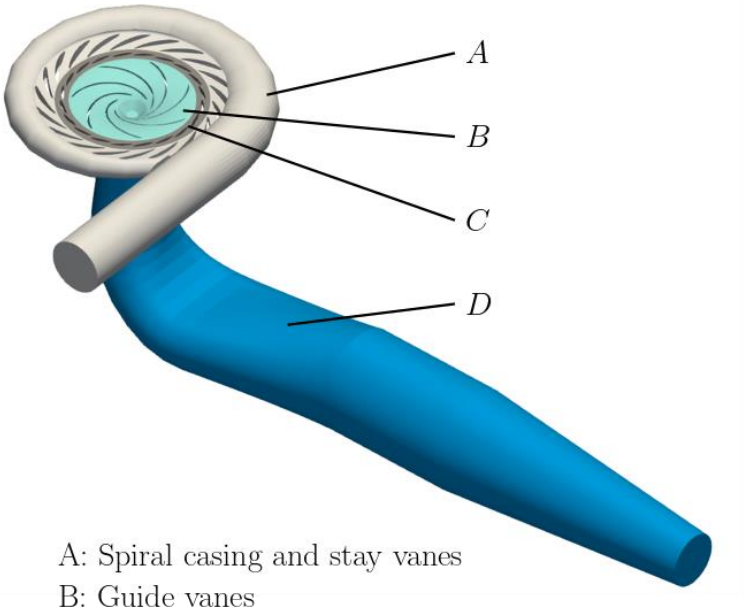
The head for the prototype-scale pump-turbine ranges between 510 [m] – 530 [m] with a maximum power output of 165 [MW].

The guide vane angle used in the present simulation is 6°, which corresponds to the setup for the synchronization point. The best efficiency point is obtained at a guide vane opening of 20°.

This case was the center of several publications on the topic of pump-turbine instability (see e.g. [59]). There it was shown that capturing the correct behaviour requires transient simulations, while the results are still dependent on the chosen turbulence model. In some cases, a specific choice for the turbulence model produced no instability at all.

The simulated geometry consists of the four main components spiral casing and stay vanes, adjustable guide vanes, runner and draft tube, shown in Figure 76. The side cavities and leakage flows are neglected.

An extension and a pinched segment were added to the end of the draft tube to suppress backflow. While the solver is able to handle a certain amount of backflow at outlet boundaries, it is recommended to avoid it.



A: Spiral casing and stay vanes
B: Guide vanes
C: Impeller
D: Draft tube

Figure 76 - Pump turbine geometry overview

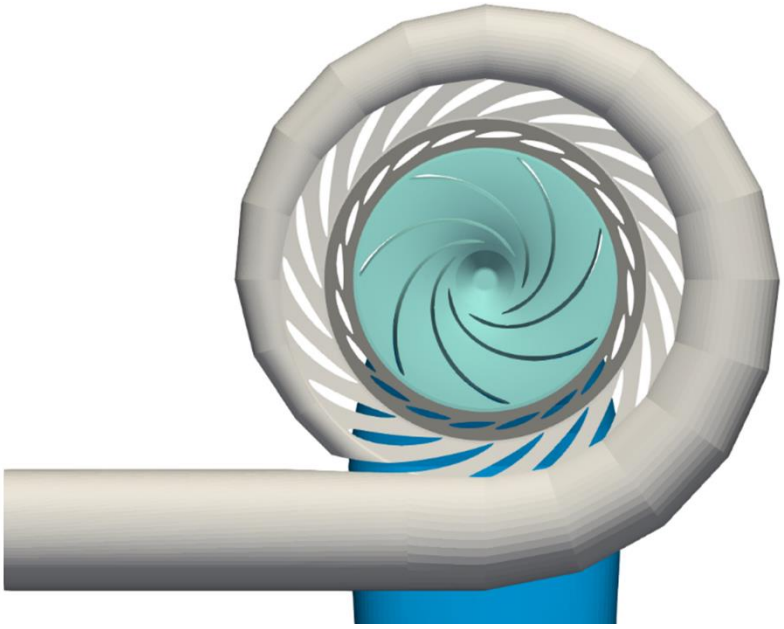


Figure 77 - Close-up of vane position and rotor geometry

The machine characteristic was simulated both in pump and turbine mode, using water as the working fluid, at a runner speed of 1200 [rpm]. Note that for the highest flow rate, the runner speed was reduced to 500 [rpm], essentially making this its own separate operating condition.

