UNIVERSITY OF SOUTHAMPTON

FACULTY OF ENGINEERING & PHYSICAL SCIENCES

Aerodynamics and Flight Mechanics

Implementation and Verification of LES models for SRT Lattice Boltzmann Methods

by

Christos Gkoudesnes

ORCID ID: 0000-0001-6576-873X

Thesis for the degree of Doctor of Philosophy

March 2021

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING & PHYSICAL SCIENCES
Aerodynamics and Flight Mechanics

IMPLEMENTATION AND VERIFICATION OF LES MODELS FOR SRT LATTICE BOLTZMANN METHODS

by Christos Gkoudesnes

In the last two decades, the Lattice Boltzmann Method (LBM) has experienced tremendous progress and rise in its application by academia and industry. This fact has led to the development of a variety of LBM solvers, both commercial and academic. Two critical factors for this success are its inexpensive numerical step and parallel scalability compared to more mainstream approaches, such as Finite Volume Navier-Stokes solvers. The main target of this project is the introduction and testing of LES models into the academic solver AMROC, developed currently at the University of Southampton. The three LES models to be considered are Constant and Dynamic Smagorinsky (CSMA & DSMA) and WALE.

Initially, three wall-free test cases, namely Forced and Decaying Homogeneous Isotropic Turbulence (FHIT & DHIT) and the Taylor-Green Vortex (TGV), were employed to verify the algorithms of the new implementations. To further improve the understanding of the models and their inter-coupling with LBM, besides the Standard (STA) Single Relaxation Time (SRT) collision model, I have also used the Regularised (REG) collision model for comparative analysis. Simultaneously, I have also investigated the effect of calculating the strain rate locally, using the non-equilibrium part, or through a finite difference stencil. An abundance of valuable data and conclusions has been obtained.

The next step was the simulation of the bi-periodic turbulent channel. In this scenario, I have validated the LES models by capturing the law of the wall. A new algorithm for imposing macrovariables in ghost cells where bounce-back-like boundary conditions are applied has been devised in parallel. This capability is of utmost importance for the LES models, particularly DSMA and WALE, in which the calculation of the eddy viscosity is based on a stencil. Therefore its calculation is possible in the first fluid cell. To further improve the ability of the solver to deal with high Re flows, a wall function has also been implemented and tested using the bi-periodic channel case.

The final test case was a square cylinder on a flat plate at an angle of 90^o. This case aimed to verify the wall function for a body not aligned to the Cartesian mesh.

Acknowledgements

First of all, I would like to thank Dr Ralf Deiterding for his supervision during the last four years. Our numerous discussions, meetings and debates have shaped this thesis.

I would like to thank the IT group at the University of Southampton and their assistance with IRIDIS supercomputer.

I would like to thank my colleagues, Antonio Reyes Barraza and Dr Jacob Turner, for their friendship and companion for the exciting discussions during lunchtimes and coffee breaks.

Finally, I would like to thank my family and particularly my fiancée for her empathy all these years.

Contents

List of Figures

List of Tables

Research Thesis: Declaration of Authorship

Print name: Christos Gkoudesnes

Title of thesis: *Implementation and Verification of LES models for SRT Lattice Boltzmann Methods*

I declare that this thesis and the work presented in it are my own and have been generated by me as the result of my own original research.

I confirm that:

1. This work was done wholly or mainly while in candidature for a research degree at this University;
2. Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
3. Where I have consulted the published work of others, this is always clearly attributed;
4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
5. I have acknowledged all main sources of help;
6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
7. Parts of this work have been published as:
    - Gkoudesnes, C. and Deiterding, R. (2019a). Evaluating the Lattice Boltzmann Method for Large Eddy Simulation with Dynamic Sub-grid Scale Models. In *Eleventh International Symposium on Turbulence and Shear Flow Phenomena, TSFP11*, Southampton
    - Gkoudesnes, C. and Deiterding, R. (2019b). Verification and Validation of Lattice Boltzmann Method Coupled with Complex Sub-grid Scale Turbulence Models. In *VI International Conference on Particle-based Methods - Fundamentals and Applications*, pages 510–521, Barcelona

- Gkoudesnes, C. and Deiterding, R. (2021). Verification of the WALE Large Eddy Simulation Model for Adaptive Lattice Boltzmann Methods Implemented in the AMROC Framework. In *Cartesian CFD Methods for Complex Applications*, number 3 in SEMA SIMAI, pages 107–127. Springer

Signature: Date: 31 March, 2021

Chapter 1

Introduction

1.1 Motivation

In recent years the lattice Boltzmann method (LBM) (Succi, 2001; Mohamad, 2011; Guo and Shu, 2013; Krüger et al., 2016) has achieved remarkable success in a variety of scientific fields. Some applications are in computational aeroacoustics (Shao and Li, 2019), multi-scale chemical engineering (Van den Akker, 2018) and fuel cells (Xu et al., 2017), micro-gaseous flows (Wang et al., 2016), porous media (He et al., 2019), multi-phase flows with heat transfer (Li et al., 2016), and turbulent flows (Aidun and Clausen, 2010; Jahanshaloo et al., 2013). There are specific advantages for each one of the above fields that make LBM a successful choice. However, there are some common benefits that LBM provides for any application. Its computationally inexpensive numerical scheme, straightforward parallelisation and close to linear parallel scalability make it a powerful alternative for subsonic flow simulations compared to the mainstream computational fluid dynamics solvers that discretise the Navier-Stokes (NS) equations and usually employ finite volume schemes. Moreover, the utilisation of Cartesian meshes, characteristic for the LBM, allows easy and automatic mesh generation and hence has the potential of reducing the time for setting up a simulation considerably, particularly with complex geometries.

There is a variety of LBM solvers available either for commercial use (XFlow; PowerFlow; ProLB) or open-source and research-oriented, such as Palabos (Latt et al., 2020) or OpenLB (Krause et al., 2021). One common attribute of these solvers is that they have been developed exclusively to implement the LBM without sharing an interface with other solvers, such as NS-based. In contrast to this condition, another LBM solver is currently developed in the University of Southampton as a new addition to a broader software. The AMROC (Adaptive Mesh Refinement in Object-oriented C ++) framework (Deiterding, 2011) implements patch-based, structured adaptive mesh refinement

(SAMR) generically for time-explicit finite volume methods. The LBM has been incorporated into AMROC by formulating it on cell-based data structures; treatment of embedded boundaries with a level-set-based ghost-fluid-type approach allows for effective handling of moving solid bodies. The interface of AMROC-LBM with the AMROC framework has ascribed the solver with two attributes. The first characteristic is that the numerical domain is decomposed into small blocks of cells of various sizes. During the execution of the LBM scheme itself, these blocks lack information such as their position in the global domain and they are unaware of whether they contain solid cells or not. Consequently, one needs to ensure a smooth interface at the borders of the blocks and, thus, a solution that is independent of their sizes. The second attribute is that the application of boundary conditions is based on ghost cells. Particularly in the case of embedded boundaries, AMROC LBM employs the *image-based ghost method*, a strategy that has been proposed only very recently in the LBM community (Tiwari and Vanka, 2012; Krüger et al., 2016). Moreover, the utilisation of ghost cells imposes a specific order to execute the various tasks during the numerical step. This order deviates from the mainstream way of implementing the method (Krüger et al., 2016), adding extra challenge in developing the software.

In the previous years, the AMROC-LBM solver has been used to simulate a variety of applications, such as thermal convection (Feldhusen et al., 2016), a flow around a high-speed train (Kin et al., 2016), studying of insect flight (Feaster et al., 2016), moving bodies (Laloglu and Deiterding, 2017) and the wake behind a wind turbine Deiterding and Wood (2016a,b). The main target of this project is to improve the capability of the solver to deal with high Reynolds turbulent flows. To achieve this goal, a variety of large eddy simulations (LES) models and a wall treatment approach have been implemented and tested against benchmark test cases such as homogeneous isotropic turbulence and bi-periodic channel. In the next section, I will discuss these topics and present the layout of this thesis.

1.2 Detailed Summary of the Chapters

In Chap. 2, I will present LBM and its implementation in the AMROC framework. Initially, in Sec. 2.1 I will refer to elements of kinetic theory and present the original Boltzmann equation. In Sec. 2.2, I will report the derivation of the LBM algorithm and the discretisation of the phase space, see Secs. 2.2.1 and 2.2.2. A critical aspect of any LBM implementation is the choice of the collision model. In the literature, there is a variety of proposed collision models with different levels of complexity and accuracy. The simplest option is the standard single relaxation time (STA SRT) (Krüger et al., 2016). One improvement of this model is the regularised SRT (Latt and Chopard, 2006). Some of the most popular is the family of Multi Relaxation Time (MRT) models (Lallemand and Luo, 2000), the ones based on central moments (Geier et al., 2006) and the entropic LBM (Boghosian et al., 2001). Moreover, researchers have suggested recently more improvements for the regularisation procedure (Malaspinas, 2015; Jacob et al., 2018). A comparison of some of them to deal with incompressible flows can be read in (Ezzatneshan, 2019) and turbulent simulations in (Nathen et al., 2018). Finally, a comprehensive review from a theoretical point of view can be found in (Coreixas et al., 2019). To avoid complexity, in this project, I will focus on the STA SRT and REG version, see Sec. 2.2.3. Another reason for these choices is that the SRT model was found to be less dissipative than other models and thus a better candidate for coupling with LES (Nathen et al., 2018). In Secs. 2.2.4 and 2.2.5, I will refer to the normalisation procedure, which rescaling lattice to physical units. Finally, I will close the chapter by explaining how the solver treats an external force, see Sec. 2.2.6.

In Chap. 3, I will present the LES models that have been coded in the AMROC-LBM solver. Thanks to its time-explicit numerical update and intrinsically low numerical dissipation, the LBM lends itself particularly to LES of engineering applications involving high Reynolds number flows. In literature, the majority of the proposed LES models are based on the eddy viscosity approximation (Malaspinas and Sagaut, 2012). I detail this approach in Sec. 3.1. The first attempt of coupling LES with LBM was the constant Smagorinsky (CSMA) model (Smagorinsky, 1963) reported in (Hou et al., 1994). Premnath et al. (2009b) improved the behaviour of CSMA in the vicinity of the wall by combining it with the Van Driest damping function. Another proposal to increase the accuracy of the model close to the wall was the shear-improved CSMA in (Jafari and Rahnama, 2011). Later, Malaspinas and Sagaut (2012) presented how to integrate the CSMA model consistently and, by extension, any turbulence model based on the eddy viscosity approach into the LBM framework. Other researchers have improved CSMA by extending it to the dynamic Smagorinsky (DSMA) model (Menon and Soo, 2004; Premnath et al., 2009a). Chen (2009) proposed an LES model for 2D turbulent simulations based on vorticity-streamfunction equations. The wall-adapting local eddy-viscosity (WALE) turbulence model (Nicoud and Ducros, 1999) has been tested in LBM and compared with CSMA by Weickert et al. (2010). WALE was found to be superior against CSMA with and without the Van Driest damping function in capturing the law of the wall. Like WALE, the Vreman (2004) model has also been tested to perform better close to the wall than CSMA (Liu et al., 2012). The Vreman model was found to be less expensive than WALE, though it was less accurate in capturing the eddy viscosity profile in the vicinity of the wall. Beyond the eddy viscosity approach, the approximate deconvolution method has also been validated for LBM simulations (Malaspinas and Sagaut, 2011). Furthermore, (PowerFlow) employs a Very LES strategy, based on the $k-\varepsilon$ RANS model, to deal with high Reynolds industrial applications (Li and Jammalamadaka, 2015). Given the available LES models, we decided to implement three of them. The first option was the standard CSMA model (Hou et al., 1994) due to its simplicity and robustness. The model is reported in Sec. 3.2 with two available versions depending on how the strain rate is evaluated, locally per cell or through a finite difference stencil. The second model was DSMA, discussed in Sec. 3.3. The final model was WALE, see Sec. 3.4. I chose this model due to its successful coupling with wall treatment for engineering applications (Simulia, 2021). Finally, I will conclude the chapter discussing their performance in the AMROC-LBM solver in Sec. 3.5.

Chap. 4 is devoted to the boundary conditions in the AMROC-LBM solver. Initially, in Sec. 4.1, I will divide the boundary conditions into two categories, namely *bounce-back-type* and *reconstruction* ones (Krüger et al., 2016). The latter group estimates the distribution functions in ghost cells based on some imposed macrovariables. On the other hand, the former group evaluates the unknown distribution functions per lattice direction. However, it deals only with the directions that point to the fluid domain, meaning that the state of the macrovariables is unknown after the application of such boundary condition. This will affect the application of the LES models that employ a stencil to estimate the eddy viscosity, such as WALE and DSMA. To deal with this issue, I have devised and proposed a novel algorithm to impose macrovariables in the ghost cells when a *bounce-back-type* boundary is used. Its implementation can be found in Sec. 4.3. In Sec. 4.2, I will detail how the AMROC-LBM solver implements the two types both for domain and embedded boundaries. In the case of the embedded boundaries, the solver utilises the *image-based ghost method*, as I have already mentioned. This is a relatively new strategy for LBM solvers proposed initially in (Tiwari and Vanka, 2012) for SRT and extended for MRT collision models in (Kaneda et al., 2014). Mozafari-Shamsi et al. (2016) have extended the method to deal with thermal boundaries, too. In all the above references, the simulations were focused on laminar cases. Therefore, the AMROC-LBM solver is one of the first attempts to employ the *image-based ghost method* for high Reynolds turbulent flows.

The chapter will close by presenting the newly implemented wall treatment. Independently of LBM or NS-based solvers, wall-resolved LES (WRLES) are still expensive for modern computers. In a rather optimistic estimation of the simulation of an airliner wing, a grid of 10^{11} points with $5 \cdot 10^6$ time steps would have to be used (Spalart et al., 1997). These numbers are still beyond the capability of modern computer systems, let alone for industrial application. Consequently, a wall-modelled LES (WMLES) strategy is vital. Choi and Moin (2012) have shown that for the simulation of a flat plate, the number of grid points N that is needed scale as $N \sim Re_{L_x}$ for WMLES, $N \sim Re_{L_x}^{13/7}$ for WRLES and $N \sim Re_{L_x}^{37/14}$ for DNS, highlighting the importance of the utilisation of a wall model[1]. Simultaneously, due to the dependency of LBM on Cartesian grids, a significant number of cells usually needs to be deployed in the vicinity of the body in order to approximate its shape accurately. Nonetheless, this can lead to prohibitively large meshes. The employment of a wall function can improve this situation. One of the first attempts of introducing a wall model in the LBM framework can be found in

[1] These resolutions were revisited based on the original estimations of Chapman (1979)

the thesis of Schneider (2015). He extended the two-layer wall function of Werner and Wengle (1993) to a three-layer approach for better accuracy in the buffer layer, and by employing the definition of Balaras et al. (1996) for the mixing length model (Wilcox, 2006), he was able to calculate the wall shear stress. Having the wall shear stress, he applied a Neumann boundary condition. Malaspinas and Sagaut (2014) proposed a robust way to implement a wall-modelled strategy in LBM with MRT collision model and body-fitted grid. To estimate the velocity in the first cell, they suggested either to use the analytical profile of Musker (1979) or resolve the turbulent boundary layer equations in a meshless way in the first fluid cell. The eddy viscosity was also estimated by employing the mixing length model with Van Driest damping function. The algorithm was extended to SRT collision models and non-body-fitted grids in (Haussmann et al., 2019). Moreover, besides the Musker velocity profile, they also tested Werner and Wengle and the three-layer profile of Schmitt (1988). The Musker velocity profile was found to be the most accurate. Moreover, recently, this approach has been evaluated successfully for a simulation of internal combustion against a finite volume solver of OpenFOAM (Haussmann et al., 2020). In (Wilhelm et al., 2018) and invoking the immersed boundary method for the Cartesian grid, an explicit power-law-based wall model was proposed to deal with non-body-fitted grids. This approach has recently been extended for flows under adverse pressure gradient, keeping its explicit formulation (Wilhelm et al., 2021). Finally, a wall-modelled strategy based on the skin friction coefficient and specialised for the cumulant LBM can be found in (Pasquali et al., 2020). This algorithm has been designed to port efficiently in general-purpose graphics processing units. The implementation of the wall treatment in the AMROC-LBM solver is based on (Malaspinas and Sagaut, 2014; Haussmann et al., 2020) and uses the Musker profile. It is described in Sec. 4.4. To my knowledge, this is the first time that this approach has been implemented using the *image-based ghost method*.

To verify the new implementations, a variety of benchmark testcases have been simulated. Firstly, homogeneous isotropic turbulence (HIT) configurations were employed to test the new LES models' capability to capture the expected shape of the power spectra. Chap. 5 details the three cases of HIT that I have simulated, namely forced HIT (FHIT), decaying HIT (DHIT) and Taylor Green vortex (TGV). In Sec. 5.1, I will describe the post-processing routine implemented to evaluate the power spectra and other turbulent statistics during simulations. In the case of FHIT, I use a model spectrum as a reference solution, detailed in Sec. 5.1.1. Moreover, I will also present different approaches to scaling the spectra, namely Kolmogorov spectra, that enabled us to compare spectra of different Re_λ. Sec. 5.2 is devoted to the FHIT case. In the beginning, Sec. 5.2.1 will report the employed force (Abdel Kareem et al., 2009). In comparison with other options (Cate et al., 2006; Valio et al., 2010), this force has the advantage that it is relatively easy to implement without the need for initialisation. I started all the simulations with a zero velocity field. On the other hand, the final Re_λ of the simulations depended on the spatial step and thus the need for the scaling of the spectra. I will start presenting the data with direct numerical simulations (DNS) in Sec. 5.2.2. Comparing the two collision models, I found that the STA model returned less dissipative energy spectra. This outcome has also been reported in (Nathen et al., 2018). However, examining the pressure fluctuation spectra, I noticed the STA model experienced instabilities in the highest wavenumbers. Consequently, I concluded that the more energetic spectra in the highest wavenumbers are due to instabilities arising from the less accurate collision step. Moreover, I found that LBM solvers need a resolution of $\kappa_{max}\eta \geq 5$ to simulate all the turbulent scales. This outcome is based on the comparison with the model spectrum. To the knowledge of the author, this is the first time that this comparison took place for the LBM. Finally, I noticed that a value of $Re_\lambda > 150$ is needed to obtain a level of around 1.6 in the inertial sub-range. For smaller Re_λ, the level was around 2, which deviates from the theoretical value of 1.5. In Sec. 5.2.3, I will carry out a convergence analysis for the three LES models, and in Sec. 5.2.4, I will compare them against DNS for two resolutions. I found that the introduction of the LES models has reduced the instabilities but not removed them in the case of the STA model. Moreover, I noticed that the contribution of the LES models diminished with increasing resolution. This is an indication of the consistency of the models. No significant discrepancies were observed among the LES models.

Due to the grid-dependence of the employed force, I decided to simulate DHIT in Sec. 5.3. This transition was found to be reasonably easy and efficient due to the FHIT data, which can be utilised as initial solutions. Given the easy implementation of the force and that it only needs around fifteen thousand iterations to obtain a fully turbulent flowfield, I suggest this methodology as an alternative to the initialisation of DHIT based on fabricated and rather complicated initial solutions (Yu et al., 2005b). This testcase enabled us to verify the LES models under coarse resolutions for which DNS failed to simulate. Three resolutions have been simulated. Examining the slopes of the turbulent kinetic energy and dissipation range, I found that they laid in the expected range (Huang and Leonard, 1994). Increasing the resolution, the LES results converged to the reference DNS data, showing the expected behaviour. Moreover, in the lowest resolution, I discovered that there was a deviation in the level of instabilities among the LES models, appearing in the pressure fluctuation spectra, for the STA model. This was not the case for the REG model. To further investigate this issue, I compared the behaviour of the two versions of the CSMA. In the case of the STA model, the way the strain rate is estimated affects the outcome, though, the discrepancies are diminished with finer resolution. However, the REG model did not experience such behaviour.

The final HIT case is the TGV reported in Sec. 5.4. Initially, four resolutions of DNS have been simulated for both STA and REG model. It was found that only the finest resolution was able to correctly capture the reference data of DeBonis (2013). In the case of the STA model, DNS failed to simulate the coarsest resolution, indicating extensive instabilities. However, examining the evolution of the dissipation rates and the spectra, there is a strong indication that the REG model is more dissipative than STA. This is in agreement with the literature (Nathen et al., 2018). For the evaluation of the LES models, a convergence analysis with three resolutions was carried out and reported in Secs. 5.4.1, 5.4.2 and 5.4.3. Particularly in the case of Sec. 5.4.2, an extensive comparison of the vorticity magnitude and the eddy viscosity field took place. I will show that both DSMA and WALE underperformed during the initial laminar phase, resulting in excess dissipation compared to CSMA. Another important outcome is that the averaged estimation of the constant of the model was unaffected by choice of collision model. Finally, the comparison of the eddy viscosity fields yielded vital information for how each model adapts to the flowfield.

The following benchmark testcase was the bi-periodic channel, discussed in Chap. 6. In the beginning, I will report the force, and the post-processing routine that was implemented to collect the mean averaged data. Afterwards, in Sec. 6.1, I will present the WRLES case of the bi-periodic channel. All three LES models accurately matched the reference data (Kim et al., 1987; Liu et al., 2012). Overall, the WALE model was found to be the most accurate. Utilising the same simulation, in Sec. 6.1.1, I will evaluate the new algorithm for imposing macrovariables in ghost cells. The results indicated that WALE has benefited from the employment of the new algorithm. In Sec. 6.2, I will introduce the two WMLES cases. The purpose of these cases is the verification of the newly implemented wall treatment. Both cases would be impossible to execute without its employment. The results will be reported in Secs. 6.2.1 and 6.3. In general, my implementations were able to capture the first-order statistics accurately. DSMA and, particularly, WALE have benefited significantly from the coupling with the wall treatment, outperforming CSMA.

The final test case, which I will discuss in Chap. 7, is the experiment reported in (Faria and Francisco, 2018; Faria et al., 2019). The purpose of this case was to test the new wall treatment for embedded boundaries. For robustness, I ran this test case only with CSMA and a relatively high value of the constant. The solver was able to predict reasonably well the mean averaged streamline velocity profiles, however with some deviation in longitudinal values.

Finally, in Chap. 8 I will summarise the outcomes of this project and suggest some future directions.

Chapter 2

Lattice Boltzmann Method

This chapter is devoted to explaining the LBM and particularly the methodology implemented in the AMROC LBM solver. In the beginning, a brief discussion about kinetic theory and the idea behind the Boltzmann Equation (BE) will take place. Having introduced the BE, I will proceed with the derivation of the LBM. In particular, I will comment on the discretisation of the phase space followed by the implementation into the solver. I will also present the rescaling of physical units to lattice ones, which is executed by the program. Finally, the force scheme that has been used during this project will be given.

2.1 Elements of Kinetic Theory

In macroscopic continuum mechanics, the motion of a fluid can be described by a sum of conservation laws accompanied by some phenomenological relations. This procedure leads to the Euler and Navier-Stokes equations.

On the other side of the spectrum, a fluid, such as air, consists of a vast number of particles/molecules. An estimation of this number can be done as

$$n = \frac{p}{k_B T}, \tag{2.1}$$

where n [m^{-3}] is the particle number density, p [Nm^{-2}] the pressure, k_B the Boltzmann constant defined as 1.38 10^{-23} m^{2}kg s^{-2}K^{-1} and T [K] the temperature. For standard conditions, the result is of the order of 10^{25} particles per volume. In a microscopic simulation, the move of these particles would be described by Newtonian mechanics. However, the direct tracking and interactions of such a huge number are impossible even with the current computational capacity. A statistical approach can be followed to deal with this obstacle, leading to the mesoscopic description and the Boltzmann Equation

(BE). To extend this discussion, the order of magnitude of the diameter (d) of the air molecules (oxygen and nitrogen) is 1 angstrom (10^{-10} [m]). Therefore, a characteristic collision cross-section can be estimated as $\sigma = \pi d^2$. Knowing σ, the mean free path can be deduced as $l = (n\sigma)^{-1}$. The mean free path is a characteristic average length that a particle has to travel to meet another particle. The above can be combined to create the non-dimensional parameter Kn, Eq. (2.2), a quantity that prescribes whether fluid can be treated as a continuum or not.

$$Kn = \frac{l}{L}, \tag{2.2}$$

where L is a characteristic macroscopic length. For a typical engineering application, it is of the order of 1 mm. Consequently, for the scenario of air Kn is of the order of 10^{-5}, much lower than the continuum limit, which is 0.1, and thus the NS equations can be applied.

Of high importance for models based on kinetic theory is the *velocity distribution function* $f(\mathbf{x}, \mathbf{e}, t)$, where $\mathbf{x}$ and $\mathbf{e}$ are the position and velocity, respectively, of a particle. Its dimensions are $[f] = [n][\mathbf{e}^{-3}] = [s^3][m^{-6}]$. In other words, it describes the possibility of a particle to be found in position $\mathbf{x}$ with velocity $\mathbf{e}$. Consequently, the three-dimensional *physical space* is replaced by the six-dimensional *phase space* (three components of space and three particle-velocity components).

A property of $f(\mathbf{x}, \mathbf{e}, t)$ is that

$$n(\mathbf{x}, t) = \frac{\rho(\mathbf{x}, t)}{m} = \iiint\limits_{-\infty}^{\infty} f(\mathbf{x}, \mathbf{e}, t)\ \partial e_1 \partial e_2 \partial e_3, \tag{2.3}$$

where ρ is the fluid density and m is the mass of an individual particle. The kinetic equation

$$\frac{\partial f}{\partial t} + \mathbf{e} \cdot \frac{\partial f}{\partial \mathbf{x}} = \Im \tag{2.4}$$

describes the evolution of $f(\mathbf{x}, \mathbf{e}, t)$ in time.

The left-hand side is called the streaming operator. Its first term is the temporal derivative of f, and the second is the advection of f (spatial acceleration). On the other hand, the right-hand side is the collision operator. The idea behind this equation is that the temporal evolution of f is the result of the streaming of the particles and their collisions with each other, which leads to loss or gain of kinetic energy.

The complexity of the above equation is due to the definition of the collision operator $\Im$. The simplest way to express this operator for Maxwellian fluids (no reactive collisions) is due to the Maxwell-Boltzmann distribution function f^{eq} that describes an equilibrium state, or in other words, the state with the higher entropy. Therefore, as time passes, the distribution function f will relax to the Maxwell-Boltzmann equilibria value with a given relaxation time τ. This method is called Single Relaxation Time (SRT) or BGK, Eq. (2.5), based on the work of Bhatnagar et al. (1954).

$$\Im = \frac{1}{\tau}(f^{eq} - f). \tag{2.5}$$

2.2 Lattice Boltzmann Equation

In the previous section, I have referred to the BE Eq. (2.4) as a continuous function in the phase space. A step towards the lattice Boltzmann Method is the discrete velocity model of BE,

$$\frac{\partial f_\alpha}{\partial t} + \mathbf{e}_\alpha \cdot \frac{\partial f_\alpha}{\partial \mathbf{x}} = \frac{1}{\tau}(f_\alpha^{eq} - f_\alpha). \tag{2.6}$$

Here, the time and space are still continuous coordinates, but the particles' velocities $\mathbf{e}$ have been replaced by a discrete set $\mathbf{e}_\alpha$ with index α denoting a lattice direction. The group of the lattice directions consists of A elements. More details about this derivation can be found in the article of Aidun and Clausen (2010) or the textbook of Krüger et al. (2016).

2.2.1 Discretisation of phase space

The first step is to choose the number A of the lattice directions and thus the lattice model. Since one wants the method to simulate flows, the number of the microscopic velocities and their directions should satisfy symmetries appearing in the NS equations and the conservation of mass and momentum (Qian et al., 1992; Latt, 2007; Krüger et al., 2016). The notation used to refer to a velocity set is $DdQA$, where d is the dimensions of the problem. There is a variety of proposed models that can be constructed for both 2D and 3D cases. In general, increasing the number of directions A leads to a more accurate model though this has the drawback of increasing the memory requirements of the simulations. In two dimensions, the two most employed models are $D2Q7$ and $D2Q9$, while in 3D $D3Q15$, $D3Q19$ and $D3Q27$. Numerical experiments with $D3Q15$ have shown that this model is unstable and inaccurate (Mei et al., 2000). Comparing the performance of $D3Q19$ and $D3Q27$ and how they affect wall-bounded turbulent flows can be found in (Kang and Hassan, 2013). The conclusion was that $D3Q19$ had returned less accurate turbulent statistics in some cases where the flow was not aligned correctly with the Cartesian grid, while $D3Q27$ has demonstrated better performance. However, $D3Q27$ was found to be 30% more expensive both in memory usage and execution time. In this project, I have focused on 3D simulations. Consequently, the $D3Q19$ model has been utilised instead of $D3Q27$ to reduce the cost of simulations.

Simultaneously with the $D3Q19$ model, I will also present $D2Q9$ since I will use it as a reference to demonstrate some features in the next chapters. Figures 2.1 shows the

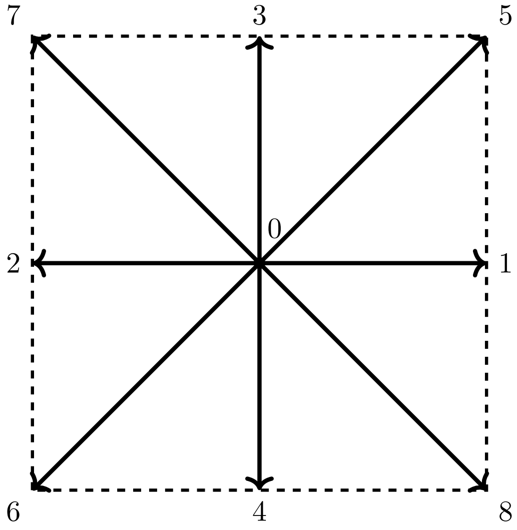


Figure 2.1: The lattice directions for the $D2Q9$ model.

discretisation of the microscopic velocities for the $D2Q9$ model. The vectors point to the eight neighbour cells, where the corresponding distribution functions will be transferred after each time step. The point at the centre indicates that this specific distribution function will not be affected by the streaming. The velocity vectors $\mathbf{e}_\alpha$ and weights w_α of $D2Q9$ read as, (He and Luo, 1997):

$$\mathbf{e}_\alpha = \begin{cases} 0, & w_\alpha = \frac{4}{9}, \quad \alpha = 0 \\ (\pm 1, 0)c, (0, \pm 1)c, & w_\alpha = \frac{1}{9}, \quad \alpha = 1, \dots, 4, \\ (\pm 1, \pm 1)c, & w_\alpha = \frac{1}{36}, \quad \alpha = 5, \dots, 8. \end{cases} \tag{2.7}$$

The discretisation of the microscopic velocities for the $D3Q19$ model is illustrated in Fig. 2.2. Examining the picture, one can distinguish the directions into two categories, excluding the central one. The first category consists of the directions that point to the cube phases and the second to the edges. This information will be useful for demonstrating the proposed algorithm for imposing macrovariables at the ghost cells in Sec. 4. For their characteristics, it yields:

$$\mathbf{e}_\alpha = \begin{cases} 0, & w_\alpha = \frac{12}{36}, \quad \alpha = 0 \\ (\pm 1, 0, 0)c, (0, \pm 1, 0)c, (0, 0, \pm 1)c, & w_\alpha = \frac{2}{36}, \quad \alpha = 1, \dots, 6, \\ (\pm 1, \pm 1, 0)c, (\pm 1, 0, \pm 1)c, (0, \pm 1, \pm 1)c, & w_\alpha = \frac{1}{36}, \quad \alpha = 7, \dots, 18 \end{cases} \tag{2.8}$$

In the case of the $D3Q27$, there would be eight further direction points to the corners of the cube. These new directions would shape the third category with velocity vectors $(\pm 1, \pm 1, \pm 1)c$.

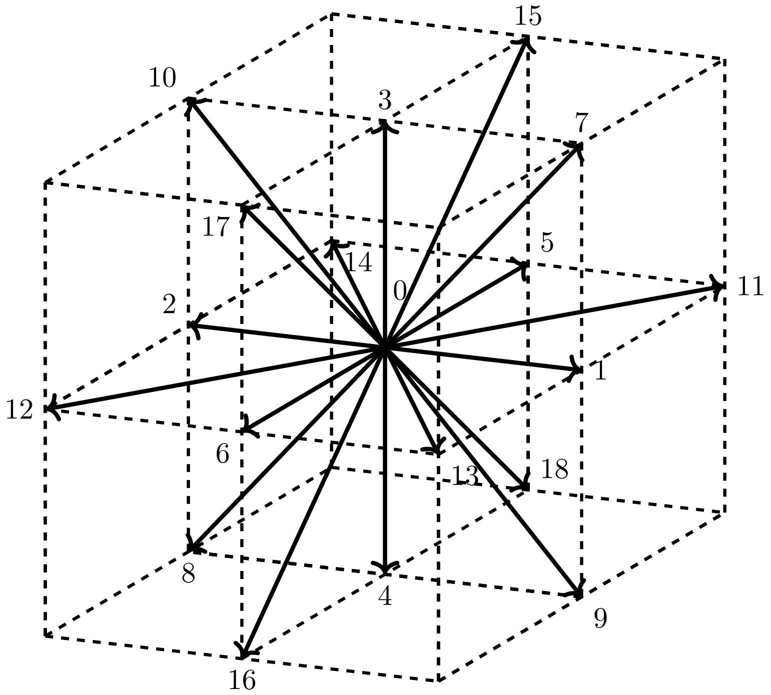


Figure 2.2: The lattice directions for the $D3Q19$ model.

To close the discretisation of the phase space, spatial Δx and temporal Δt steps need to be introduced. In the LBM framework, these two variables are related as $c = \Delta x/\Delta t$, where c is the lattice velocity (Dellar, 2013). The time-space discretisation that has been implemented in AMROC follows the common practice of a two-step procedure (Guo and Shu, 2013). The first step, namely *streaming*, is a time explicit transport equation of the form

$$\mathscr{S}: \qquad \tilde{f}_\alpha(\mathbf{x} + \mathbf{e}_\alpha \Delta t, t + \Delta t) = f_\alpha(\mathbf{x}, t). \tag{2.9}$$

The second step is the computation of the contribution of the collision operator. Introducing the term of the non-equilibrium part $\tilde{f}_\alpha^{neq}(\mathbf{x}, t) = \tilde{f}_\alpha(\mathbf{x}, t) - f_\alpha^{eq}(\mathbf{x}, t)$, this step can be expressed as

$$\mathscr{C}: \qquad f_\alpha(\mathbf{x}, t + \Delta t) = \tilde{f}_\alpha(\mathbf{x}, t) - \frac{\Delta t}{\tau}\tilde{f}_\alpha^{neq}(\mathbf{x}, t) = f_\alpha^{eq}(\mathbf{x}, t) + (1 - \frac{\Delta t}{\tau})\tilde{f}_\alpha^{neq}(\mathbf{x}, t). \tag{2.10}$$

The notation $\tilde{f}_\alpha$ refers to the intermediate state of the distribution function after the streaming and before the collision. The equilibrium function used in AMROC reads

$$f_\alpha^{eq}(\mathbf{x}, t) = w_\alpha \rho \left[1 + \frac{\mathbf{e}_\alpha \cdot \mathbf{u}}{c_s^2} + \frac{(\mathbf{e}_\alpha \cdot \mathbf{u})^2}{2c_s^4} - \frac{\mathbf{u} \cdot \mathbf{u}}{2c_s^2} \right] + O(\mathbf{u}^3). \tag{2.11}$$

Equation (2.11) is a *low-Mach number approximation* (Frisch et al., 1987) of the Maxwell-Boltzmann distribution function. As a consequence of this approximation, the Lattice Boltzmann Equations (LBE), Eq. (2.6), can only be used for low Ma numbers, usually

Ma ≤ 0.2, and thus incompressible flows. The notation c_s is the physical speed of sound related to c as $c_s = c/\sqrt{3}$.

The macrovariables density ρ [kg m^{-3}] and velocity field $\mathbf{u}$ [m s^{-1}] can be calculated through the zeroth and first moments of the distribution function yielding

$$\rho(\mathbf{x}, t) = \sum_{\alpha} f_{\alpha}(\mathbf{x}, t) = \sum_{\alpha} f_{\alpha}^{eq}(\mathbf{x}, t), \tag{2.12a}$$

$$\rho(\mathbf{x}, t) u_i(\mathbf{x}, t) = \sum_{\alpha} e_{\alpha i} f_{\alpha}(\mathbf{x}, t) = \sum_{\alpha} e_{\alpha i} f_{\alpha}^{eq}(\mathbf{x}, t), \tag{2.12b}$$

where u_i is the component of the velocity in the direction i. The kinematic viscosity ν can be estimated as (Hénon, 1987)

$$\nu = c_s^2 \Delta t (\tau - 0.5). \tag{2.13}$$

The above procedure leads to the SRT LBM model (Qian et al., 1992). The numerical integration leads to first order accuracy in time and second order accuracy in space (Junk, 2001; Junk et al., 2005). A Chapman-Enskog expansion of the aforementioned LBM model (Hudong et al., 1992; Hou et al., 1994) leads to the athermal weakly compressible Navier-Stokes set of equations

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_i}{\partial x_i} = 0, \tag{2.14a}$$

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{\partial p}{\partial x_j} + \nu \frac{\partial^2 u_i}{\partial x_j^2}, \tag{2.14b}$$

where the pressure p can be evaluated from the equation of state as $p = \rho c_s^2$. Though the $D3Q19$ model respects the conservation of mass and momentum, it does not preserve the energy, and thus the current set-up is only valid for athermal flows. With the introduction of a second distribution function to handle temperature variations, a two-population model needs to be employed for thermal flows (Krüger et al., 2016).

Besides the zeroth and first moments of the distribution function, the second order moments can also yield useful information (Krüger et al., 2009). The convective term of the momentum flux tensor, appearing in NS equations, can be calculated based on the equilibrium part as

$$\Pi_{ij}^{eq} = \sum_{\alpha} e_{\alpha i} e_{\alpha j} f_{\alpha}^{eq} = \rho c_s^2 \delta_{ij} + \rho u_i u_j \ . \tag{2.15}$$

On the other hand, the shear stress tensor σ_{ij} is related to the non-equilibrium part as

$$\Pi_{ij}^{neq} = \sum_{\alpha} e_{\alpha i} e_{\alpha j} f_{\alpha}^{neq} = -\frac{c_s^2 \tau}{\nu} \sigma_{ij} \ . \tag{2.16}$$

From Eq. (2.16), one can also devise a relation to estimating the strain rate S_{ij} that reads

$$S_{ij} = -\frac{1}{2\rho c_s^2 \tau} \Pi_{ij}^{neq}. \tag{2.17}$$

This last equation provides the opportunity to calculate the strain rate locally per cell without applying an FD stencil. This will be very useful in some LES models that I will discuss in the next chapter.

Mass and momentum conservation imposes some restrictions on the first and second moments of the non-equilibrium part. These restrictions can be derived from Eq. (2.12) and read

$$\sum_\alpha f_\alpha^{neq}(\mathbf{x}, t) = 0, \tag{2.18a}$$

$$\sum_\alpha e_{\alpha i} f_\alpha^{neq}(\mathbf{x}, t) = 0, \tag{2.18b}$$

meaning that the collision step should not alter the macrovariables density ρ and velocity field $\mathbf{u}$. However, it will affect the calculation of variables such as the strain rate Eq. (2.17).

2.2.2 Implementation in AMROC

At this point, I will comment on the implementation of the LBM algorithm in the AMROC framework. First of all, the implementation is based on a cell-centred approach rather than the more frequently employed node-based strategy (Schneider, 2015; Latt et al., 2020). This behaviour has been inherited by constructing the AMR capability in the software (Deiterding, 2011). The cell-based formulation is mandatory for AMR algorithms to be conservative in kinetic energy and density at interfaces of different levels.

Another important artefact of the AMROC interface is the decomposition of the domain into smaller blocks of variable sizes. This fragmentation benefits memory allocation by restricting access to big chunks of memory that could create traffic jams to the different caches and registers of processors, thus reducing the overall performance. However, this choice also affects the design and coding of the algorithms implemented in AMROC. For the communication of blocks with each other, external ghost cells surrounding them are utilised. The number of layers of external ghost cells depends on the needs of the employed method.

To further assist the discussion, Algorithm 2.1 illustrates the implementation of the numerical step for LBM. At the beginning of the numerical step, the program estimates and imposes values, in this case, the distribution functions, in ghost cells utilised to model boundary conditions. There are two kinds of boundary conditions. The first one is the domain boundaries appearing on the edges of the numerical domain. The second kind is the embedded boundaries that represent solid bodies inside the domain. Finally, a synchronisation also takes place to update the values of external ghost cells that are part of other blocks.

The next step is time integration. The program parses all the blocks and executes streaming, Eq. (2.9), and collision, Eq. (2.10), in the cells of each block. It is essential to mention that the function *Step* is executed locally per block, meaning that the block has limited information about its position on the whole domain, and it is unaware of whether to contain or not domain or embedded boundaries. The last observation plays a significant role in applying algorithms based on stencils, as is the case for some LES models. Finally, the step finishes with the calculation and storage of the outputs.

It is important to comment here that the order of the various tasks, which appear in Algorithm 2.1, deviates for the mainstream implementation (see Krüger et al., 2016, page 67). The most common order is the execution of the collision, followed by streaming and then the application of the boundaries. This deviation will lead to significant challenges, as I will discuss in the next chapters.

A simplified representation of the decomposition of the 2D domain into blocks can be seen in Fig. 2.3. This part of the domain consists of four blocks. There is one layer of external ghost cells, as the dashed lines surrounding the blocks illustrate. Blocks 1 and 2 share a domain boundary on the left, pictured with a lighter colour. On the other hand, blocks 3 and 4 share a solid body, which is part of a circle. The cells that represent the surface of the body are depicted in a lighter tone. During the application of the boundary conditions, the program will impose the new distribution functions in the regions with the lighter tone and update the one layer of external ghost cells. In the next phase, the streaming will occur both in the interior region of blocks and the external layer of ghost cells. This also means that it will be applied in the internal ghost cells, the embedded boundaries, and the rest of the solid cells.

On the other hand, collision will occur only in the blocks' internal region, excluding the external ghost cells. However, it will also apply to all solid cells. In other words, during the execution of the *step* function, blocks 3 and 4 have no information about the solid body. From the above discussion, it is clear that it is of utmost importance to correctly set-up the ghost cells' values since one cannot affect them after this stage.

During this discussion, I have assumed a uniform grid. About the AMR implementation, the interested reader can refer to (Deiterding, 2011).

Algorithm 2.1: Execution of a numerical step in AMROC

```
Function Boundary Conditions
    Result: Set-up the values in ghost cells
    begin
        Apply domain boundaries;                          /* e.g. inlet, outlet */
        repeat
            Apply embedded boundaries;                    /* e.g. no-slip wall */
            Update ghost cells at the boundaries of blocks
        until completion;
    end
end
forall Processors do
    forall Blocks in the current processor do
        Function Step
            Result: Evolve one $\Delta t$ and update the values in the current block
            begin
                Execute streaming;                        /* Eq. (2.9) */
                Execute collision;                        /* STA SRT version Eq. (2.10) */
            end
        end
    end
end
Function Output
    Result: Store the data in binary or ASCII format
    begin
        forall Processors do
            forall Blocks in the current processor do
                Calculate the user-specific variables.
            end
        end
    end
end
```

2.2.3 LBM with regularised pre-collision distribution function

In Eq. (2.17), I have shown that the non-equilibrium part of the distribution function is related to the strain rate. Consequently, the symmetries that characterise the strain rate should also be reflected in the non-equilibrium part. Moreover, these imposed symmetries are vital for satisfying the conservation of mass and momentum for the collision step, Eq. (2.18). Nonetheless, due to the discretisation of the phase space, Eq. (2.10) has been found to underperform, particularly for high Re and high Ma number flows. One countermeasure that has been proposed early in the history of LBM is the regularisation of the non-equilibrium part (Latt and Chopard, 2006).