The Reynolds number, based on the rotational speed of the impeller, was $12.6 \cdot 10^6$, which is extremely high for Lattice Boltzmann simulations.

The total physical simulated time was 1 [s], corresponding to 20 impeller revolutions. Evaluation of statistical quantities was started after 10 revolutions and collected over another 10.

Two different grids were generated for this case: grid A consisting of $92 \cdot 10^6$ grid points and grid B with $170 \cdot 10^6$. Note however, that in the coarser mesh A, the boundary resolution of the draft tube is effectively finer than the one in mesh B. Figure 78 to Figure 81 show close-ups of the grid in the pump-turbine runner and the surrounding areas. Also here, artifacts in the grid denote the domain decompositions.

Figure 78 - Pump-turbine mesh, side view, grid A

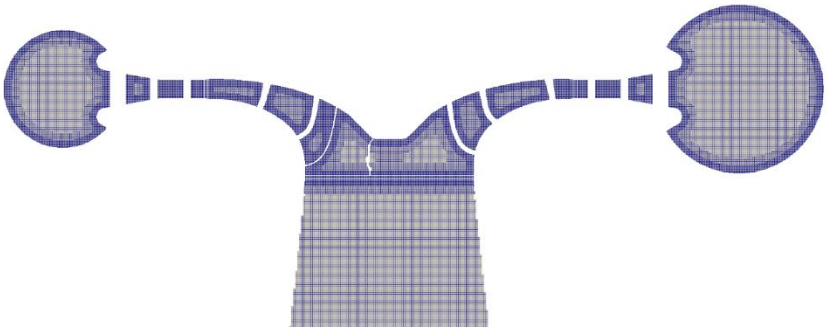


Figure 79 - Pump-turbine mesh, side view, grid B

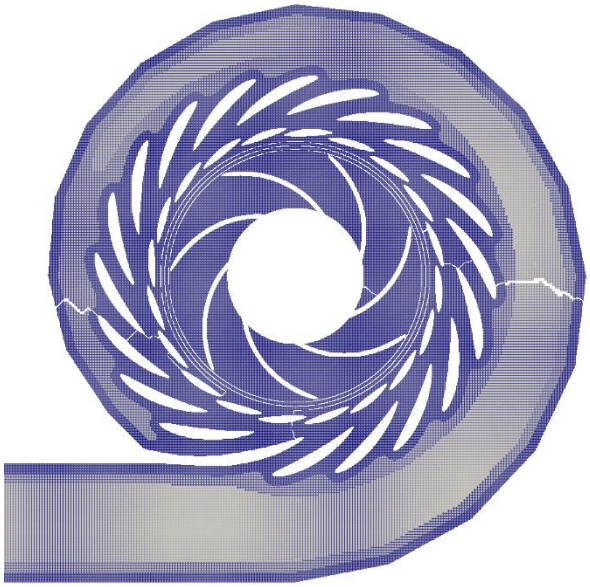


Figure 80 - Pump-turbine mesh, top view, grid A

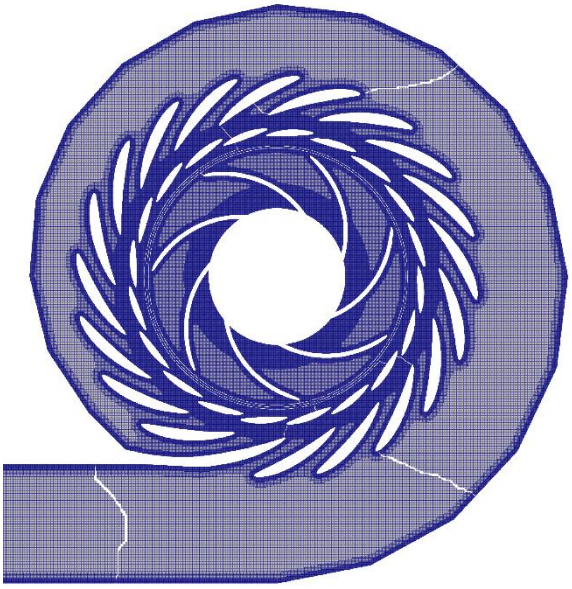


Figure 81 - Pump-turbine mesh, top view, grid B

The simulation results were evaluated in terms of the dimensionless runner speed k_{u1} and the dimensionless flow rate k_{cm1}, which are defined as follows:

$$k_{u1} = \frac{n\,\pi D}{\sqrt{2gH}} \tag{21}$$

$$k_{cm1} = \frac{4Q}{D^2\pi\sqrt{2gH}} \tag{22}$$

Where n is the runner speed, D its diameter, Q the volume flow rate and H the head.

The simulation results are compared to measurements performed by HSLU in 2017 and reported e.g. in [59] in Figure 82.

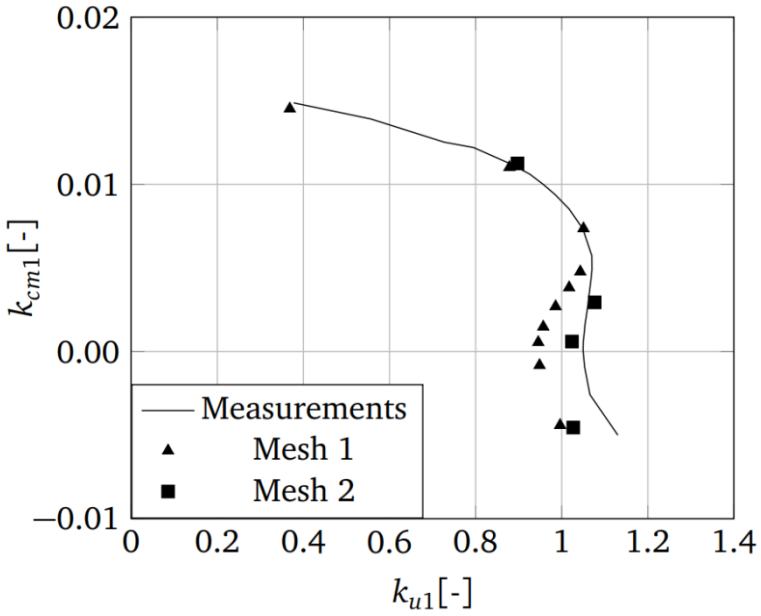


Figure 82 - Pump-turbine characteristic

The agreement with measurements away from the zero flow rate region is good, and the results of both grids clearly capture an unstable S-shape in the characteristic. However, as the flow rate gets closer to zero the results show a relatively large deviation from the measured values. Grid B, which resolves the runner and the vaneless space more accurately, shows much better agreement.

In reverse pump mode, i.e. the runner rotating in turbine direction, but with a negative flow rate, both grids show a significant deviation from the measurements. This could be due to the outlet pinch acting as a diffuser in this flow direction, which, if separation occurs, could lead to increased losses. In all cases, the outlet section is included in the loss calculation, as it is currently not possible to accurately record data on arbitrary surfaces in the domain, which will have an effect on the calculated values.

The discrepancies will be analyzed in more detail in the future, including a part-by-part investigation of pressure drops in the pump-turbine components and the draft tube sections.

It has to be noted that such a simulation has never been performed before with the Lattice Boltzmann Method and these first results presented here are very promising for future analysis and developments.

4 Conclusions

In this work, a new CFD simulation software, based on the Lattice Boltzmann Method was developed. The code was tested on a series on verification and validation cases and is able to deliver accurate results of turbulent flows in complex geometries.

The code was applied to the simulation of full turbomachines in the form of fans, pumps and pump-turbines. The simulations presented in this report are the first of their kind in the context that they were presented and have shown that the Lattice Boltzmann Method can provide a viable alternative to classical RANS-based simulation methods.

Further work will be required, both in applying the presented method and improving the existing software.

5 Outlook and next steps

The CFD solver developed in this project is ready for application to a wide variety of test cases in the context of hydraulic machines. Experience will need to be generated for the use of this new method and best practice procedures will have to be developed. All simulation results presented here will be revisited and further refined in the near future.

In terms of technical next steps, the focus will lie on improving the solver performance and reducing the memory requirements. For both of these steps, the potential for large improvement steps is very high, and first ideas are currently being evaluated.

The topic of multiphase and multicomponent simulations is also worth revisiting, also in the context of free-surface situations, as they occur in Pelton runners.

6 National and international cooperation

The current project was carried out for the most part at HSLU in Horw, Lucerne, but includes a collaboration with both the ETH Zurich and the USI in Lugano. The authors would like to thank Prof. Dr. I. Karlin and Prof. Dr. Krause, along with members of their respective groups, for their valuable input and fruitful discussions.

7 Publications

D. Roos Launchbury, E. Casartelli and L. Mangani, “Lattice Boltzmann simulation of the ERCOFTAC pump impeller”, *IOP Conference Series: Earth and Environmental Science*, vol. 1079, 2022.

8 References

[1] O. Braun, B. Horisberger, N. Ruchonnet, A. Taruffi and A. Gehrer, “Validation of CFD Analysis of Acoustic Effects in Pump-Turbine Runners,” *IOP Conference Series: Earth and Environmental Science,* vol. 240, 2019.

[2] R. Blanc-Coquand, S. Lavigne and J. L. Deniau, “Experimental and Numerical Study of Pressure Fluctuations in High Head Pump-Turbine,” in *Proceedings of the 20th IAHR Symposium on Hydraulic Machinery and Systems*, 2000.

[3] J. Yan, J. Koutnik, U. Seidel and B. Hübner, “Compressible Simulation of Rotor-Stator Interaction in Pump-Turbines,” *International Journal of Fluid Machinery and Systems,* vol. 3, 2010.

[4] U. Ješe, V. Novotný and A. Skoták, “Development Trends in the Field of Reversible Pump-Turbines - Study of Pumping and Generating Mode Off-Design Conditions,” *IOP Conf. Series: Earth and Environmental Science,* vol. 240, 2019.

[5] P. L. Bhatnagar, E. P. Gross and M. Krook, “A Model for Collision Processes in Gases. I. Small Amplitude Processes in Charged and Neutral One-Component Systems,” *Physical Review,* vol. 94, no. 3, p. 511–525, May 1954.

[6] I. Karlin, A. Ferrante and H. C. Öttinger, “Perfect Entropy Functions of the Lattice Boltzmann Method,” *EPL (Europhysics Letters),* vol. 47, p. 182, January 1999.

[7] M. Klein, “An Attempt to Assess the Quality of Large Eddy Simulations in the Context of Implicit Filtering,” *Flow, Turbulence and Combustion,* vol. 75, pp. 131-147, 2005.

[8] S. B. Pope, “Ten Questions Concerning the Large-Eddy Simulation of Turbulent Flows,” *New Journal of Physics,* vol. 6, 2004.

[9] T. Krüger, H. Kusumaatmaja, A. Kuzmin, O. Shardt, G. Silva and E. M. Viggen, The Lattice Boltzmann Method, Principles and Practice, Springer International Publishing, 2017.

[10] I. V. Karlin, S. Ansumali, C. Frouzakis and S. S. Chikatamarla, “Elements of the Lattice Boltzmann Method. I. Linear Advection Equation,” *Communications in Computational Physics,* vol. 1, pp. 616-655, August 2006.

[11] Y. H. Qian, D. d'Humières and P. Lallemand, “Lattice BGK Models for Navier-Stokes Equation,” *Europhysics Letters (EPL),* vol. 17, p. 479–484, February 1992.

[12] M. Geier, M. Schönherr, A. Pasquali and M. Krafczyk, “The Cumulant Lattice Boltzmann Equation in Three Dimensions: Theory and Validation,” *Computers & Mathematics with Applications,* vol. 70, pp. 507-547, 2015.