Based on the Chapman-Enskog procedure, the f_α can be expanded as $f_\alpha = f_\alpha^{(0)} + \varepsilon f_\alpha^{(1)} + \varepsilon^2 f_\alpha^{(2)} + \ldots$, where for the zeroth order holds $f_\alpha^{eq} = f_\alpha^{(0)}$. An important assumption is

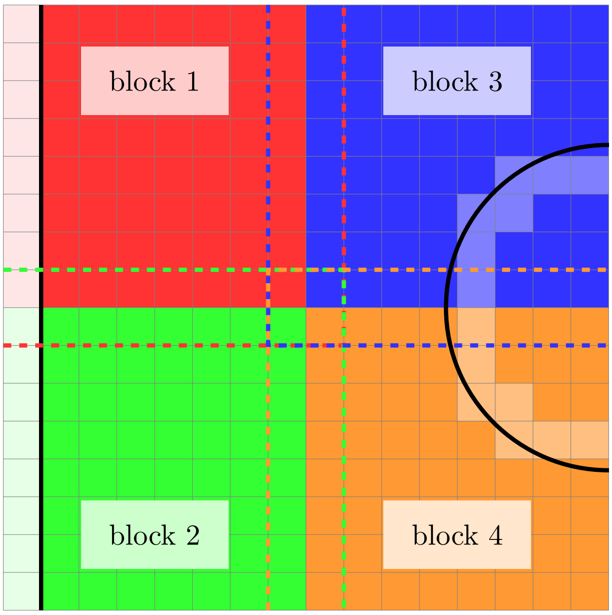


Figure 2.3: Decomposition of the numerical domain into blocks.

$f_\alpha^{neq} \approx f_\alpha^{(1)}$. The NS equations can be derived using the first two terms of the expansion, namely $f_\alpha^{(0)}$ and $f_\alpha^{(1)}$. A way to improve the accuracy of the non-equilibrium part and avoid higher order contributions is to impose directly on it the symmetry of the strain rate. Using Eq. (2.16), one can end up with a relation for $f_\alpha^{(1)}$ that reads

$$(f_\alpha^{neq})_{reg} \approx f_\alpha^{(1)} = \frac{w_\alpha}{2c_s^4} Q_{\alpha ij} \Pi_{ij}^{neq}, \tag{2.19}$$

where $Q_{\alpha ij} = e_{\alpha i} e_{\alpha j} - c_s^2 \delta_{ij}$. This extra step will alter the computation of the collision step as

$$\mathscr{C}: \qquad (f_\alpha)_{reg}(\mathbf{x}, t + \Delta t) = f_\alpha^{eq}(\mathbf{x}, t) + (1 - \frac{\Delta t}{\tau})(\tilde{f}_\alpha^{neq})_{reg}(\mathbf{x}, t). \tag{2.20}$$

This procedure leads to the REG SRT collision model. The only significant change in the algorithm is the execution of Eq. (2.19) before the application of the collision step as one can see in Algorithm 2.2.

2.2.4 Normalisation, from lattice units to physical ones

The user interface of the AMROC LBM solver has been designed to accept input parameters in physical dimensions. However, the algorithms run internally in lattice units. Consequently, a normalisation procedure needs to take place to transform the data from physical to lattice units. To achieve this goal, a variety of scaling factors are calculated.

In the case of the length, the lattice symmetry imposes that the lattice spatial and temporal step should be one ($\delta x = \delta y = \delta z = \delta t = 1$). Consequently, the size of a specific dimension in the lattice space is the number of cells N_c $[-]$ in this direction.

Algorithm 2.2: The function *step* for the REG version

```
forall Processors do
    forall Blocks in the current processor do
        Function Step
            Result: Evolve one $\Delta t$ and update the values in the current block
            begin
                Execute streaming;                          /* Eq. (2.9) */
                Apply regularisation;                       /* Eq. (2.19) */
                Execute collision;          /* REG SRT version Eq. (2.20) */
            end
        end
    end
end
```

On the other hand, a length L $[m]$ describes this dimension in the physical space. A normalization factor L_0 can be defined as

$$L = L_0 N_c \implies L_0 = \frac{L}{N_c} = \Delta x. \tag{2.21}$$

The lattice symmetry also applies to the physical space meaning that ($\Delta x = \Delta y = \Delta z$).

The next scaling factor that is calculated by the program is the one for the velocity U_0. The user needs to specify the physical speed of sound c_s as input. From theory, it is known that the lattice speed of sound $\check{c}_s$ for the $D3Q19$ model equals $1/\sqrt{3}$. Therefore, one can devise the factor as

$$U_0 = \frac{c_s}{\check{c}_s}. \tag{2.22}$$

Combining the two factors, one can estimate the time scaling T_0 as

$$T_0 = \frac{L_0}{U_0}. \tag{2.23}$$

In a previous section, I have mentioned that Δx and Δt are related through the equation $c = \Delta x/\Delta t$. The same relation stands true for the lattice units and reads $\check{c} = \delta x/\delta t = 1$. Combining these two, one can conclude that the physical, temporal step can be calculated as $\Delta t = T_0$. This analysis also shows that the temporal step depends strongly on the spatial one. Moreover, normalisation for the density is also needed. Imposing the lattice density $\check{\rho}$ as 1, one can set the factor as $\rho_0 = \rho_\infty$, i.e., the physical density of the employed liquid. One can deduce more scaling factors with the above normalisations, such as for viscosity $\nu_0 = L_0^2/T_0$.

Finally, one needs to non-dimensionalise the relaxation time τ. However, instead of focusing on τ, AMROC LBM calculates the relaxation frequency ω as

$$\omega = \frac{\Delta t}{\tau} = \frac{c_s^2 \Delta t}{\nu + \frac{c_s^2 \Delta t}{2}}. \tag{2.24}$$

Examining this equation, one can conclude that the maximum value that ω can achieve is 2. In order for this to happen, the viscosity ν should be zero. This last condition is not valid and can lead to unstable simulations (Succi, 2001). However, if one replaces c_s and ν with the physical values of air, it is clear that ω will reach a value very close to 2, which may lead to unstable solutions (Krüger et al., 2016).

2.2.5 Accelerating simulations through rescaling

Through normalisation, the program provides two options to accelerate the flow internally by increasing Δt and keeping the same Δx. The first option is available by examining Eqs. (2.22) and (2.23). Decreasing the value of the physical speed of sound c_s will also reduce the value of the scaling factor U_0. The latter effect will lead to a proportional increase of T_0 and the physical time step Δt. However, there are two side-effects. Firstly, if the value of the physical speed of sound must remain constant, such as in aeroacoustic simulations, this trick is not valid. The second drawback is more general. By the definition of Ma number, both in physical and lattice space,

$$Ma = \frac{u_\infty}{c_s} = \frac{u_\infty / U_0}{c_s / U_0} = \frac{\check{u}_\infty}{\check{c}_s}, \tag{2.25}$$

it is expected that by decreasing the physical speed of sound, the non-dimensional parameter will increase. Nonetheless, this may lead to unstable simulations, as I have mentioned previously, and thus this sets a limit.

The first option does not need to alter the coding of the previously mentioned algorithms. This is not true for the second one, which is based on the introduction of an accelerating parameter, called the *speed-up factor* s. The idea besides this parameter originates from the definition of Re number as

$$Re = \frac{u_\infty L}{\nu} = \frac{\check{u}_\infty N_c}{\check{\nu}}. \tag{2.26}$$

One can multiply both numerator and denominator of the second fraction with s without altering the Re number value. This will lead to alterations of lattice velocities as $(\check{u}_\infty)_{sp} = s \cdot \check{u}_\infty$ and viscosity as $(\check{\nu})_{sp} = s \cdot \check{\nu}$. However, the physical velocity u_∞ should stay unaffected; otherwise, this will change the output. To achieve this goal, the definition of the velocity scaling factor needs to be altered as

$$U_0 = \frac{u_\infty}{(\check{u}_\infty)_{sp}}. \tag{2.27}$$

This alteration will lead to similar behaviour as in the case of the first option. Instead of decreasing the numerator (physical speed of sound), one can increase the denominator (velocity in lattice space).

Another important observation is that both options will affect the estimation of the relaxation frequency ω, Eq. (2.24). Either decreasing c_s or increasing ν will lead to a decrease of the value of ω and thus improving the stability of simulations.

2.2.6 Application of external force

Till now, I have reported the LBM implementation in AMROC without the application of an external force. In this project, there were two cases where a force should be employed. Therefore, in this section, I will discuss its implementation.

Guo et al. (2002a) have commented on a variety of potential implementations of force schemes in LBM with different levels of accuracy and complexity and how they affect the consistency of the method to replicate the NS equations. The accuracy of these schemes is based on the characteristics of the force itself. A force field with large spatial and temporal gradients and high values of its magnitude tends to be more inaccurate for some implementations. The current implementation in AMROC LBM may lead to such inaccuracies. However, during this project, the forces that I have employed are either with smooth spatial derivatives and low magnitude, Sec. 5.2, or without spatial derivatives at all, Chap. 7. Therefore, I kept this formula that reads

$$\mathscr{C}: \qquad f_\alpha(\mathbf{x}, t+\Delta t) = f_\alpha^{eq}(\mathbf{x}, t) + (1-\omega)\tilde{f}_\alpha^{neq}(\mathbf{x}, t) + \frac{w_\alpha(\mathbf{e}_\alpha \cdot \mathbf{F})}{c_s^2}. \tag{2.28}$$

The above formula is valid for both collision models. The force is applied after the collision step. The implemented algorithm in AMROC can be seen in Algorithm 2.3.

Algorithm 2.3: The function *step* with application of external force

```
forall Processors do
    forall Blocks in the current processor do
        Function Step
            Result: Evolve one $\Delta t$ and update the values in the current block
            begin
                Execute streaming;                                   /* Eq. (2.9) */
                if STA SRT then
                    Execute collision;                  /* STA SRT version Eq. (2.10) */
                else
                    Apply regularisation;                           /* Eq. (2.19) */
                    Execute collision;                  /* REG SRT version Eq. (2.20) */
                end
                Add the force;                              /* Last term of Eq. (2.28) */
            end
        end
    end
end
```

Chapter 3

Large Eddy Simulations in AMROC-LBM

Up to this point, a discussion has taken place about the LBM implementation without any handling of turbulence modelling. This would force us to low Re numbers, while real engineering applications would be prohibitively expensive. To surpass this obstacle, three turbulence models of LES have been implemented in AMROC, namely CSMA, DSMA and WALE. In this section, a presentation of the current available LES models and their implementation in AMROC will be given.

3.1 LBM and Turbulence Modeling

It was the pioneering work of Hou et al. (1994) that introduced the idea of a subgrid model for LBM based on the LES model of Smagorinsky (1963). Following the LES approach, a space filtering operation is applied to the distribution function yielding

$$\overline{f}_\alpha(\mathbf{x},t) = \iiint f_\alpha(\mathbf{x},t)\mathbf{G}(\mathbf{x},\mathbf{x}')d\mathbf{x}'. \tag{3.1}$$

For the convenience and efficiency to apply it in the physical space, the *box filter* has been employed as the filter function $G(\mathbf{x},\mathbf{x}')$ that reads

$$\mathbf{G}(\mathbf{x},\mathbf{x}') = \begin{cases} \dfrac{1}{\Delta x}, & |\mathbf{x}-\mathbf{x}'| < \dfrac{\Delta x}{2} \\ 0, & |\mathbf{x}-\mathbf{x}'| \geq \dfrac{\Delta x}{2}. \end{cases} \tag{3.2}$$

The filtering also affects other quantities, such as the velocity field, $(\mathbf{u} \rightarrow \overline{\mathbf{u}})$.

In the equations presented in the previous chapter, the relaxation time τ is a global variable depending only on the physical speed of sound c_s, the viscosity of the gas ν and the time step Δt. To introduce a sub-grid scale model in LBM, the relaxation time is transformed into a local variable computed based on the local resolved quantities, called the effective relaxation time τ_{eff}. There is no other change in the equations presenting in the previous chapter. Therefore, under these conditions, the collision step reads

$$\mathscr{C}: \qquad \overline{f}_\alpha(\mathbf{x}, t+\Delta t) = \overline{f}_\alpha^{\,eq}(\mathbf{x}, t) + \left(1 - \frac{\Delta t}{\tau_{eff}}\right)\overline{\tilde{f}}_\alpha^{\,neq}(\mathbf{x}, t). \tag{3.3}$$

The same alteration applies to the REG SRT model.

Hou et al. (1994) have stated that altering the relaxation time is analogous to changing the mean free path of the particles. Invoking the mixing length theory of Prandtl, one can argue that altering the mean free path is equivalent to changing the viscosity, leading to the idea of a turbulent viscosity ν_t. From Eq. (2.24), one can show that the calculation of τ_{eff} is

$$\tau_{eff} = \frac{\nu_{eff} + \dfrac{c_s^2 \Delta t}{2}}{c_s^2}, \tag{3.4}$$

where ν_{eff} is the effective viscosity which can be estimated as

$$\nu_{eff}(\mathbf{x}, t) = \nu + \nu_t(\mathbf{x}, t). \tag{3.5}$$

The general equation that describes the eddy viscosity ν_t reads

$$\nu_t = (C\Delta x)^2 OP_{LES}, \tag{3.6}$$

where C is the constant of the specific LES model. The above equation has been derived based on dimensional analysis, and OP_{LES} refers to the time scale of turbulence.

3.2 Constant Smagorinsky

In the case of CSMA (Smagorinsky, 1963), the eddy viscosity reads

$$\nu_t = (C_S \Delta x)^2 |\overline{S}|, \tag{3.7}$$

where C_S is the constant of the model, a user-defined value usually varied from 0 to 0.2. Examining the equation is clear that

$$OP_{\text{CSMA}} = |\overline{S}| = \sqrt{2\overline{S}_{ij}\overline{S}_{ij}}, \tag{3.8}$$

with $|\overline{S}|$ the intensity of the strain rate and $\overline{S}_{ij}$ the filtered strain rate

$$\overline{S}_{ij} = \frac{1}{2}\left(\frac{d\overline{u}_i}{dx_j} + \frac{d\overline{u}_j}{dx_i}\right). \tag{3.9}$$

LBM provides two possible ways to estimate the eddy viscosity for CSMA. The first option is to calculate the strain rate based on a finite difference stencil. This capability is currently available for the AMROC LBM solver. The implementation, which is scaled on lattice units, is based on a central finite difference scheme that reads

$$\frac{d\bar{\check{u}}_i}{d\check{x}_j} \approx \frac{\bar{\check{u}}_{i+1} - \bar{\check{u}}_{i-1}}{2}. \tag{3.10}$$

The velocity field $\bar{\mathbf{u}}$ is calculated from Eq. (2.12). Having estimated the strain rate, it is straight forward to calculate its intensity Eq. (3.8) and then the eddy viscosity Eq. (3.7). In this way, it is evident that communication between neighbour cells is mandatory even during the collision step. One important note here is that the velocity field used to estimate the strain rate is evaluated before the streaming. This is possible in AMROC because the software provides two sets of distribution functions with the same values at the beginning of the *step* function. Then, the streaming affects only one of them, leaving the other to represent the state before it. Therefore, I employ the second set for the evaluation of stencil calculations for the turbulence models. Though this means that an error of the order of Δt is introduced, this condition is vital for the correct values of macrovariables in ghost cells that have been calculated during the boundary conditions.

However, a second option allows the algorithm to keep its locality for the collision step. Starting from Eq. (3.4), one can substitute Eq. (3.5) and Eq. (3.7) yielding

$$\tau_{eff} = \frac{\nu + (C_S \Delta x)^2 |\bar{S}| + \dfrac{c_s^2 \Delta t}{2}}{c_s^2} \quad [s], \tag{3.11}$$

in physical units. As it has been shown in the previous chapter, LBM provides a way to estimate the strain rate locally per cell using the non-equilibrium part, Eq. (2.17). With the addition of τ_{eff} this equation now reads

$$\bar{S}_{ij} = -\frac{1}{2\rho c_s^2 \tau_{eff}} \bar{\Pi}_{ij}^{neq} \quad \left[\frac{1}{s}\right]. \tag{3.12}$$

The $\bar{\Pi}_{ij}^{neq}$ is calculated from Eq. (2.16). Using the definition of the intensity of the strain rate Eq. (3.8), one can derive

$$|\bar{S}| = \frac{1}{2\rho c_s^2 \tau_L^\star} |\bar{\Pi}^{neq}| \quad \left[\frac{1}{s}\right]. \tag{3.13}$$

It is apparent that to compute $|\bar{S}|$ one needs to know τ_{eff} which creates an implicit problem for the calculation of the latter. However an explicit function can be derived if one replaces τ_{eff} in the previous equation by Eq. (3.11) and solves for $|\bar{S}|$, i.e.

$$|\bar{S}|^2 + \frac{\tau c_s^2}{C_S^2 \Delta x^2} |\bar{S}| - \frac{|\bar{\Pi}^{neq}|}{2\rho C_S^2 \Delta x^2} = 0. \tag{3.14}$$

The above equation is a quadratic one for the unknown $|\overline{S}|$. Therefore, it makes sense to keep only the positive root since intensity is by construction a positive quantity, hence

$$|\overline{S}| = \frac{-\tau c_s^2 + \sqrt{\tau^2 c_s^4 + 2\rho^{-1} C_S^2 \Delta x^2 |\overline{\Pi}^{neq}|}}{2C_S^2 \Delta x^2} \quad \left[\frac{1}{s}\right]. \tag{3.15}$$

The final step is to substitute Eq. (3.15) into Eq. (3.11) and after some algebra one attains

$$\tau_{eff} = \frac{\tau}{2} + \sqrt{\frac{\tau^2}{4} + \frac{C_S^2 \Delta x^2 |\overline{\Pi}^{neq}|}{2\rho c_s^4}} \quad [s]. \tag{3.16}$$

The above equation is a local per cell calculation contrary to the first option. Dimensional analysis confirms the consistency of the above derivations. The lattice version of Eq. (3.16) that has been implemented in AMROC LBM solver reads

$$\frac{\tau_{eff}}{\Delta t} = \frac{1}{2\Delta t}\left(\tau + \sqrt{\tau^2 + \frac{18 C_S^2 |\overline{\check{\Pi}}^{neq}|}{\check{\rho}}}\right) \quad [-]. \tag{3.17}$$

The Algorithm 3.1 reports the implementation of the CSMA model in the AMROC LBM solver.

Algorithm 3.1: The function *step* with CSMA turbulence model

```
forall Processors do
    forall Blocks in the current processor do
        Function Step
            Result: Evolve one $\Delta t$ and update the values in the current block
            begin
                Execute streaming;                          /* Eq. (2.9) */
                Estimate $\tau_{eff}$;     /* Stencil Eq. (3.4) or locally Eq. (3.17) */
                if STA SRT then
                    Execute collision;          /* STA SRT version Eq. (3.3) */
                else
                    Apply regularisation;                  /* Eq. (2.19) */
                    Execute collision;          /* REG SRT version Eq. (3.3) */
                end
                if external force then
                    Add the force;                /* Last term of Eq. (2.28) */
                end
            end
        end
    end
end
```

3.3 Dynamic Smagorinsky

The implementation of the dynamic Smagorinsky model in LBM is based on the work of Premnath et al. (2009a) and follows the idea of Germano et al. (1991) with the modification of Lilly (1992). The target is to replace the constant C_S of the CSMA model with a function that takes into consideration local effects, $(C_S \rightarrow C_S(\mathbf{x}, t))$. The role of this function is to modify the length scale of the model.

To create this function, I will introduce the concept of test filtering besides the standard grid-filtering Eq. (3.1). The test filtering occurs in a coarser grid under consideration $\widehat{\Delta x}/\Delta x = 2$, that is, two times the physical spatial spacing. From now on, the notation $\hat{\overline{\phi}}$ will refer to a test filter variable.

The grid-filter Sub-Grid-Scale (SGS) stress tensor τ_{ij} can be estimated as:

$$\tau_{ij} = \overline{u_i u_j} - \overline{u}_i \overline{u}_j. \tag{3.18}$$

The first term of the Right Hand Side (RHS) represents the Reynolds stresses, which are unknown and thus, modelling is needed. Models based on the eddy viscosity assumption correlate the anisotropic part of the previous tensor with the strain rate $\overline{S}_{ij}$ and a scalar variable that is the eddy viscosity, Eq. (3.6), hence yielding

$$\tau_{ij} - \frac{\delta_{ij}}{3}\tau_{kk} = -2\nu_t \overline{S}_{ij}. \tag{3.19}$$

The notation δ_{ij} refers to the delta of Kronecker. One can similarly compute an SGS stress tensor based on the test-filter scale as

$$T_{ij} - \frac{\delta_{ij}}{3}T_{kk} = -2(C_S\widehat{\Delta x})^2|\hat{\overline{S}}|\hat{\overline{S}}_{ij} = -2\widehat{\nu}_t\hat{\overline{S}}_{ij}. \tag{3.20}$$

The strong assumption that is imposed in the above equations, and is also inherited by the DSMA model, is that the constant C_S is invariant for both scales, standard filtering and test filtering. The T_{ij} can be computed as

$$T_{ij} = \widehat{\overline{u_i u_j}} - \hat{\overline{u}}_i \hat{\overline{u}}_j, \tag{3.21}$$

where again, the first term of the RHS is unknown. However, one can combine Eq. (3.18) and Eq. (3.21) to derive

$$L_{ij} = T_{ij} - \widehat{\tau}_{ij} = \widehat{\overline{u}_i\overline{u}_j} - \hat{\overline{u}}_i \hat{\overline{u}}_j, \tag{3.22}$$

which is the Germano identity (Germano et al., 1991). In the above relation, all terms are known. By also combining Eq. (3.19) and Eq. (3.20), one has

$$L_{ij} - \frac{\delta_{ij}}{3}L_{kk} = -2C_S^2 M_{ij}, \tag{3.23}$$

where

$$M_{ij} = \widehat{\Delta x}^2 |\hat{\overline{S}}| \hat{\overline{S}}_{ij} - \Delta x^2 \widehat{|\overline{S}| \overline{S}_{ij}}. \tag{3.24}$$

The above two relations create a set of five independent equations for the one unknown C_S. The system is overdetermined, and thus a least square minimisation procedure can be applied following the modification of Lilly (1992). The error function R that has to be minimised is

$$R = (L_{ij} - \frac{\delta_{ij}}{3} L_{kk} + 2C_S^2 M_{ij})^2, \tag{3.25}$$

which leads to

$$C_S^2 = -\frac{1}{2} \frac{\langle L_{ij} M_{ij} \rangle}{\langle M_{ij} M_{ij} \rangle}. \tag{3.26}$$

In theory, $\langle \cdot \rangle$ refers to averaging in homogeneous directions and, if the problem is statistically stationary, in time, too. Nonetheless, I apply this procedure locally in each cell because identifying homogeneous directions in a general situation is either difficult or impossible. This restriction can introduce some instabilities due to large local values of the constant (Lilly, 1992). To alleviate this problem, the constant is truncated as $0 \leq C_S \leq 0.23$. Moreover, the term $\delta_{ij}(L_{kk}M_{kk})/3$ has been omitted from the numerator since M_{ij} is analogous to the strain rate, which is traceless for incompressible fluids. This statement is partially correct for the LBM, which is a weakly compressible method.

The filtering is founded on the trapezoidal filter as it is suggested by Premnath et al. (2009a) due to its natural fitting with Cartesian grids. Below is the algorithm, divided into three consecutive steps.

$$\overline{\phi}^{\star}_{i,j,k} = \frac{1}{4}(\overline{\phi}_{i+1,j,k} + 2\overline{\phi}_{i,j,k} + \overline{\phi}_{i-1,j,k}), \tag{3.27a}$$

$$\overline{\phi}^{\star\star}_{i,j,k} = \frac{1}{4}(\overline{\phi}^{\star}_{i,j+1,k} + 2\overline{\phi}^{\star}_{i,j,k} + \overline{\phi}^{\star}_{i,j-1,k}), \tag{3.27b}$$

$$\hat{\overline{\phi}}_{i,j,k} = \frac{1}{4}(\overline{\phi}^{\star\star}_{i,j,k+1} + 2\overline{\phi}^{\star\star}_{i,j,k} + \overline{\phi}^{\star\star}_{i,j,k-1}). \tag{3.27c}$$

To avoid using two layers of external ghost cells, one-sided test filtering has been coded for the variables in the first layer of ghost cells. This formula, in the case of the left phase of the domain, reads

$$\overline{\phi}^{\star}_{i,j,k} = \frac{1}{2}(\overline{\phi}_{i+1,j,k} + \overline{\phi}_{i,j,k}). \tag{3.28}$$

The above equation replaces any one of the three steps in Eq. (3.27). In that way, one ensures that reasonable values will be calculated in the first layer of ghost cells that will lead to accurate values of C_S to the first layer of inner cells. The test-filter is applied for the variables $\overline{u}_i$, $\overline{u}_i\overline{u}_j$ and $|\overline{S}|$, $\overline{S}_{ij}$ and their multiplication. Finally, having calculated the constant C_S, Eq. (3.17) is called to compute the effective relaxation time during the collision phase.

Algorithm 3.2 reports the implementation of DSMA in the AMROC LBM solver. The constant C_S is estimated at the beginning of the function *step*, that is, before the streaming. This is important because the streaming will alter the values in ghost cells that have been calculated during boundary conditions, see Algorithm 2.1. However, in this way, the calculation of C_S lags one Δt from its use, that is, during the collision step. Additionally, by employing the data before the streaming, the algorithm deals with the post-collision state of the distribution functions. This contrasts with the use of the pre-collision values during the estimation of τ_{eff}. Moreover, the strain rate $\overline{S}_{ij}$ and its intensity $|\overline{S}|$, appearing in the second term of Eq. (3.24), are estimated locally per cell invoking Eq. (3.12). The τ_{eff}, which emerges in this equation, is calculated based on the data of the previous time step to avoid an implicit problem. On the other hand, the test-filtering strain rate $\hat{\overline{S}}_{ij}$ and its intensity $\hat{|\overline{S}|}$ are estimated based on finite differences, Eq. (3.10), of the test filter velocity $\hat{\overline{u}}_i$.

Algorithm 3.2: The function *step* with DSMA turbulence model

```
forall Processors do
    forall Blocks in the current processor do
        Function Step
            Result: Evolve one $\Delta t$ and update the values in the current block
            begin
                Calculate the constant $C_S$ per cell;          /* Eq. (3.26) */
                Execute streaming;                               /* Eq. (2.9) */
                Estimate $\tau_{eff}$;                           /* Eq. (3.17) */
                if STA SRT then
                    Execute collision;               /* STA SRT version Eq. (3.3) */
                else
                    Apply regularisation;                        /* Eq. (2.19) */
                    Execute collision;               /* REG SRT version Eq. (3.3) */
                end
                if external force then
                    Add the force;                    /* Last term of Eq. (2.28) */
                end
            end
        end
    end
end
```

3.4 The WALE model

The idea of the DSMA model was to replace the constant C_S in Eq. (3.6) by a local function. On the other hand, the idea in the WALE (Wall-Adapting Local Eddy-viscosity) model is to replace $|\overline{S}|$ with a more advanced operator that can handle the damping of the eddy viscosity in the vicinity of the wall (Nicoud and Ducros, 1999) effectively. The new operator is a function both of the strain rate S_{ij} and the rotation rate Ω_{ij} as can be seen below:

$$OP_{\text{WALE}} = \frac{(\mathcal{J}_{ij}\mathcal{J}_{ij})^{\frac{3}{2}}}{(\overline{S}_{ij}\overline{S}_{ij})^{\frac{5}{2}} + (\mathcal{J}_{ij}\mathcal{J}_{ij})^{\frac{5}{4}}}, \tag{3.29}$$

where $\mathcal{J}_{ij}$ reads

$$\mathcal{J}_{ij} = \overline{S}_{ik}\overline{S}_{kj} + \overline{\Omega}_{ik}\overline{\Omega}_{kj} - \frac{1}{3}\delta_{ij}(\overline{S}_{mn}\overline{S}_{mn} - \overline{\Omega}_{mn}\overline{\Omega}_{mn}). \tag{3.30}$$

Therefore the eddy viscosity can be computed as

$$\nu_t = (C_w\Delta)^2 OP_{WALE}, \tag{3.31}$$

where C_w is the constant of the model and is equal to 0.5. To compute the two rates, central finite differences are used for the derivatives, Eq. (3.10). Similar to the CSMA model based on stencil, the τ_{eff} is estimated through Eq. (3.4) using Eq. (3.5) for the effective viscosity. Furthermore, the algorithm is based on the second set of distribution functions unaffected by the streaming. Consequently, an error of order Δt is also introduced for the case of WALE. Compared to the DSMA model, the WALE model does not need truncation corrections besides the situation where all the velocities are zero, i.e. during the initialisation of the flow field, in which scenario nu_t is set to zero. Other researchers have investigated the accuracy of this model with the LBM; see for instance (Weickert et al., 2010; Liu et al., 2012).

Algorithm 3.3: The function *step* with WALE turbulence model

```
forall Processors do
    forall Blocks in the current processor do
        Function Step
            Result: Evolve one $\Delta t$ and update the values in the current block
            begin
                Execute streaming;                              /* Eq. (2.9) */
                Estimate $\tau_{eff}$;                          /* Eq. (3.4) */
                if STA SRT then
                    Execute collision;          /* STA SRT version Eq. (3.3) */
                else
                    Apply regularisation;                      /* Eq. (2.19) */
                    Execute collision;          /* REG SRT version Eq. (3.3) */
                end
                if external force then
                    Add the force;                /* Last term of Eq. (2.28) */
                end
            end
        end
    end
end
```

3.5 On the performance and implementation of turbulence models in AMROC

Besides the estimation of the constant $C_S(\mathbf{x}, t)$ in the case of DSMA, the only difference in performance among the LES models is due to the calculation of τ_{eff}. With a local estimation of τ_{eff}, CSMA is the only model that keeps the locality of the algorithm during the collision step and thus is the fastest option. CSMA with stencil and WALE need to communicate with neighbour cells, leading to cache misses and thus reducing the performance. The size of blocks can improve this condition. In one case, WALE was 17% more expensive than local CSMA (Gkoudesnes and Deiterding, 2021). On the other hand, for the estimation of C_S in DSMA, the algorithm forces the solver to parse the domain multiple times to estimate the test filtered quantities. This condition makes DSMA the most expensive model by almost doubling the wall time needed to run a simulation compared to the local CSMA. However, the extra cost tends to be accompanied by more accuracy, as discussed during the results in the next chapters.

All the above models affect the simulation by altering the relaxation time on a local basis. Nonetheless, in the literature, one can find other ways to implement turbulence models in LBM. There is the possibility to add the effect of the turbulence modelling through either an external force (Malaspinas and Sagaut, 2012)or a mixed way (Xia et al., 2015). These two methods enable a model that is not based on the Boussinesq assumption incorporated into LBM (Xia et al., 2015). There is also the approximate deconvolution model with its specific implementation in LBM (Malaspinas and Sagaut, 2011).

Chapter 4

On the Boundary Conditions in AMROC LBM

The aim of this chapter is threefold. Initially, I will mention briefly two general types of boundary conditions that LBM employs to model boundaries. Then I will report how the AMROC LBM solver treats and applies boundaries. Additionally, I will present two domain boundaries that have been implemented during this project. The purpose of these open boundaries is to model inlet and outlet conditions.

Secondly, I will show the proposed algorithm for imposing macrovariables in ghost cells. Some boundary conditions do not impose the quantities density ρ and velocity field $\mathbf{u}$ in ghost cells directly. The absence of accurate macrovariables in those cells can affect the accuracy of the LES models based on a stencil.

The final section will discuss the newly implemented wall treatment and its integration into the solver. Contrary to the other boundary conditions, the wall treatment is the only boundary in the AMROC LBM applied to the first layer of fluid cells avoiding the ghost cells. I will also comment on how this algorithm interacts with the LES models; see Chap. 3.

4.1 The two Types of Boundary Conditions for LBM

The purpose of a boundary condition in the LBM framework is to estimate the missing distribution functions f_α that have to be propagated either from outside of the domain or from a solid surface to a fluid cell. This situation is illustrated in Fig. 4.1 for the $D2Q9$ lattice model as an example. The vertical red line represents the boundary. Any cell that lies in the grey area, with coordinates $\mathbf{x_{BC}}$, is a boundary cell, and thus one needs to take action to calculate the coloured distribution functions. The left image shows the state of the distribution functions exactly after applying the boundary conditions.

That is, the coloured distributions have just been estimated and are known. The right image describes the situation after the streaming step. The coloured distributions have been propagated, and unknown values occupy their previous positions. The application of the next boundary phase will recalculate them before the next streaming step.

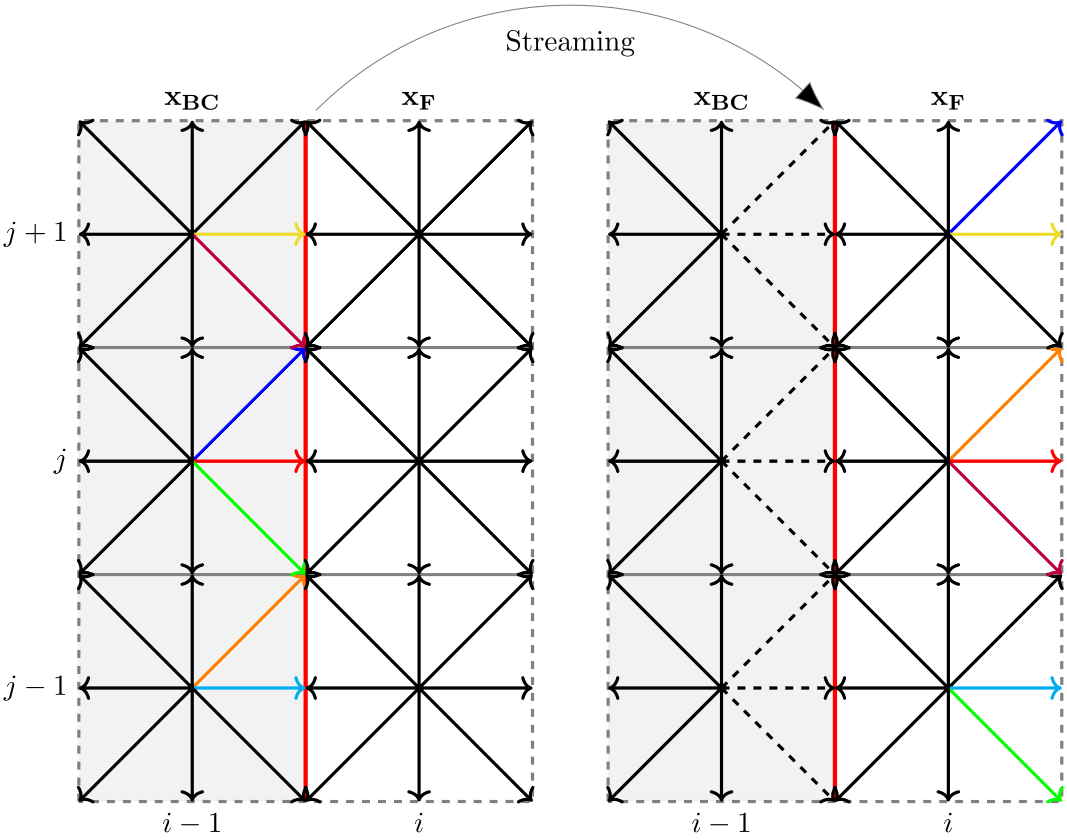

Streaming
$\mathbf{x}_{\mathbf{BC}}$
$\mathbf{x}_{\mathbf{F}}$
$j+1$
j
$j-1$
$i-1$
i
$\mathbf{x}_{\mathbf{BC}}$
$\mathbf{x}_{\mathbf{F}}$
$i-1$
i

Figure 4.1: Sketch of domain boundaries for *D2Q9* model in the AMROC LBM solver. The vertical red line represents the domain boundary. The left and right pictures displace the states of the distribution functions before and after the streaming, respectively.

In a solver based on NS equations, there would be three macrovariables, namely density and two velocities, that should be imposed or calculated at the boundary cell. However, in the case of LBM, there is more degree of freedoms (distribution functions), thus creating an underdetermined system of equations (moments) without a unique solution. One can distinguish the boundary conditions in LBM into two categories based on whether they are applied directly to the distribution functions or tried to impose a set of macrovariables (Malaspinas and Sagaut, 2014). The first case, I call it *bounce-back-type*, estimates the unknown distribution functions per lattice direction and is based on reflections of the distribution functions on the boundary line. For instance, the calculation of the blue direction in Fig. 4.1 is independent of the calculation of the red one. Some examples of this kind are the standard bounce-back boundary conditions for modelling slip and no-slip wall (Krüger et al., 2016; Bouzidi et al., 2001). A critical characteristic of this type is that it does not affect the lattice directions outside of the domain or to other solid cells. Therefore, the resulting macrovariables in these cells are unknown after applying the boundary condition. In LES models based on a finite difference stencil, this situation would lead to inaccurate estimation of eddy viscosity close to boundaries. To avoid this issue, the author has devised a new algorithm that enables the users to employ the rest of the distribution functions to impose macrovariables by manipulating the moments Eq. (2.12). I will present it in a later section of this chapter.

The second category of boundary conditions in LBM focuses on the *reconstruction* of the distribution functions $f_\alpha(\mathbf{x_{BC}})$ in boundary cells. The formula that describes this approach reads

$$f_\alpha(\mathbf{x_{BC}}) = f_\alpha^{eq}(\mathbf{x_{BC}}) + f_\alpha^{neq}(\mathbf{x_{BC}}). \tag{4.1}$$

The basic idea is to estimate the equilibrium functions $f_\alpha^{eq}(\mathbf{x_{BC}})$ based on macrovariables that are either imposed or calculated depending on the type of boundary condition. The next step is to estimate the non-equilibrium part $f_\alpha^{neq}(\mathbf{x_{BC}})$ usually through an extrapolation scheme from neighbouring fluid cells (Zhao-Li et al., 2002; Guo et al., 2002b). The second step is not mandatory for the boundary to be effective. However, without applying it, the boundary is restricted to be only first order accurate (Krüger et al., 2016), in contrast to the second order of accuracy of the numerical scheme, Sec. 2.2.1. Moreover, by construction, the algorithms of this kind of boundaries impose macrovariables automatically in boundary cells. Therefore, no additional action is needed to improve the accuracy of LES.

4.2 Boundary Conditions in AMROC-LBM

As mentioned before in Chap. 1, the application of boundary conditions in AMROC is based on ghost cells. AMROC provides two distinct functions, see Algorithm 2.1, one for the application of domain boundaries and the other for the embedded cells representing a solid body. In the case of the domain boundaries, conditions, such as slip and no-slip wall, are based on a *bounce-back-type* strategy. The implementation is similar to the *full-way bounce-back* method reported in (Krüger et al., 2016, chap 5). The concept of this boundary implementation in the AMROC-LBM solver is depicted in Fig. 4.2. The sketch describes the time evolution of a specific lattice velocity that crosses the domain boundary. Initially, at $t-\Delta t$ the streaming step will propagate the distribution function to the ghost cell located at $\mathbf{x_{BC}}$. As mentioned before in Sec. 2.2.1, the collision step is not applied in the ghost cells, thus leaving its value unaffected at $t - \Delta t$. At the beginning of the next step t, the boundary condition will be applied at the opposite direction, which points to the fluid cell at $\mathbf{x_F}$. Finally, the streaming will proper the new distribution function back in the fluid domain. One significant difference between my implementation and the one reporting in (Krüger et al., 2016, chap 5) is that they propagate the post-collision distribution function while I stream the pre-collision value. At the next step, they combine collision and boundary condition to alter the distribution function at the ghost cell. Finally, they stream back to the fluid domain the post-collision value while streaming the pre-collision one, which will collide at the fluid cell.

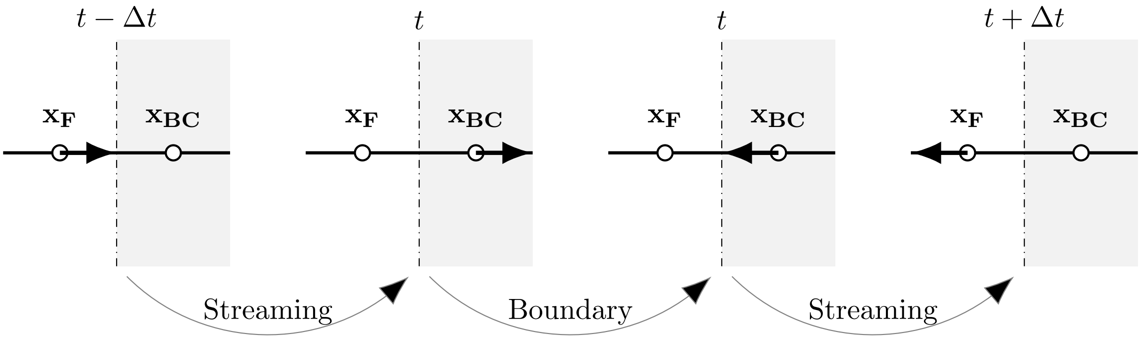


Figure 4.2: Application of the *full-way bounce-back* method in AMROC LBM for the modelling of slip and no-slip wall at the domain boundaries.

On the other hand, open boundaries such as inlet and outlet are applied following the *reconstruction* approach, Eq. (4.1). In the case of an inlet boundary, the vector of velocities is given, and the density $\rho(\mathbf{x_{BC}})$ at the boundary is unknown. Therefore, the density of the first neighbour normal fluid cell $\rho(\mathbf{x_F})$ is extrapolated as

$$\rho(\mathbf{x_{BC}}) = \rho(\mathbf{x_F}). \tag{4.2}$$

Examining Fig. 4.1 for example, the fluid cell that lies in the normal direction of the ghost cell with coordinates $(i-1, j)$ is the cell at (i, j). In the case of the outlet, the density $\rho(\mathbf{x_{BC}})$ is imposed, and the velocity field needs to be extrapolated from the normal neighbour fluid cell. Following the same methodology, the estimation of the velocity field in the ghost cell is

$$\mathbf{u}(\mathbf{x_{BC}}) = \mathbf{u}(\mathbf{x_F}). \tag{4.3}$$

The macrovariables in the fluid cell $[\rho(\mathbf{x_F}), \mathbf{u}(\mathbf{x_F})]$ can be calculated by invoking Eq. (2.12). Having obtained $[\rho(\mathbf{x_{BC}}), \mathbf{u}(\mathbf{x_{BC}})]$, one can proceed to estimate the equilibrium part $f_\alpha^{eq}(\mathbf{x_{BC}})$ of the distribution function by applying Eq. (2.11).

For the extrapolation of the non-equilibrium part $f_\alpha^{neq}(\mathbf{x_{BC}})$, two models have been implemented during this project. The first strategy has been proposed in (Zhao-Li et al., 2002) and reads

$$f_\alpha^{neq}(\mathbf{x_{BC}}) = f_\alpha(\mathbf{x_F}) - f_\alpha^{eq}(\mathbf{x_F}) = f_\alpha^{neq}(\mathbf{x_F}). \tag{4.4}$$

The only difference with the reference is that I extrapolate the post-collision values and not the pre-collision. The above methodology is of second order accuracy aligning with the order of the method, Sec. 2.2.1. Though this strategy recovers the density and the momentum correctly, it does not correct the stress tensor Π_{ij}^{neq}. One way to fix this is to apply the regularisation procedure, see Sec. 2.2.3, as proposed in (Latt et al., 2008). Since they applied the boundary on the first fluid cell, they used the method proposed in (Zou and He, 1997) to estimate the missing non-equilibrium distribution functions locally in this cell. In my case, I extract information from the neighbour fluid cell where the non-equilibrium part is known. Consequently, one can again employ the extrapolation scheme of (Zhao-Li et al., 2002) and the final formula reads

$$f_\alpha^{neq}(\mathbf{x_{BC}}) = \frac{w_\alpha}{2c_s^4} Q_{\alpha ij} \Pi_{ij}^{neq} \quad \text{where,} \quad \Pi_{ij}^{neq} = \sum_\alpha e_{\alpha i} e_{\alpha j} f_\alpha^{neq}(\mathbf{x_F}). \tag{4.5}$$

This boundary yields no benefits when it is combined with the REG collision model seeing that the post-collision value of $f_\alpha^{neq}(\mathbf{x_F})$ is already regularised. Finally, having calculated the non-equilibrium part either through option one, Eq. (4.4) or two Eq. (4.5), one can invoke Eq. (4.1) to reconstruct the distribution function at the ghost cell.

In the case of the embedded boundary conditions, the AMROC-LBM solver employs the *image-based ghost method*, (see Krüger et al., 2016, chap 11). Again, one can categorise these boundary conditions into *bounce-back-type* and *reconstruction*. The most accurate *bounce-back-type* condition implemented into the solver is the Bouzidi model, (Bouzidi et al., 2001). The author has not implemented this boundary, and it is not used in any test case in this thesis, and thus it will not be presented here. However, the interested reader can learn about its implementation in (Gkoudesnes and Deiterding, 2021).