[13] S. S. Chikatamarla, S. Ansumali and I. V. Karlin, “Entropic Lattice Boltzmann Models for Hydrodynamics in Three Dimensions,” *Physical Review Letters,* vol. 97, no. 1, p. 010201, July 2006.

[14] O. Malaspinas, M. O. Deville and B. Chopard, “Towards a Physical Interpretation of the Entropic Lattice Boltzmann Method,” *Physical Review E,* vol. 78, p. 066705, 2008.

[15] F. Bösch, "Entropic Lattice Boltzmann Models for Fluid Dynamics," ETH Zurich, Zurich, 2017.

[16] F. Bösch, S. S. Chikatamarla and I. V. Karlin, “Entropic Multirelaxation Lattice Boltzmann Models for Turbulent Flows,” *Physical Review E,* vol. 92, no. 4, p. 043309, October 2015.

[17] D. d'Humières, P. V. Coveney, S. Succi, I. Ginzburg, M. Krafczyk, P. Lallemand and L. Luo, “Multiple-Relaxation-Time Lattice Boltzmann Models in Three Dimensions,” *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences,* vol. 360, pp. 437-451, 2002.

[18] B. Dorschner, "Entropic Lattice Boltzmann Method for Complex Flows," ETH Zurich, Zurich, 2018.

[19] J. Latt, B. Chopard, O. Malaspinas, M. Deville and A. Michler, “Straight Velocity Boundaries in the Lattice Boltzmann Method,” *Physical Review E,* vol. 77, no. 5, p. 056703, May 2008.

[20] H. Grad, “On the Kinetic Theory of Rarefied Gases,” *Communications on Pure and Applied Mathematics,* vol. 2, pp. 331-407, 1949.

[21] S. S. Chikatamarla, S. Ansumali and I. V. Karlin, “Grad's Approximation for Missing Data in Lattice Boltzmann Simulations,” *Europhysics Letters (EPL),* vol. 74, p. 215, January 2007.

[22] B. Dorschner, S. S. Chikatamarla, F. Bösch and I. V. Karlin, “Grad's Approximation for Moving and Stationary Walls in Entropic Lattice Boltzmann Simulations,” *Journal of Computational Physics,* vol. 295, pp. 340-354, August 2015.

[23] S. Péron, C. Benoit, T. Renaud and I. Mary, “An Immersed Boundary Method on Cartesian Adaptive Grids for the Simulation of Compressible Flows around Arbitrary Geometries,” *Engineering with Computers,* vol. 37, 2021.

[24] N. Frapolli, S. S. Chikatamarla and I. V. Karlin, “Entropic Lattice Boltzmann Model for Gas Dynamics: Theory, Boundary Conditions, and Implementation,” *Phys. Rev. E,* vol. 93, no. 6, p. 063302, June 2016.

[25] Y. Feng, S. Guo, J. Jacob and P. Sagaut, “Solid Wall and Open Boundary Conditions in Hybrid Recursive Regularized Lattice Boltzmann Method for Compressible Flows,” *Physics of Fluids,* vol. 31, p. 126103, December 2019.

[26] C. Feuchter, O. Wagner, A. Stief and T. Beisswenger, “Turbulent Flow Simulations Around a Surface-Mounted Finite Cylinder Using an Entropic Multi-Relaxation Lattice Boltzmann Method,” *Fluid Dynamics Research,* vol. 51, p. 055509, September 2019.

[27] B. Dorschner, S. S. Chikatamarla and I. V. Karlin, “Entropic Multirelaxation-Time Lattice Boltzmann Method for Moving and Deforming Geometries in three dimensions,” *Phys. Rev. E,* vol. 95, no. 6, p. 063306, June 2017.

[28] E. K. Far, M. Geier and M. Krafczyk, “Simulation of Rotating Objects in Fluids with the Cumulant Lattice Boltzmann Model on Sliding Meshes,” *Computers & Mathematics with Applications,* vol. 79, pp. 3-16, 2020.

[29] R. Zhang, C. Sun, Y. Li, R. Satti, R. Shock, J. Hoch and H. Chen, “Lattice Boltzmann Approach for Local Reference Frames,” *Communications in Computational Physics,* vol. 9, p. 1193–1205, 2011.

[30] A. Kupershtokh, “New Method of Incorporating a Body Force Term into the Lattice Boltzmann Equation,” in *Proc. 5th International EHD Workshop*, 2004.

[31] OPTIMAD, *bitpit: Open Source Library for Scientific High Performance Computing,* 2023.

[32] G. Karypis and V. Kumar, “METIS* A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices,” 1997.

[33] C. R. Trott, D. Lebrun-Grandié, D. Arndt, J. Ciesko, V. Dang, N. Ellingwood, R. Gayatri, E. Harvey, D. S. Hollman, D. Ibanez, N. Liber, J. Madsen, J. Miles, D. Poliakoff, A. Powell, S. Rajamanickam, M. Simberg, D. Sunderland, B. Turcksin and J. Wilke, “Kokkos 3: Programming Model Extensions for the Exascale Era,” *IEEE Transactions on Parallel and Distributed Systems,* vol. 33, pp. 805-817, 2022.

[34] H. C. Edwards, C. R. Trott and D. Sunderland, “Kokkos: Enabling Manycore Performance Portability Through Polymorphic Memory Access Patterns,” *Journal of Parallel and Distributed Computing,* vol. 74, pp. 3202-3216, 2014.

[35] L. Dagum and R. Menon, “OpenMP: An Industry Standard API for Shared-Memory Programming,” *Computational Science & Engineering, IEEE,* vol. 5, p. 46–55, 1998.

[36] J. Nickolls, I. Buck, M. Garland and K. Skadron, “Scalable Parallel Programming with CUDA: Is CUDA the Parallel Programming Model That Application Developers Have Been Waiting For?,” *Queue,* vol. 6, p. 40–53, March 2008.

[37] E. Gabriel, G. E. Fagg, G. Bosilca, T. Angskun, J. J. Dongarra, J. M. Squyres, V. Sahay, P. Kambadur, B. Barrett, A. Lumsdaine, R. H. Castain, D. J. Daniel, R. L. Graham and T. S. Woodall, “Open MPI: Goals, Concept, and Design of a Next Generation MPI Implementation,” in *Proceedings, 11th European PVM/MPI Users' Group Meeting*, Budapest, 2004.

[38] J. W. Deardorff, “A Numerical Study of Three-Dimensional Turbulent Channel Flow at Large Reynolds Numbers,” *Journal of Fluid Mechanics,* vol. 41, p. 453–480, 1970.

[39] F. Ducros, N. Franck and T. Poinsot, “Wall-Adapting Local Eddy-Viscosity Models for Simulations in Complex Geometries,” *Numerical Methods for Fluid Dynamics VI,* January 1998.

[40] F. Nicoud, H. Baya Toda, O. Cabrit, S. Bose and J. Lee, “Using Singular Values to Build a Subgrid-Scale Model for Large Eddy Simulations,” *Physics of Fluids,* vol. 23, p. 085106, 2011.

[41] F. R. Menter, “Improved Two-Equation k-omega Turbulence Models for Aerodynamic Flows,” 1992.

[42] X. Han and S. Krajnović, “An Efficient Very Large Eddy Simulation Model for Simulation of Turbulent Flow,” *International Journal for Numerical Methods in Fluids,* vol. 71, pp. 1341-1360, 2013.

[43] A. J. Musker, “Explicit Expression for the Smooth Wall Velocity Distribution in a Turbulent Boundary Layer,” *AIAA Journal,* vol. 17, pp. 655-657, 1979.

[44] M. O. Deville, An Introduction to the Mechanics of Incompressible Fluids, Springer International Publishing, 2022.

[45] U. Ghia, K. N. Ghia and C. T. Shin, “High-Re Solutions for Incompressible Flow Using the Navier-Stokes Equations and a Multigrid Method,” *Journal of Computational Physics,* vol. 48, pp. 387-411, 1982.

[46] R. D. Moser, J. Kim and N. N. Mansour, “Direct Numerical Simulation of Turbulent Channel Flow up to Re_τ = 590,” *Physics of Fluids,* vol. 11, pp. 943-945, 1999.