The *reconstruction* boundaries follow the same idea as in the case of the domain boundaries presented above. Nonetheless, an essential difference is how one estimates the extrapolated values, both macrovariables and non-equilibrium part, through the image point. The approach followed by AMROC-LBM solver is similar to the strategy reported in (Tiwari and Vanka, 2012). To assist the explanation of the algorithm, Fig. 4.3 displays two possible scenarios of embedded boundaries and how the solver deals with the estimation of the values at the image point X. The algorithm can be decomposed into steps:

1. The first step consists of distinguishing fluid and solid cells. The AMROC framework provides the level-set mechanism based on which every cell keeps track of its distance to its closest solid surface. For instance, in Fig. 4.3, for the cell K, this distance is the length KM. Moreover, this distance has a sign which depends on whether the cell exists in the enclosed region from the solid surface or not. Any cell centre in the enclosed region has a negative sign, and the algorithm treats it as a solid cell. The level set can also be used to estimate the normal to the wall (see Deiterding, 2011, chap 3). Having both the distance and the normal, the image

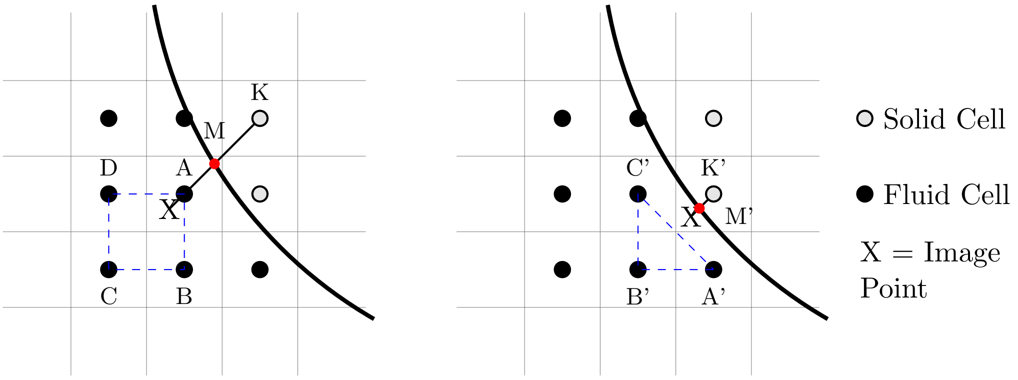


Figure 4.3: Two scenarios for the estimation of the image point X. The region described by the blue dashed line represents the bilinear interpolation.

point coordinates can be located at the same distance in the opposite direction of the solid surface.

2. In the general situation, the image point will not coincide with the centre of a fluid cell. Therefore, an interpolation scheme is utilised to estimate the values at that point. AMROC employs the bilinear spatial interpolation, and its formula can be found in (see Deiterding, 2011, chap 3). One crucial observation is that, in the left image, AMROC will use the information in all four corners of the blue rectangle for the interpolation. However, in the right image, where one corner is a solid cell, the interpolation will see only the three fluid corners and thus, the scheme will transform into an extrapolation approach. This approximation is less accurate than the approach proposed in (Tiwari and Vanka, 2012), where the solid corner is replaced by the point M′ at the solid surface.

3. When AMROC executes the image point algorithm, it has no access to the LBM equations, such as moments Eq. (2.12). Consequently, the interpolated values at the image point are the vector of state, that is the distribution functions $f_\alpha(\mathbf{x}, t)$. This situation is in contrast to the strategy in (Tiwari and Vanka, 2012), where the interpolated values are the macrovariables, density and velocity field, estimated in the four corners. In this case, the macrovariables at the image point are estimated based on the interpolated values of the distribution functions during the execution of the embedded boundaries, see Algorithm 2.1. Having the interpolated macrovariables, one can also estimate the interpolated equilibrium part of the distribution function by using Eq. (2.11) at the image point.

4. The macrovariables at the solid cell is a function of the interpolated values at the image point. The specific boundary condition dictates the formula of the equations. Having the macrovariables, one can calculate the equilibrium part of the distribution function invoking Eq. (2.11), at the ghost cell. Afterwards, the non-equilibrium part at the image point can be estimated by subtracting the interpolated equilibrium part from the interpolated distribution functions. Then,

the non-equilibrium part of the distribution function at the ghost cell can be calculated using Eq. (4.4) or Eq. (4.5). Finally, the reconstruction of the ghost cell's distribution function is due to Eq. (4.1).

The above strategy has also been applied to implement the wall function, as I will discuss in Sec. 4.2.

4.3 An Algorithm for Imposing Macrovariables in Ghost Cells with Bounce-back Boundaries

As mentioned before in Chap. 3, various LES models, such as WALE and DSMA, estimate the eddy viscosity based on a stencil. Moreover, it has been also commented that during the execution of the numerical scheme update step, the solver cannot distinguish between fluid cell and solid cell. Consequently, the phase of the boundary conditions discussed in the previous section is of utmost importance to secure correct macrovariables in each ghost cell. In the case of the *reconstruction* boundaries, this condition is automatically guaranteed. However, this is not the case for the *bounce-back-type* boundaries. To resolve this issue, I propose here a new algorithm that can be called after the application of a *bounce-back-type* boundary and will ensure that invoking the moments, Eq. (2.12), it will result in correct macrovariables in ghost cells. The algorithm is decomposed into steps as follows:

1. The first step of applying this algorithm is the identification of the *free* lattice directions, that, is, the directions that are not affected by the boundary condition and point either outside of the domain or to other ghost cells. I index this group of lattice directions with l. I will refer to the directions that participate in the boundary condition as *non-free* ones, and their notation will be k. Assuming the groups of non-free and free directions have K and L elements, I have $A = K + L$, with $A = 19$ for the $D3Q19$ and $A = 9$ for $D2Q9$ lattice models. For instance, in the case of the ghost cell at $(i-1, j)$ in Fig. 4.1, the three coloured directions are the *non-free* group and the rest six directions the *free* group. For more clarity, this situation is also depicted in the left image of Fig. 4.4 with the three black *non-free* and six red *free* directions.

2. Based on the *non-free* group, one can estimate a partial density and velocity field as

$$(\delta\rho)_K = \sum_k f_k, \quad \forall\ k \in K,\ \textit{non-free directions}, \tag{4.6a}$$

$$(\delta\rho\mathbf{u})_K = \sum_k \mathbf{e}_k f_k, \quad \forall\ k \in K,\ \textit{non-free directions}. \tag{4.6b}$$

Assuming the macrovariables density ρ_0 and velocity field $\mathbf{u}_0$, the idea of the algorithm is to use the *free* directions to impose them in the ghost cell. However, the only available equations are the moments, Eq. (2.12), three for 2D and four for 3D cases. The scenario of Fig. 4.4 shows that the unknown variables, six $free$ distributions, outnumber the equations. This situation leads to the underdetermined problem

$$(\delta\rho)_L = \rho_0 - (\delta\rho)_K = \sum_l f_l, \quad \forall\ l \in L,\ \textit{free directions}, \tag{4.7a}$$

$$(\delta\rho\mathbf{u})_L = \rho_0\mathbf{u}_0 - (\delta\rho\mathbf{u})_K = \sum_l \mathbf{e}_l f_l, \quad \forall\ l \in L,\ \textit{free directions}. \tag{4.7b}$$

An efficient way to resolve this issue is using the equilibrium function Eq. (2.11), which the imposed macroscopic quantities can estimate, ρ_0 and $\mathbf{u}_0$. In this way, I can divide the group of the $free$ directions L into two smaller groups. The first group, called the *equilibrium* E, consists of the distribution functions in which the equilibrium values will be imposed. The second group is called the *moments* M and consists of the distribution functions that will be calculated in such a way that invoking Eq. (2.12) will results in the expected macrovariables, ρ_0 and $\mathbf{u}_0$. For this to be possible, the group M must have as many elements as the number of the equations, that is, $M = 3$ for 2D and $M = 4$ for 3D. In other words, the restriction for this algorithm to be applicable is $L \geq M$, that is, the number of the $free$ directions must be equal or larger than the equations of moments. Moreover, it is apparent that $E = L - M$. Another important comment is that the central distribution, f_0, must always belong to the *moment* group. Given the indexing of the lattice directions in AMROC-LBM, see Sec. 2.2.1, an easy way to determine the additional elements of the group M is to pick the last, based on indexing, available $M - 1$ elements of the group L. The right image of Fig. 4.4 illustrates this strategy. The last available element is f_8 and has been flagged as the group's second element M together with f_0. On the other hand, f_7 is not available, and thus the final pick is f_6.

3. Having identify the elements of M, the next move is to impose the equilibrium values in the elements of the group E, that is $f_1 \to f_1^{eq}$, $f_2 \to f_2^{eq}$ and $f_4 \to f_4^{eq}$. One can also calculate partial density and velocity field as

$$(\delta\rho)_E = \sum_e f_e, \quad \forall\ e \in E,\ \textit{equilibrium directions}, \tag{4.8a}$$

$$(\delta\rho\mathbf{u})_E = \sum_k \mathbf{e}_e f_e, \quad \forall\ e \in E,\ \textit{equilibrium directions}. \tag{4.8b}$$

4. At this point, I end up with a system of equations that is now solvable. Starting from the momentum equation, it reads

$$(\delta\rho\mathbf{u})_M = \rho_0\mathbf{u}_0 - (\delta\rho\mathbf{u})_K - (\delta\rho\mathbf{u})_E = \sum_m \mathbf{e}_m f_m \quad \forall\ m \in M \setminus \{f_0\}. \tag{4.9}$$

This linear system is of the type $\mathbf{b} = \mathbf{Af}$, with the vector $\mathbf{b}$ the values of the partial moments $(\delta\rho\mathbf{u})_M$ and the equivalent matrix A the elements of the lattice velocities $\mathbf{e}_m$. Solving this system, with the exemption of the central value f_0, one will obtain all the unknown distributions, in this case, f_6 and f_8. To calculate the central value, initially, one has to estimate the partial density

$$(\delta\rho)_{M-1} = \sum_m f_m, \quad \forall\ m \in M \setminus \{f_0\}. \tag{4.10}$$

The final step is to apply the equation of the zeroth moment as

$$f_0 = \rho_0 - (\delta\rho)_K - (\delta\rho)_E - (\delta\rho)_{M-1}. \tag{4.11}$$

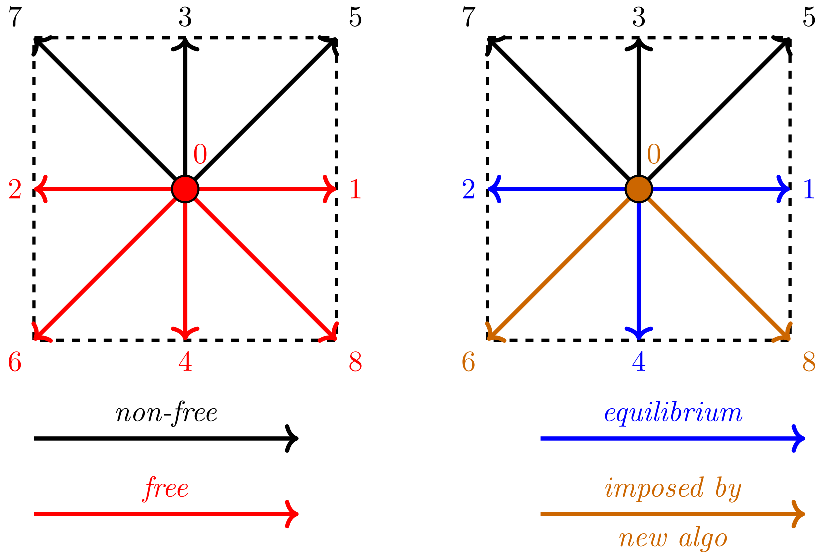


Figure 4.4: Illustration of the new algorithm for imposing macrovariables for the $D2Q9$ lattice model. The left image shows the ghost cell state before applying the new algorithm and the right image after it.

In the case of the $D3Q19$ model, the situation is more complicated. Figure 4.5 presents the state of the distribution functions after applying the new algorithm. I assume that this ghost cell has a domain boundary on the top phase, hence the five $non - free$ directions. The strategy of determining the *momentum* group by parsing the $free$ directions in descending order is crucial in this case. For instance, if one picks instead of f_{14} the distribution function f_4, then the equivalent matrix A in the linear system of equations in Eq. (4.9) will be singular, since one column will be zeroes. Therefore, the descending order restricts any lattice direction pointing to a phase, Eq. (2.8), to participate in the *momentum* group and thus avoiding the singular matrix. In this way, this algorithm can be applied without extra consideration for embedded boundaries, in which case the number of $free$ directions may depend on the location of the ghost cell. On the other hand, only directions point to edges in the group M leads to a matrix A that needs factorisation. Consequently, to resolve the arising system of 3 × 3 equations, an LU decomposition strategy is employed. In the case of $D3Q15$ or $D3Q27$ lattice models, by picking at least one available diagonal direction, one can guarantee that the matrix A is not singular.

It is important to comment that though the proposed algorithm succeeds in imposing macrovariables, it cannot impose a correct non-equilibrium part of the distribution function. It is to be expected that the way the boundary condition itself has imposed the distribution functions and the new algorithm, the conservativeness of the method, Eq. (2.18), does not hold any more. The application of the collision step in such a cell will yield an error. However, in the AMROC-LBM solver, we do not rely on collision in ghost cells, so this inaccuracy does not affect the implementation. As for the LES models, WALE expects to see correct macrovariables in adjacent cells and ignores the values of the non-equilibrium part.

Consequently, the proposed algorithm is sufficient for the WALE model. On the other hand, the estimation of the constant of DSMA needs accurate values of the non-equilibrium part besides macrovariables in neighbour cells. Therefore, the proposed strategy improves the DSMA model's performance but cannot entirely eliminate the error close to boundaries.

Up to this point, I have presented the way to impose macrovariables to a ghost cell in the case of a bounce-back boundary. The next question is how to estimate these macrovariables, density ρ_0 and velocity components $[u_0, v_0, w_0]$. The answer to this question depends on the boundary condition. Currently, the implementation categorises the boundaries into three types, namely slip, no-slip and sliding wall. The last option is a boundary with a moving frame parallel to the solid surface.

For all three options, the density ρ_0 at the boundary is extrapolated based on Eq. (4.2). In an embedded boundary, one extrapolates the density estimated by the interpolated distribution functions at the image point. Since all three types describe wall boundaries, the normal velocity at the solid surface should be zero. To describe the step, I will assume that the normal velocity aligns with v_0. To achieve the zero normal velocity at the wall, I impose $v_0 = -v_F$, where v_F is the velocity normal to the wall at the normal neighbour fluid cell.

The other two components, u_0 and w_0 are extrapolated invoking Eq. (4.3) in the case of the slip wall. On the other hand, for the no-slip and sliding wall boundaries, I apply the wall function algorithm, see Sec. 4.4. The goal is to estimate the wall shear stress τ_w

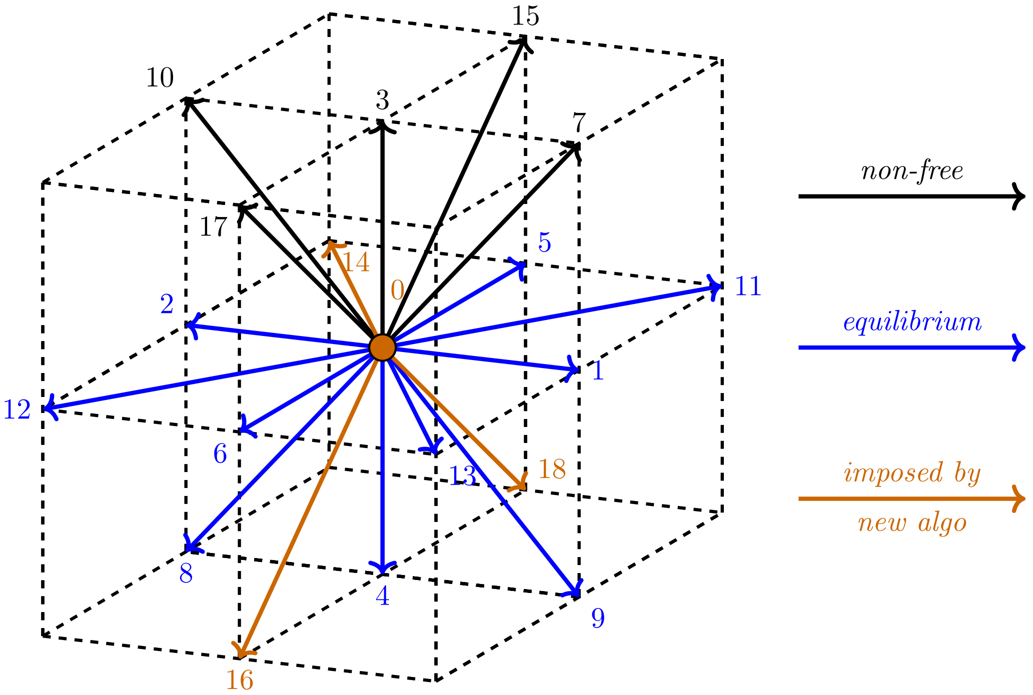


Figure 4.5: The lattice directions for the $D3Q19$ model.

using data from the normal neighbour fluid cell. Having estimated τ_w, one can calculate the velocity derivative normal to the wall at the solid surface as

$$\frac{\partial U}{\partial n} = \frac{\tau_w}{\rho_0 \nu}. \tag{4.12}$$

The notation U refers to the streamwise velocity, which is the vector sum of the components in x and z directions. From this definition, one can estimate the streamwise velocity U_F in the fluid cell. Approximating the above derivative with central finite differences between ghost cell and fluid cell, the only unknown is the streamwise velocity in the ghost cell U_{BC}. Having calculated U_{BC}, one can decompose it into its Cartesian components u_0 and w_0. In the case of the sliding wall, I initially subtract the components of the wall's velocity from the velocities in the first fluid cell. In this way, I transform the absolute frame to a relative frame of reference in which the wall appears static. Then I can invoke the wall function procedure. In the end, I add back the components of the wall velocity to the values at the ghost cell to return to the absolute frame of reference.

For curved boundaries, one can rotate the Cartesian system of coordinates to align one axis with the normal wall direction. All the above procedures can then be executed in the new reference system using the values at the image point. In the end, it is straightforward to estimate the Cartesian components. Finally, Table 4.1 summarises the methodology applied for each boundary condition.

	slip wall	no-slip wall	sliding wall
ρ_0	extrapolation	extrapolation	extrapolation
u_0	extrapolation	wall function	wall function*
v_0	opposite sign	opposite sign	opposite sign
w_0	extrapolation	wall function	wall function*

Table 4.1: The estimation of the imposed macrovariables for *bounce-back-type* boundaries. The table describes the case for the y-axis parallel to the normal direction. The notation * indicates the application in a relative frame.

4.4 Wall Treatment

The wall treatment that has been implemented in the AMROC-LBM solver follows the approach proposed in (Malaspinas and Sagaut, 2014; Haussmann et al., 2019). The difference between this boundary and the boundaries presented in the previous sections is that the wall treatment is applied to the first layer of fluid cells and not in ghost cells. The situation is depicted in Fig. 4.6 for both domain and embedded boundaries. In the case of the domain boundary, since I apply the wall treatment in position $\mathbf{x_{F1}}$, I now need access to the second layer of fluid cells at position $\mathbf{x_{F2}}$. Similarly, for the curved boundaries, the image point is now further inside the fluid domain. Its position has been estimated as two times the distance to the wall $MF1$ in the normal to the wall direction.

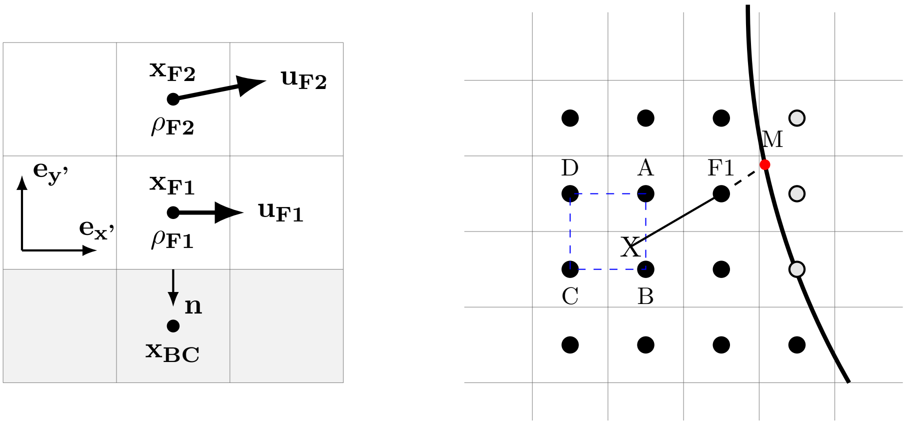


Figure 4.6: The application of the wall treatment in AMROC-LBM for domain boundaries (left) and embedded boundaries (right).

There is a vital discrepancy between the implementation in AMROC-LBM solver and the references (Malaspinas and Sagaut, 2014; Haussmann et al., 2019). The difference lies in the order of the execution of the various functions during a time step. In references, it is proposed to start the step with collision, followed by streaming and then applying the wall treatment. However, this order deviates from the current implementation in AMROC, see Algorithm 2.1. The result is that the wall treatment in the AMROC-LBM solver deals with the post-collision state of the distribution functions $f_\alpha(\mathbf{x}, t)$.

Nonetheless, (Malaspinas and Sagaut, 2014; Haussmann et al., 2019) deal with the post-streaming state $\tilde{f}_\alpha(\mathbf{x}, t)$. Discussing the algorithm, it will be apparent that this alteration will affect the outcome. As mentioned before in Sec. 2.2.1, the collision, Eq. (2.10), should not affect the values of the macrovariables. However, it does change the values of the non-equilibrium part $f_\alpha^{neq}(\mathbf{x}, t)$.

To simplify the explanation of the method, I will focus on a domain boundary; see the left image of Fig. 4.6. The algorithm of the wall treatment is summarised in Algorithm 4.1. It consists of several steps that read:

1. The first step is the estimation of the macrovariables $[\rho_{F2}, \mathbf{u_{F2}}]$ in the second fluid cell at position $\mathbf{x_{F2}}$.

2. The idea of this wall treatment is based on the concept of the **canonical boundary layer** (Malaspinas and Sagaut, 2014). This boundary layer theory is intrinsic 2D. Consequently, a coordinate system $(\mathbf{e_{x'}}, \mathbf{e_{y'}})$ where one axis is aligned with the streamwise direction, $\mathbf{e_{x'}}$, and the other axis with the normal $\mathbf{n}$ to the wall should be constructed and employed. The vector basis parallel to the normal can be found as $\mathbf{e_{y'}} = -\mathbf{n}$, while one can proceed to estimate the other vector basis as

$$\mathbf{e}_{x'} = \frac{\mathbf{u_{F2}} - (\mathbf{u_{F2}} \cdot \mathbf{n})\mathbf{n}}{||\mathbf{u_{F2}} - (\mathbf{u_{F2}} \cdot \mathbf{n})\mathbf{n}||}. \tag{4.13}$$

3. Having estimated the new coordinate system, the streamwise velocity u'_{F2} can be calculated as

$$u'_{F2} = \mathbf{e}_{x'} \cdot \mathbf{u_{F2}}. \tag{4.14}$$

4. The next step is the estimation of the wall shear stress τ_w. To do so, the wall function of Musker has been employed and reads

$$\begin{aligned} u^+ &= 5.424 \arctan(0.119760479041916168 y^+ - 0.488023952095808383) \\ &+ 0.434 \log\left(\frac{(y^+ + 10.6)^{9.6}}{(y^{+\ 2} - 8.15 y^+ + 86)^2}\right) - 3.50727901936264842. \end{aligned} \tag{4.15}$$

In the above equation, the velocity scaled in wall units u^+ is calculated as

$$u^+ = \frac{u'}{u_\tau}, \quad \text{where} \quad u_\tau = \sqrt{\frac{\tau_w}{\rho}} \tag{4.16}$$

is the friction velocity. Moreover, the relation describing y^+ is

$$y^+ = \frac{u_\tau\, y}{\nu}. \tag{4.17}$$

The solution of Eq. (4.15) for the wall shear stress extraction is an implicit problem. A Newton-Raphson solver is called to estimate the final solution numerically.

5. Having estimated the wall shear stress τ_w, based on data provided by the second fluid cell at $\mathbf{x_{F2}}$, one can utilise again Eq. (4.15) to estimate the streamwise velocity u'_{F1} in the first fluid cell at $\mathbf{x_{F1}}$ position. This time, the problem is explicit, and there is no need for a numerical method to solve it.

6. Afterwards, one can calculate $\mathbf{u_{F1}}$ in Cartesian coordinates as

   $$\mathbf{u_{F1}} = u'_{F1}\ \mathbf{e}_{x'}. \tag{4.18}$$

   Based on this strategy, there is no normal to the wall component of the first fluid cell's velocity.

7. The next step is the estimation of the density, ρ_{F1}, and non-equilibrium part of the distribution function. In the case of ρ_{F1}, I follow the approach of Haussmann et al. (2019) which is identical to Eq. (4.2), that is an extrapolation from the neighbour fluid cell at $\mathbf{x_{F2}}$.

   In the non-equilibrium part, I also tried to apply the extrapolation described in Eq. (4.4), but I noticed that this had detrimental effects on the outcome. As mentioned at the beginning of this section, I extrapolate the post-collision values in contrast to the post-streaming ones (Haussmann et al., 2019). After some experiments, I have concluded that the most robust and accurate solution is achieved by omitting the non-equilibrium part. However, by avoiding the non-equilibrium part, this boundary condition is only first order accurate.

8. The final step is the reconstruction of the distribution function in the first fluid cell Eq. (4.1). Having calculated the macrovariables $[\rho_{F1}, \mathbf{u_{F1}}]$, one can call Eq. (2.11) to estimate the equilibrium values.

After applying the boundary condition, the solver will stream the now known distribution functions from the first layer of fluid cells to the interior domain. The collision will also occur in these cells, but the outcome will be erased in the subsequent application of the boundary condition. One side effect of this procedure is that the output occurs before applying the boundary conditions, see Algorithm 2.1, which may result in erroneous data in the first layer of fluid cells.

In the LES models, WALE is unaffected because its eddy viscosity is estimated based on the state of macrovariables that occurs before the streaming. As mentioned before, this is possible because the AMROC framework provides two vectors of states of which the only one is affected by streaming. On the other hand, DSMA application in the second layer of interior cells will see zero values of non-equilibrium part in some adjacent cells resulting in some inaccuracy during the constant estimation. Though, it should be mentioned that a wall treatment that will be able to apply accurate values of non-equilibrium part, in combination with the imposition of accurate macrovaribales in the first layer of ghost cells, can lead to a successful application, without inaccuracies, of DSMA in the vicinity of solid boundaries.

Algorithm 4.1: Implementation of wall treatment approach in AMROC-LBM

```
Function Wall Treatment
    Data: The macroscopic variables $\rho_{F2}$ and $\mathbf{u_{F2}}$, the normal $\mathbf{n}$, the viscosity $\nu$ and
          the distances at the wall for the first and second cell $(y_{F1}, y_{F2})$
    Result: The unknown distributions at the boundary located in $\mathbf{x_{F1}}$
    begin
        Compute the macrovariables $[\rho_{F2}, \mathbf{u_{F2}}]$ from the previous step;
        Calculate the streamwise basis vector $\mathbf{e}_{x'}(\mathbf{n}, \mathbf{u_{F2}})$, Eq. (4.13);
        Compute the streamwise velocity $u'_{F2}$, Eq. (4.14);
        Estimate $\tau_w$ by solving an implicit problem, Eq. (4.15);
        Compute the streamwise velocity at the first cell $u'_{F1}$, through Eq. (4.15);
        Compute the velocity vector at the boundary $\mathbf{u_{F1}}$, Eq. (4.18);
        Extrapolate the density at the wall using Eq. (4.2);
        Reconstruct the unknown distributions based on equilibrium values;
    end
end
```

Chapter 5

Wall free cases

Currently, there is no deterministic approach to describe turbulence. Therefore, there are no general analytical solutions that can be used for the verification of turbulence models. As a consequence, the proof of correctness of every implementation of turbulence models usually takes place against experimental data or code-to-code comparisons. However, there are a few cases for which an analytic solution seems to fit the results under some strong assumptions. One such case is Homogeneous Isotropic Turbulence (HIT), where the shape of the energy spectra, particularly in the inertial subrange and dissipation range, is supposed to follow a universal behaviour.

In order to simulate HIT, there are mainly two ways. The first option is the addition of a fictitious external body force that will distort the large eddies of the flowfield. This concept leads to the Forced HIT (FHIT) case. After some initial time, this extra energy will be in equilibrium with the viscous dissipation. As long as the flow is provided with this extra kinetic energy, the flowfield will be in a statistically stationary state, and thus statistical in time analysis can be applied. Consequently, FHIT enables us to have sufficient time for statistical analysis of the flowfield. Simultaneously, another advantage is that FHIT has the potential to simulate higher Re numbers than DHIT.

The second option is Decaying HIT (DHIT), in which an initial solution, representing some characteristics of turbulent flow, is allowed to decay. After some initial time, this solution will be turned into a real turbulent flow. After this transient time, one can start analysing the flow statistically. However, compared to the FHIT, in this case, there is a limited window for analysis. In literature, there is a variety of such attempts with LBM (Peng et al., 2010; Yu et al., 2005a). The drawback of this procedure is the flow statistics are time-dependent because of the decay. Thus, averaging in time is not suitable. Therefore, one can study the dynamic response of a turbulence model. Furthermore, compared to FHIT, DHIT tends to simulate smaller Re numbers for the same resolution. In this chapter, I will present results for both FHIT and DHIT cases.

Finally, the Taylor Green Vortex (TGV) has been chosen as the third benchmark case for this chapter. TGV enables us to verify the turbulence models during the transition from the laminar to the turbulent field. Moreover, this test case can be used to examine the numerical dissipation of the employed numerical scheme or turbulent model by comparing the dissipation rate with the dissipation rate based on enstrophy.

Besides the verification of the LES models, I will also examine the coupling of collision models, particularly SRT and REG, with the turbulent models and how they affect each other. Another investigation will examine the difference of estimating the strain rate locally through the non-equilibrium part or using a finite difference stencil. Some of the results have been also published in (Gkoudesnes and Deiterding, 2019a,b, 2021).

In the next section, I will present the background and essential tools to analyse isotropic turbulence flow. Simultaneously, I will show the implemented routine for statistical and spectral analysis of the flowfield on-the-fly. For the validation of DNS spectra in the case of FHIT, the model spectrum of Pope (2000) will be employed. Furthermore, I will also discuss the force and its implementation that I have used in FHIT.

5.1 Spectral and Statistical Analysis of Turbulent Flows

According to Richardson (Pope, 2000), turbulent flows obey the *energy cascade* concept based on which the flow is decomposed into eddies of variable dimensions (wavenumbers). Following this idea, the largest eddies are responsible for the turbulent energy production by *stealing* energy from the main flow. On the other hand, in the smallest eddies, this energy is dissipated to heat. A transfer mechanism, obeying the universal power-law spectrum based on Kolmogorov's hypotheses, is responsible for supplying the highest wavenumbers (smaller eddies) with energy from the lowest wavenumbers (large eddies). The above theory can be illustrated more accurately for *homogeneous isotropic* turbulence, because isotropy is the primary assumption, and homogeneity is essential for accurate statistical analysis in space and time.

The shape of the energy spectrum, particularly in the inertial subrange, can be determined and derived from the $-\frac{5}{3}$ Kolmogorov law, based on the second similarity hypothesis. Initially, this outcome enables us to validate the current implementation of the Lattice Boltzmann method in AMROC as a DNS solver. Simultaneously, the main contribution of an LES model is the modelling of the dissipation in the smallest scales or largest wavenumbers in the spectrum. This range is called the dissipation one. In this way, one can evaluate the LES modelling effect in the dissipation range by comparing LES with DNS. Therefore, in this chapter, the test cases offer an unmatched chance to examine and verify the behaviour of the implemented LES models without the interaction of boundary conditions or AMR interfaces.

It is evident from the above discussion that the spectral representation of the velocity field is vital for post-processing. Consequently, a function has been implemented and employed to obtain on-the-fly spectra and other turbulent statistics. MPI communication has been applied to parallelised this piece code. Algorithm 5.1 shows the implemented method *Compute_Statistics* that is responsible for the statistical and spectral analysis of the flowfields presented in this chapter. The library FFTW3 (Frigo, 1999) is used, as a black box, to deal with the Forward Fourier Transform of the components of the physical velocity and pressure fluctuations. This transformation is from real data (physical velocity) to complex (spectral velocity). Invoking Fourier theory, it is known that there is symmetry around the zero wavenumber for the spectral components under these circumstances, and thus only half of them need to be computed and saved. Therefore, I used the specialised plan of the FFTW3 for such transformations instead of the regular forward transform, almost halving the memory requirements and speeding up the process. In this way, I stored all the wavenumbers in x-direction and y-direction but only the positive part of the z-direction.

However, FFTW3 stores the wavenumbers in the standard “in-order” output ordering. In other words, the positive wavenumbers are stored in the first half of the array and the negative ones in the second part in backward order. This representation is not ideal for the looping in the x-direction and y-direction. To handle this obstacle, I shifted the second half of the indexes by the number of the cells in these directions (N). Moreover, FFTW3 computes an unnormalised discrete Fourier transformation. Therefore, the Fourier coefficients are normalised by dividing them by the total number of cells (N^3).

Having both the spectral velocities and pressure, normalised and ordered, one can continue with the computation of their spectra. In turbulence analysis the *velocity-spectrum tensor* $\Phi_{ij}(\boldsymbol{\kappa})$, where $\boldsymbol{\kappa}$ is the discrete wavenumber vector, contains valuable information (see p. 221 of (Pope, 2000)). The subscripts i and j describe the velocity components in the physical space. Moreover, one can extrude the information of its direction in physical space as $\mathbf{e} = \boldsymbol{\kappa}/|\boldsymbol{\kappa}|$ and its lengthscale as $\ell = 2\pi/|\boldsymbol{\kappa}|$ where $|\boldsymbol{\kappa}| = \sqrt{\kappa_x^2 + \kappa_y^2 + \kappa_z^2} = \kappa$ is the magnitude of the wavenumber vector. However, because it is a second order tensor it is difficult to interpret this information. Consequently, a less informative although simpler expression is used, namely the *energy-spectrum function* $E(\kappa)$. The relation between these two variables can be seen in Eq. (5.1) (or Eq. 6.188 in (Pope, 2000))

$$E(\kappa) = \oint \frac{1}{2}\Phi_{ii}(\boldsymbol{\kappa})\ \partial\mathcal{S}(\kappa). \tag{5.1}$$

Therefore, $E(\kappa)$ contains no directional information as a result of using only the trace Φ_{ii} and integrating over the sphere $\mathcal{S}(\kappa)$ with a radius the magnitude κ. Assuming a sphere, the above surface integral can be computed analytically as

$$\oint \partial\mathcal{S}(\kappa) = 4\pi\kappa^2, \tag{5.2}$$

Algorithm 5.1: Statistical and spectral analysis of the flowfield

```
Function Compute_Statistics
    Data: The components of the velocity (u, v, w) and the pressure fluctuations ($\delta p$)
    Result: 3D Spectra and other statistical variables
    begin /* Set-up phase */
        Compute the plans for FFTW3; /* Real to Complex Fourier Transform */
    end
    begin /* Estimate the contribution of each processor */
        Compute $< |u|^2 >$ and then $u_{rms}$ and $k$;
        Execute the four Fourier transforms, $[u(\mathbf{x}) \mapsto \hat{u}(\boldsymbol{\kappa})]$;
        for $\kappa_x = -\frac{N}{2} + 1$ to $\frac{N}{2} - 1$ do
            for $\kappa_y = -\frac{N}{2} + 1$ to $\frac{N}{2} - 1$ do
                for $\kappa_z = 0$ to $\frac{N}{2}$ do
                    Compute $|\boldsymbol{\kappa}| = \sqrt{\kappa_x^2 + \kappa_y^2 + \kappa_z^2}$ and $\kappa_{bin} = round(\boldsymbol{\kappa})$ ;
                    Check if the wavenumber $\kappa_{bin}$ is in the range $(0, \kappa_{max}]$;
                    Compute the kinetic energy per bin $E(\kappa_{bin}) = \sum \frac{1}{2}\hat{R}_{ii}(\boldsymbol{\kappa})$;
                    Compute the pressure spectrum per bin $\Delta\hat{P}(\kappa_{bin}) = \delta\hat{p}(\boldsymbol{\kappa})\delta\hat{p}^{\star}(\boldsymbol{\kappa})$;
                    Normalize the spectral velocities and pressure fluctuations by $N^3$;
                end
            end
        end
    end
    begin /* Gather the results in the master processor */
        for $\kappa_{bin} = \kappa_0$ to $\kappa_{max}$ do
            Compute the turbulent kinetic energy of Fourier modes $\left(\sum E(\kappa_{bin})\right)$;
            Compute the dissipation rate $\varepsilon = 2\nu \sum \kappa_{bin}^2 \hat{E}(\kappa_{bin})$;
        end
        forall $x_i = 0$ to $N - 1$ do
            Compute the magnitude of the velocity;
            Compute the standard deviation $u_{rms} = \sqrt{\frac{|u|^2}{3N^3}}$; /* assuming isotropy */
            Compute the turbulent kinetic energy in the physical space;
        end
    end
    Compute Kolmogorov scales $(\eta, u_\eta, \tau_\eta)$;
    Compute Taylor microscale and Reynolds number;
end
```

yielding for $E(\kappa)$

$$E(\kappa) = 2\pi\kappa^2\ \Phi_{ii}(\boldsymbol{\kappa}). \tag{5.3}$$

Finally, to estimate Eq. (5.3), one should invoke the relation of $\Phi_{ij}(\boldsymbol{\kappa})$ with the covariance of the velocity. Assuming discretisation of the domain, this relation reads

$$\Phi_{ij}(\boldsymbol{\kappa}) \equiv \hat{R}_{ij}(\boldsymbol{\kappa}), \tag{5.4}$$

and the trace $\hat{R}_{ii}$ can be estimated as

$$\hat{R}_{ii}(\boldsymbol{\kappa}) \equiv \widehat{u}_i^*(\boldsymbol{\kappa})\widehat{u}_i(\boldsymbol{\kappa}) = \widehat{u}_i(-\boldsymbol{\kappa})\widehat{u}_i(\boldsymbol{\kappa}). \tag{5.5}$$

The notation $\widehat{u}_i^*$ is the conjugate of $\widehat{u}_i$. Given that the velocity field is a real function in the physical space, its Fourier coefficients experience some symmetry, as the above equation illustrates. Finally, Eq. (5.3) transforms to

$$E(\kappa) = 2\pi\kappa^2\hat{R}_{ii} = 2\pi\kappa^2\widehat{u}_i^*(\boldsymbol{\kappa})\widehat{u}_i(\boldsymbol{\kappa}). \tag{5.6}$$

One way to smooth the shape of the spectra is by grouping the wavevectors in bins around specific wavenumbers κ_{bin}, which, in general, are integer multiples of the lowest one ($\kappa_0 = 2\pi/\mathcal{L}$, where $\mathcal{L}$ is the length of the domain). Implemented this way, a check takes place to discard the zero wavenumber and the insignificant ones, whose magnitudes are higher than the Nyquist criterion ($|\boldsymbol{\kappa}| > \kappa_{max} = N/2$). Therefore, Eq. (5.6) now reads

$$E(\kappa_{bin}) = 2\pi\kappa_{bin}^2\langle\hat{R}_{ii}\rangle_{bin} = 2\pi\kappa_{bin}^2\langle\widehat{u}_i^*(\boldsymbol{\kappa})\widehat{u}_i(\boldsymbol{\kappa})\rangle_{bin}, \tag{5.7}$$

where $\langle\cdot\rangle_{bin}$ denotes the ensemble average over all the contributing wavevectors $|\boldsymbol{\kappa}|$ for each κ_{bin}. Finally, a possible implementation reads

$$E(\kappa_{bin}) = \frac{2\pi\kappa_{bin}^2}{N_{con}}\ \widehat{u}_i^*(\boldsymbol{\kappa})\widehat{u}_i(\boldsymbol{\kappa}) = b_f * \widehat{u}_i^*(\boldsymbol{\kappa})\widehat{u}_i(\boldsymbol{\kappa}), \quad \forall\ |\boldsymbol{\kappa}| \in \kappa_{bin}, \tag{5.8}$$

where N_{con} is the number of contributing wavevectors for the current bin and b_f is the binning factor. Having computed the energy spectrum function, the turbulent kinetic energy k can be computed as

$$k_{sp} = \sum_{\kappa_{bin}} E(\kappa_{bin}). \tag{5.9}$$

Another way to compute the spectra is directly through the kinetic energy of the Fourier modes. For this reason, an alternative way to relate $\Phi_{ij}(\boldsymbol{\kappa})$ with $E(\kappa_{bin})$ is (see also Eq. 6.189 in (Pope, 2000))

$$E(\kappa) = \iiint_{-\infty}^{\infty} \frac{1}{2}\Phi_{ii}(\boldsymbol{\kappa})\delta(|\boldsymbol{\kappa}| - \kappa)\ \partial\boldsymbol{\kappa}. \tag{5.10}$$

Now, assuming discretisation of the domain and invoking Eq. (5.4), the above equation can be rewritten as

$$E(\kappa) = \frac{1}{2}\hat{R}_{ii}(\boldsymbol{\kappa}) = \hat{E}(\boldsymbol{\kappa}), \tag{5.11}$$

where $\hat{E}(\boldsymbol{\kappa})$ is the *kinetic energy of the Fourier mode* (Pope, 2000). In this case, the binning procedure yields

$$E(\kappa_{bin}) = \sum_{\kappa_{bin}} \frac{1}{2} \hat{R}_{ii}(\boldsymbol{\kappa}), \quad \forall\ |\boldsymbol{\kappa}| \in \kappa_{bin}. \tag{5.12}$$

Finally, the turbulent kinetic energy can be computed similarly as in Eq. (5.9) as

$$k_{fr} = \sum_{\kappa_{bin}} E(\kappa_{bin}). \tag{5.13}$$

The previous estimations of the turbulent kinetic energy are based on spectral space. One can also calculate the turbulent kinetic energy in physical space. Assuming homogeneous isotropic turbulence, the two-point correlation function reads

$$R_{ij}(\mathbf{r} = 0) = \langle u_i(\mathbf{x}) u_j(\mathbf{x}) \rangle_{\mathcal{L}} = u_{rms}^2 \delta_{ij}, \tag{5.14}$$

where $\langle \cdot \rangle_{\mathcal{L}}$ denotes the ensemble-averaged over all the domain (cells), and u_{rms} is the standard deviation of the velocity field. It is evident that under these assumptions, the deviatoric part of the Reynold stresses is zero. From the previous equation, one can determine the u_{rms} as

$$u_{rms} = \sqrt{\frac{\langle u_i u_i \rangle_{\mathcal{L}}}{3}}, \tag{5.15}$$

and the kinetic energy as

$$k_{ph} = \frac{\langle u_i u_i \rangle_{\mathcal{L}}}{2} = \frac{3u_{rms}^2}{2}. \tag{5.16}$$

To verify the implementation of the Algorithm 5.1, one needs to show that the estimation of the value of the turbulent kinetic energy is the same both in physical and spectral space. One can find both ways to compute the turbulent kinetic energy in the spectral space in the literature. In order for the Eqs. (5.8) and (5.12) to be equal, the below equation should be valid:

$$N_{con} = 4\pi\kappa_{bin}^2 \tag{5.17}$$

Obviously, this equation will be true in the limit of a continuous spectral space, in which scenario the number of contributors will be equal with the surface of the sphere with radius the wavenumber $|\boldsymbol{\kappa}|$. Consequently, it is expected that Eq. (5.8) is only an approximation to Eq. (5.12). To further asses these two options, Fig. 5.1 shows the evolution in time of the kinetic energy computed by Eqs. (5.9), (5.13) and (5.16) for the test case of FHIT ($N = 128, A = 10^{-4}, \nu = 3 \cdot 10^{-5}\ m^2\ s^{-1}$).