[47] M. Lee and R. D. Moser, “Direct Numerical Simulation of Turbulent Channel Flow up to Re_τ = 5200,” *Journal of Fluid Mechanics,* vol. 774, p. 395–415, June 2015.

[48] P. Smirnov and F. Menter, “Sensitization of the SST Turbulence Model to Rotation and Curvature by Applying the Spalart–Shur Correction Term,” *Journal of Turbomachinery-transactions of The Asme - J TURBOMACH-T ASME,* vol. 131, October 2009.

[49] M. Stieglmeier, C. Tropea, N. Weiser and W. Nitsche, “Experimental Investigation of the Flow Through Axisymmetric Expansions,” *Journal of Fluids Engineering-transactions of The Asme - J FLUID ENG,* vol. 111, pp. 464-471, December 1989.

[50] F. Qin, A. Mazloomi Moqaddam, Q. Kang, D. Derome and J. Carmeliet, “Entropic Multiple-Relaxation-Time Multirange Pseudopotential Lattice Boltzmann Model for Two-Phase Flow,” *Physics of Fluids,* vol. 30, p. 032104, March 2018.

[51] Z. X. Wen, Q. Li, Y. Yu and K. H. Luo, “Improved Three-Dimensional Color-Gradient Lattice Boltzmann Model for Immiscible Two-Phase Flows,” *Phys. Rev. E,* vol. 100, no. 2, p. 023301, August 2019.

[52] Z. Wang, R. Örlü, P. Schlatter and Y. M. Chung, “Direct Numerical Simulation of a Turbulent 90° Bend Pipe Flow,” *International Journal of Heat and Fluid Flow,* vol. 73, pp. 199-208, 2018.

[53] M. Ubaldi, P. Zunino, G. Barigozzi and A. Cattanei, “An Experimental Investigation of Stator Induced Unsteadiness on Centrifugal Impeller Outflow,” *Journal of Turbomachinery,* vol. 118, pp. 41-51, January 1996.

[54] J. F. Combes, “Test Case U3: Centrifugal Pump with a Vaned Diffuser,” 1999.

[55] W. Merzkirch, Fluid Mechanics of Flow Metering, Springer-Verlag, 2005.

[56] H. N. O. Petit, “Numerical Investigations of Unsteady Flow in a Centrifugal Pump with a Vaned Diffuser,” *International Journal of Rotating Machinery,* vol. 2013, 2013.

[57] O. Petit, P. A. G. E. Maryse, M. Beaudoin and H. Nilsson, “The ERCOFTAC Centrifugal Pump OpenFOAM Case-Study,” January 2009.

[58] S. Xie, “Studies of the Ercoftac Centrifugal Pump with Openfoam,” 2010.

[59] E. Casartelli, L. Mangani, A. Del Rio and A. Schmid, “Capturing the S-Shape of Pump-Turbines by CFD Simulations Using an Anisotropic Turbulence Model,” in *Fluids Engineering Division Summer Meeting*, 2019.

[60] M. Zorn, K. Khalfaoui and S. Riedelbauch, “Towards Accurate Numerical Fluid Dynamics of Pump-Turbine Runners in Deep Partial Load Operation,” *IOP Conference Series: Earth and Environmental Science,* vol. 1079, p. 012086, September 2022.

[61] K. Yamamoto, “Hydrodynamics of Francis Turbine Operation at Deep Part Load Condition,” 2017.

[62] F. Thaler, “Multi-GPU Two-Component Fluid Mixtures with the Lattice Boltzmann Method,” 2016.

[63] G. Tauzin, “Implicit Sub-Grid Scale Modeling within the Entropic Lattice Boltzmann Method in Homogeneous Isotropic Turbulence,” 2019.

[64] W. Strunk and E. B. White, The Elements of Style, 4th ed., Longman Publishers, 1899.

[65] M. Shan, C. Zhu, X. Zhou, C. Yin and Q. Han, “Investigation of Cavitation Bubble Collapse Near Rigid Boundary by Lattice Boltzmann Method,” *Journal of Hydrodynamics, Ser. B,* vol. 28, pp. 442-450, 2016.