Moreover, Fig. 5.2 shows the difference between the computation of the turbulent kinetic energy based on physical space and the other two variations. This plot confirms that the computation of the turbulent kinetic energy based on the kinetic energy of the Fourier

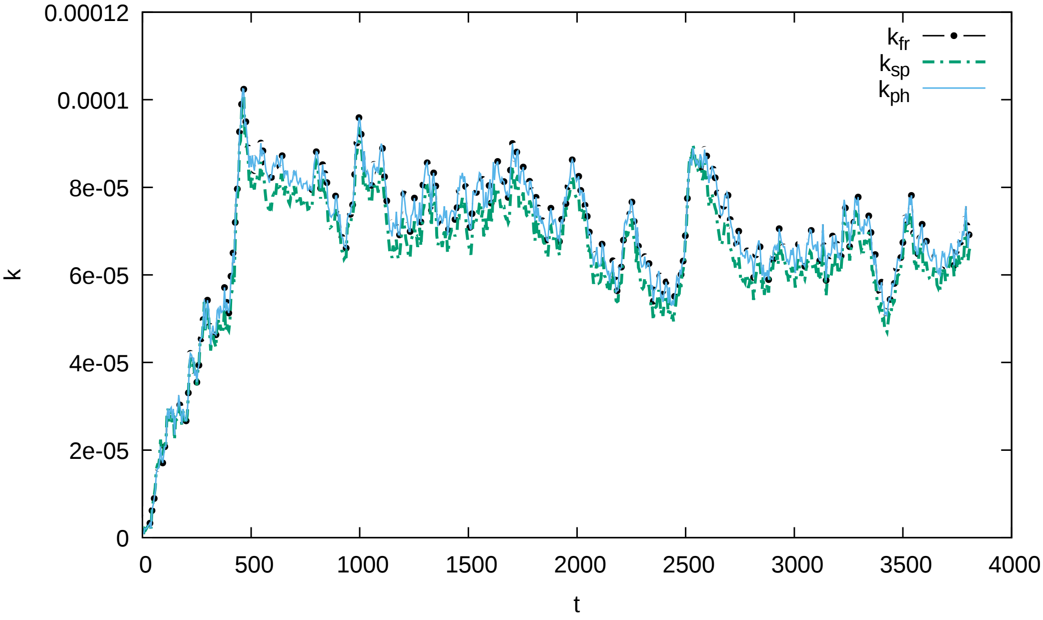


Figure 5.1: Evolution of kinetic energy in physical time.

modes Eq. (5.13) is indeed more accurate. Therefore, Eq. (5.13) has been implemented and used in Algorithm 5.1.

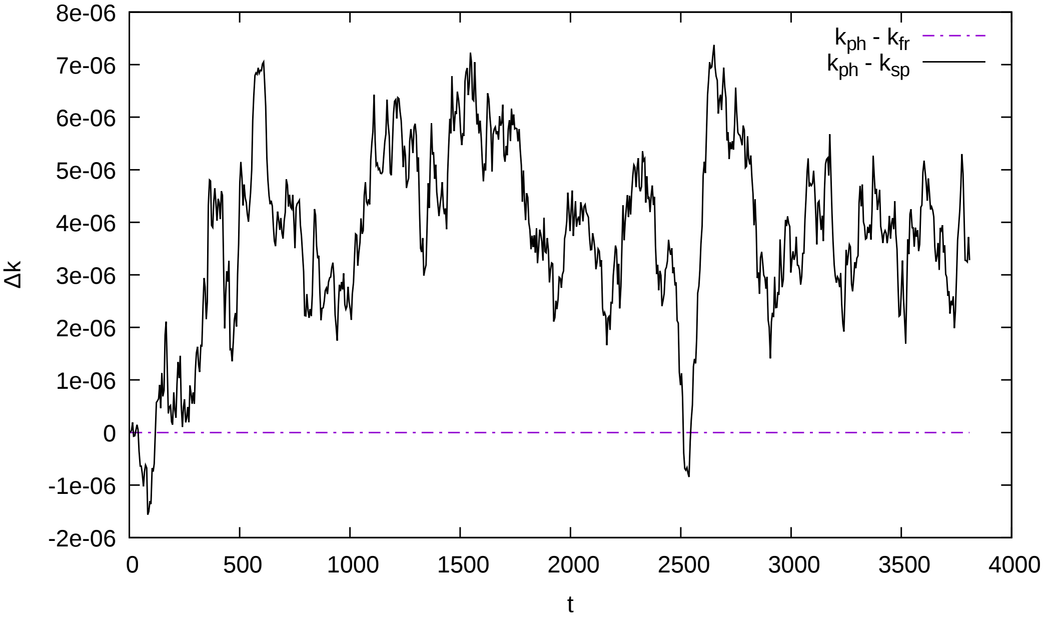


Figure 5.2: Differences among the various estimations of the turbulent kinetic energy.

Similar to the velocity, the pressure spectrum can be computed as

$$\Delta P(\kappa_{bin}) = \sum_{\kappa_{bin}} \delta\hat{p}(\boldsymbol{\kappa})\delta\hat{p}^{\star}(\boldsymbol{\kappa}), \quad \forall\ |\boldsymbol{\kappa}| \in \kappa_{bin}, \tag{5.18}$$

where $\delta\hat{p}$ are the Fourier modes of the pressure fluctuations calculated as

$$\delta p = (\rho - \rho_0)c_s^2. \tag{5.19}$$

The notation ρ_0 is the base density. The dissipation ε of the flow can be estimated as

$$\varepsilon = 2\nu \sum_{\kappa_{bin}} \kappa_{bin}^2 E(\kappa_{bin}), \tag{5.20}$$

where ν is the kinematic viscosity. Finally, a variety of turbulence statistics can be computed based on the dissipation. The Kolmogorov scales can be defined as

$$\eta \equiv \sqrt[4]{\frac{\nu^3}{\varepsilon}}, \tag{5.21a}$$

$$u_\eta \equiv \sqrt[4]{\varepsilon\nu}, \tag{5.21b}$$

$$\tau_\eta \equiv \sqrt{\frac{\nu}{\varepsilon}}. \tag{5.21c}$$

Last but not least, the Taylor length scale is calculated as

$$\lambda = \sqrt{15}\tau_\eta u_{rms}. \tag{5.22}$$

Having computed λ, the Taylor-scale Reynolds number, Re_λ, can be defined as

$$Re_\lambda = \frac{u_{rms}\lambda}{\nu}. \tag{5.23}$$

5.1.1 A model spectrum and Kolmogorov spectra

Many researchers have combined Kolmogorov theory and experimental data to create model spectra that seem to fit well with the latter. In this project and in the case of FHIT, I will adopt the one of Pope (2000). It can be described as (see also Eq. 6.246 in (Pope, 2000)):

$$E(\kappa) = C\varepsilon^{2/3}\kappa^{-5/3} f_L(\kappa L) f_\eta(\kappa\eta). \tag{5.24}$$

The non-dimensional functions f_L and f_η shape the spectrum in the energy-containing range and the dissipation range, respectively, while they do not affect it in the inertial subrange. The f_L can be proposed as

$$f_L(\kappa L) = \left(\frac{\kappa L}{\sqrt{(\kappa L)^2 + c_L}}\right)^{\frac{5}{3}+p_0}, \tag{5.25}$$

and the f_η as

$$f_\eta(\kappa\eta) = \exp\left\{ -\beta\left(\sqrt[4]{(\kappa\eta)^4 + c_\eta^4} - c_\eta \right)\right\}. \tag{5.26}$$

The coefficients of the above model are, $C = 1.5$, the Kolmogorov constant, $c_L \approx 6.78$, $c_\eta \approx 0.4$ and $\beta = 5.2$. Finally, p_0 specifies the slope of the spectra in the low wavenumbers, and usually, it has the value of either 2 or 4. A sketch of the energy spectrum based on the current model can be seen in Fig. 5.3. L is a length describing the large eddies and is defined as

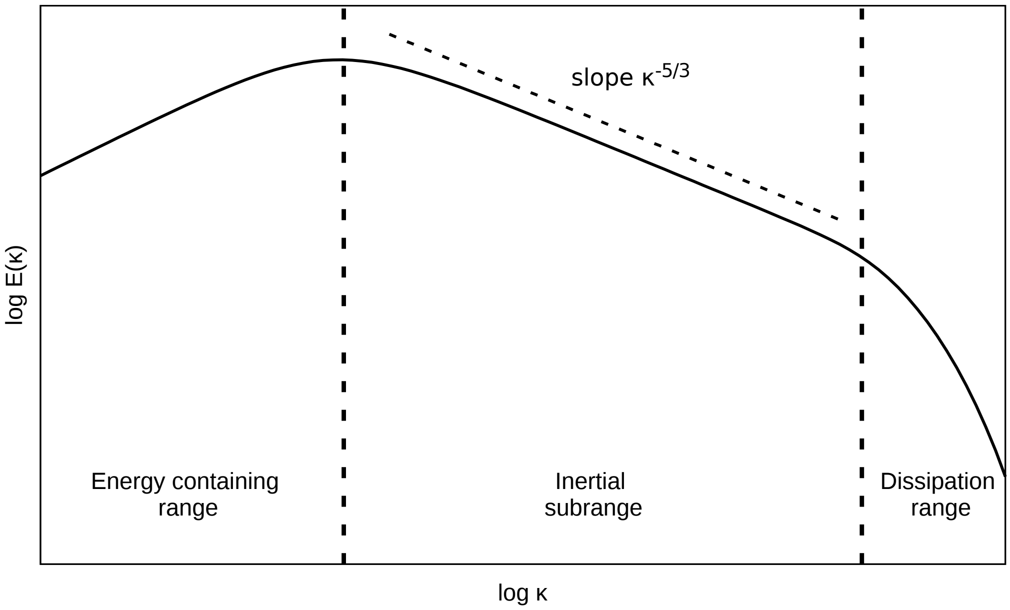


Figure 5.3: Sketch of the energy spectrum based on the model one Eq. (5.24).

$$L = \frac{k^{3/2}}{\varepsilon}. \tag{5.27}$$

Finally, another important length scale for the energy-containing region is the *longitudinal integral length scale* L_{11} computed as (under the assumption of isotropy)

$$L_{11} = \frac{3\pi}{4}\frac{\int \frac{E(\kappa)}{\kappa}\partial\kappa}{k}. \tag{5.28}$$

The model spectrum has been designed to reflect a solution to the Navier-Stokes equations. In the following section, it will be shown that the addition of an external force will alter the shapes of the spectra at the lower wavenumbers, that is, in the energy-containing range.

It is essential to provide the different scaling of the spectra, namely Kolmogorov spectra, that enable us to compare spectra of different Re_λ. From the first similarity hypothesis of Kolmogorov (Pope, 2000), one can assume that the inertial subrange and the dissipation range form a universal equilibrium range where velocity statistics are dependent only on κ, ε and ν.

In this way, the dissipation range can be described by non-dimensionalising $E(\kappa)$ with ε and ν (through η)

$$\phi(\kappa\eta) = \frac{E(\kappa)}{\sqrt[4]{(\varepsilon\nu^5)}} = \frac{E(\kappa)}{u_\eta^2\eta}, \tag{5.29}$$

where $\phi(\kappa\eta)$ is the *Kolmogorov spectrum function.* On the other hand, the second similarity hypothesis states that the inertial subrange spectra should be independent of ν, thus only a function of ε and a length-scale. This implies that the spectra tend to the constant C (Kolmogorov constant) in the inertial subrange as they depart from the dissipation range. A suitable scaling from this range is based on ε and κ, and it defines the *compensated Kolmogorov spectrum function* $\Psi(\kappa\eta)$ as

$$\Psi(\kappa\eta) = E(\kappa)\varepsilon^{-2/3}\kappa^{5/3}. \tag{5.30}$$

Contrary to the universal equilibrium range, the energy-containing range has no universal behaviour and is characterised by a particular flow. Another difficulty in the current report is that the forcing affected primarily the large scales making this range artificial compared to the NS and the model spectrum. The suitable non-dimensionalisation is based on the integral length scale L_{11} and turbulent kinetic energy k.

5.2 Forcing Homogeneous Isotropic Turbulence

The numerical domain for all simulations in this chapter is a cube of length $\mathcal{L} = 2\pi$. Due to the need for spectral analysis, all boundaries are periodic, aligning with the Fourier restriction. The lowest resolvable wavenumber is $\kappa_0 = 2\pi/\mathcal{L} = 1$, while the largest one is dependent on the resolution hence $\kappa_{max} = N\kappa_0/2$, where N is the number of cells in one direction. In this way, the domain consists of N^3 cells.

According to Pope (2000), there is a limit of the minimum size of the cube of eight L_{11}, which can be expressed as

$$\kappa_0 L_{11} = \frac{\pi}{4} \approx 0.8. \tag{5.31}$$

The above outcome is because the peak of the spectrum appears to be at $\kappa L_{11} \approx 1.3$ and there is only 5% of the total energy that occupies the range below the value of 0.8. On the other hand, there is a resolution limit based on the spatial step Δx. In the case of a pseudo-spectral method a value of $\kappa_{max}\eta \geq 1.5$ or $\eta/\Delta x \geq 1/2.1$ is assumed a good criterion. However, in literature (Peng et al., 2010), it is mentioned that the previous limit cannot be applied for LBM, which is only a second order method. They proposed more restrictive ones as $\kappa_{max}\eta \geq \pi$. Based on my results and using the model spectrum as a reference, I have estimated that a value of

$$\frac{\eta}{\Delta x} \geq 1.6 \qquad \Longrightarrow \kappa_{max}\eta \geq 5, \tag{5.32}$$

seems to be more accurate.

Furthermore, it is also crucial to check if the simulation time is adequate for reasonable statistical analysis and particularly when one averages the data. The ratio of the averaging time t_{aver} with the **turbulence turn-over time** is a helpful indication

$$t_{aver}\frac{u_{rms}}{L_{11}} \geq 10. \tag{5.33}$$

5.2.1 The Force Scheme

In Sec. 2.2.6, I have shown how to apply a body force $\mathbf{F}$ in the LBM framework. Here, I will define the formula of $\mathbf{F}$ that has been used in the case of the FHIT. In literature, a variety of forcing methods has been documented with different advantages and disadvantages. For instance, the spectral force due to Cate et al. (2006) allows the user to specify a priori some turbulence characteristic of the final flowfield. However, being spectral means that the force has to be computed in the Fourier space every time step and then transformed back to the physical space. Consequently, this requirement increases the complexity of the code and adds extra computational time and memory.

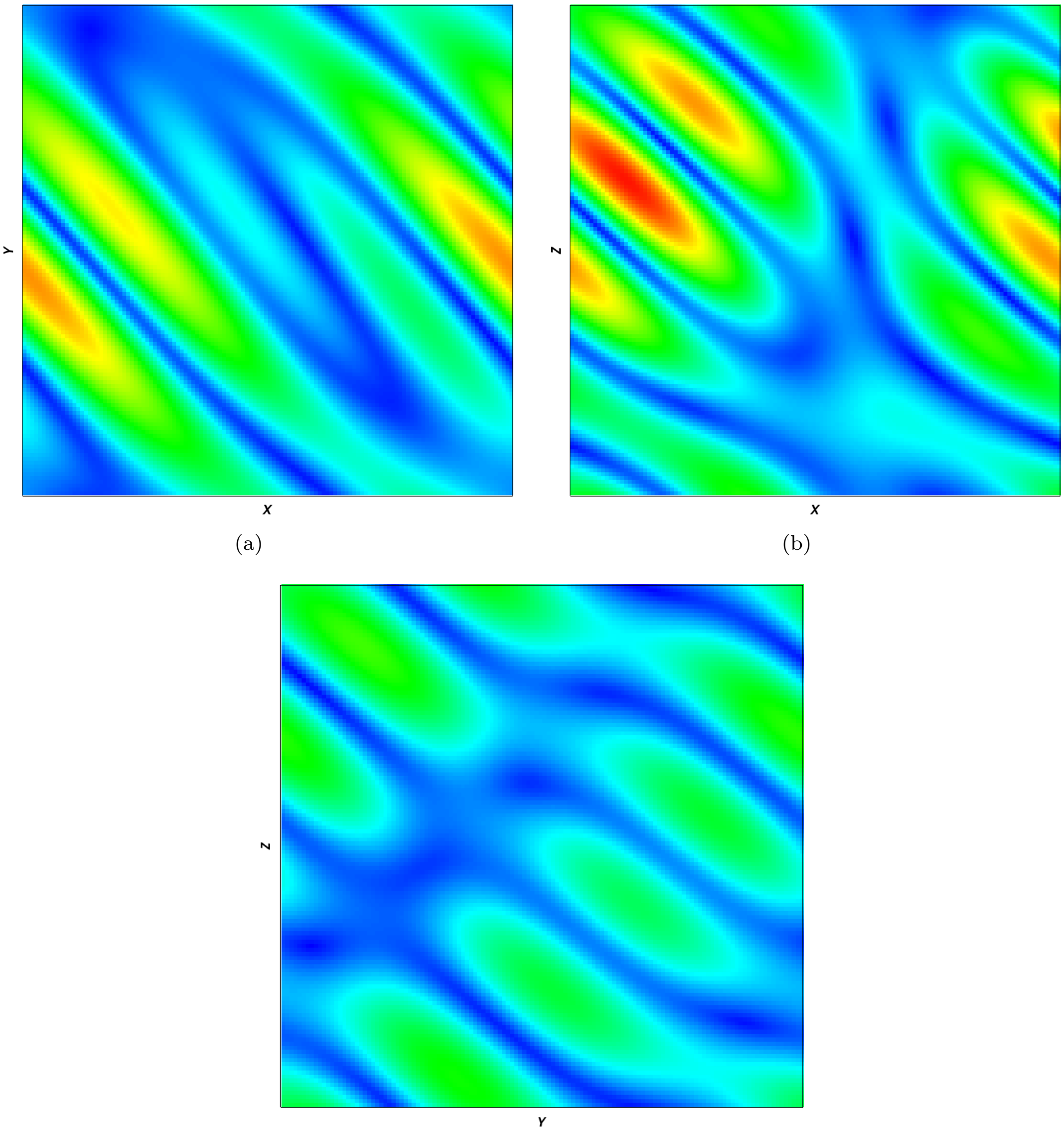
To address the extra complexity, Valio et al. (2010) proposed a linear forcing which allows the user to specify either the final turn-over time or the final dissipation rate. However, though this scheme reduces the complexity, the resulting force is analogous to the velocity meaning that an appropriate initial velocity field is a prerequisite.

To avoid the difficulties of the previous two methods, I have decided to employ a third option proposed in (Abdel Kareem et al., 2009). This method keeps a relatively simple mathematical algorithm without the need for complicated initialisation. However the disadvantage of this proposal is that the user cannot specify any turbulent characteristic of the final flowfield. Moreover, as I will show later, the final Re_λ depends on the spatial step, and thus a normalisation of the spectra presented in the previous Sec. 5.1.1 needs to be used to compare different resolutions. The formula for the estimation of $\mathbf{F}$ reads

$$\mathbf{F} \equiv \begin{cases} F_x = 2\rho A\left(\frac{\kappa_y\kappa_z}{|\kappa|^2}\right)G(\kappa_x,\kappa_y,\kappa_z,\phi), \\ F_y = -\rho A\left(\frac{\kappa_x\kappa_z}{|\kappa|^2}\right)G(\kappa_x,\kappa_y,\kappa_z,\phi), \\ F_z = -\rho A\left(\frac{\kappa_x\kappa_y}{|\kappa|^2}\right)G(\kappa_x,\kappa_y,\kappa_z,\phi), \end{cases} \tag{5.34}$$

with $G(\kappa_x, \kappa_y, \kappa_z)$ computed as

$$G(\kappa_x, \kappa_y, \kappa_z) = \sin\left(\frac{2\pi x}{L}\kappa_x + \frac{2\pi y}{L}\kappa_y + \frac{2\pi z}{L}\kappa_z + \phi\right). \tag{5.35}$$



(a)

(b)

(c)

Figure 5.4: Slices at 50% of the cube with N=128 cells after the first iteration.

In the above equations, ρ is the density of the fluid, A is the forcing amplitude, L is the length of the domain (assuming a cube) and $\phi(\kappa_x, \kappa_y, \kappa_z)$ is the random phase different for each combination of the wavevector. $\mathbf{F}$ is applied only to the lower wavenumbers $(0 < \boldsymbol{\kappa} \leq 2)$.

As far as the initial solution is concerned, every simulation is started from a zero velocity field with imposed unit density. Since one can start from a zero velocity field, there is no need to initialise the non-equilibrium part of the distribution function. This is an

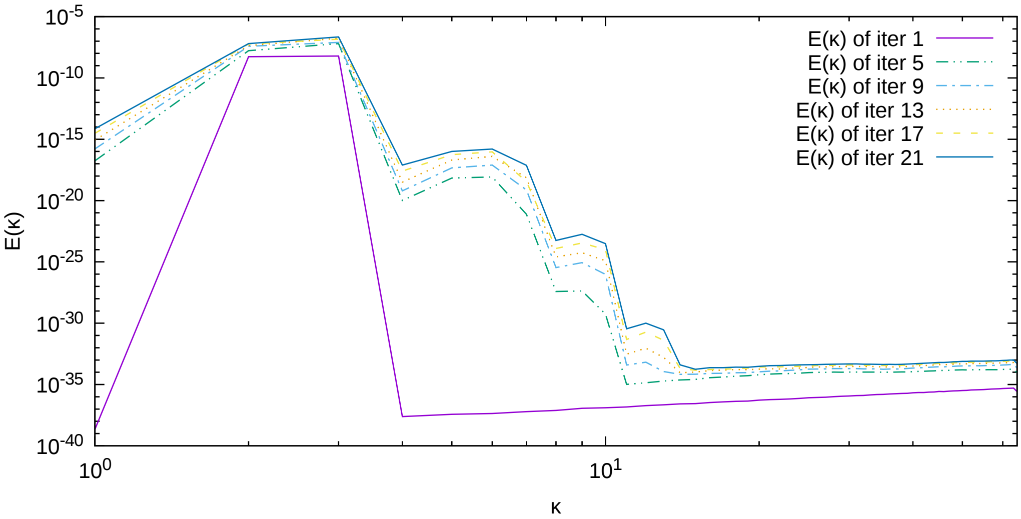
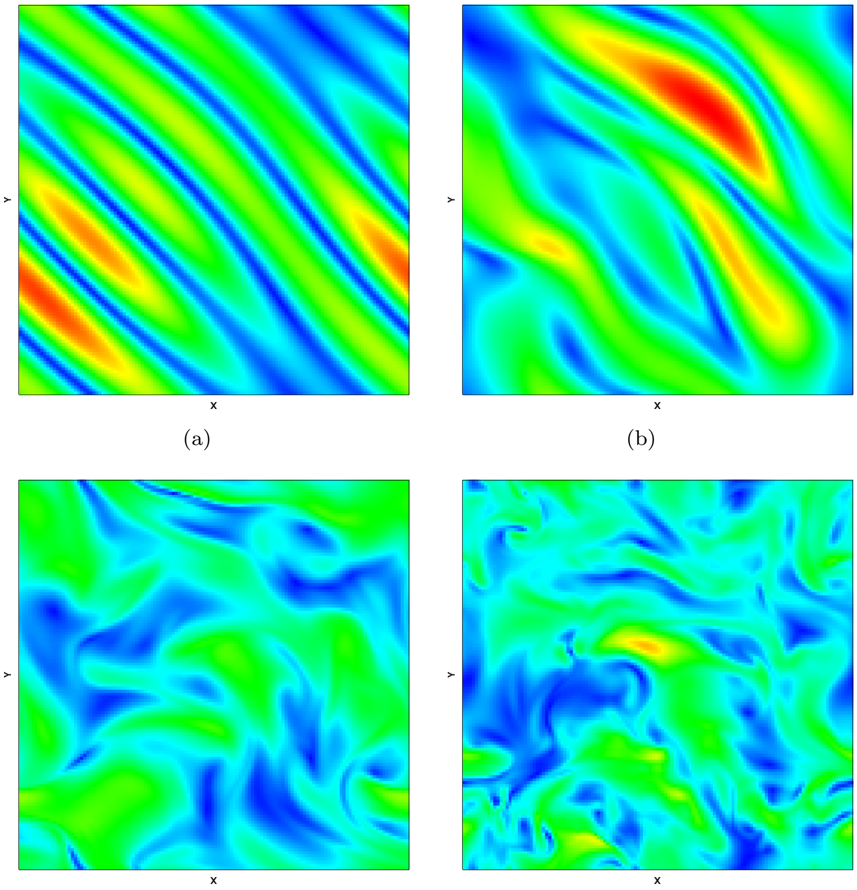


Figure 5.5: Evolution of Power spectra at the beginning of the simulation. ($N = 128,\ A = 10^{-4},\ \nu = 6 \cdot 10^{-5}\ \mathrm{m^2\ s^{-1}}$)

extra advantage of this force compared to other approaches. If an initial field requires initialisation of the non-equilibrium part, but one does not provide it, this is an inconsistency and will lead to instabilities arising during the initial phase. Then, one needs to allow some time for these instabilities to dissipate.

Another benefit of the zero initial velocity field is the possibility to have a representation of the forcing by plotting slices of the first iteration. Figure 5.4 shows the slices at the centre of the cube in the three planes, XY, XZ and YZ, of the velocity magnitude. The forcing affects only the large scales of the flow. To further support its establishment, the continuous line in Fig. 5.5 shows the energy spectrum after the first iteration. It is evident that the fictitious kinetic energy is applied only at the lower wavenumbers, and for the choice of $(0 < \boldsymbol{\kappa} \leq 2)$, only the $2 \leq |\boldsymbol{\kappa}| \leq 3$ wavenumbers are triggered, with almost the same amount of energy. As time passes, this external energy will be dissipated, expanding in this way the spectrum to the higher wavenumbers, a procedure that can be observed in Fig. 5.5. I present these results as a verification for the implementation of the forcing.

Figure 5.6 shows the magnitude of the velocity at the slice of 50% of the z-axis for four different iterations in time. These pictures demonstrate the initial stretching of the large artificial eddies and their gradual decomposition into smaller ones until a real turbulent flowfield appears. Similarly, the growth of the energy in the largest wavenumbers in Fig. 5.7 describes the appearance of smaller and smaller eddies as time passes.



(a)

(b)

(c)

(d)

Figure 5.6: Slices of velocity magnitude at 50% of the z axis of cube with N=128 cells after 5.6(a) 1000, 5.6(b) 7000, 5.6(c) 12000 and 5.6(d) 15000 iterations

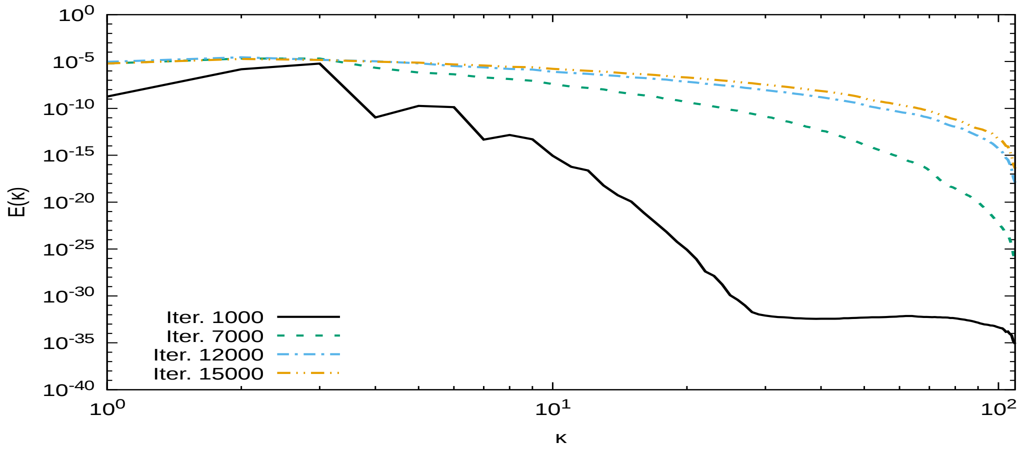


Figure 5.7: Evolution of power spectra in time. (N = 128, A = 10^{-4}, ν = 3 10^{-5} m^2 s^{-1})

N	32	128	256	512
Re_λ	46	58	65	68
L_{11}	1.02	0.92	0.89	0.88
λ	0.57	0.42	0.37	0.34
η	0.04	0.03	0.02	0.02
$\eta/\Delta x$	0.2	0.57	0.97	1.72
$\kappa_{max}\eta$	0.68	1.8	3.03	5.41
L_{11}/u_{rms}	251.44	133.32	102.64	88.05
$t_{aver}u_{rms}/L_{11}$	60.75	131.44	93.13	20.56

Table 5.1: Characteristics of the turbulent field for DNS for four resolutions with the STA collision model.

N	32	128	256	512
Re_λ	56	60	66	70
L_{11}	1.07	0.92	0.9	0.85
λ	0.68	0.43	0.37	0.34
η	0.05	0.03	0.02	0.02
$\eta/\Delta x$	0.24	0.57	0.97	1.67
$\kappa_{max}\eta$	0.75	1.81	3.01	5.26
L_{11}/u_{rms}	262.15	131.97	102.17	81.95
$t_{aver}u_{rms}/L_{11}$	58.27	132.8	68.95	31.15

Table 5.2: Characteristics of the turbulent field for DNS for four resolutions with the REG collision model.

5.2.2 Direct Numerical Simulations

Initially, DNS data were collected to evaluate the capability of the solver to deal with turbulent cases. For all cases, the input parameters were $A = 10^{-4}$ and the value of viscosity $\nu = 5 \cdot 10^{-5}$ m^2 s^{-1}. Four resolutions were simulated, and a comparison between STA and REG collision models was also carried out. In Table 5.1 and Table 5.2, one can see some characteristics of the flowfields for STA and REG, respectively. Examining the Re number, it is clear that Re tends to have higher values as the resolution increases, though convergence is also indicated. This behaviour mainly originates from the employed force, as mentioned before. Comparing the data between the two tables shows no significant discrepancies between the two collision models with the exemption of the lowest resolution $N32$. It appears that the STA model has estimated a smaller value of Re_λ than the REG model. This divergence is one key point that I will search and investigate in this chapter.

Inspecting the three lengths, namely L_{11}, λ and η, one can see a convergence towards smaller values the higher the resolution. This indicates the existence of smaller and smaller scales of turbulence in higher resolutions, an expected outcome. I have reported previously that the value of $\kappa_0 L_{11}$ should be smaller than 0.8 for better accuracy. In this case, the use of the force affects the large eddies, which leads to artificially larger values for the length L_{11}. Furthermore, the values of $\kappa_{max}\eta$ getting larger and larger as the resolution increases indicate the better representation of the small scales. Finally, the row before the last one reports the eddy turnover times, while the last one shows how many eddy turnover times have been spent for averaging. All the simulations have run more than ten times which is assumed to be the minimum threshold for accurate statistics.

To identify the initial transient period and the onset of the averaging one, a variety of turbulent characteristics were monitored over the simulation time. Figures 5.8, 5.9 and 5.10 show the turbulent kinetic energy k, dissipation rate ε and Kolmogorov length scale η, respectively, for the resolution of N 32 for both STA and REG models. First of all, the onset of the transient time was chosen based on the dissipation rate and turbulent kinetic energy, variables that need more time to transit compared to Kolmogorov scales. Comparing the two collision models, in the case of k, they seem to estimate the same averaged value. This indicates that the force adds the same amount of kinetic energy for both STA and REG. However, examining the dissipation rate, it is obvious that the STA model has returned a higher value. A higher value of ε means that more kinetic energy is removed in the case of STA than REG over time, explaining the lower value of the Re_λ. In the case of η, the difference between the two models is much smaller.

To further enrich this investigation, I also provide for comparison the same plots but for the resolution of $N128$ in Figs. 5.11, 5.12 and 5.13. In this scenario, there is no clear difference between the estimated averaged values for the two models, indicating that the reduction of the spatial step has alleviated the issue. Another important piece of information is that the averaged value of the turbulent kinetic energy is three times larger than in the resolution of $N32$. This last outcome shows the grid-dependence of the employed force and explains the higher values of Re_λ for finer meshes.

In Fig. 5.14 shows the Kolmogorov spectra for the DNS above. As mentioned before, this normalisation is suitable for comparing the dissipation range of different Re_λ simulations. The ideal behaviour is for all of them to converge in the highest wavenumbers to the model spectrum solution. It is clear that as the resolution increases, the results approach more and more the ideal solution. The resolution of $N512$ fits precisely. Therefore, based on this finding and using the model spectrum as a threshold, one can conclude that $\kappa_{max}\eta \geq 5$ is needed to accurately capture all the small scales in the case of LBM DNS. Comparing the two collision models shows that STA simulations have predicted spectra closer to the expected one for all the resolutions. At first glance, this observation could lead to the outcome that the STA model is less dissipative than the REG one. In the case of $N512$, the STA solution has overestimated the energy level in the highest wavenumbers, while the REG solution fits precisely. The mismatching of the spectra with the model in the lowest wavenumbers is due to the employment of the force.

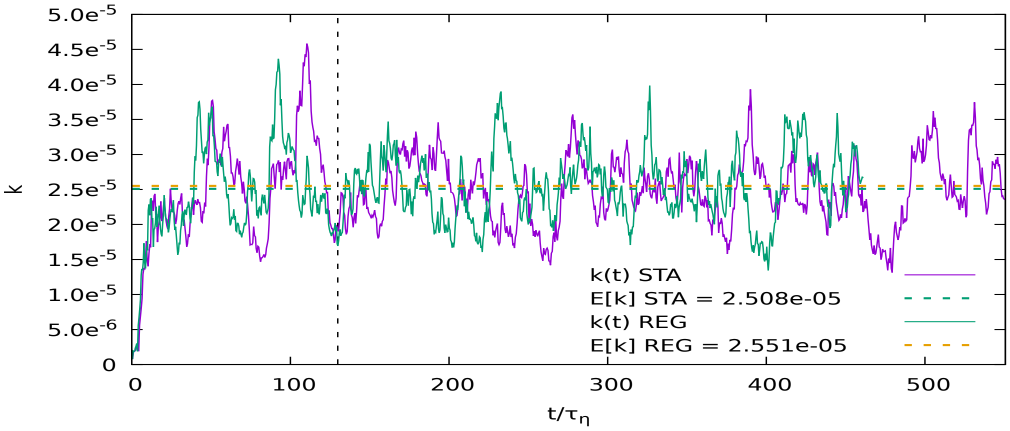


Figure 5.8: Evolution of turbulent kinetic energy for STA and REG DNS of $N = 32$ resolution.

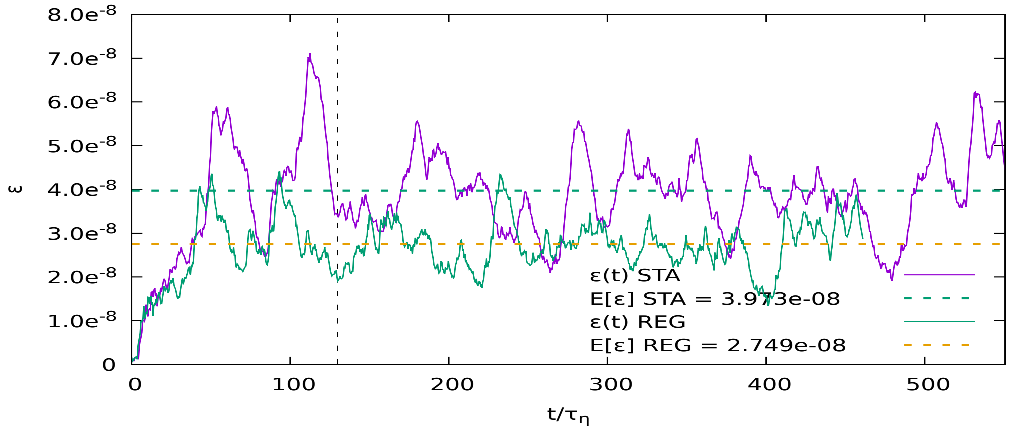


Figure 5.9: Evolution of dissipation rate for STA and REG DNS of $N = 32$ resolution.

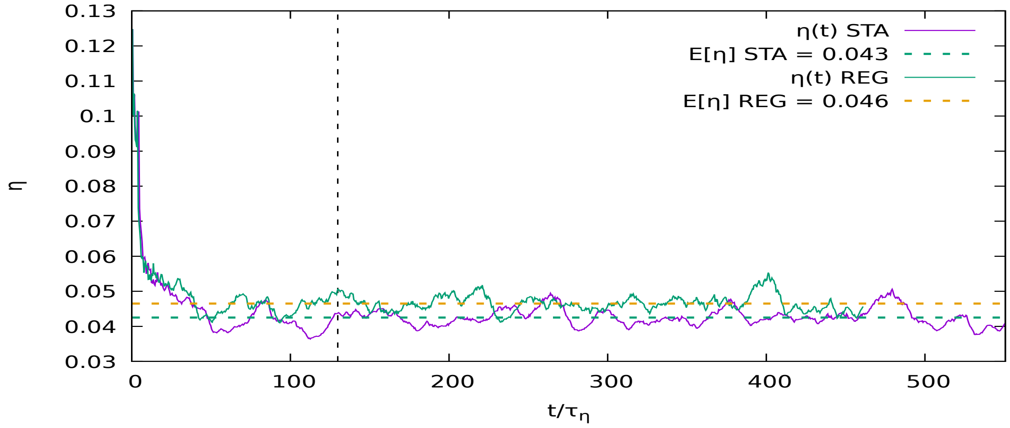


Figure 5.10: Evolution of Kolmogorov length scale for STA and REG DNS of $N = 32$ resolution.

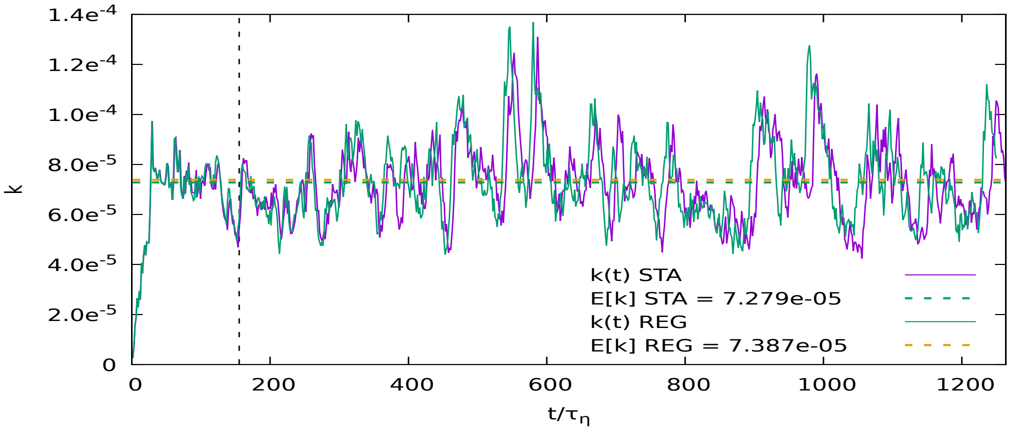


Figure 5.11: Evolution of turbulent kinetic energy for STA and REG DNS of $N = 128$ resolution.

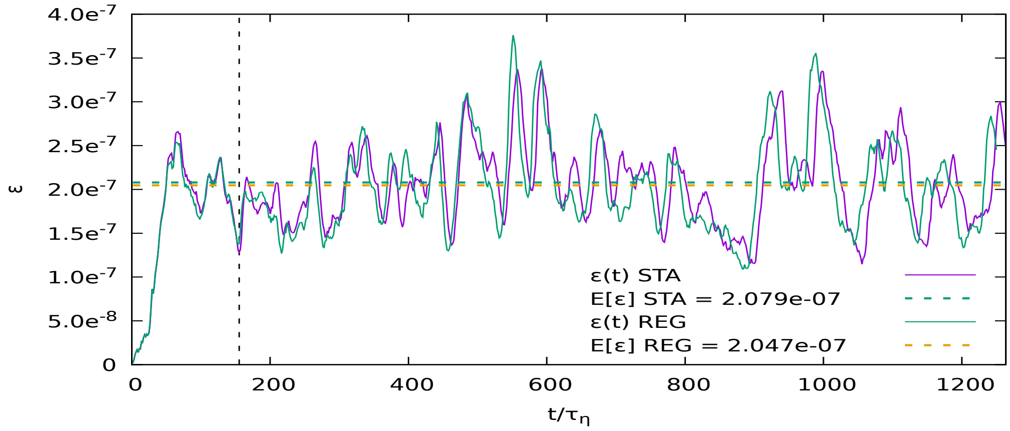


Figure 5.12: Evolution of dissipation rate for STA and REG DNS of $N = 128$ resolution.

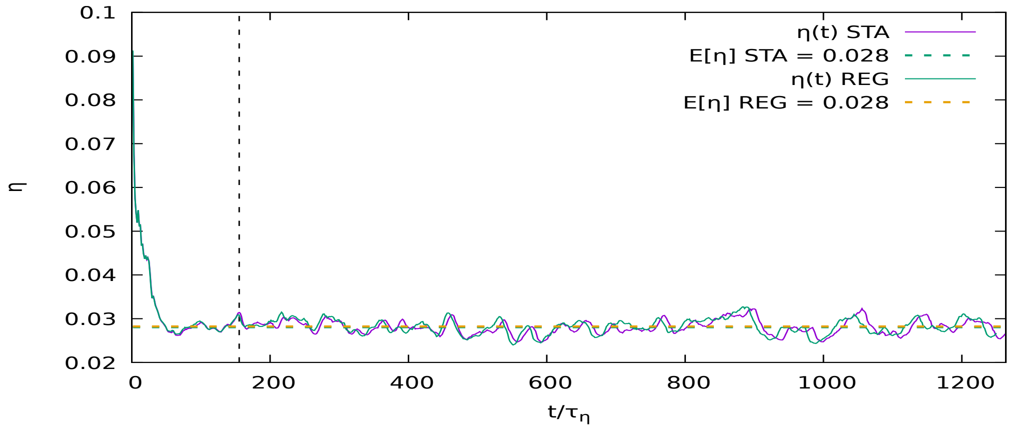


Figure 5.13: Evolution of Kolmogorov length scale for STA and REG DNS of $N = 128$ resolution.

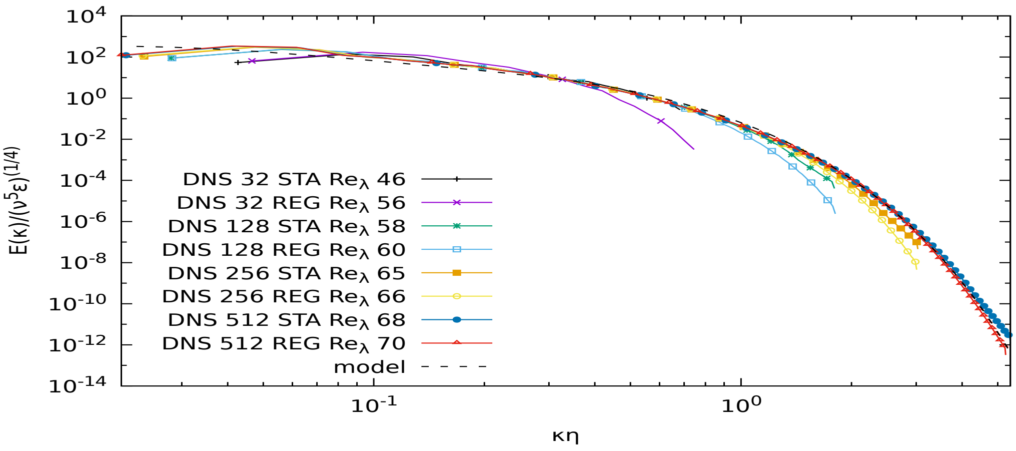


Figure 5.14: Kolmogorov energy spectra of DNS compared against the model spectrum.

Similar conclusions can be extracted by inspecting Fig. 5.15. In this plot, the spectra have been normalised with the turbulent kinetic energy and the integral length scale to investigate the energy-containing range. This plot's critical outcome is the coincidence of all the spectra in this range, indicating that the force produces similar flow fields independently of resolution. This is a validation of the current procedure of comparing the spectra using normalisations.

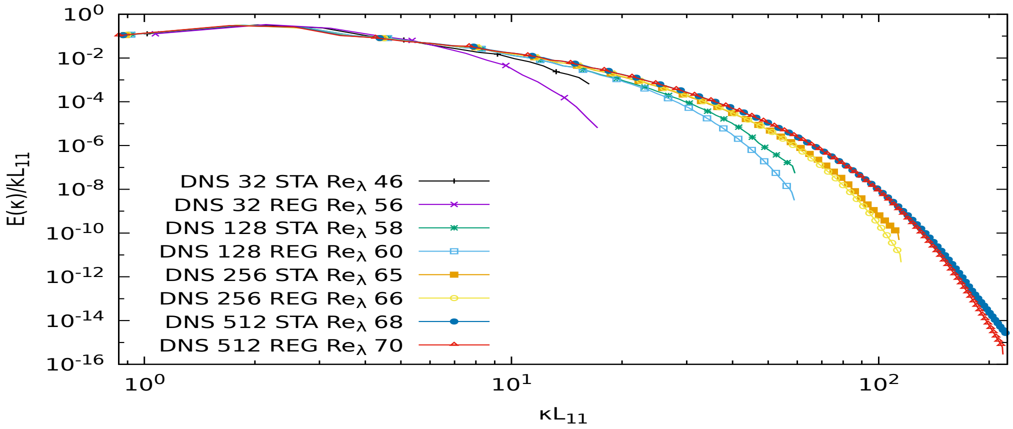


Figure 5.15: Energy spectra of DNS normalised for comparison of the energy-containing range.