[66] R. Phogat, N. González Díez, J. Smeulers, D. Casalino and F. Avallone, “Analysis of Aeroacoustic Phenomena in Centrifugal Compressors Using a Lattice Boltzmann Flow

Simulation Approach," in *Proceedings of the ASME Turbo Expo 2020: Turbomachinery Technical Conference and Exposition*, 2020.

[67] S. Leclaire, M. Reggio and J.-Y. Trépanier, "Isotropic Color Gradient for Simulating Very High-Density Ratios with a Two-Phase Flow Lattice Boltzmann Model," *Computers & Fluids,* vol. 48, pp. 98-112, 2011.

[68] J. Latt, O. Malaspinas, D. Kontaxakis, A. Parmigiani, D. Lagrava, F. Brogi, M. B. Belgacem, Y. Thorimbert, S. Leclaire, S. Li, F. Marson, J. Lemus, C. Kotsalos, R. Conradin, C. Coreixas, R. Petkantchin, F. Raynaud, J. Beny and B. Chopard, "Palabos: Parallel Lattice Boltzmann Solver," *Computers & Mathematics with Applications,* vol. 81, pp. 334-350, 2021.

[69] A. Kummerländer, S. Avis, H. Kusumaatmaja, F. Bukreev., D. Dapelo, S. Großmann, N. Hafen, C. Holeksa, A. Husfeldt, J. Jeßberger, L. Kronberg, J. E. Marquardt, J. Mödl, J. Nguyen, T. Pertzel, S. Simonis, L. Springmann, N. Suntoyo, D. Teutscher, M. Zhong and M. J. Krause, *OpenLB Release 1.5: Open Source Lattice Boltzmann Code,* Zenodo, 2022.

[70] A. Krämer, K. Küllmer, D. Reith, W. Joppich and H. Foysi, "Semi-Lagrangian Off-Lattice Boltzmann Method for Weakly Compressible Flows," *Physical Review E,* vol. 95, no. 2, p. 023305, February 2017.

[71] I. V. Karlin and S. S. Chikatamarla, "Elements of the Lattice Boltzmann Method II: Kinetics and Hydrodynamics in One Dimension," *Communications in Computational Physics,* vol. 2, pp. 196-238, April 2007.

[72] S. Joss, "Lattice Boltzmann Method (LBM) to Solve Multiflow Problems," 2022.

[73] S. Joss, "Grid Refinement for the Lattice Boltzmann Method," 2021.

[74] M. Januszewski and M. Kostur, "Sailfish: A Flexible Multi-GPU Implementation of the Lattice Boltzmann Method," *Computer Physics Communications,* vol. 185, p. 2350–2368, September 2014.

[75] L. Hanimann, L. Mangani, E. Casartelli and M. Widmer, "Steady-State Cavitation Modeling in an Open Source Framework: Theory and Applied Cases," in *16th International Symposium on Transport Phenomena and Dynamics of Rotating Machinery*, Honolulu, 2016.

[76] G. Falcucci, S. Ubertini, G. Bella and S. Succi, "Lattice Boltzmann Simulation of Cavitating Flows," *Communications in Computational Physics,* vol. 13, pp. 685-695, 2013.

[77] B. Dorschner, N. Frapolli, S. S. Chikatamarla and I. V. Karlin, "Grid Refinement for Entropic Lattice Boltzmann Models," *Phys. Rev. E,* vol. 94, no. 5, p. 053311, November 2016.

[78] C. E. Brennen, Cavitation and Bubble Dynamics, Cambridge University Press, 2013.

[79] R. Benzi, S. Succi and M. Vergassola, "The Lattice Boltzmann Equation: Theory and Applications," *Physics Reports,* vol. 222, pp. 145-197, 1992.

[80] M. Bauer, S. Eibl, C. Godenschwager, N. Kohl, M. Kuron, C. Rettinger, F. Schornbaum, C. Schwarzmeier, D. Thönnes, H. Köstler and U. Rüde, "waLBerla: A Block-Structured High-Performance Framework for Multiphysics Simulations," *Computers & Mathematics with Applications,* vol. 81, p. 478–501, January 2021.