In Fig. 5.16 shows the compensated Kolmogorov spectra, a normalisation suitable for comparison of the inertial sub-range. In the lowest-resolution simulations, both STA and REG have overestimated the peak, which seems to be due to the application of the force. However, all other resolutions have returned the same level around this range with a value close to 2. From theory, it is known that the expected value should be around 1.5. To investigate this mismatching, I have also run another DNS with higher Re_λ of about 165 and with a resolution of $N128$. The result is shown in Fig. 5.17. To increase the Re_λ, the viscosity was reduced to $\nu = 10^{-5}$. Though this simulation has returned a value of $\kappa_{max}\eta < 1$ and thus it is under-resolved, the inertial sub-range level is much closer to the theoretical value. Therefore, the high value of the level is due to the relatively low Re_λ in the previous simulations, and it does not seem to be an imperfection of the current LBM solver.

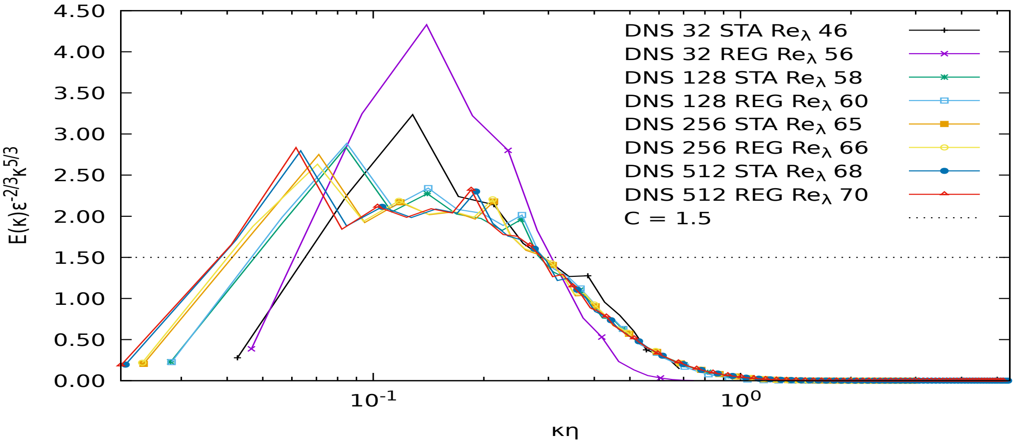


Figure 5.16: Compensated Kolmogorov energy spectra of DNS.

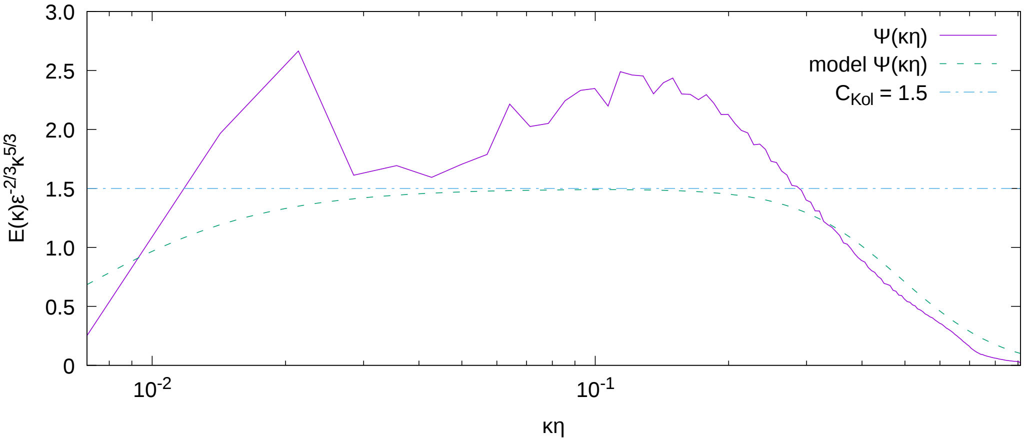


Figure 5.17: Compensated energy spectrum of DNS.

Up to this point, I have focused on the energy spectra and a variety of their normalisations. To further investigate the collision models' differences, I have also calculated the pressure fluctuation spectra presenting in Fig. 5.18. The first observation is that increasing the resolution, an increasing number of eddies are captured by the simulations enlarging the values in the whole range of the wavenumbers, though a convergence seems to take place, too. However, the most interesting outcome is the excess of the values appearing in the highest wavenumbers in the STA model for all resolutions, pointing to numerical instabilities. Consequently, one can conclude that the less accurate estimation of the non-equilibrium part in the STA model case leads to the appearance of more than small natural scales in the flowfield. One can combine the knowledge from the previous energy spectra with the pressure fluctuation ones. The apparent better accuracy of the STA model, in the case of the energy spectra, originated by the excess of small scales that in reality are instabilities arising from the less accurate collision step. Therefore, just by examining the energy spectra, one could be led to the wrong assumption that SRT is less dissipative than REG. It seems that by decreasing the spatial step, the situation gets improved. In the lowest resolution, the excess in the highest wavenumbers is comparable with the values in the lowest ones. In the case of $N32$, the dissipation rate's immense value is due to excess in the highest wavenumbers.

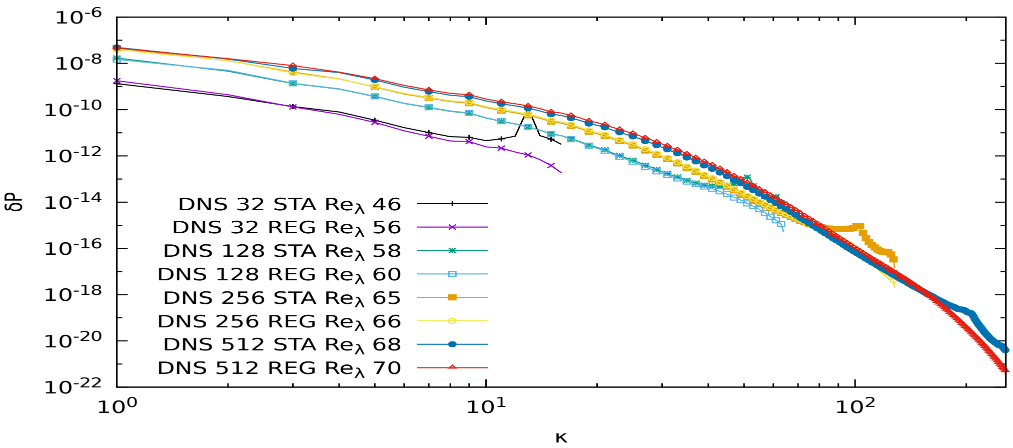


Figure 5.18: Pressure fluctuation spectra of DNS.

5.2.3 Large Eddy Simulations

In the previous section I have discussed the performance of the AMROC LBM solver to deal with turbulence without the use of turbulence models. This section will evaluate the newly implemented turbulence models and their coupling with the two collision models. The first results will be about CSMA, followed by DSMA and finally WALE. I will examine three resolutions, namely $N128, N64$ and $N32$.

Constant Smagorinsky

For all the CSMA simulations, the value of the constant C was set to 0.1. Following the same procedure as in the DNS case, Table 5.3 reports some characteristics of the flowfields for all the resolutions and both STA and REG models. Again, the final Re_λ is resolution-dependent. In the case of the lowest resolution $N32$, the addition of the turbulent model has increased the estimated Re_λ, particularly in the STA model value of $\kappa_{max}\eta$. The latter is an indication of how well the flowfield has been resolved. Therefore, one can assume that CSMA has benefited the results, particularly for STA. On the other hand, in the resolution $N128$, both STA and REG have yielded identical numbers that are similar to the REG DNS of the same resolution.

Figure 5.19 shows the Kolmogorov energy spectra of CSMA simulations for the three resolutions compared against the model spectrum. The same pattern as in DNS can be

	STA			REG		
N	32	64	128	32	64	128
Re_λ	51	55	60	58	58	60
L_{11}	1.07	0.98	0.92	1.12	0.98	0.92
λ	0.63	0.51	0.43	0.71	0.54	0.43
η	0.05	0.03	0.03	0.05	0.04	0.03
$\eta/\Delta x$	0.23	0.35	0.58	0.24	0.36	0.58
$\kappa_{max}\eta$	0.72	1.11	1.81	0.77	1.15	1.82
L_{11}/u_{rms}	264.56	180.17	133	276.12	181.57	132.61
$t_{aver}u_{rms}/L_{11}$	57.74	94.6	128.1	55.32	93.87	128.44

Table 5.3: Characteristics of the turbulent field for CSMA ($C = 0.1$) for three resolutions for both the STA and REG collision models.

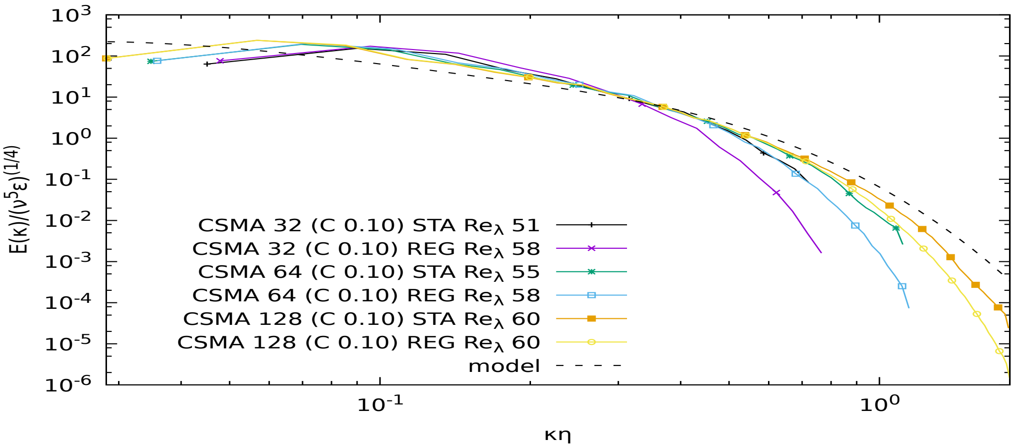


Figure 5.19: Kolmogorov energy spectra of CSMA compared against the model spectrum.

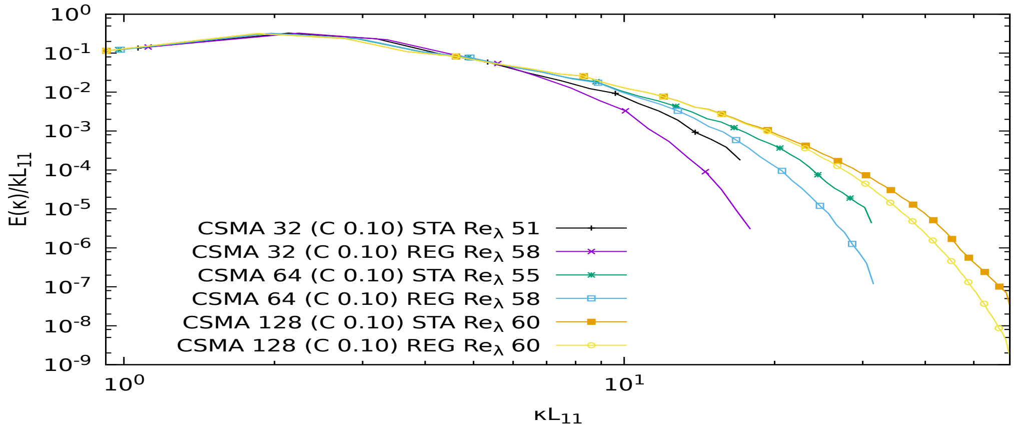


Figure 5.20: Energy spectra of CSMA normalised for comparison of the energy-containing range.

observed with the STA results to be closer to the model, though increasing the resolution reduces the difference. Moreover, Fig. 5.20 plots the energy spectra normalised for comparison of the energy-containing range. All the curves coincide in the low wavenumbers,

	STA			REG		
N	32	64	128	32	64	128
Re_λ	52	57	60	58	61	61
L_{11}	1.08	1.0	0.93	1.12	1.01	0.93
λ	0.64	0.52	0.43	0.72	0.56	0.44
η	0.05	0.03	0.03	0.05	0.04	0.03
$\eta/\Delta x$	0.23	0.36	0.58	0.24	0.37	0.58
$\kappa_{max}\eta$	0.73	1.12	1.81	0.77	1.16	1.83
L_{11}/u_{rms}	267.44	181.45	132.49	274.57	184.55	133.56
$t_{aver}u_{rms}/L_{11}$	57.12	83.11	132.27	55.64	81.71	123.3

Table 5.4: Characteristics of the turbulent field for DSMA for three resolutions for both the STA and REG collision models.

proving that the turbulent model has not affected the large scales as is expected.

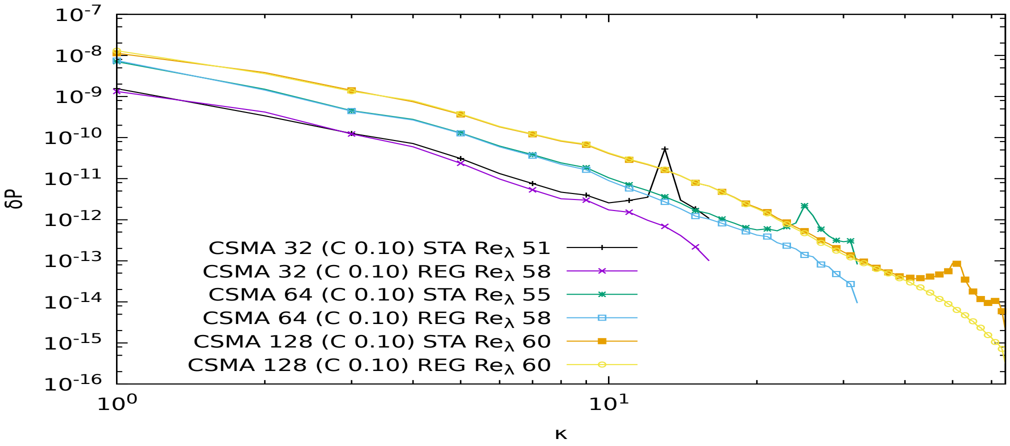


Figure 5.21: Pressure fluctuation spectra of CSMA.

Finally, Fig. 5.21 shows the pressure fluctuation spectra for the case of CSMA. The turbulence model was unable to eradicate the numerical instabilities in the case of the STA model.

Dynamic Smagorinsky

Table 5.4 presents characteristics of the flowfield of the DSMA simulations. Compared with CSMA data, DSMA has predicted higher values of Re_λ in almost all cases. This is the first indication of less dissipation added due to the turbulence model, which is expected to improve the overall accuracy. Moreover, DSMA has also slightly improved the values of $\kappa_{max}\eta$.

Similarly to the CSMA case, Figs. 5.22, 5.23 and 5.24 are given to examine the grid convergence of the DSMA turbulence model. The same trends as in the case of CSMA are observed here, too.

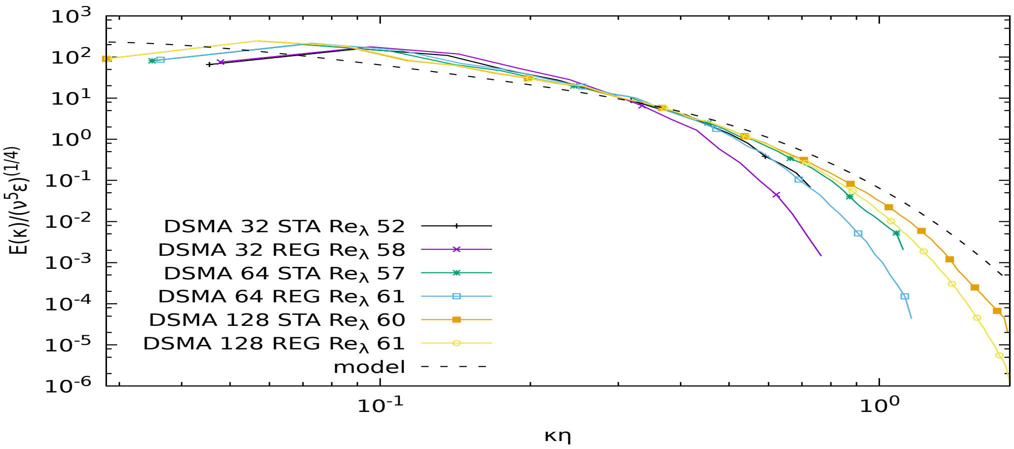


Figure 5.22: Kolmogorov energy spectra of DSMA compared against the model spectrum.

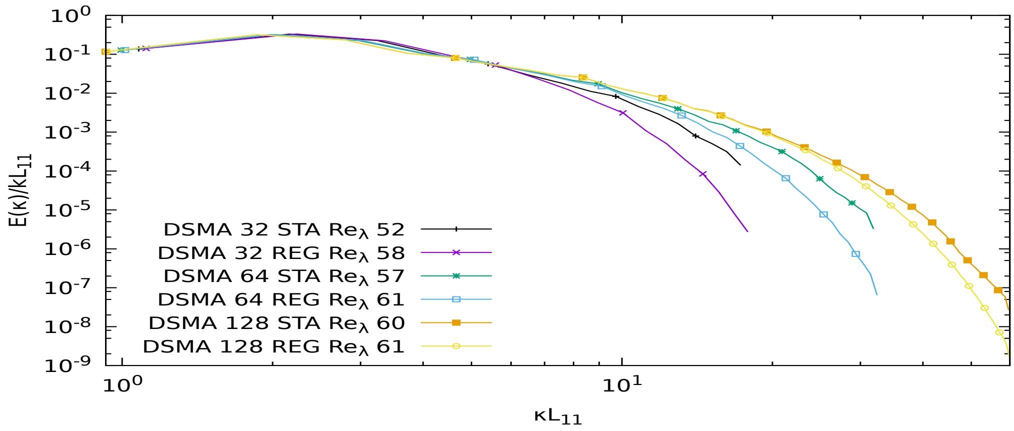


Figure 5.23: Energy spectra of DSMA normalised for comparison of the energy-containing range.

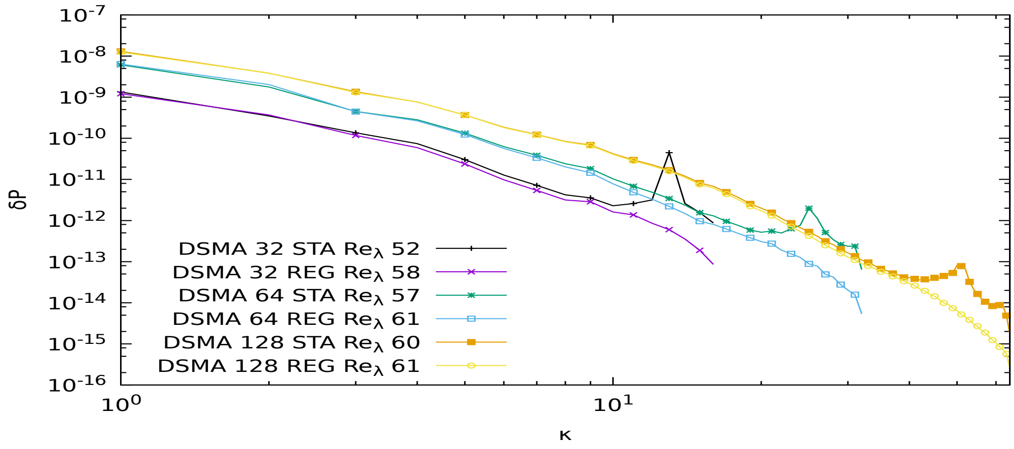


Figure 5.24: Pressure fluctuation spectra of DSMA.

WALE

Table 5.5 presents characteristics of the flowfield of the WALE simulations. The WALE data are much closer to DSMA, indicating similar behaviour for this specific test case.

	STA			REG		
N	32	64	128	32	64	128
Re_λ	52	56	60	59	60	61
L_{11}	1.09	0.99	0.93	1.11	1.	0.94
λ	0.64	0.52	0.43	0.72	0.54	0.44
η	0.05	0.04	0.03	0.05	0.04	0.03
$\eta/\Delta x$	0.23	0.36	0.58	0.24	0.36	0.58
$\kappa_{max}\eta$	0.73	1.12	1.82	0.77	1.15	1.84
L_{11}/u_{rms}	268.56	181.24	133.88	273.23	182.38	134.98
$t_{aver}u_{rms}/L_{11}$	56.88	134.34	130.89	55.91	82.68	129.83

Table 5.5: Characteristics of the turbulent field for WALE for three resolutions for both the STA and REG collision models.

Again, Figs. 5.25, 5.26 and 5.27 are given to examine the grid convergence of WALE. There are no obvious differences between WALE and the other two models.

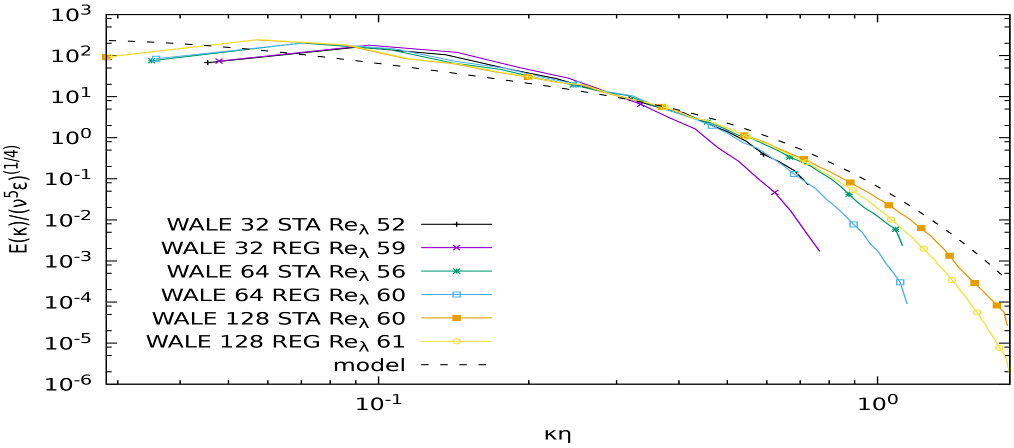


Figure 5.25: Kolmogorov energy spectra of WALE compared against the model spectrum.

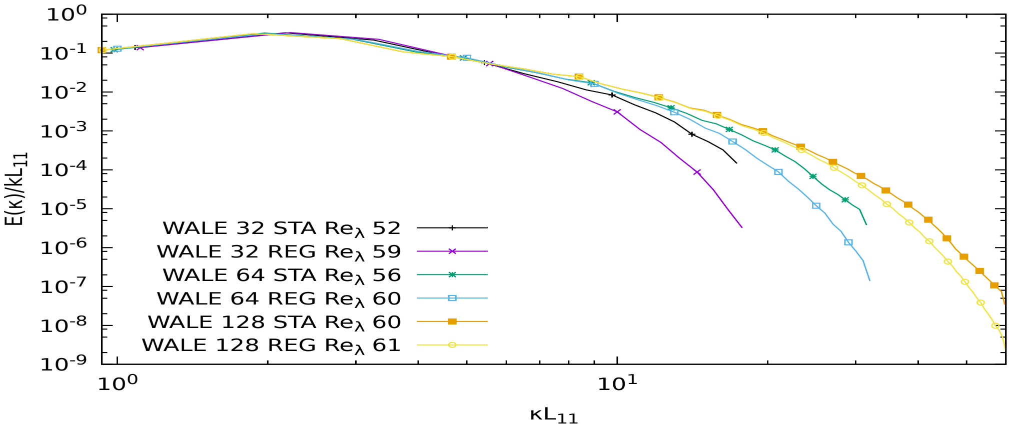


Figure 5.26: Energy spectra of WALE normalised for comparison of the energy-containing range.

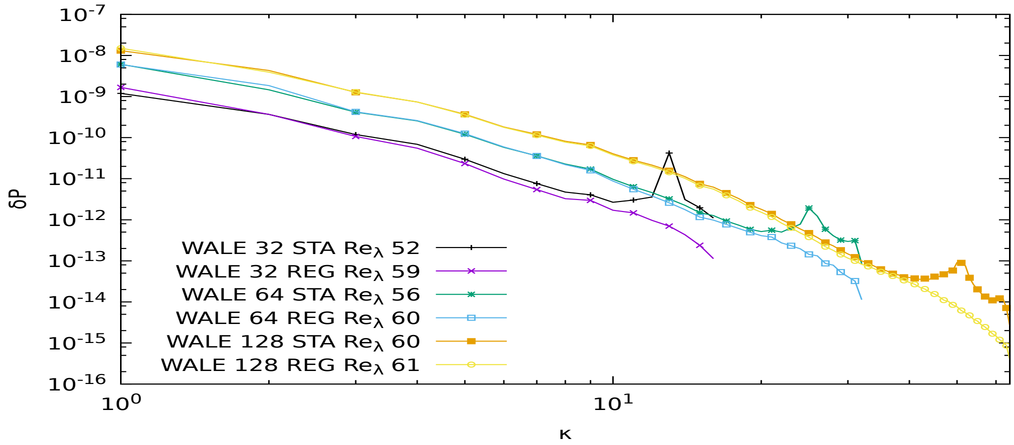


Figure 5.27: Pressure fluctuation spectra of WALE.

5.2.4 Comparison of LES and DNS

In the previous section, I performed a grid convergence analysis for the three turbulence models. All of them returned similar results both in the case of STA and REG collision models. In this section, I will compare them against each other and the DNS data. To do so, the results of two resolutions, namely the $N32$ and $N128$, will be presented.

N 32 Resolution

Figures 5.28 shows the Kolmogorov energy spectra for the $N32$ resolution for both collision models. All the STA versions have predicted more energy in the highest wavenumbers compared to REG counterparts. Moreover, there is a noticeable difference in STA between DNS and all three LES, with DNS to predict more energetic small scales. However, examining the pressure fluctuation spectra in Fig. 5.29, one can observe that DNS has also predicted more instabilities in the high wavenumbers compared to LES. Therefore, one can conclude that the addition of the LES models in the case of STA has benefited the results by reducing the instabilities. There are no significant differences among the LES data. On the other hand, in the case of REG, both DNS and LES energy spectra coincide, while there is only a small reduction for LES in the case of pressure fluctuation spectra for high wavenumbers. This last deviation seems to be the only effect of the addition of the LES models for the REG collision model.

N128 Resolution

Following the same procedure, Fig. 5.30 presents the Kolmogorov energy spectra for $N128$ resolution. Again, the STA versions have predicted more energetic small scales, though the difference is smaller than before. Furthermore, in this case, there are no apparent discrepancies between STA DNS and LES. The latter can also be confirmed by

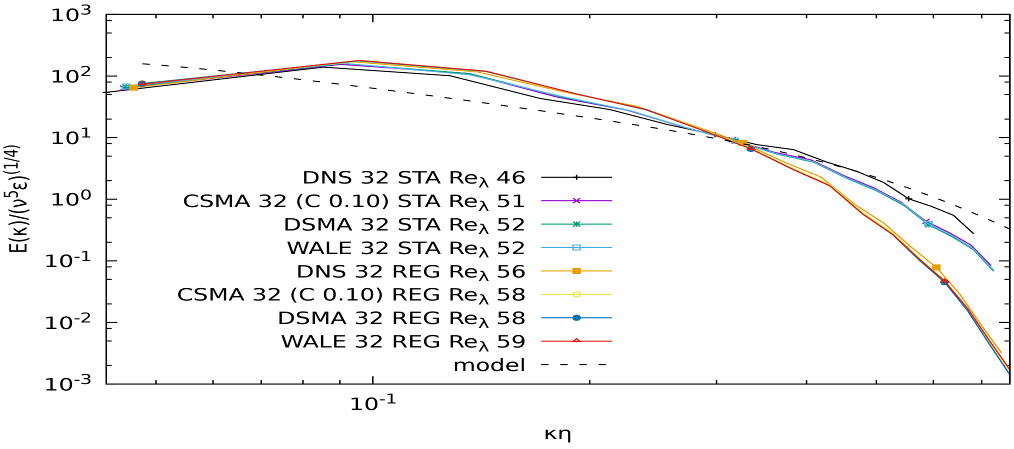


Figure 5.28: Comparison of Kolmogorov energy spectra for DNS and LES with N32 resolution.

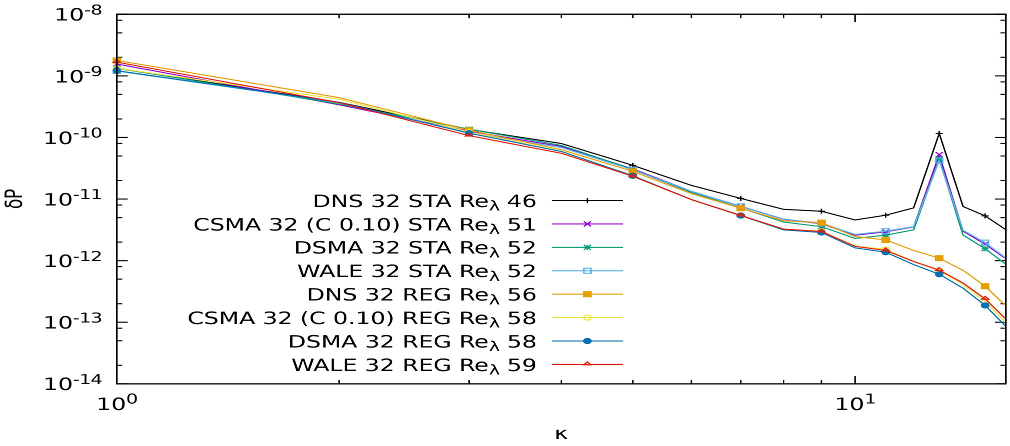


Figure 5.29: Comparison of pressure fluctuation spectra for DNS and LES with N32 resolution.

examining the pressure fluctuation spectra in Fig. 5.31, where the STA curves coincide. However, compared to REG, all STA versions have estimated instabilities in the highest wavenumbers. In the case of REG, in both plots, all the curves match.

Consequently, though the FHIT test case mainly enabled us to compare the two collision models STA and REG, it did not provide a challenging benchmark for verifying and evaluating the LES models. To surpass this obstacle, the DHIT case was employed.

5.3 Decaying Isotropic Turbulence

In the previous section, I have examined the test case of FHIT. I have shown that the final Re_λ and thus the input of kinetic energy depended strongly on the resolution and weakly on the collision and the application of LES models. This behaviour was an obstacle for the detailed verification of the LES models. Consequently, to have a proper

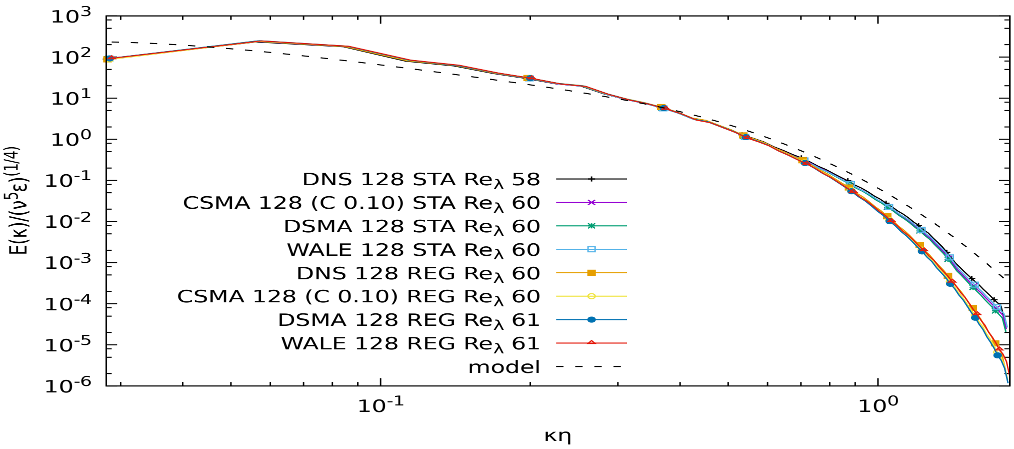


Figure 5.30: Comparison of Kolmogorov energy spectra for DNS and LES with N128 resolution.

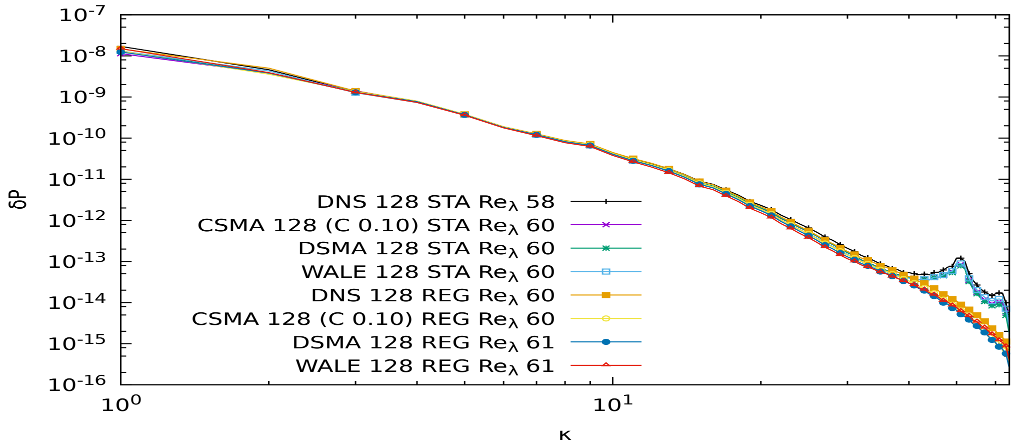


Figure 5.31: Comparison of pressure fluctuation spectra for DNS and LES with N128 resolution.

benchmark test case and avoid any influence of the force on results, the test case of DHIT was employed next. In this scenario, the simulation of FHIT has been restarted without applying the force. To ensure a fair comparison, all simulations in this section have been initialised based on DNS with the resolution of 512^3 running with the REG collision model. As I have shown in the previous section, this combination has returned the most accurate results in the energy and pressure fluctuation spectra. All simulations in this section have run for a final time of 1000 time units. As reference data, I have used DNS to resolve $N512$ with the REG collision model.

Initially, I will present the results for three resolutions, namely $N32$, $N64$ and $N128$ for all three turbulence and the two collision models. The initial solutions for these cases were based on suitable local volume averaging. Afterwards, I will investigate the effect of estimating the strain rate locally or using a stencil. Finally, I will close this section showing contours of vorticity magnitude of instantaneous flowfields.

N32 Resolution

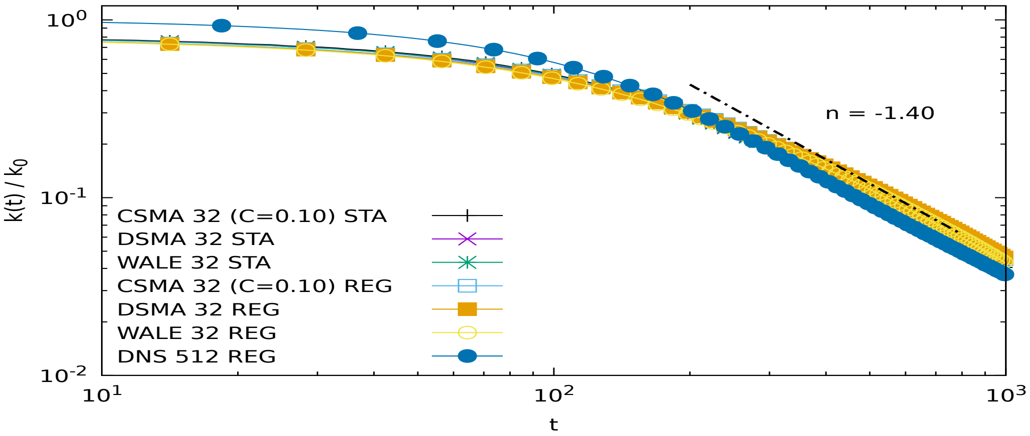


Figure 5.32: Evolution of the turbulent kinetic energy k for CSMA with $C = 0.1$, DSMA and WALE of a resolution of 32^3 cells for both STA and REG SRT. The DNS of 512^3 resolution with REG SRT has been added as a reference.

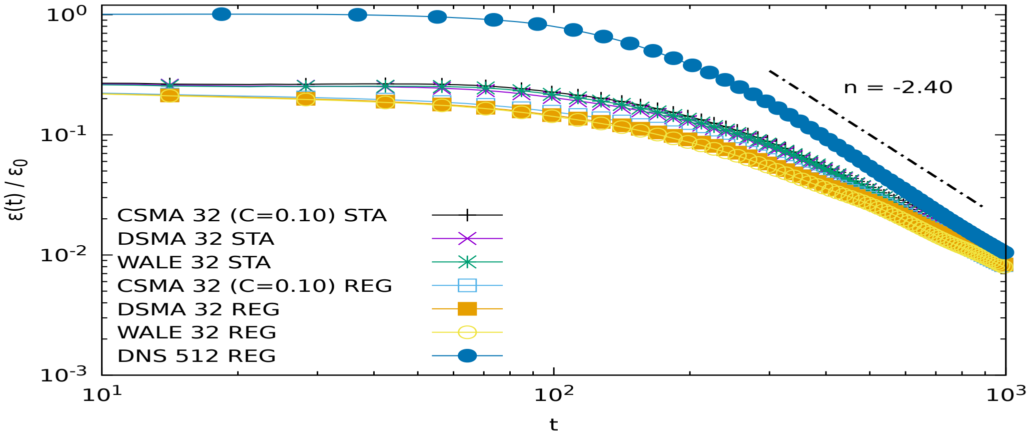


Figure 5.33: Evolution of dissipation rate ε for CSMA with $C = 0.1$, DSMA and WALE of a resolution of 32^3 cells for both STA and REG SRT. The DNS of 512^3 resolution with REG SRT has been added as a reference.

Figure 5.32 presents the evolution of the turbulent kinetic energy k for the resolution $N32$. All the combinations of turbulent and collision models have predicted identical curves. However, they show some divergence for the DNS reference due to coarse resolution, particularly during the first 100 time units. Moreover, Fig. 5.33 plots the evolution of the dissipation rate ε. In this case, there is an observable discrepancy between STA and REG results, with the former to estimate higher values over the whole duration. There is no difference between the performance of the LES models. However, there is a significant discrepancy between the lower resolution LES results and the DNS data. This is expected since, in the finer DNS, a higher number of small eddies will be simulated, and thus more kinetic energy will be transferred to them resulted in higher values of the dissipation rate.

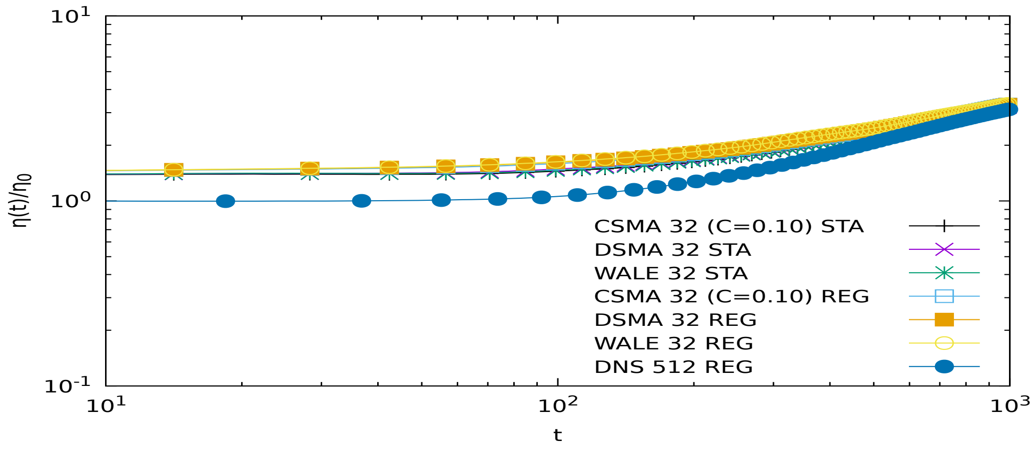


Figure 5.34: Evolution of Kolmogorov length scale η for CSMA with $C = 0.1$, DSMA and WALE of a resolution of 32^3 cells for both STA and REG SRT. The DNS of 512^3 resolution with REG SRT has been added as a reference.

From the theory of decaying homogeneous isotropic turbulence, one can anticipate that power-laws of the type $k \sim (t + t_0)^{-n}$ and $\varepsilon \sim (t + t_0)^{-n-1}$ can describe the slopes in the current plots. Therefore, I have also estimated the exponents based on the reference DNS results with $n = 1.4$ for the case of the turbulent kinetic energy and $n = 2.4$ for the dissipation rate. These values are in the expected range predicted by the literature (Huang and Leonard, 1994). Only the former slope seems to be simulated correctly in the case of LES.

Finally, in Fig. 5.34 I plot the evolution of the Kolmogorov length scale η to enrich the comparison. All the LES curves collapse together, indicating that the collision models have no significant effect on this variable. However, there is an apparent discrepancy between them and the DNS data. Due to the finer resolution, the latter has estimated much smaller eddies hence the lower value of η.

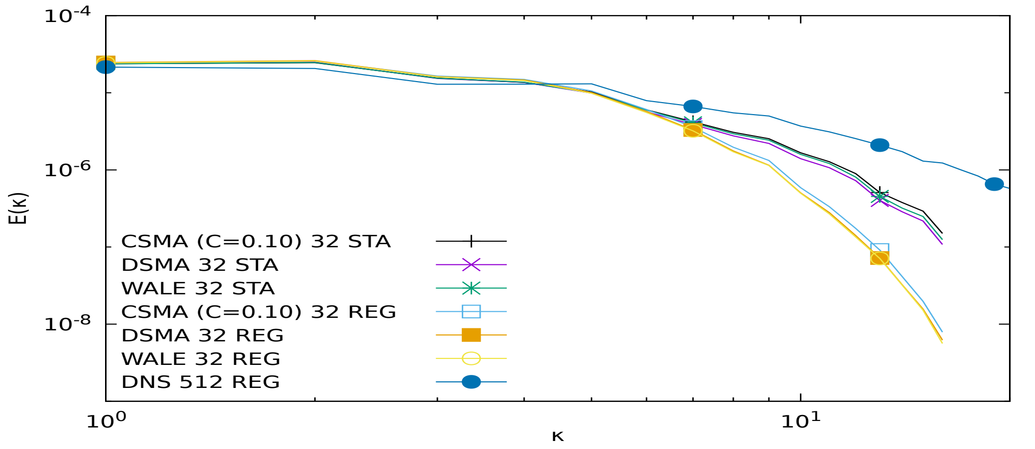


Figure 5.35: Instantaneous energy spectra of CSMA ($C = 0.1$), DSMA and WALE for both STA and REG SRT for the resolution of 32^3 cells at $t = 98.17$ time units. The curve of the REG DNS on 512^3 cells are shown as a reference.

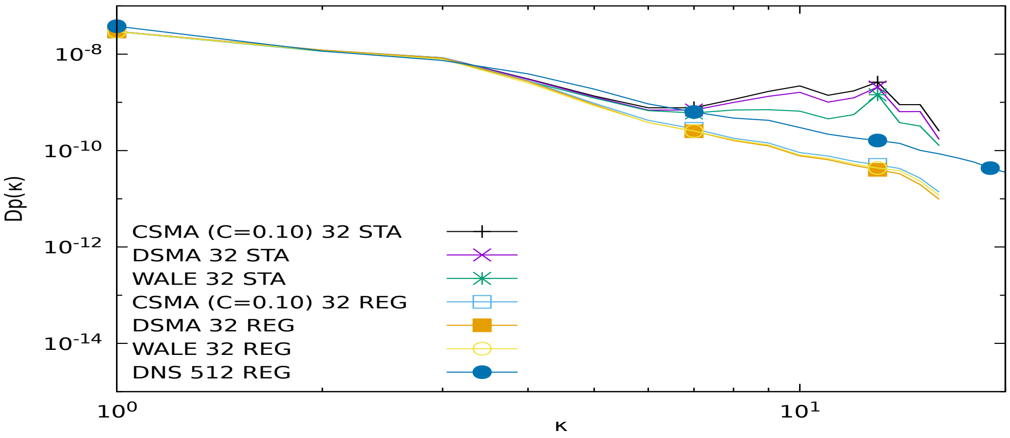


Figure 5.36: Instantaneous pressure fluctuation spectra of CSMA ($C = 0.1$), DSMA and WALE for both STA and REG SRT for the resolution of 32^3 cells at $t = 98.17$ time units. The REG DNS on 512^3 cells are shown as a reference.

Up to this point, I have examined the evolution in time of some quantities. To improve my investigation, Figs. 5.35 and 5.36 show the instantaneous, at $t = 98.17$, energy and pressure fluctuation spectra. Again the reference solution is based on the DNS REG of $N512$ resolution. The curves of the LES models coincide with the reference in the low wavenumbers for both plots. This is an indication that neither LES nor collision models have affected the large scales of the flowfield. However, similar to the FHIT case, the STA versions have estimated more energetic small scales than REG. Simultaneously, the same pattern of instabilities appearing in high wavenumbers for the pressure fluctuation spectra in the STA versions. This is the reason for the higher values of the dissipation rate ε compared to REG. One significant difference is that the LES models have estimated different instabilities levels with WALE returning the closest solution to the reference and thus better performance. On the other hand, there is no obvious difference among REG LES. Nonetheless, the REG versions have underestimated the values of pressure fluctuations in the small scales significantly, pointing to a dissipative behaviour in the case of a highly coarse mesh.

N64 Resolution

Following the same order as in the case of $N32$, Figs. 5.37, 5.38 and 5.39 report the time evolution of turbulent kinetic energy k, dissipation rate ε and Kolmogorov length scale η for $N64$ resolution. By increasing the resolution, all results have been improved, converging to the reference data. Moreover, the discrepancy between STA and REG versions in the plot of dissipation rate is much smaller.

Examining the spectra in Figs. 5.40 and 5.41, a better convergence towards the reference data is also apparent. One significant distinction in this resolution is that all STA LES results have estimated the same instabilities level in high wavenumbers for the

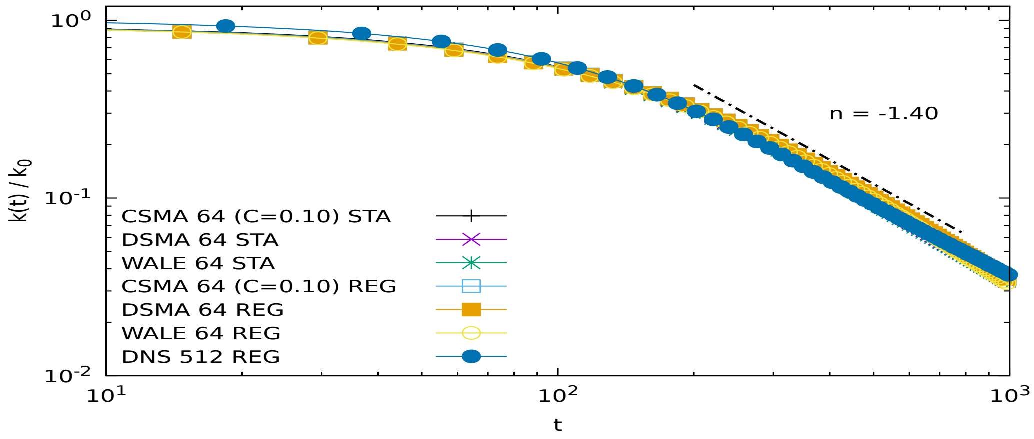


Figure 5.37: Evolution of the turbulent kinetic energy k for CSMA with $C = 0.1$, DSMA and WALE of a resolution of 64^3 cells for both STA and REG SRT. The DNS of 512^3 resolution with REG SRT has been added as a reference.

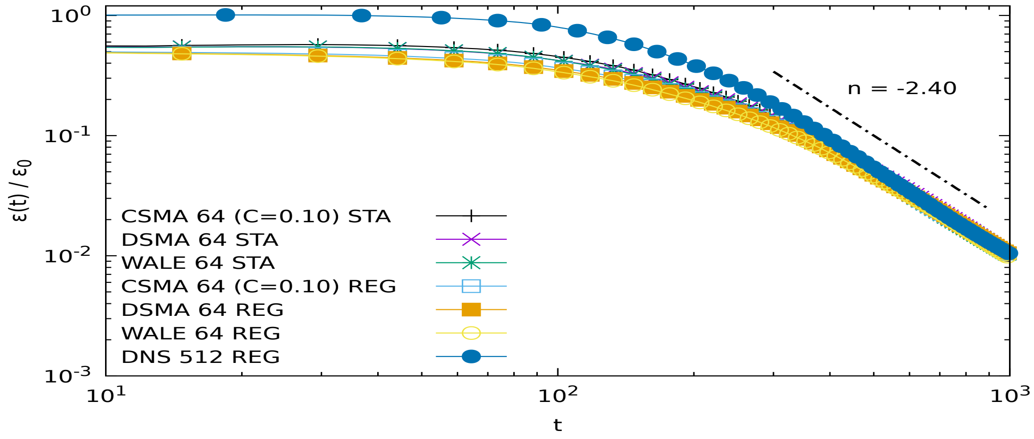


Figure 5.38: Evolution of dissipation rate ε for CSMA with $C = 0.1$, DSMA and WALE of a resolution of 64^3 cells for both STA and REG SRT. The DNS of 512^3 resolution with REG SRT has been added as a reference.

pressure fluctuation spectra. Another important observation is that both STA and REG versions for some wavenumbers underestimate the pressure fluctuation spectra, though STA results seem to be closer to the reference up to the point where instabilities pollute the curves.

Consequently, one can argue that the addition of the regularisation procedure has increased the scheme's dissipation. Simultaneously, this procedure is essential to eradicate the instabilities, which seems to be the main reason for the apparent better performance of the STA versions in the energy spectra.

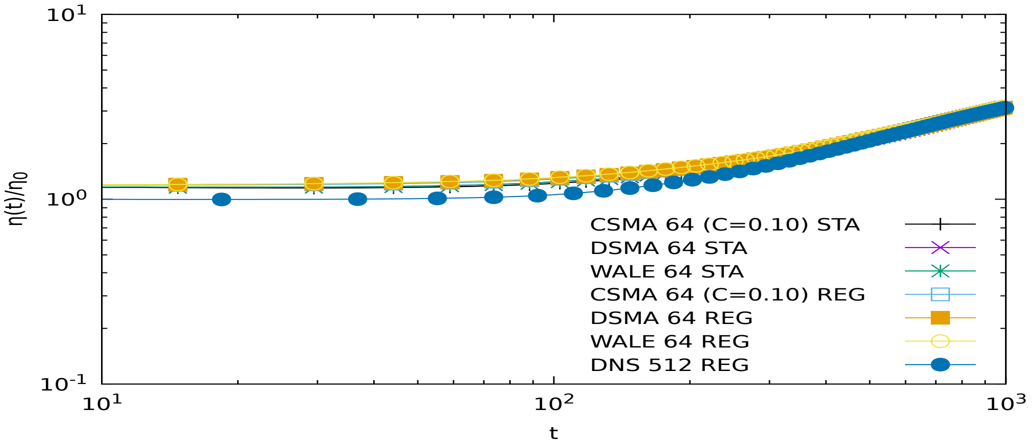


Figure 5.39: Evolution of Kolmogorov length scale η for CSMA with $C = 0.1$, DSMA and WALE of a resolution of 64^3 cells for both STA and REG SRT. The DNS of 512^3 resolution with REG SRT has been added as a reference.

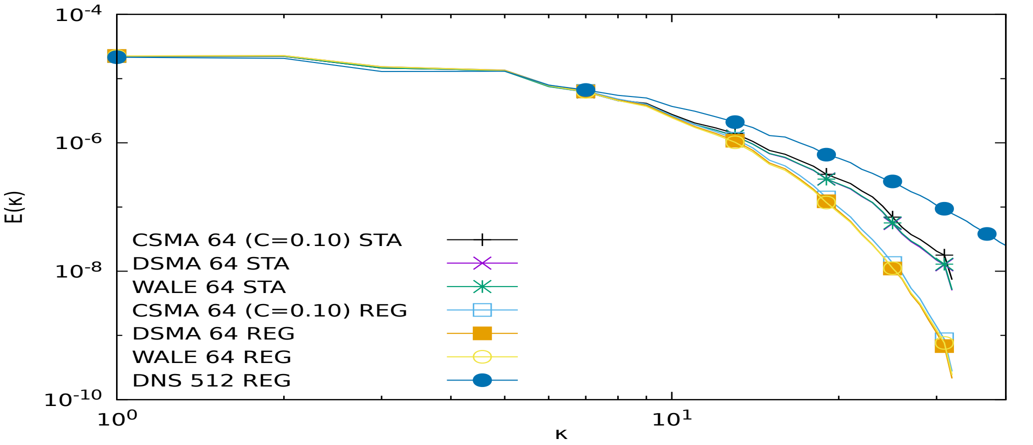


Figure 5.40: Instantaneous energy spectra of CSMA ($C = 0.1$), DSMA and WALE for both STA and REG SRT for the resolution of 64^3 cells at $t = 98.17$ time units. The curve of the REG DNS on 512^3 cells are shown as a reference.

N128 Resolution

Figures 5.42, 5.43 and 5.44 show the turbulent kinetic energy k, dissipation rate ε and Kolmogorov length scale η, respectively, for the resolution $N128$. It is eminent that all the curves collapsed with the reference data. There is a small discrepancy in the initial stage of the dissipation rate with LES to slightly under-predict the values. This is because more small eddies were simulated by the finer DNS, thus increasing the value of ε. Moreover, as in the $N64$, there is no difference between STA and REG versions.

Finally, Figs. 5.45 and 5.46 report the instantaneous energy and pressure fluctuation spectra for this resolution. In the case of the energy spectra, the LES curves converge closer to the reference DNS. Once more, the STA versions have predicted an apparent less dissipative behaviour. However, examining the pressure fluctuation spectra, the curves

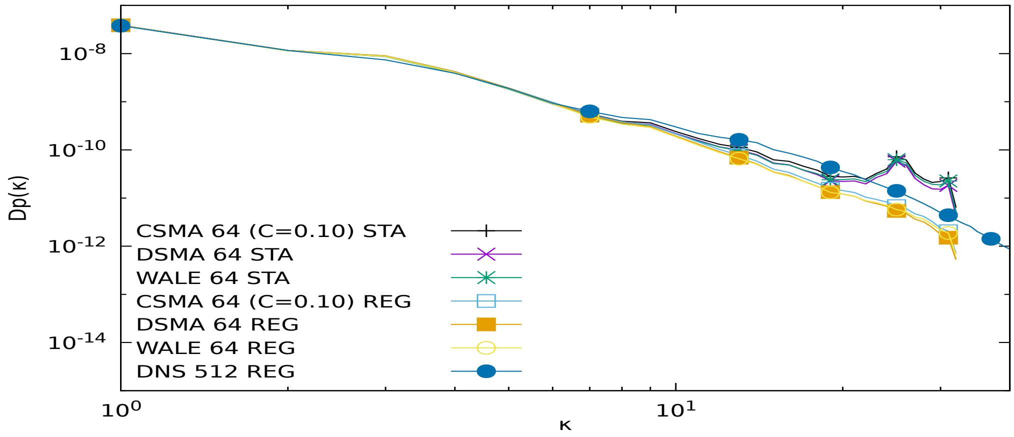


Figure 5.41: Instantaneous pressure fluctuation spectra of CSMA ($C = 0.1$), DSMA and WALE for both STA and REG SRT for the resolution of 64^3 cells at $t = 98.17$ time units. The REG DNS on 512^3 cells are shown as a reference.

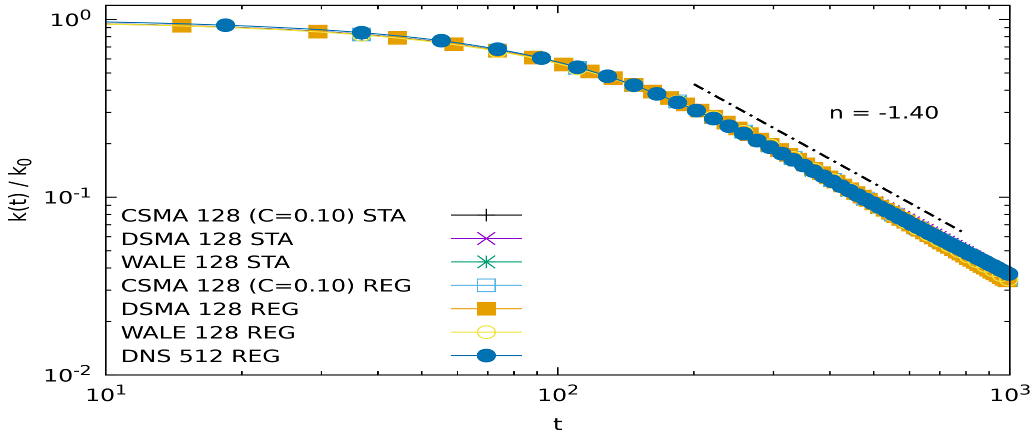


Figure 5.42: Evolution of the turbulent kinetic energy k for CSMA with $C = 0.1$, DSMA and WALE of a resolution of 128^3 cells for both STA and REG SRT. The DNS of 512^3 resolution with REG SRT has been added as a reference.

of both STA and REG versions collapse with the reference except for the instabilities in high wavenumbers for STA LES. Consequently, in this case, it is much clearer that the less dissipative behaviour of STA appearing in the energy spectra is the outcome of the instabilities rather than an attribute of the method.

Currently, I have demonstrated that the case of DHIT enabled us to perform a grid convergence analysis for the three LES models with both collision schemes. I have shown that by increasing the resolution, all LES models can converge to the reference data. Observing the energy spectra, the STA LES has returned more energetic small scales and closer to the reference independently of the resolution. This observation has been recently also reported by Nathen et al. (2018). Nonetheless, examining the pressure fluctuation spectra, I have noticed that STA versions exaggerate the high wavenumbers, introducing instabilities and thus polluting the solution. This behaviour is the outcome

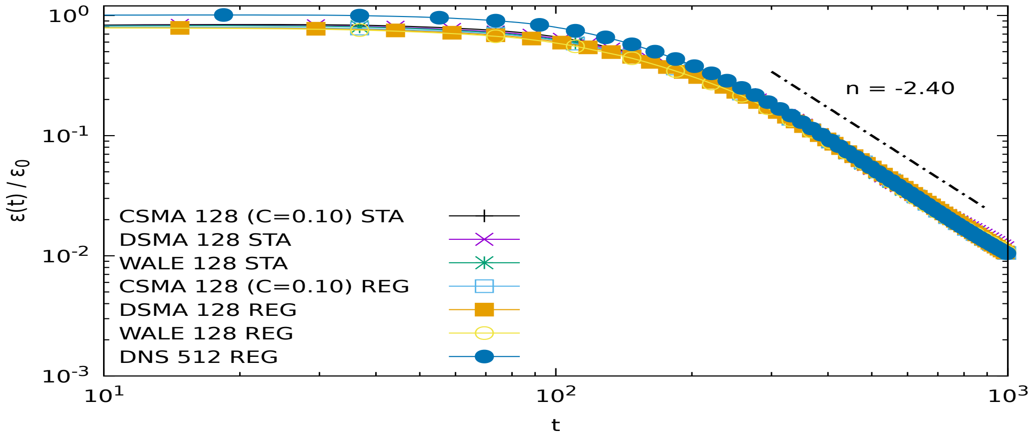


Figure 5.43: Evolution of dissipation rate ε for CSMA with $C = 0.1$, DSMA and WALE of a resolution of 128^3 cells for both STA and REG SRT. The DNS of 512^3 resolution with REG SRT has been added as a reference.

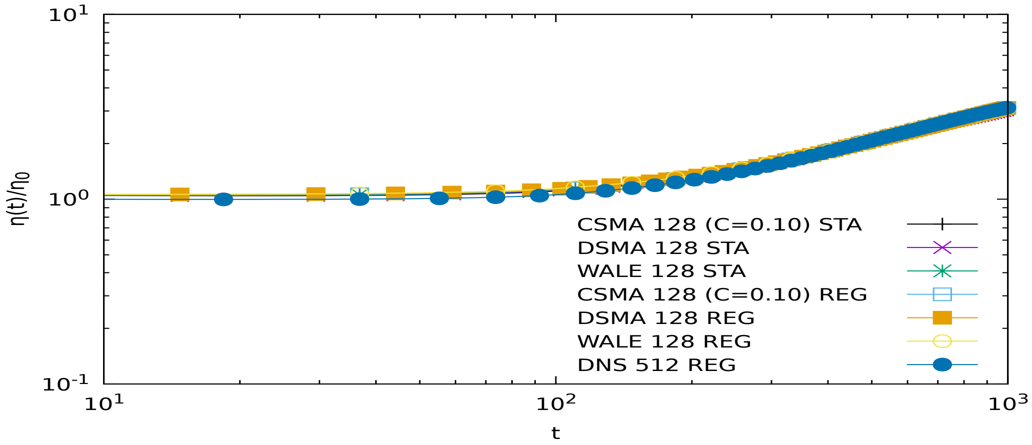


Figure 5.44: Evolution of Kolmogorov length scale η for CSMA with $C = 0.1$, DSMA and WALE of a resolution of 128^3 cells for both STA and REG SRT. The DNS of 512^3 resolution with REG SRT has been added as a reference.

of numerical error arising during the collision step. Therefore, one can conclude that the STA model's apparent better performance in the energy spectra is based on the appearance of more small scales rather than a less dissipative behaviour.

Independently of the resolution, the curves of all REG versions match each other. On the other hand, STA WALE has calculated a lower level of instabilities in the lowest resolution case. This is not the case for the other resolutions. This observation indicates that LES models may be affected and have different behaviour under different collision models.

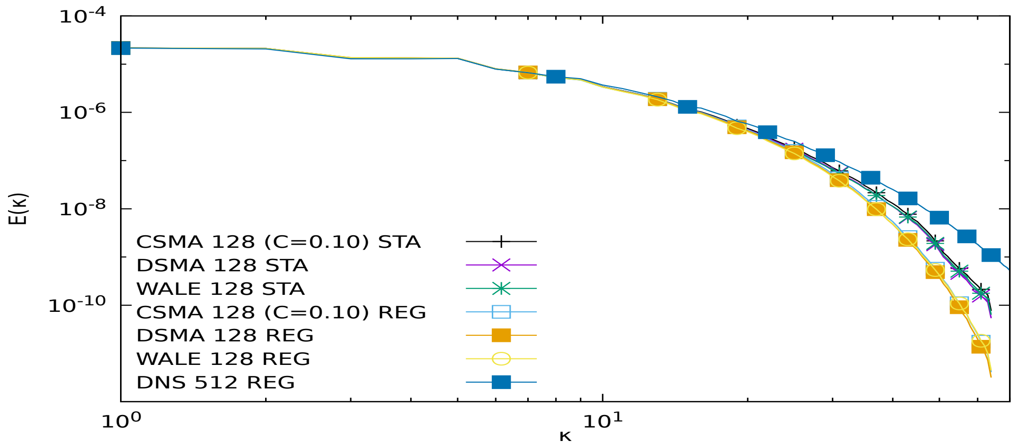


Figure 5.45: Instantaneous energy spectra of CSMA ($C = 0.1$), DSMA and WALE for both STA and REG SRT for the resolution of 128^3 cells at $t = 98.17$ time units. The curve of the REG DNS on 512^3 cells are shown as a reference.

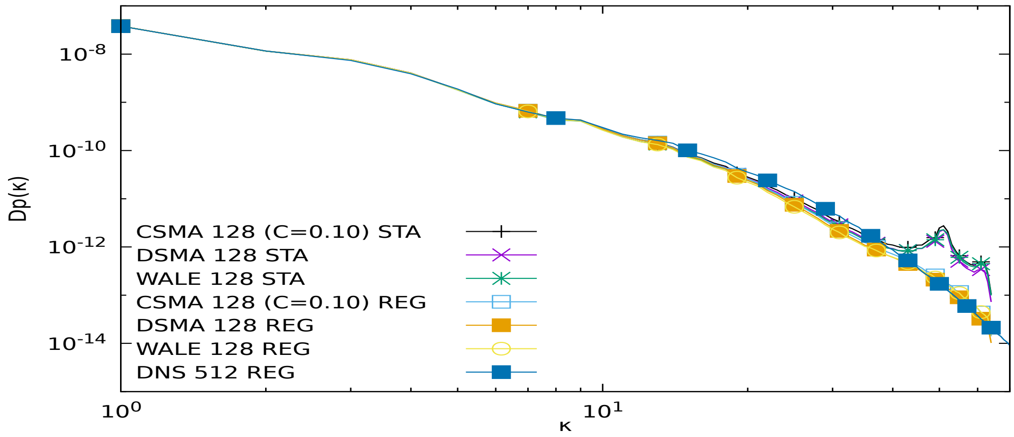


Figure 5.46: Instantaneous pressure fluctuation spectra of CSMA ($C = 0.1$), DSMA and WALE for both STA and REG SRT for the resolution of 128^3 cells at $t = 98.17$ time units. The REG DNS on 512^3 cells are shown as a reference.

Comparison of local vs stencil (STE) CSMA

In the previous section, I have pointed that the collision model may affect the performance of the LES model. To further investigate this observation, I have decided to examine the effect of estimating the eddy viscosity locally per cell or through a stencil. However, the only LES model that permits us to check the difference is CSMA. This is possible because the strain rate can be estimated either locally through the non-equilibrium part or using FD stencil, see Sec. 3.2. On the other hand, the WALE model needs to calculate the rotation rate, which cannot be estimated locally, while DSMA needs information from neighbour cells.

Initially, I will focus on the STA collision model. Figure 5.47 shows the evolution of the turbulent kinetic energy for three resolutions. Again, DNS REG of $N512$ is used as the

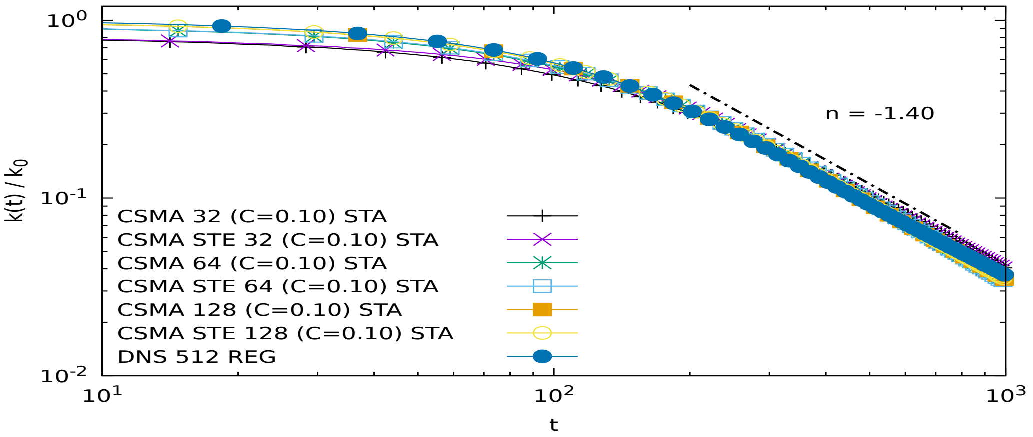


Figure 5.47: Evolution of the turbulent kinetic energy k for CSMA with $C = 0.1$ STA SRT for both local and stencil-based estimation of strain rate. The DNS of 512^3 resolution with REG SRT has been added as a reference.

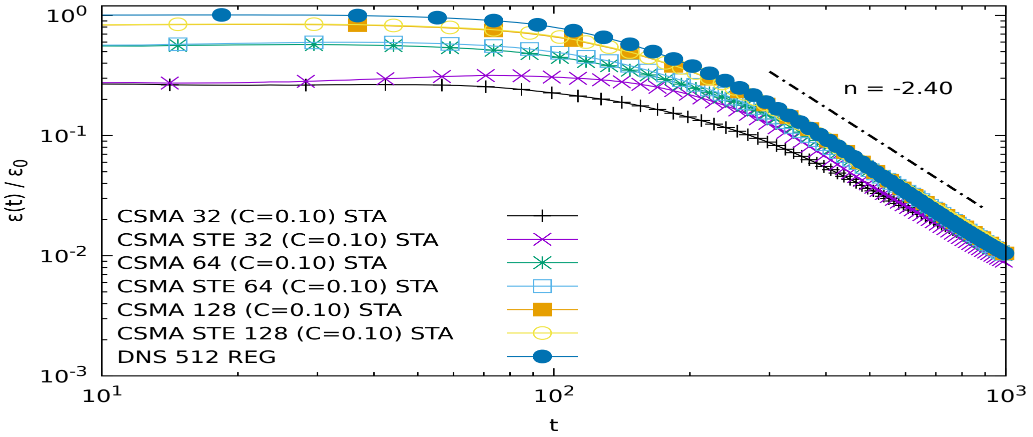


Figure 5.48: Evolution of dissipation rate ε for CSMA with $C = 0.1$ STA SRT for both local and stencil-based estimation of strain rate. The DNS of 512^3 resolution with REG SRT has been added as a reference.

reference data. It is evident that as the resolution increases, the results converge to the benchmark, as is expected. There is no apparent discrepancy between local and STE CSMA. In Fig. 5.48 one can see the evolution of the dissipation rate. In this case, STE CSMA has estimated higher values for almost all the time in the coarsest resolution. As the resolution increases, this deviation diminishes with the higher resolution showing no evident difference. The Kolmogorov length scale evolution in Fig. 5.49 reveals no distinction between the two options in question.

Figure 5.50 plots the energy spectra. The coarser the grid, the bigger the difference between local and STE CSMA to predict more energetic small scales. However, observing the pressure fluctuation spectra Fig. 5.51, one can conclude that again, the reason for the apparent better performance is the exaggeration of the instabilities in high wavenumbers.

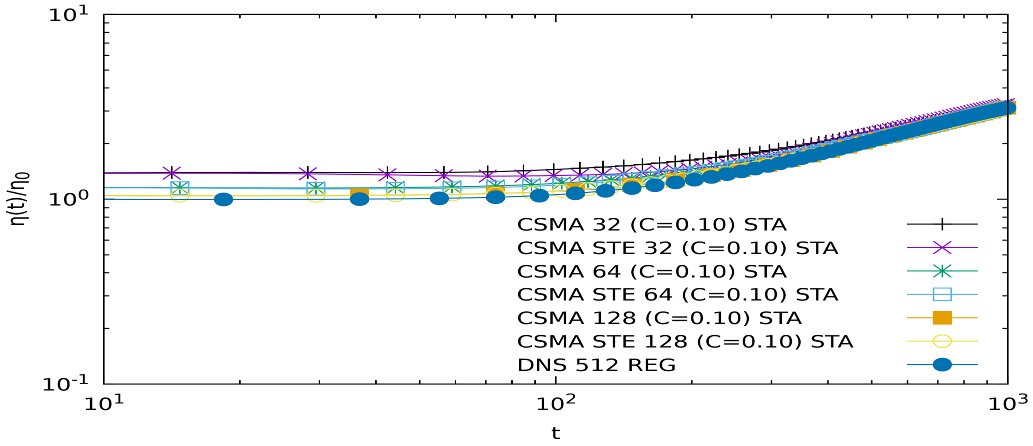


Figure 5.49: Evolution of Kolmogorov length scale η for CSMA with $C = 0.1$ STA SRT for both local and stencil-based estimation of strain rate. The DNS of 512^3 resolution with REG SRT has been added as a reference.

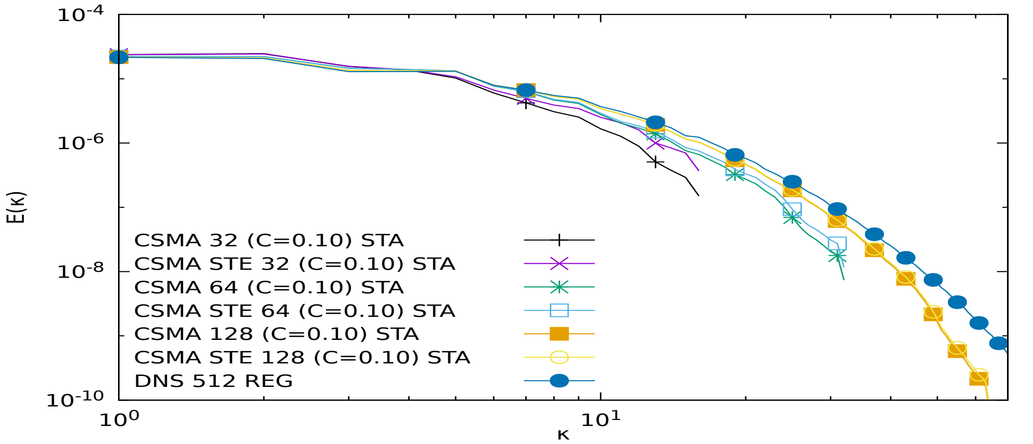


Figure 5.50: Instantaneous energy spectra of CSMA ($C = 0.1$) STA SRT for both local and stencil-based estimation of strain rate. The curve of the REG DNS on 512^3 cells are shown as a reference.

As the resolution increases, this discrepancy between the two options getting smaller and smaller, and there is no evident distinction in the finest resolution.

To investigate if the last observation is also experienced by the REG model, Figs. 5.52, 5.53 and 5.54 present the evolution of turbulent kinetic energy, dissipation rate and Kolmogorov length scale for the case of the REG model. In all three figures, the local and STE CSMA curves collapse, indicating that the REG model is not affected by the way the strain rate is estimated. This conclusion is also supported by the energy and pressure fluctuation spectra appearing in Figs. 5.55 and 5.56, respectively. It is evident that in these plots, there are no differences between local and STE CSMA.

Finally, I will close this section by presenting contours of vorticity magnitude $|\omega|$ and how the different models affect the outcome. In Fig. 5.57, one can see the instantaneous flowfield at $t = 98.17$ time units for the case of CSMA and resolution $N32$. It is evident

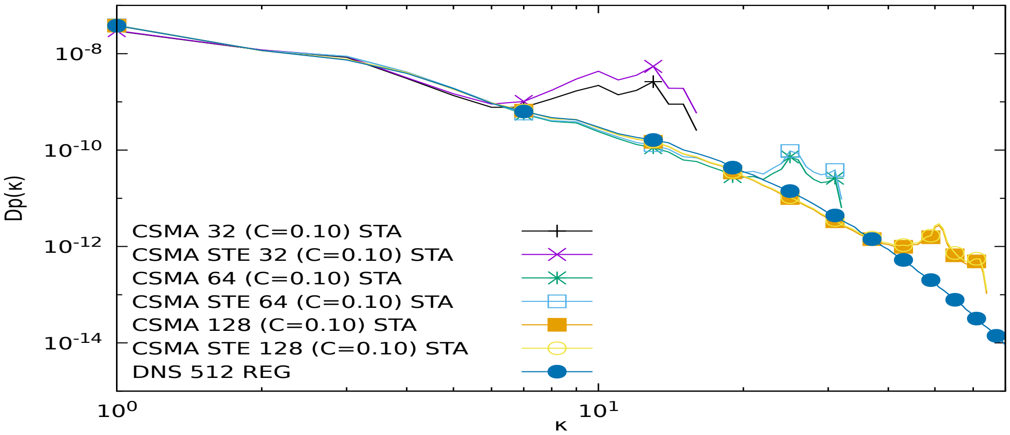


Figure 5.51: Instantaneous pressure fluctuation spectra of CSMA ($C = 0.1$) STA SRT for both local and stencil-based estimation of strain rate. The REG DNS on 512^3 cells are shown as a reference.

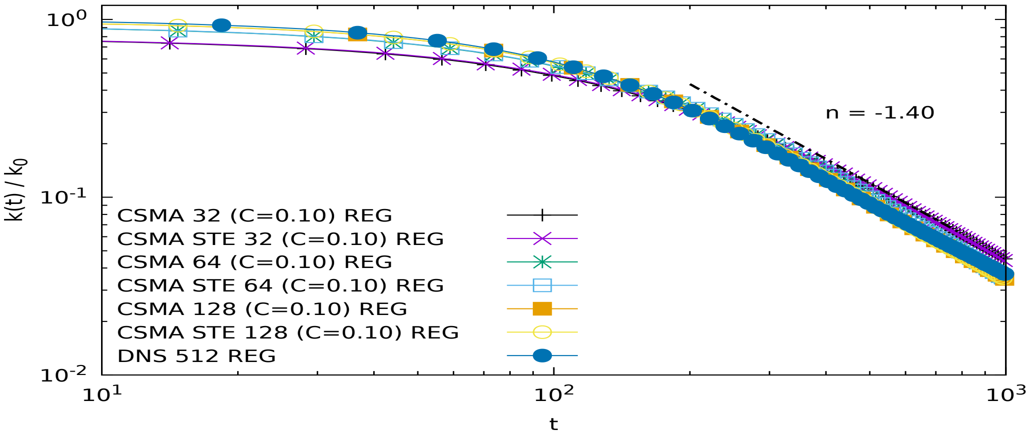


Figure 5.52: Evolution of the turbulent kinetic energy k for CSMA with $C = 0.1$ REG SRT for both local and stencil-based estimation of strain rate. The DNS of 512^3 resolution with REG SRT has been added as a reference.

that in the case of the STA model, there is a discrepancy between the local and STE CSMA, with the latter to predict significantly more eddies. On the other hand, the vorticity fields yield no obvious differences in the case of REG. Comparing STA with REG, the latter has predicted a more dissipative field. The illustration is different in Fig. 5.58 for the $N128$ resolution, where all combinations of models have returned a similar solution. The above observations are a direct confirmation of the previous results.

Figure 5.59 shows the comparison for the DSMA model. In the resolution $N32$, STA and REG have simulated considerably different vorticity fields, with the former to return a somewhat more energetic. As for the other resolution, there is no eminent distinction in the case of $N128$. The same pattern is also observed by the WALE model in Fig. 5.60.

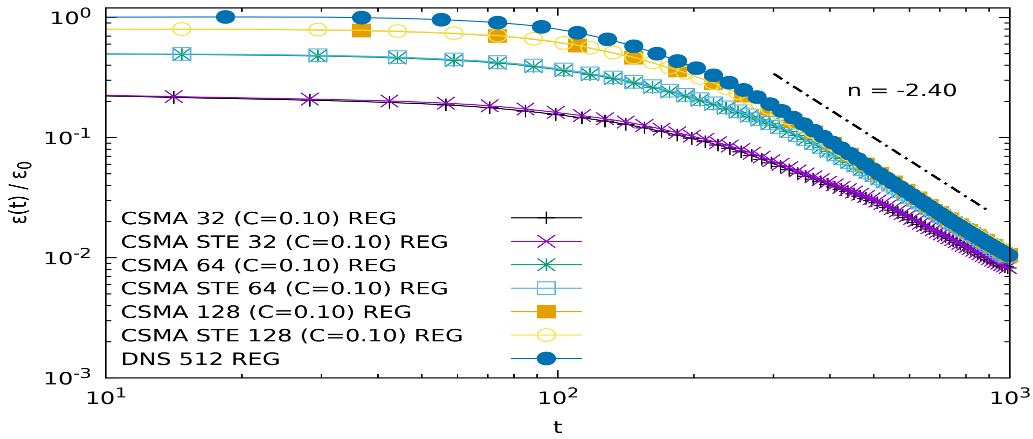


Figure 5.53: Evolution of dissipation rate ε for CSMA with $C = 0.1$ REG SRT for both local and stencil-based estimation of strain rate. The DNS of 512^3 resolution with REG SRT has been added as a reference.

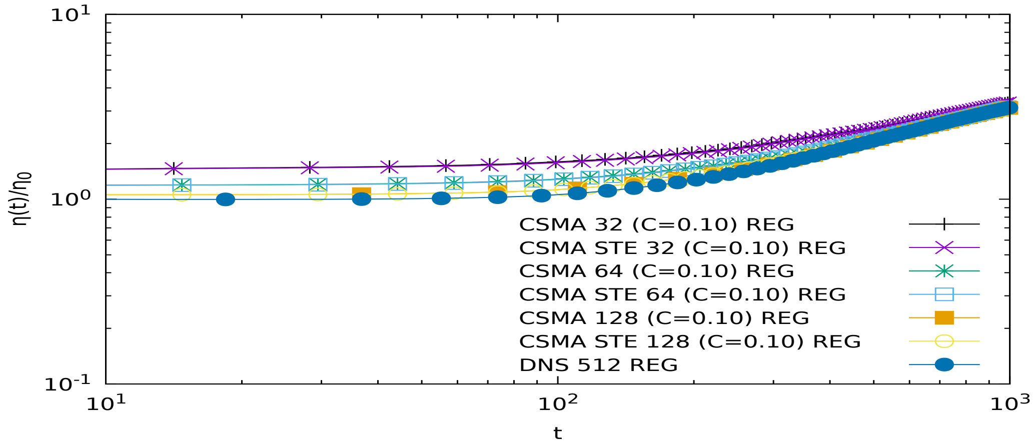


Figure 5.54: Evolution of Kolmogorov length scale η for CSMA with $C = 0.1$ REG SRT for both local and stencil-based estimation of strain rate. The DNS of 512^3 resolution with REG SRT has been added as a reference.

By evaluating the turbulence models' performance, all three models were able to accurately capture the majority of the eddies appearing in the reference DNS data for the high resolution. However, there is a significant divergence between LES and DNS and LES for the low resolution, and no clear outcome can be concluded.

5.4 Taylor Green Vortex

The TGV case was chosen as the final wall-free benchmark for the newly implemented LES models. Again the domain was a periodic cube with a length equal to $2\pi L$, where $L = 1$. There is no external force, while the initial conditions read

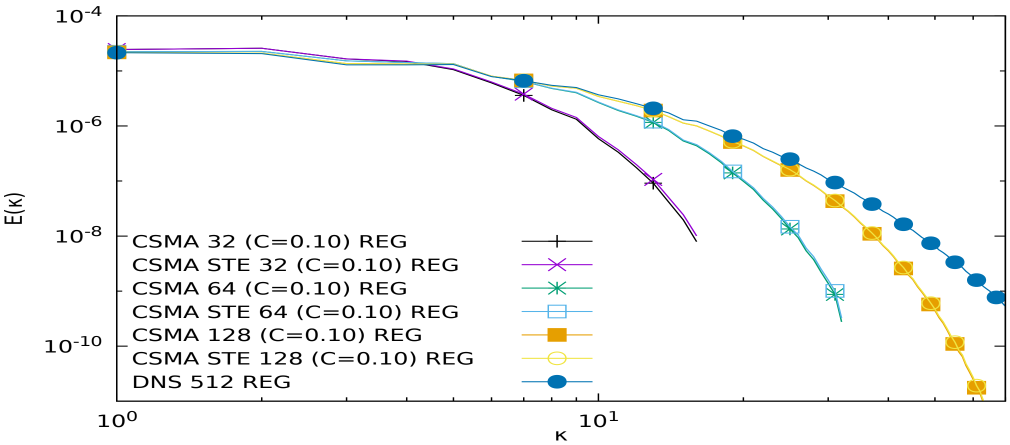


Figure 5.55: Instantaneous energy spectra of CSMA ($C = 0.1$) REG SRT for both local and stencil-based estimation of strain rate. The curve of the REG DNS on 512^3 cells are shown as a reference.

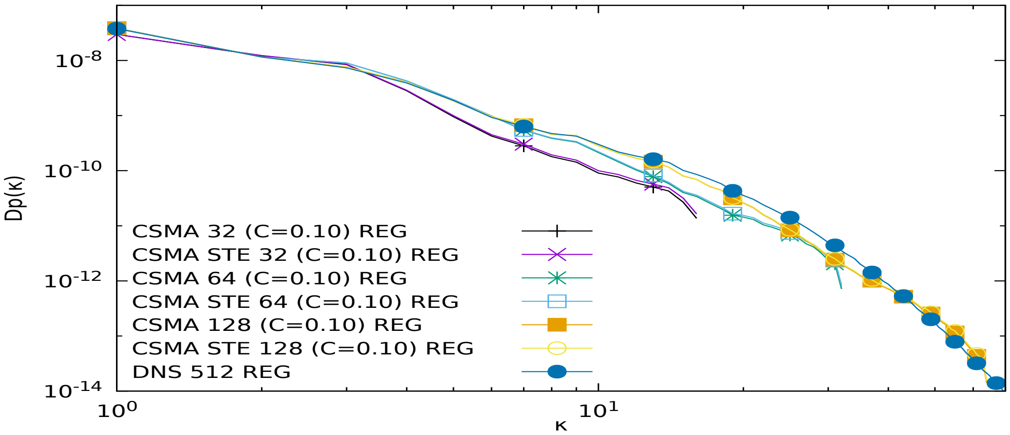


Figure 5.56: Instantaneous pressure fluctuation spectra of CSMA ($C = 0.1$) REG SRT for both local and stencil-based estimation of strain rate. The REG DNS on 512^3 cells are shown as a reference.

$$\begin{aligned}
\rho(\mathbf{x}, t_0) = \rho_0 + \frac{\rho_0 U_0^2}{16 c_s^2}\left[\cos\left(\frac{2x}{L}\right) + \cos\left(\frac{2y}{L}\right)\right]\left[\cos\left(\frac{2z}{L}\right) + 2\right],\\
u(\mathbf{x}, t_0) = U_0 \sin\left(\frac{x}{L}\right)\cos\left(\frac{y}{L}\right)\cos\left(\frac{z}{L}\right),\\
v(\mathbf{x}, t_0) = -U_0 \cos\left(\frac{x}{L}\right)\sin\left(\frac{y}{L}\right)\cos\left(\frac{z}{L}\right),\\
w(\mathbf{x}, t_0) = 0.
\end{aligned} \tag{5.36}$$

In the above, $\rho_0 = 1$ and $U_0 = 0.1$ leading to $\mathrm{Ma} \approx 0.17$ at the start of the simulation. To achieve $\mathrm{Re} = U_0 L/\nu = 1600$, the viscosity ν was set to $6.25 \cdot 10^{-5}$. By having estimated the macrovariables locally per cell, the microvariables were calculated by applying the equilibrium function Eq. (2.11). All the simulations ran for twenty-time units.

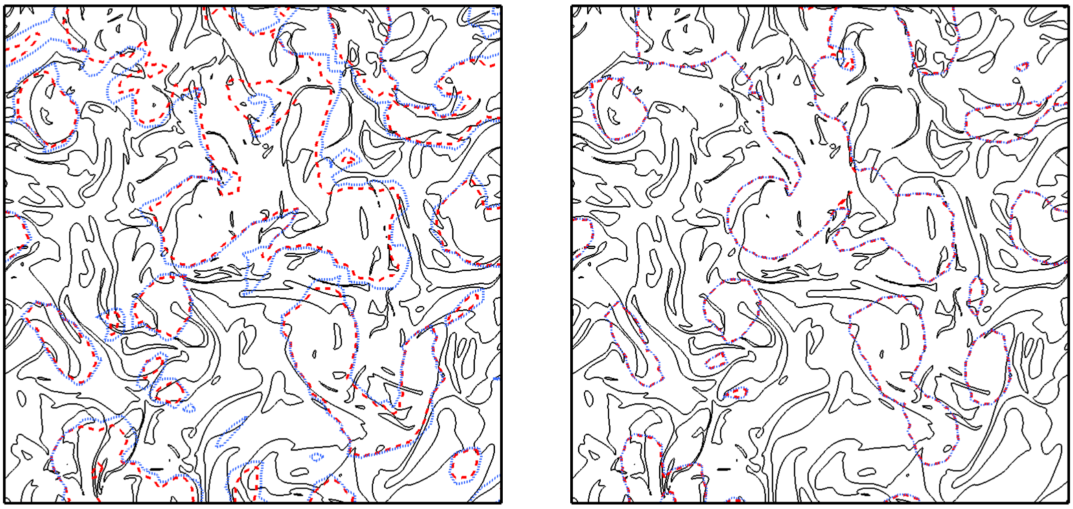


Figure 5.57: Contours of vorticity magnitude ($|\omega| = 0.05$) at $t = 98.17$ time units on 32^3 cells, for CSMA with $C = 0.1$ STA (left) and REG (right). The red dashed line is the version with localised estimation of the eddy viscosity while the blue dotted the one based on stencil. The black solid line is the DNS with a resolution of 512^3 given as a reference.

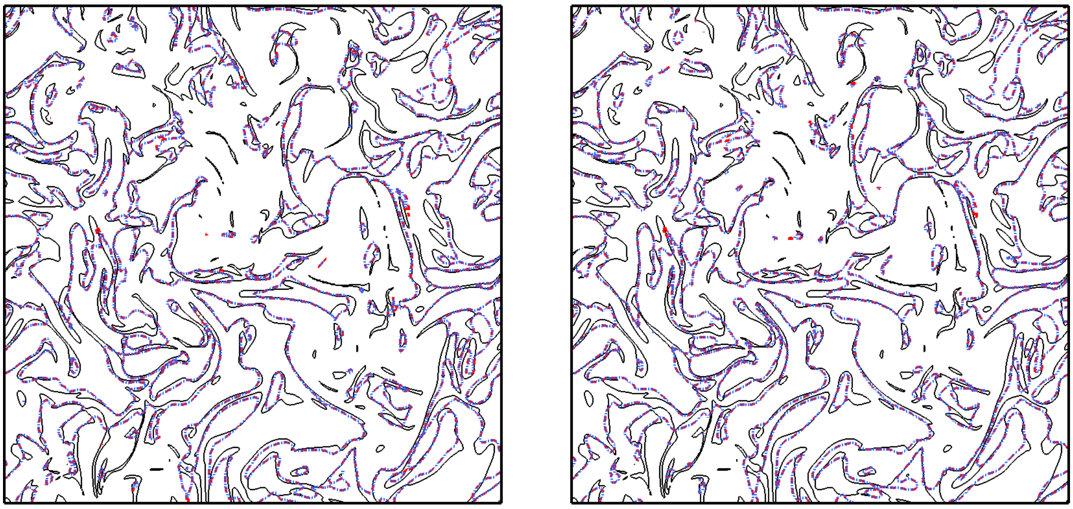


Figure 5.58: Contours of vorticity magnitude ($|\omega| = 0.05$) at $t = 98.17$ time units on 128^3 cells, for CSMA with $C = 0.1$ STA (left) and REG (right). The red dashed line is the version with localised estimation of the eddy viscosity while the blue dotted the one based on stencil. The black solid line is the DNS with a resolution of 512^3 given as a reference.

It is essential to mention that the non-equilibrium part of the distribution function was set to zero. This decision has added two effects that need to be taken into consideration in the following discussion. The first is the addition of a second order numerical error in the initial field. It is expected that as time passes, this error will be dissipated. Moreover, LES models, such as CSMA and DSMA, that base their calculations on the non-equilibrium part will see initially a zero eddy viscosity field that gradually will rise to more reasonable values. These effects will be diminished by increasing the resolution.

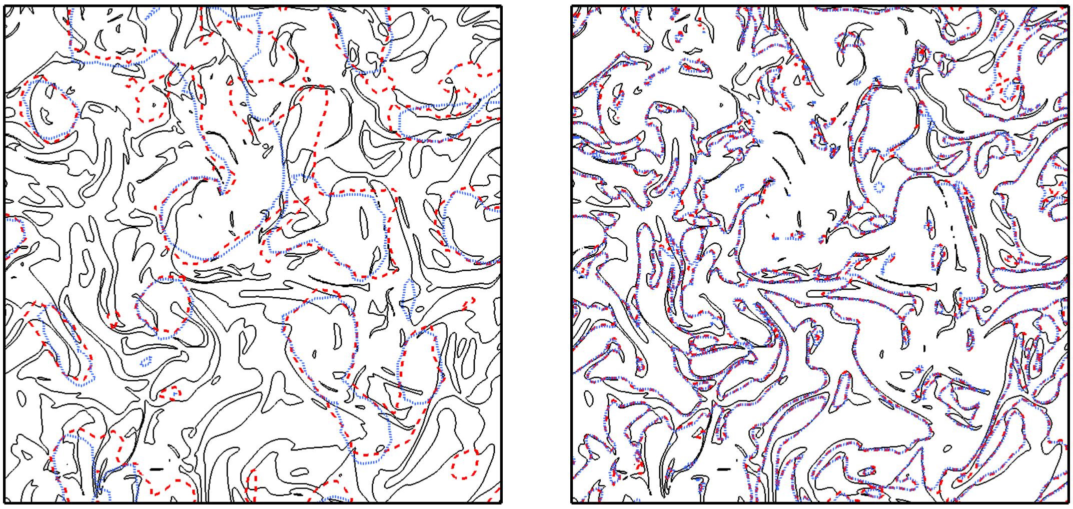


Figure 5.59: Contours of vorticity magnitude ($|\omega| = 0.05$) at $t = 98.17$ time units for DSMA on 32^3 (left) and 128^3 (right) cells. The red dashed line is the STA model and the blue dotted the REG. The black solid line is the DNS with a resolution of 512^3 given as a reference.

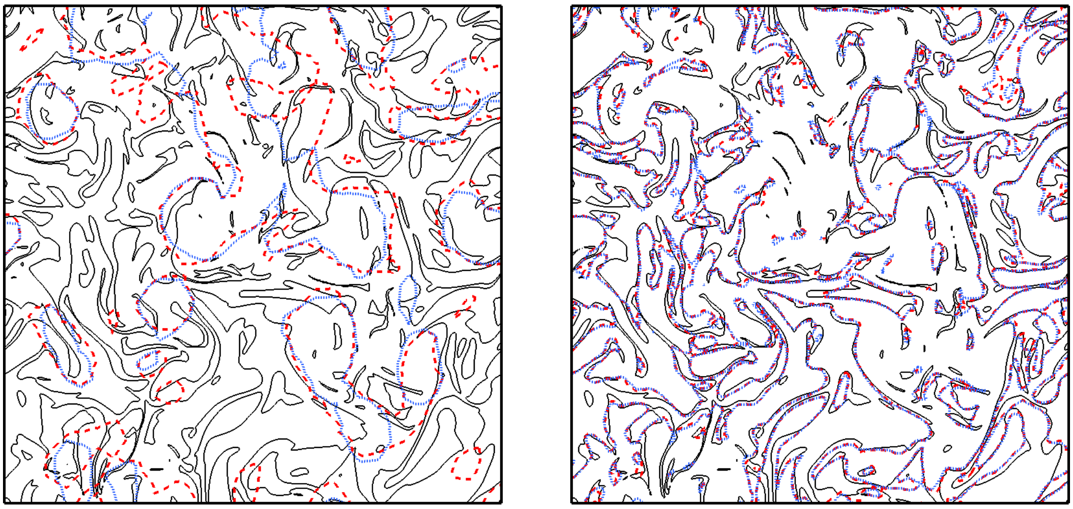


Figure 5.60: Contours of vorticity magnitude ($|\omega| = 0.05$) at $t = 98.17$ time units for WALE on 32^3 (left) and 128^3 (right) cells. The red dashed line is the STA model and the blue dotted the REG. The black solid line is the DNS with a resolution of 512^3 given as a reference.

Due to the imposed initial conditions, large scale vortices appear in the flowfield. This phase is characterised by the absence of small scales, and thus turbulence. As the time passes, the vortices will start to break into smaller eddies, which finally will be dissipated. Therefore, this test case has an initial inviscid part followed by the transition to turbulence and finishes with turbulence decay. The above procedure imposes new challenges for the turbulence models.

Following the same path as in FHIT, I will begin the discussion comparing four DNS resolutions against the reference data of DeBonis (2013), which is an NS DNS with

$N512$ resolution based on a 13-point dispersion-relation-preserving scheme after Bogey & Bailley. Figure 5.61 shows the evolution of kinetic energy for the four resolutions of the STA model. Only the highest resolution can match the reference for the whole time. The other two resolutions, $N128$ and $N256$ follow the reference solution for the first ten time units, but then they express a steeper slope at the beginning of the last phase, i.e. the decay of turbulence. Finally, the simulation with the lowest resolution has crashed almost in the beginning.

For a more comprehensive discussion, I will also report the evolution of two dissipation rates. The first one, plotted with a solid line, is the temporal derivative of the kinetic energy ε, while the other, plotted with a dashed line, is the dissipation rate ε_ζ calculated based on enstrophy ζ Eq. (5.37).

$$\varepsilon_\zeta = 2\nu\zeta, \quad \text{where} \quad \zeta = \frac{1}{\rho_0\Omega}\int_\Omega \rho\frac{|\omega|^2}{2}d\Omega. \tag{5.37}$$

The former describes the total dissipation while the latter is associated with the small scales, and thus it is related to turbulence. Furthermore, the difference between these two variables links to dissipation related to the numerical scheme.

Figure 5.62 reports the two dissipation rates for the case of STA DNS. Only the total dissipation of the highest resolution fits well with the reference, though it has slightly under-predicted the peak and the values at the end of the simulation. The curves of ε_ζ coincide with the total dissipation almost all the time except for around the peak, pointing to a small contribution of numerical dissipation. The total dissipation of the $N256$ DNS follows closely, but it begins to diverge after the peak, at around $t^* = 9$. Due to lower resolution, the presence of numerical dissipation is more evident in this case. However, its most significant contribution appears in the case of $N128$ DNS, though it was not sufficient to correctly capture the shape of the curve, particularly around the peak. Moreover, the steepest slope after the second peak indicates a much higher pace of attenuation of small scales compared to the previous two resolutions. The results express the expected convergence behaviour and the reduction of the numerical dissipation as the resolution increases.

At this point, the discussion will focus on the REG model. Figure 5.63 shows the evolution of the kinetic energy for the same four resolutions for REG DNS. Compared with the STA versions, there is no evident discrepancy among the three finer resolutions, except for the curve of $N128$ that seems to be closer to the reference one. An interesting comment is that the REG collision model has succeeded in running the lowest resolution. However, the curve fails to follow the other data for the whole duration.

Examining the dissipation rates in Fig. 5.64, they follow the same converging pattern with higher resolution. Nonetheless, all the REG curves have underperformed compared

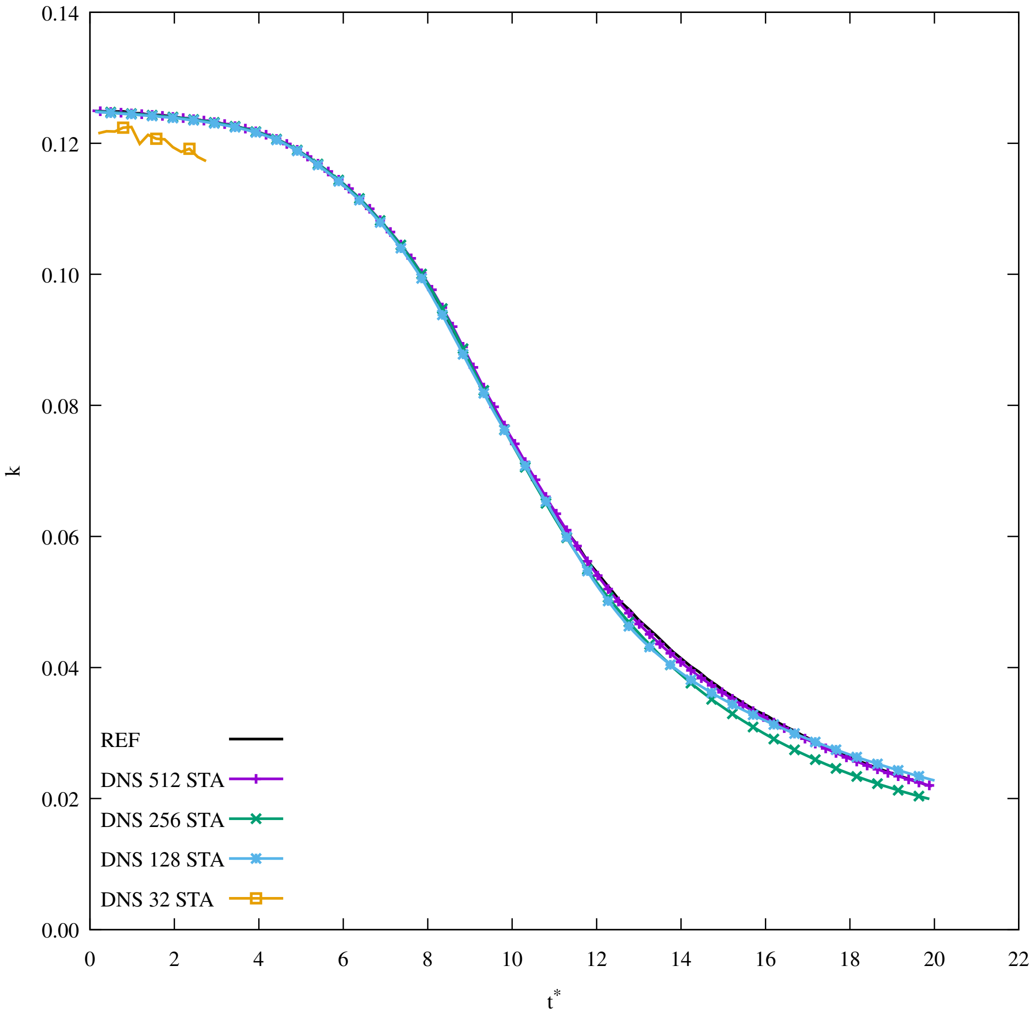


Figure 5.61: Evolution of kinetic energy for DNS STA for a variety of resolutions. The reference solution is based on (DeBonis, 2013).

to their STA counterparts. In the case of the highest resolution, it appears that REG has returned a slightly lower peak. The situation is similar for the $N256$ case though the difference is more eminent than STA. On the other hand, the lowest resolution, besides the first two time units, has failed to predict the value of the peak and its position. However, the most striking discrepancy can be observed for the $N128$ resolution. The REG curve starts to diverge from the reference data as early as around $t^* = 4$ while the STA version follows the data reasonably well up to around $t^* = 8$. This period is characterised by the break down of the large vortices into smaller and smaller, signalling the transition to turbulence. There is also difference in the peak for ε_ζ with REG returns a value around $7 \cdot 10^{-3}$ while STA $8 \cdot 10^{-3}$. This reduction indicates significant damping of the turbulent scales in the simulation of REG DNS.

To further investigate this behaviour, I also report the instantaneous energy and pressure fluctuation spectra at four different time steps around the peak in Figs. 5.65 and

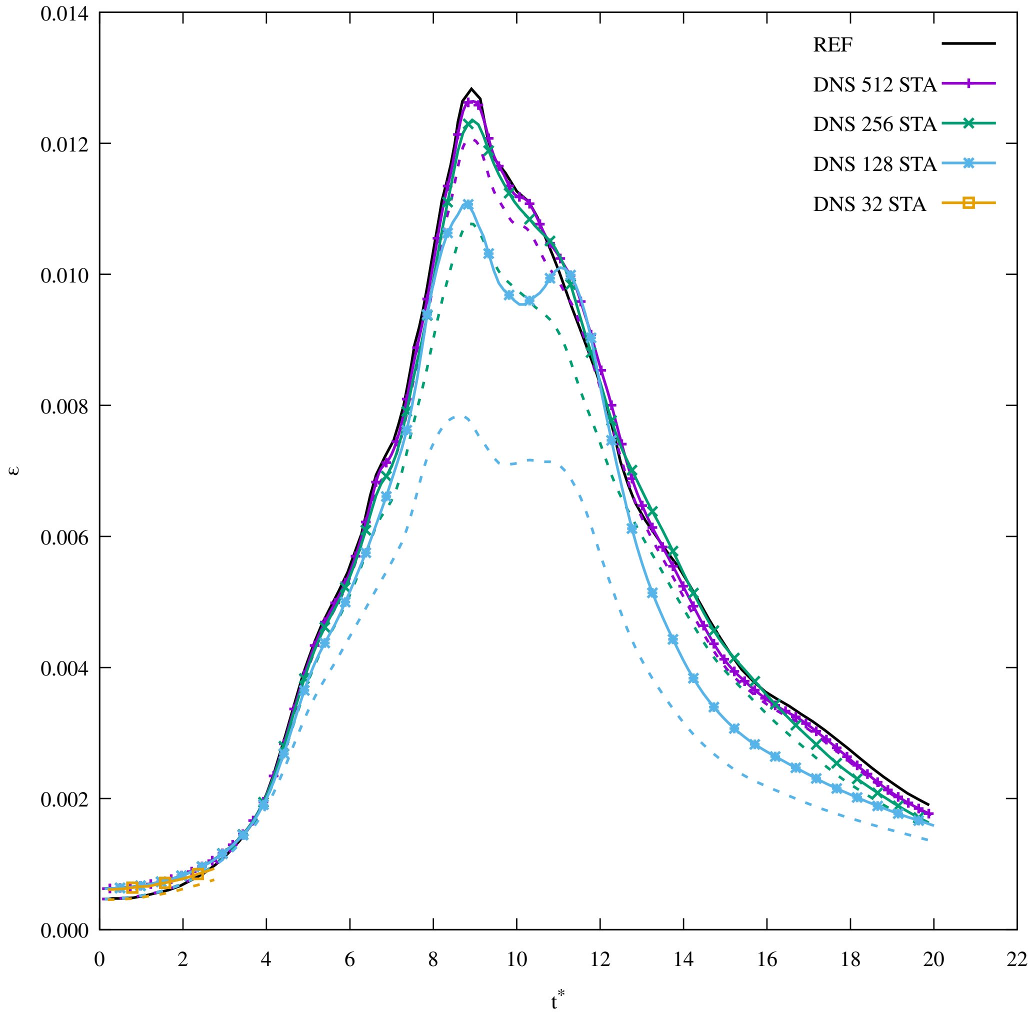


Figure 5.62: Evolution of dissipation rate (solid) and dissipation rate based on enstrophy (dashed) for DNS STA for a variety of resolutions. The reference solution is based on (DeBonis, 2013).

5.66, for both STA and REG DNS for $N128$ resolution. In the case of energy spectra, the STA versions have predicted more energetic small scales, a similar situation reporting in the previous two test cases. However, in this scenario, discrepancies can be observed over the whole spectrum of wavenumbers for the case of $t^* = 12$. Examining the pressure fluctuation spectra, similarly to the previous two sections, the STA curves suffer from numerical instabilities at the highest wavenumbers. Nevertheless, the two collision models start to diverge from each other even at the low wavenumbers as time passes. Finally, it appears that the REG simulation has damped more the small scales in the range $20 < \kappa < 40$. It seems that this region has not been affected by numerical instabilities in the case of the STA model.

By compiling all the information from the three testcases, there is a strong indication that the STA collision model is less dissipative than the REG one, although this observation is exaggerated by an artificial increase of the small scales due to inaccuracies. In

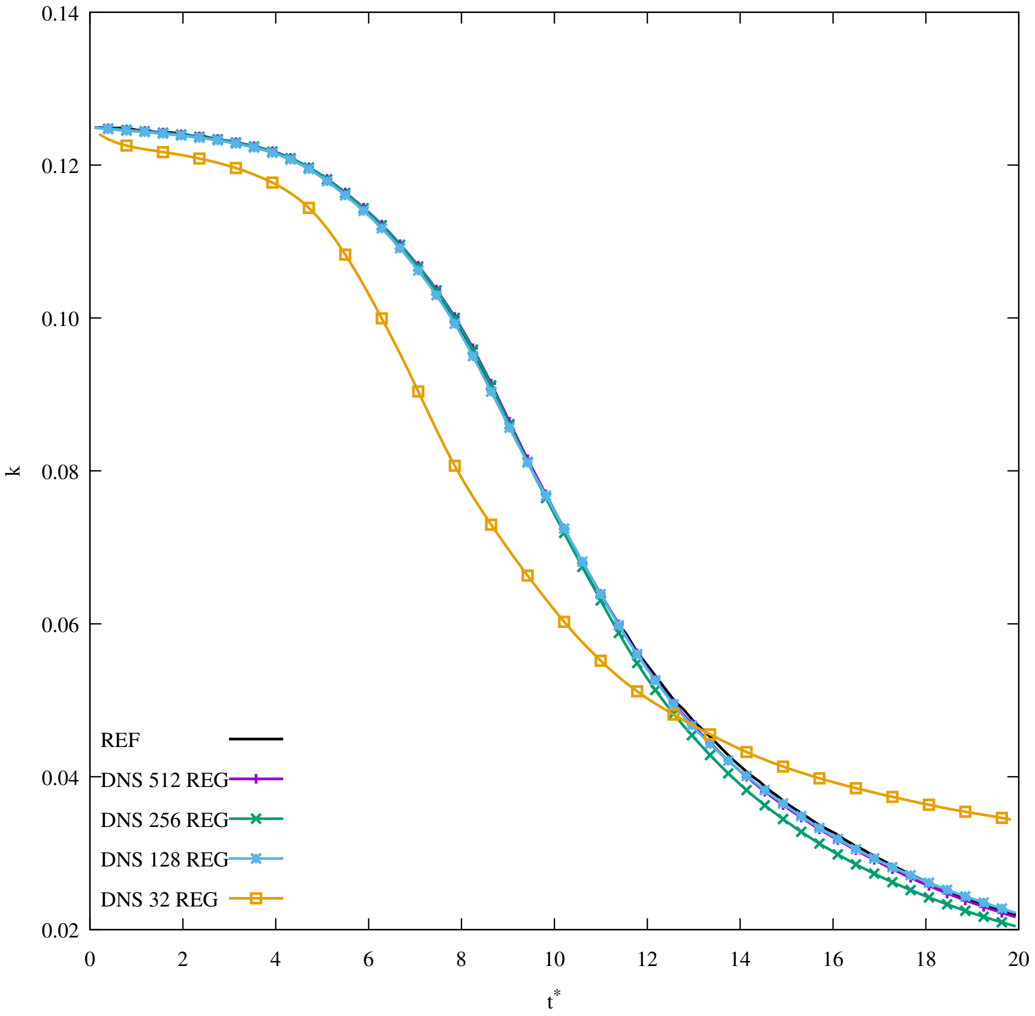


Figure 5.63: Evolution of kinetic energy for DNS REG for a variety of resolutions. The reference solution is based on (DeBonis, 2013).

a different interpretation, the regularisation procedure successfully removes the numerical instabilities in the STA model, but it tends to attenuate some of the flow's physical scales. Simultaneously, though this mechanism seems to only affect the small eddies, in a transient case, such as the break down of the large vortices in TGV, some alterations in medium and large ones may also lead to some alterations over time. The divergence between the two models amplifies with lower resolutions.

5.4.1 Resolution N 256

I will finish this section by presenting a convergence analysis for the LES models and their performance simulating the TGV test case. As mentioned before, the initial TGV phase is characterised by the absence of turbulence and the second phase by the transition. Nonetheless, there is no explicit mechanism to inform the LES models that they should turn off during these phases. Consequently, it is expected that the introduction of the

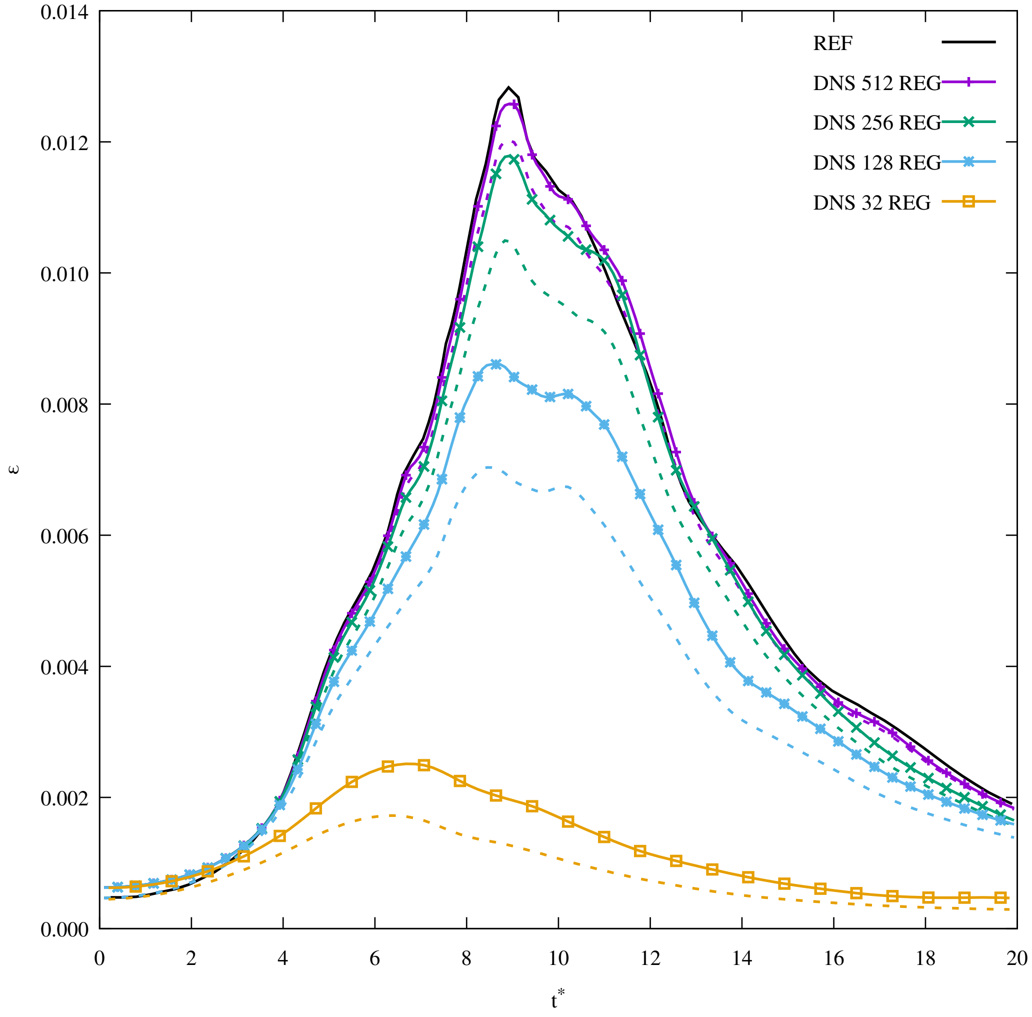


Figure 5.64: Evolution of dissipation rate (solid) and dissipation rate based on enstrophy (dashed) for DNS REG for a variety of resolutions. The reference solution is based on (DeBonis, 2013).

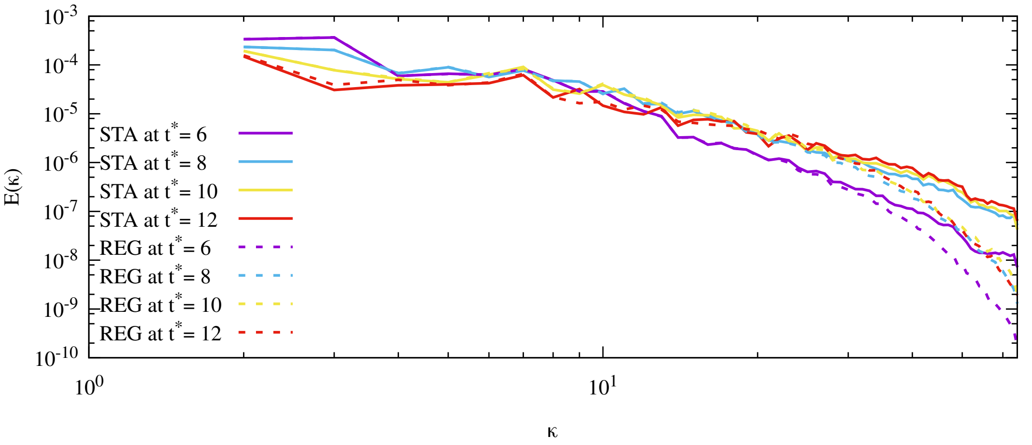


Figure 5.65: Instantaneous energy spectra of DNS at different time steps for the resolution of $N128$.

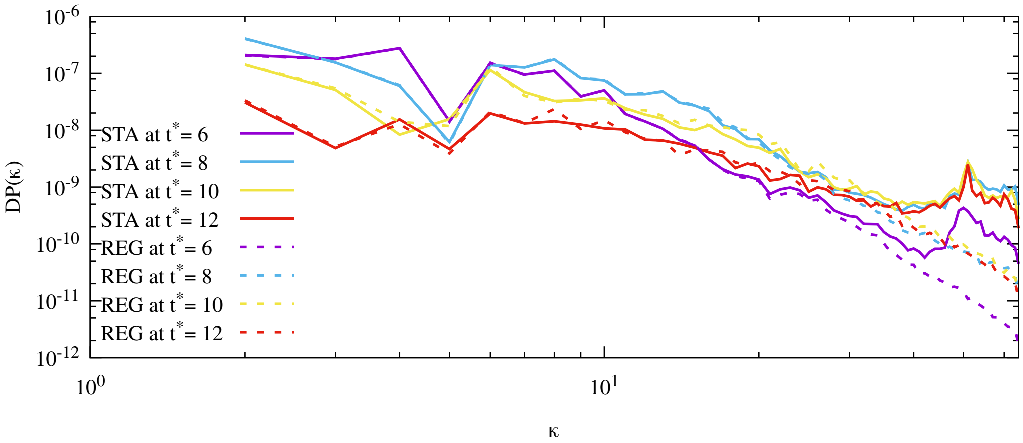


Figure 5.66: Instantaneous pressure fluctuation spectra of DNS at different time steps for the resolution of $N128$.

sub-grid scale modelling will affect, by dampening, the initial large vortices and their break down.

The highest resolution that was used for this situation was the $N256$. As reference data, I will use the DNS of the exact resolution. Figure 5.67 shows the evolution of the kinetic energy for the current resolution. All curves coincide with each other, though some divergence appears in the last phase of turbulence decay. In particular, the WALE model returns the most considerable disparity against the DNS data.

To better understand these differences, Fig. 5.68 reports the dissipation rates. First of all, this plot provides a useful direct comparison of the two DNS versions, with the STA model predicted a larger peak and thus indicating the existence of a larger number of small scales in the flow. Moreover, subgrid-scale modelling has damped the structures in the flow field, reducing the LES curves' shape. As in the case of DNS, the STA LES versions have returned higher values. The behaviour is identical for both the dissipation rates.

Analysing the performance of the LES models, it is evident that the CSMA, with $C = 0.1$, has returned minor dissipative behaviour. In theory, DSMA should perform the same or slightly better compared to CSMA given that they share the exact mechanism to calculate the eddy viscosity, but DSMA can also alter the value of the constant C locally. Nonetheless, some affect this estimation. First of all, in the current implementation, I have prevented backscattering of the turbulent energy by blocking the value of C to be negative. It has been shown that in this case, backscattering may improve results, leading to a less dissipative functioning, (DeBonis, 2013). The reason for imposing the minimum value of C at zero is robustness. Negative values of C have the potential to reduce the value of ω to such levels that may render the simulation unstable. This final outcome can worsen by the second factor: the locality of the constant calculation. Theoretically, the calculation of C should take place in homogeneous directions. Nevertheless, the

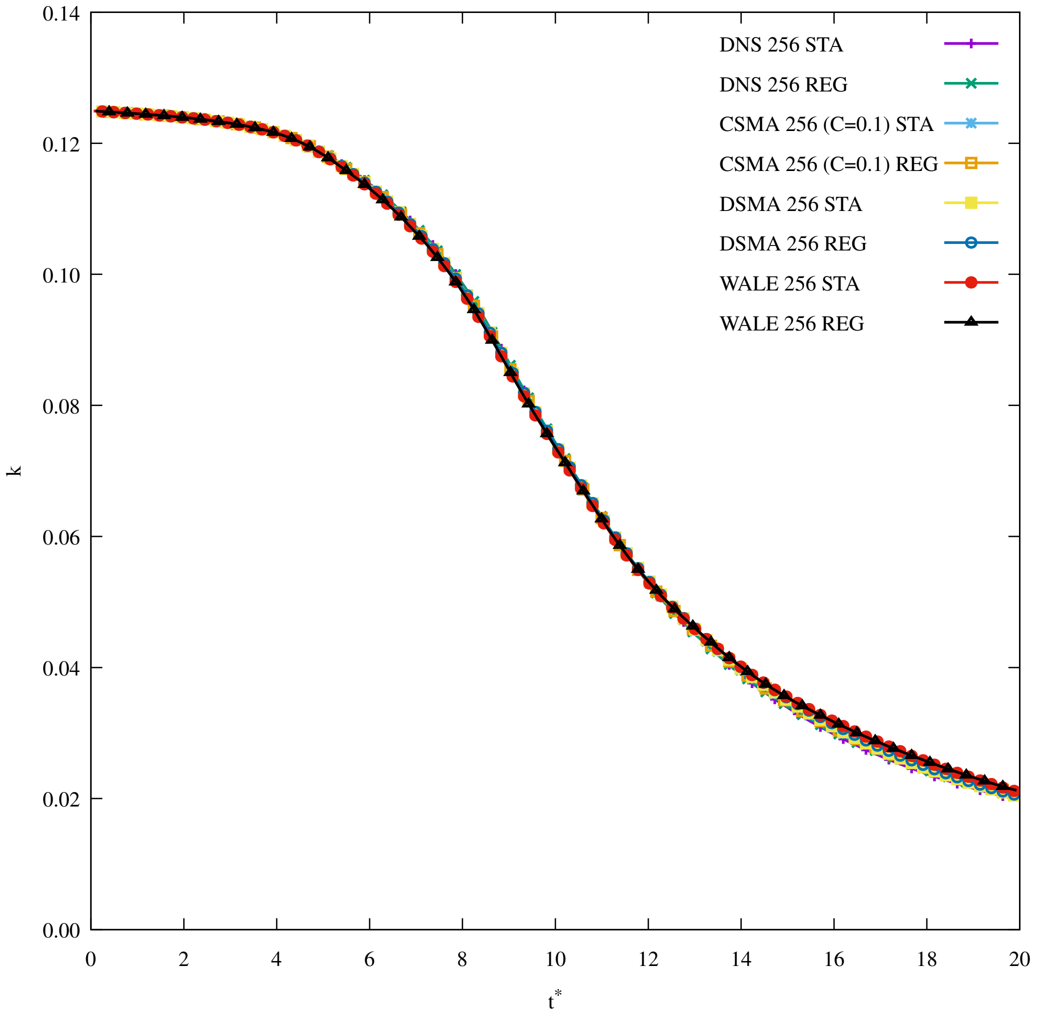


Figure 5.67: Evolution of kinetic energy for all LES models and DNS of both STA and REG versions for the resolution of $N256$.

identification of homogeneity is complicated in a general case, and thus the current implementation is based locally per cell. The third factor is the imposed macrovariables in the ghost cells which forcing us to calculate the constant using the data from the previous time step, introducing, in that way, an error analogous to Δt.

In the case of WALE, the last factor is also applicable. On the other hand, the WALE model has a different mechanism for calculating the eddy viscosity, i.e. taking into account the rotation rate. Consequently, it appears that the WALE model is more sensitive and triggers more dissipation during the second phase, that is, the transition to turbulence. Finally, none of the LES models seems to reduce better the disparity between total and enstrophy based dissipation rate. This means that none of the models has affected the numerical dissipation of the scheme for this resolution.

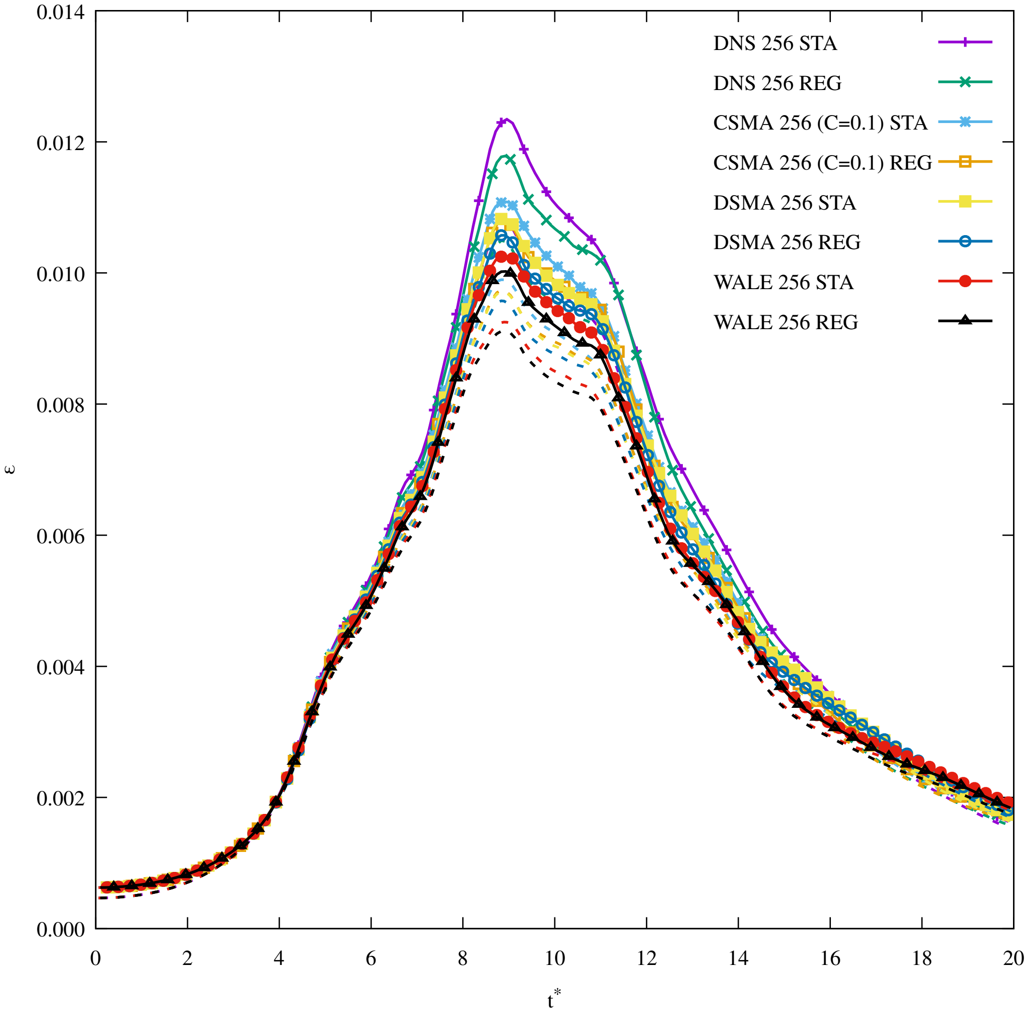


Figure 5.68: Evolution of dissipation rate (solid) and dissipation rate based on enstrophy (dashed) for all LES models and DNS of both STA and REG versions for the resolution of $N256$.

5.4.2 Resolution N 128

We will proceed with the $N128$ resolution. In Fig. 5.69, one can see the evolution of the kinetic energy. In this scenario, the divergence between DNS and LES curves is more obvious, starting after $t^{\star} = 4$. Again, the WALE model returns the largest difference from DNS. Figure 5.70 plots the dissipation rates. The discrepancy between STA and REG DNS is highlighted in this figure. The LES models have performed similarly to the highest resolution, though the disparity among them and against the DNS reference has been enlarged. Furthermore, STA versions have estimated higher numerical dissipation values as the difference between the two dissipations specifies.

At this point, I will analyse the response of the three LES models to the three different phases of the TGV case by examining the vorticity field and how the eddy viscosity adapted to it. To do so, three different times will be studied. The first instance will

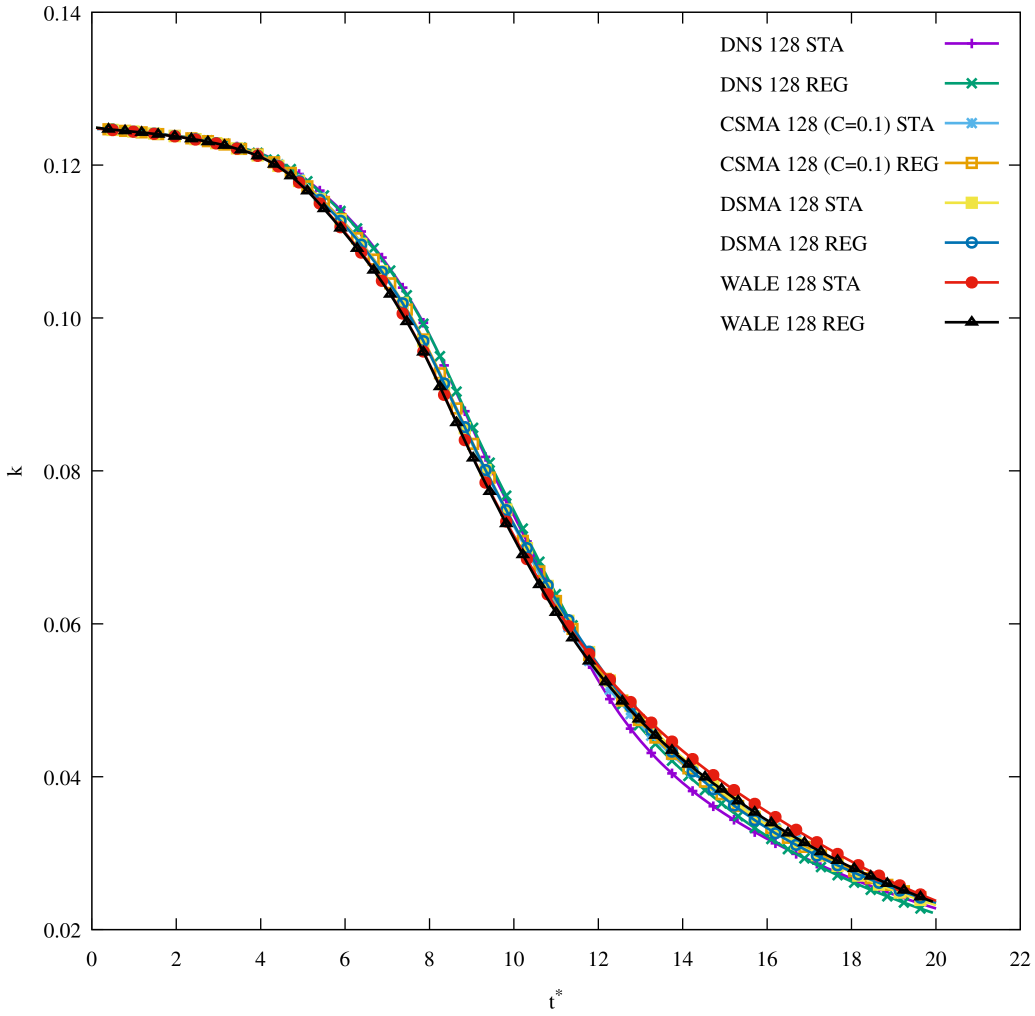


Figure 5.69: Evolution of kinetic energy for all LES models and DNS of both STA and REG versions for the resolution of $N128$.

be at $t^\star = 3$, which is in the inviscid phase, characterised by the absence of turbulence. The second time is $t^\star = 9$, which is the peak of the dissipation rate, signalling the end of the transition to turbulence. Finally, $t^\star = 15$ has been chosen as a representative of the final phase, the decay of turbulence. The last phase's performance is also illustrative of the other two test cases, FHIT and DHIT. To non-dimensionalise the results, I have divided the eddy viscosity by the value of the physical one.

Starting with CSMA, Figs. 5.71, 5.72 and 5.73 report the vorticity field for the three times respectively. Data for both collision models are given for comparison. It is evident that as time passes, smaller and smaller scales appear in the flowfield while large vortices break down. Figures 5.74, 5.75 and 5.76 show the instantaneous eddy viscosity fields corresponding to the same models and times. The eddy viscosity field is in good agreement with vorticity for all times, with the least accurate to be during the initial phase, indicating CSMA's difficulty to deal with the laminar case.

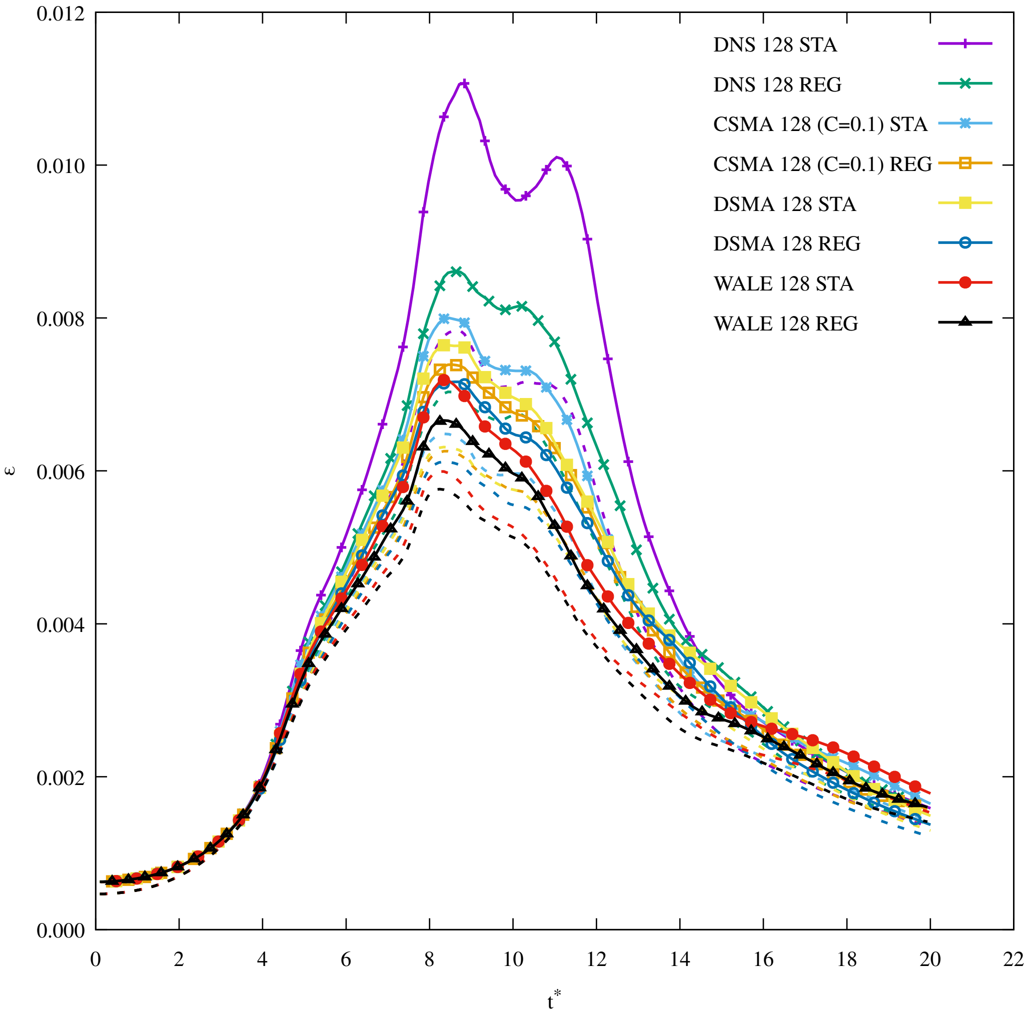


Figure 5.70: Evolution of dissipation rate (solid) and dissipation rate based on enstrophy (dashed) for all LES models and DNS of both STA and REG versions for the resolution of $N128$.

By comparing the two collision models, at $t^\star = 3$, there is no apparent discrepancy. The range between the minimum and the maximum value is two orders of magnitude. The next time, both maximum values have been increased, following the increase of vorticity, though the STA model has estimated a more significant rise. Furthermore, examining the flow structures, the REG version presents a smoother flowfield, indicating fewer small structures than the STA one. Nonetheless, some of these small scales are the result of numerical instabilities. The same differences are observed for $t^\star = 15$, though at this time, both flowfields are full of small structures pointing to a fully turbulent field.

The next model to be discussed is DSMA. Figures 5.77, 5.78 and 5.79 plot the vorticity fields. Besides the range of the values, there are no obvious discrepancies between DSMA and CSMA. On the other hand, the situation is different in the case of eddy viscosity fields reporting in Figs. 5.80, 5.81 and 5.82. Though both CSMA and DSMA share the same way to calculate the eddy viscosity, it is clear that the dynamic behaviour of

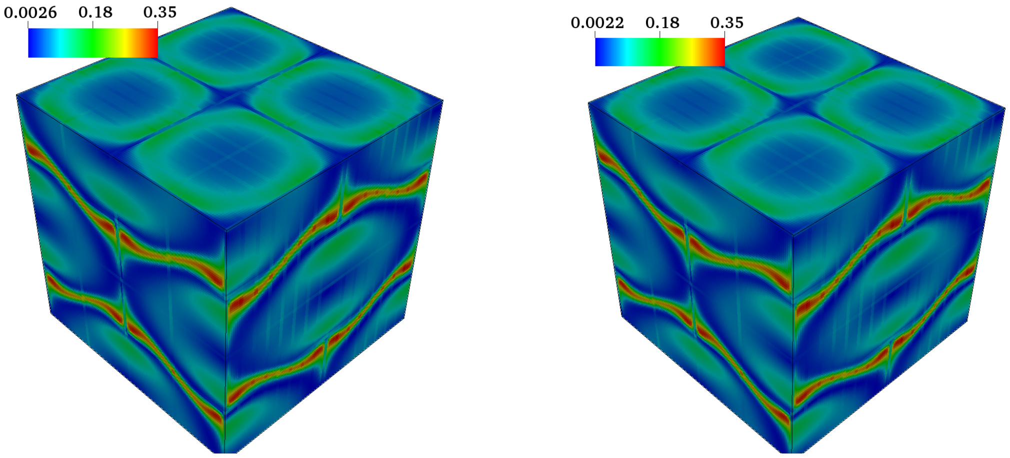


Figure 5.71: Instantaneous vorticity field for the resolution of $N128$ for CSMA ($C = 0.1$) STA (left) and REG (right) at dimensional time 3.

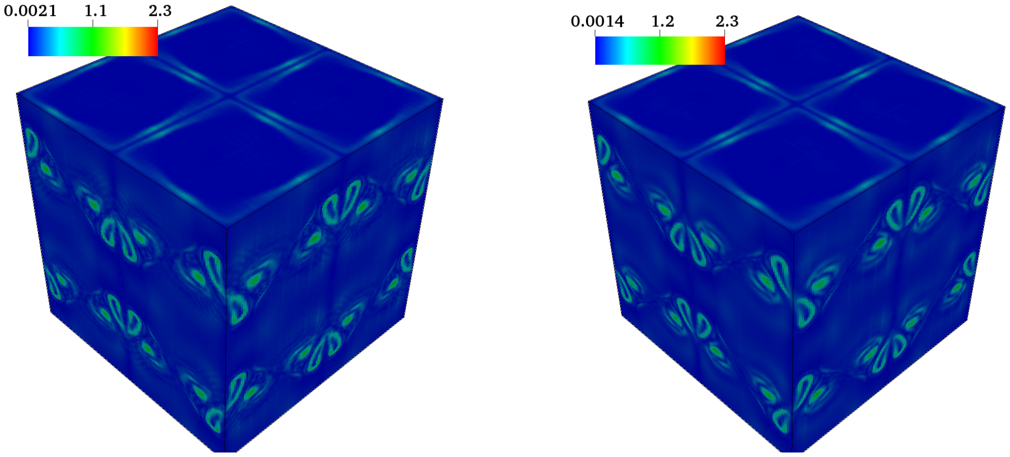


Figure 5.72: Instantaneous vorticity field for the resolution of $N128$ for CSMA ($C = 0.1$) STA (left) and REG (right) at dimensional time 9.

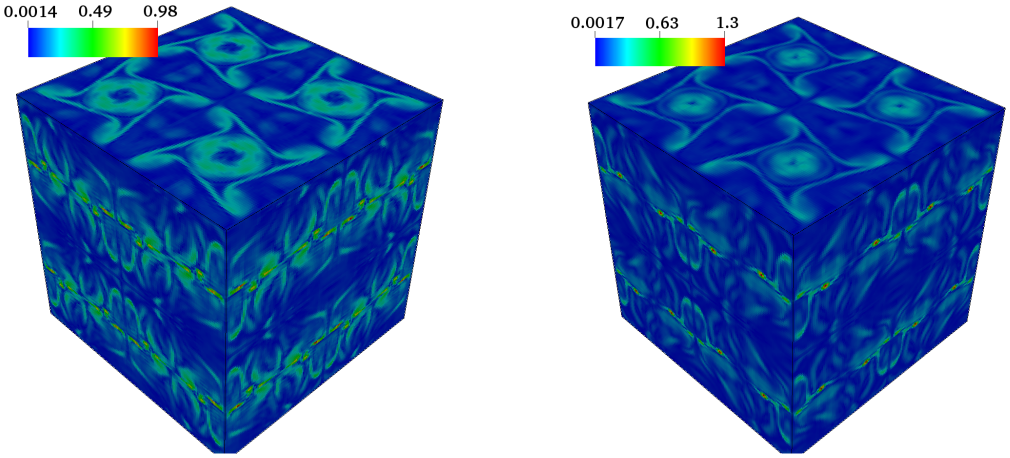


Figure 5.73: Instantaneous vorticity field for the resolution of $N128$ for CSMA ($C = 0.1$) STA (left) and REG (right) at dimensional time 15.

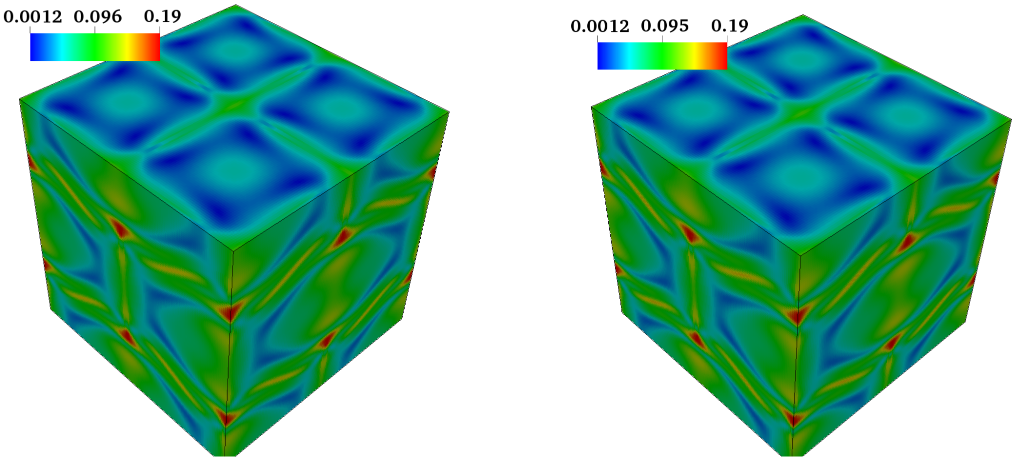


Figure 5.74: Instantaneous eddy viscosity field for the resolution of N 128 for CSMA ($C = 0.1$) STA (left) and REG (right) at dimensional time 3.

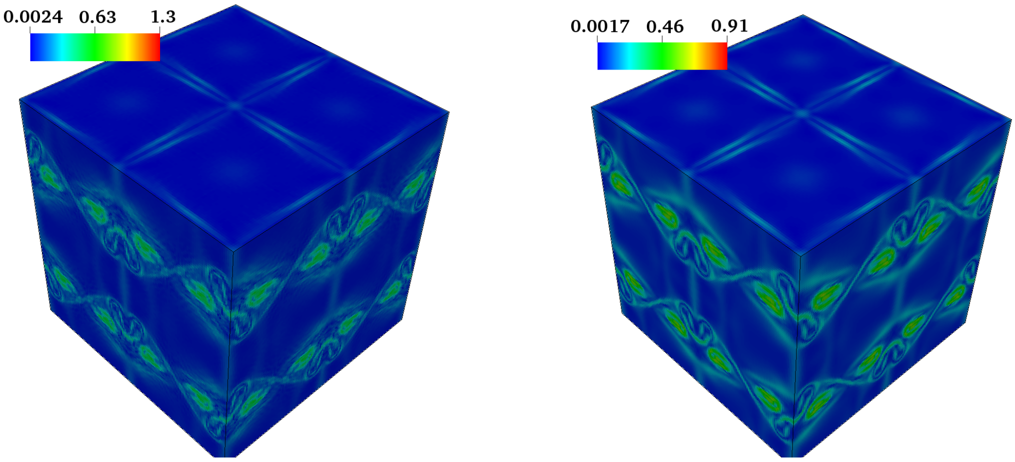


Figure 5.75: Instantaneous eddy viscosity field for the resolution of N 128 for CSMA ($C = 0.1$) STA (left) and REG (right) at dimensional time 9.

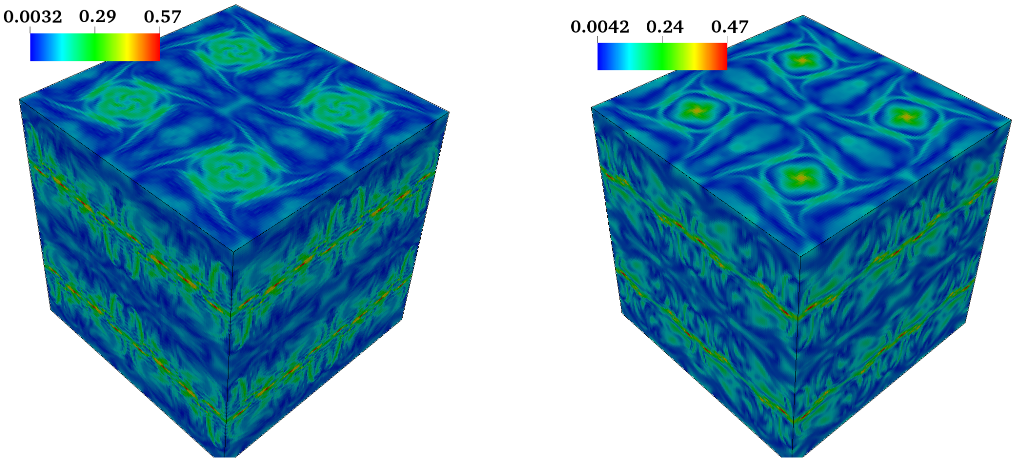


Figure 5.76: Instantaneous eddy viscosity field for the resolution of N 128 for CSMA ($C = 0.1$) STA (left) and REG (right) at dimensional time 15.

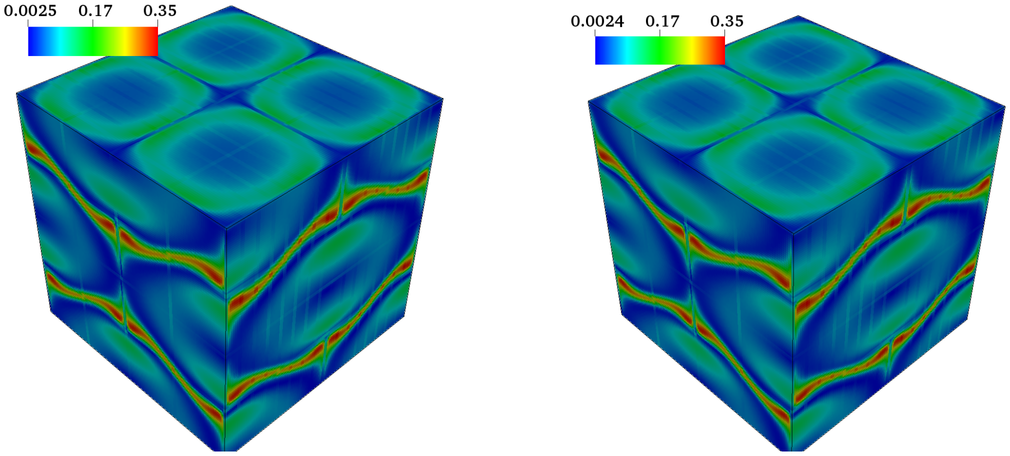


Figure 5.77: Instantaneous vorticity field for the resolution of $N128$ for DSMA STA (left) and REG (right) at dimensional time 3.

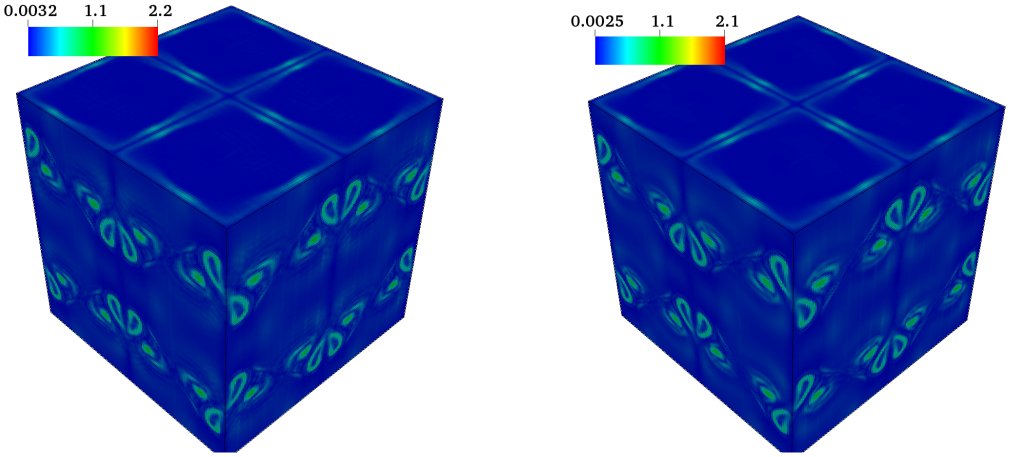


Figure 5.78: Instantaneous vorticity field for the resolution of $N128$ for DSMA STA (left) and REG (right) at dimensional time 9.

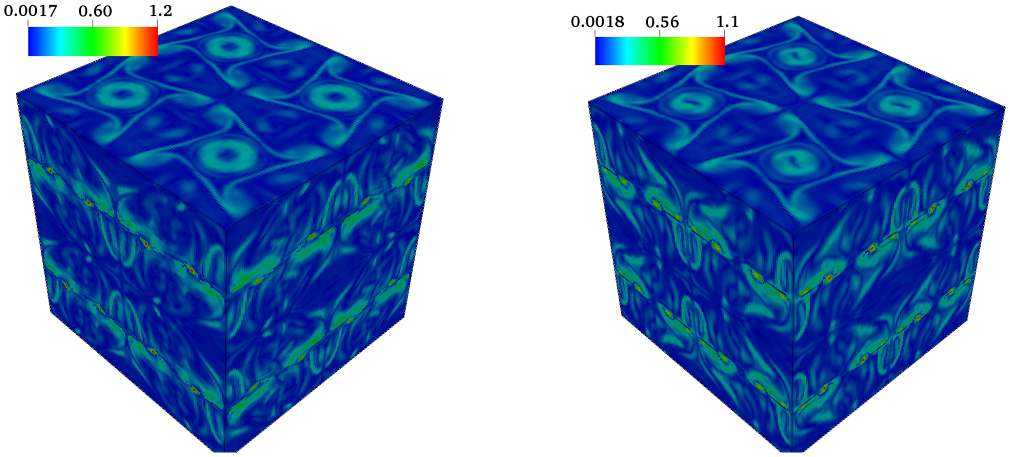


Figure 5.79: Instantaneous vorticity field for the resolution of $N128$ for DSMA STA (left) and REG (right) at dimensional time 15.

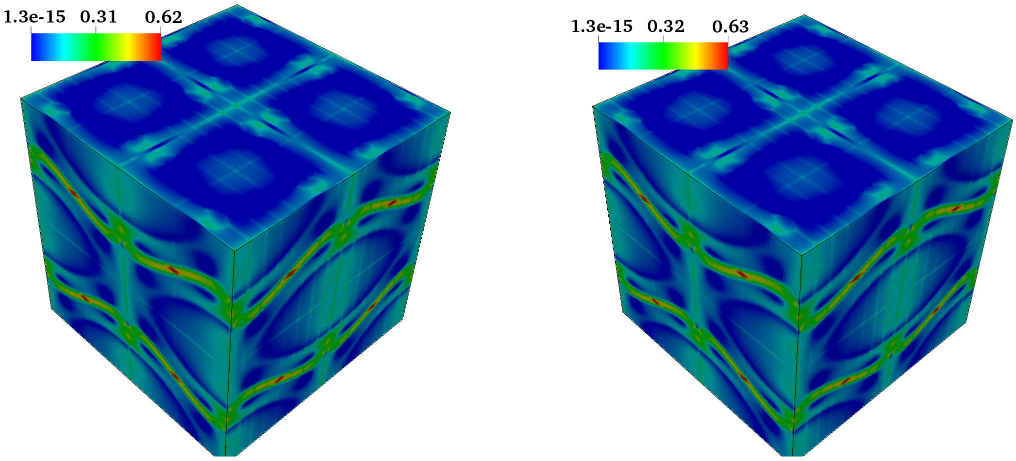


Figure 5.80: Instantaneous eddy viscosity field for the resolution of N 128 for DSMA STA (left) and REG (right) at dimensional time 3.

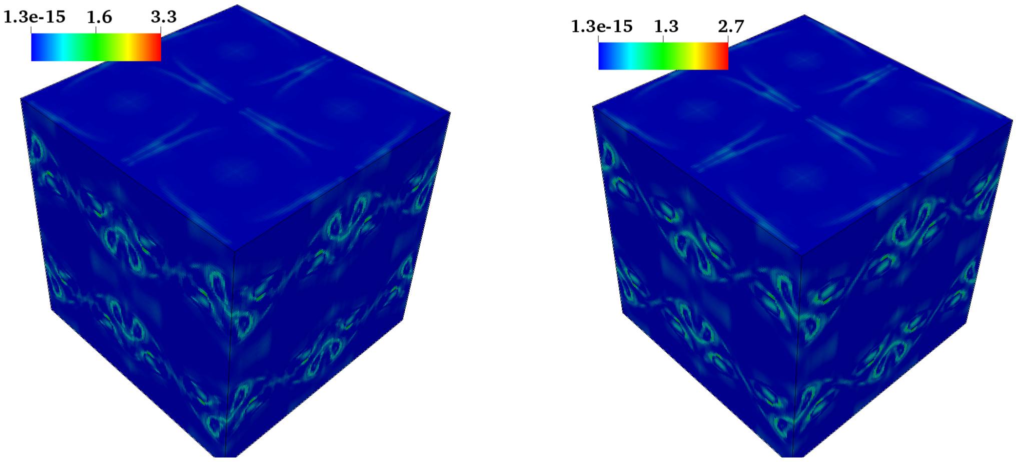


Figure 5.81: Instantaneous eddy viscosity field for the resolution of N 128 for DSMA STA (left) and REG (right) at dimensional time 9.

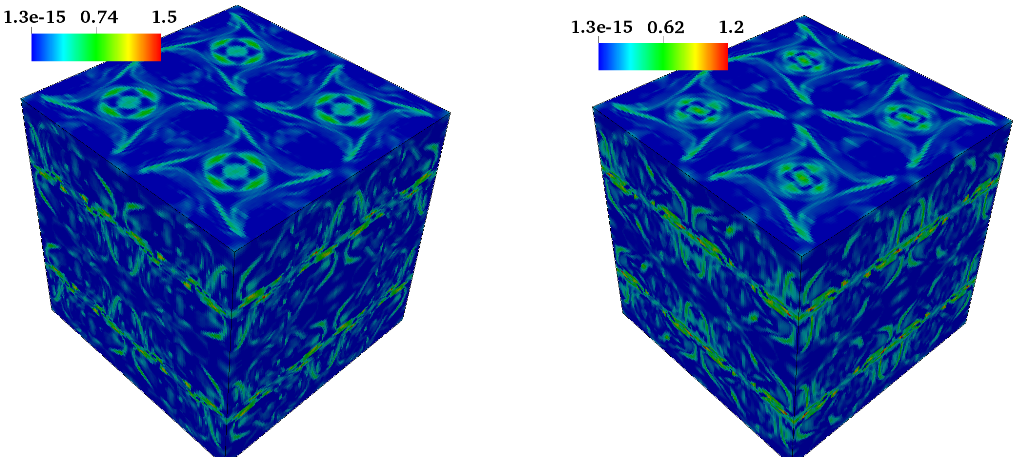


Figure 5.82: Instantaneous eddy viscosity field for the resolution of N 128 for DSMA STA (left) and REG (right) at dimensional time 15.

the constant C has altered the outcome significantly. First of all, at $t^\star = 3$, the eddy viscosity field calculated by DSMA is in a better agreement with vorticity.

Moreover, the range between minimum and maximum value has expanded with the former value to be around the machine error and close to zero. This enormous disparity is essential for better adaptability to different regions of the flowfield. In the other two time instances, DSMA has returned a smoother solution with more distinguishable features than the more blurring versions of CSMA.

Therefore, DSMA appears to deliver a more accurate and less dissipative eddy viscosity field, though, recalling the dissipation rates, CSMA has performed slightly better. To investigate this issue further, Figs. 5.83, 5.84 and 5.85 present the instantaneous values of the constant C estimating at the same three time instances. In general, the algorithm has identified successfully in all cases the regions of high vorticity. Examining the behaviour at $t^\star = 3$, it is evident that most of the domain has a value around 0.1, a case similar to CSMA. Moreover, the algorithm has characterised the vortices correctly as regions of low turbulence. However, some local extrema do not correspond to regions of high vorticity.

Given that it is still in the first phase, one can conclude that the algorithm has erroneously marked part of the domain as a high turbulent one. By looking at the eddy viscosity field, not all of these high-value regions translated to local extrema for the eddy viscosity. Nonetheless, this miscalculation is expected to result in excess dissipation. As for the collision models, there is no apparent discrepancy between the STA and REG version. This observation is also accurate for the next instance at $t^\star = 9$. This time, the instantaneous values of the constant are a better representation of the corresponding vorticity field.

Additionally, the part of the domain marked with low values has increased compared to the first instance. Consequently, this is a piece of solid evidence that the algorithm cannot distinguish between laminar and turbulent flow accurately. Finally, investigating the last instance, in this case, the algorithm is capable of capturing the turbulent field and the existence of small scales. Like in the previous instance, the patterns are identical to the corresponding vorticity and eddy viscosity fields. Finally, it appears that there are some noticeable differences between the STA and REG models, with the former to estimate more regions having extreme values.

To conclude the current discussion, Fig. 5.86 reports the evolution of the average value of the constant C estimated per saved time step for both collision models. This plot verifies the previous observation about the poor performance of the model during the first phase. Nonetheless, after $t^\star = 4$, it seems to return the expected behaviour. During the second phase, the transition to turbulence adopts a relatively low and constant value of around 0.07. Then, at the peak of the dissipation and the beginning of the final phase,

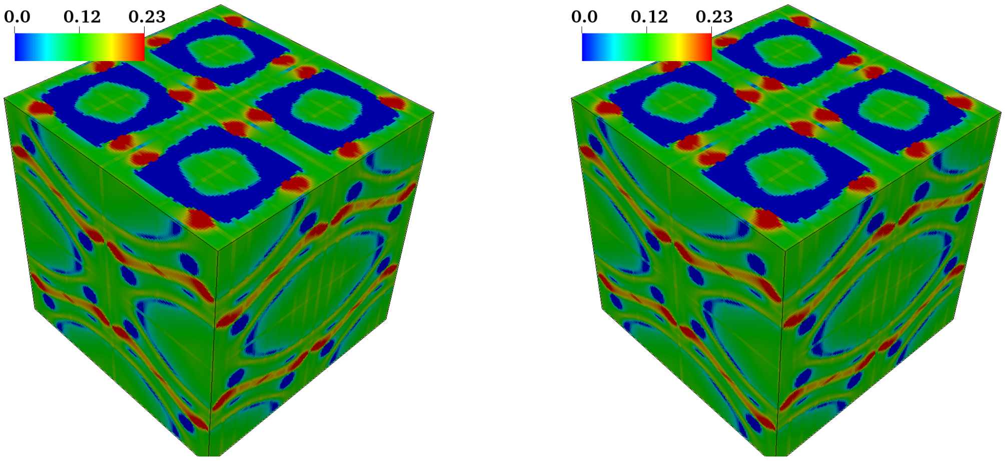


Figure 5.83: Instantaneous values of the constant C for DSMA for the resolution of N 128 for STA (left) and REG (right) at dimensional time 3.

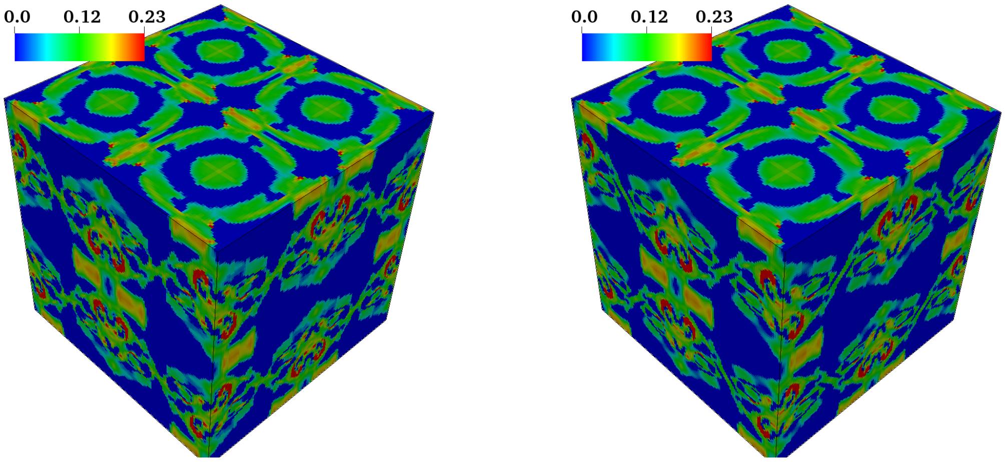


Figure 5.84: Instantaneous values of the constant C for DSMA for the resolution of N 128 for STA (left) and REG (right) at dimensional time 9.

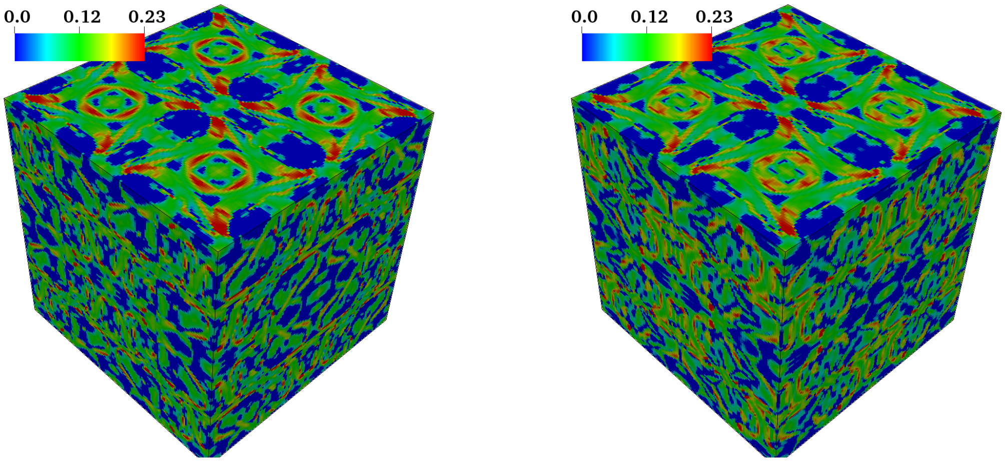


Figure 5.85: Instantaneous values of the constant C for DSMA for the resolution of N 128 for STA (left) and REG (right) at dimensional time 15.

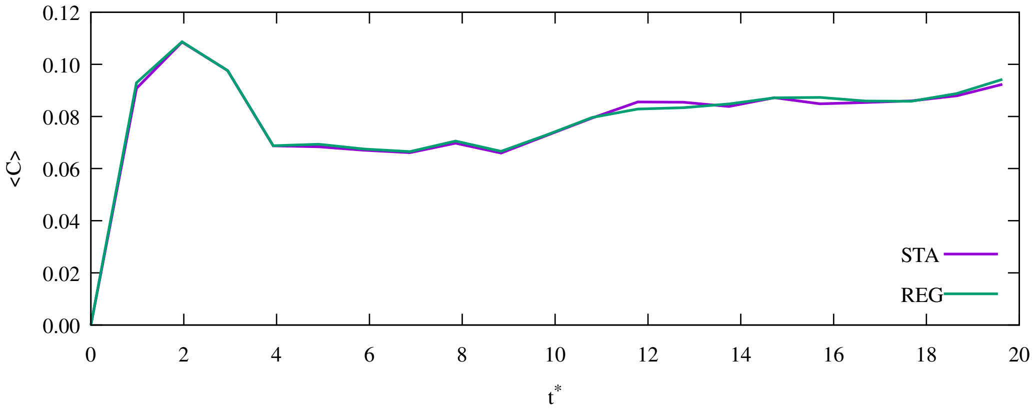


Figure 5.86: Evolution of the domain average value of the constant C for the DSMA model at $N128$ resolution both for STA and REG models.

it rises rapidly and then continues to increase with a slower pace up to a value of around 0.09.

Consequently, though for the majority of the time, it keeps an average value lower than CSMA, the excess of dissipation added during the initial stage appears to play an essential role in its more unsatisfactory performance compared to CSMA. Another important observation is that both collision models have estimated identical curves, with some slight discrepancies during the last phase. Consequently, though the STA model has been shown to return values closer to the extrema, the average picture is similar to the REG one.

The extra dissipation during the initial phase can be restricted by reducing the threshold's value for the maximum available value, that is $C = 0.23$. However, this may reduce the robustness of the model and thus, acquiring data from more simulations is vital for such a decision. To identify the importance of the threshold, I have run the TGV case without applying it, and the value of C reached unnaturally high values greater than one. The last outcome is the product of calculating the constant locally per cell and not averaging in homogeneous directions, which in this case are all three directions. One can speculate that applying this averaging would have resulted in a smoother field, reducing the excess of dissipation in some regions. On the other hand, by setting the minimum value at zero, I have prevented the backscattering of the energy from smaller scales back to the largest ones. This could also benefit the performance of DSMA, but, at the same time, it would reduce its robustness.

Finally, I will proceed with the WALE model. Figures 5.87, 5.88 and 5.89 report its vorticity field. Again, besides the values of minimum and maximum, there are no discrepancies with the other models. The situation is different in the case of the eddy viscosity field as one can see in Figs. 5.90, 5.91 and 5.92 Firstly, it appears that the majority of the domain is covered by low values. Simultaneously, there are some distinguishable spots of high eddy viscosity values. This behaviour also seems to be the

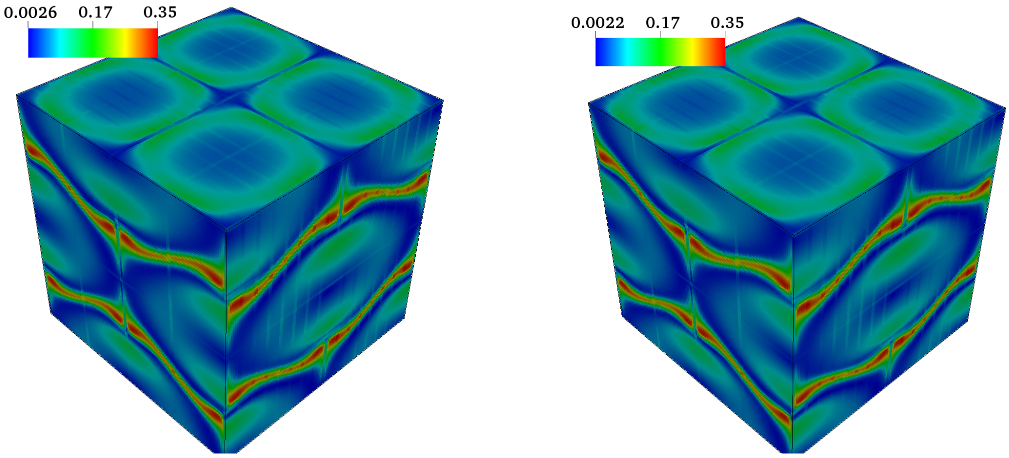


Figure 5.87: Instantaneous vorticity field for the resolution of $N128$ for WALE STA (left) and REG (right) at dimensional time 3.

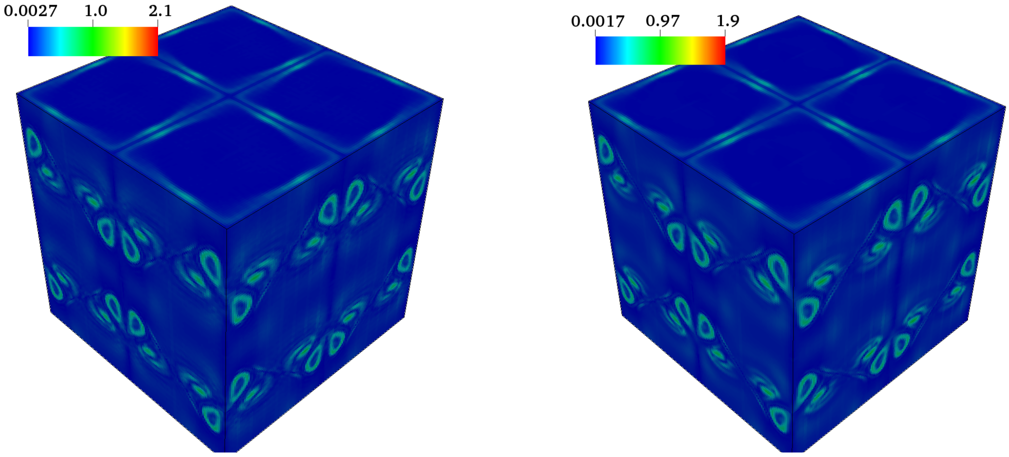


Figure 5.88: Instantaneous vorticity field for the resolution of $N128$ for WALE STA (left) and REG (right) at dimensional time 9.

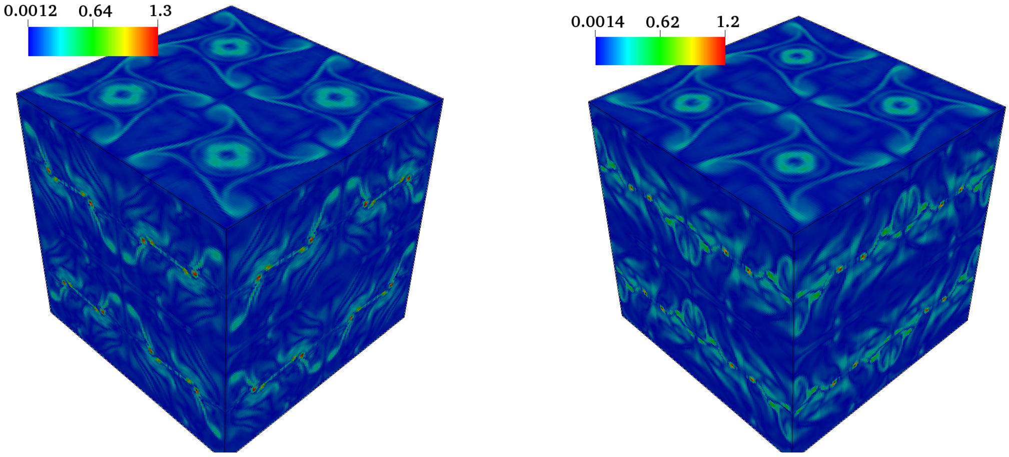


Figure 5.89: Instantaneous vorticity field for the resolution of $N128$ for WALE STA (left) and REG (right) at dimensional time 15.

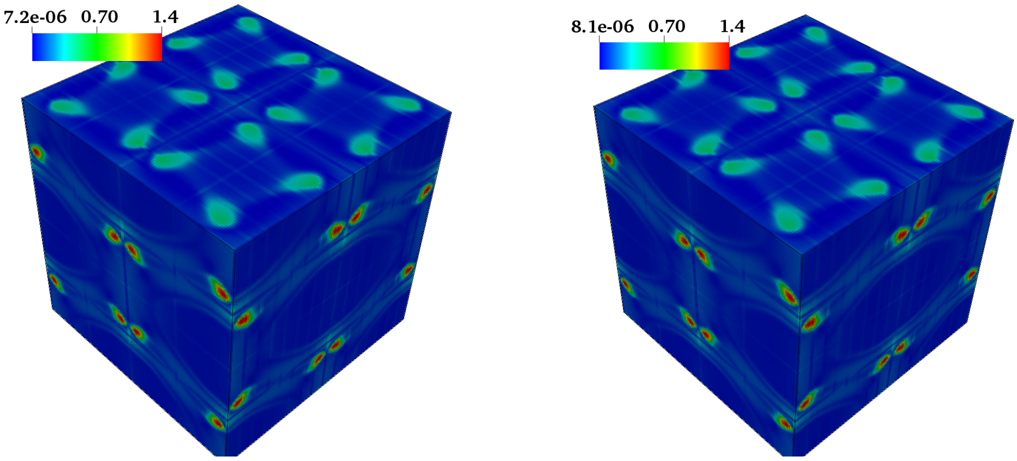


Figure 5.90: Instantaneous eddy viscosity field for the resolution of N 128 for WALE STA (left) and REG (right) at dimensional time 3.

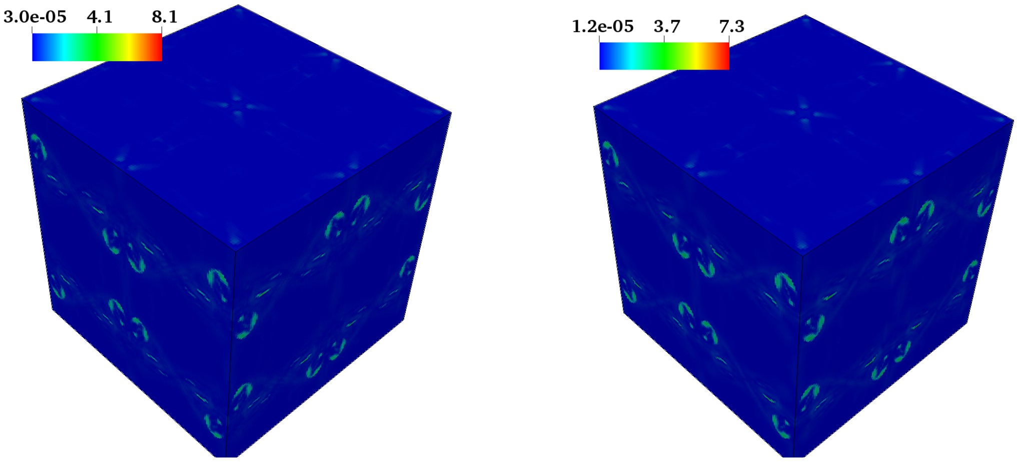


Figure 5.91: Instantaneous eddy viscosity field for the resolution of N 128 for WALE STA (left) and REG (right) at dimensional time 9.

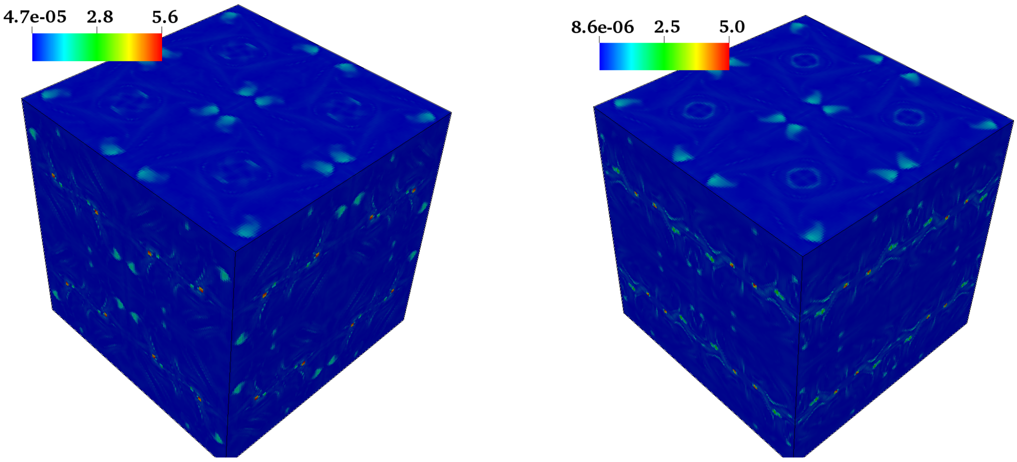


Figure 5.92: Instantaneous eddy viscosity field for the resolution of N 128 for WALE STA (left) and REG (right) at dimensional time 15.

case for the other two time instances. The high values coincide with the vorticity fields, except for some regions during the first instance.

Interestingly, these are the same regions falsely reported by the DSMA algorithm with large constant values. Examining the disparity of the range, WALE has predicted the minimum value two or three orders of magnitude smaller than CSMA, though not as small as DSMA. However, in the case of the maximum, it has returned the highest value of all the models. These high values are located in some isolated cells. On the comparison between STA and REG, as in the previous cases, there is no significant difference during the initial phase both in the range of values and in the field. However, in the subsequent two phases, STA has returned slightly bigger values, while the fields remain reasonably identical.

Therefore, one can conclude that the WALE model behaves noticeably different from the other two. While the other two models estimate a somewhat diffusive eddy viscosity field, WALE returns a more concentrated one. To further enhance the comparative analysis, Fig. 5.93 reports the evolution of the maximum value of the ratio of viscosities found in the domain for all the turbulence and collision models. The WALE model appears to have calculated the most significant maximum values during all the period, and particularly during the initial phase. It also experiences the most considerable disparity between STA and REG version.

It is essential to discuss how the initial conditions and the zero value non-equilibrium part have affected the results. First of all, examining Fig. 5.93 it is clear that the zero value of eddy viscosity at the beginning of the simulation for both CSMA and DSMA is due to this decision. Therefore, it appears that these two turbulence models have been benefited from the initial conditions resulting in a less dissipative behaviour during the initial phase and thus an apparent better performance compared to WALE. However, there are no apparent discrepancies in the vorticity fields obtained by all three models during the initial phase. Consequently, one can argue that this inaccuracy plays no vital role in the verification procedure.

Taking everything into account, one can speculate the capability of the turbulence models to deal with a turbulence field in general. CSMA has returned the lowest disparity between minimum and maximum value, with the former to be two or three orders of magnitude larger than the second higher. Moreover, it has also returned the most diffusive eddy viscosity field. Therefore, having the eddy viscosity spread in a relatively homogeneous way adds robustness and a way to deal with any numerical instabilities arising in the domain. On the other hand, the excess of dissipation in the whole domain also means a deterioration of the accuracy and damping of physical scales. This behaviour can be improved using the DSMA model. Its ability to alter the constant C locally per cell results in having the most extensive range of values for the eddy viscosity, thus expecting to enhance its adaptability with different flowfield regions. However, its most significant drawback is its expensive step since the algorithm needs to parse the domain multiple times to apply the filtering. Finally, the candidate that seems to provide both accuracy and reasonably expensiveness is the WALE model. Its eddy viscosity field is more concentrated than CSMA, resulting in a less dissipative behaviour but at the same time with reduced robustness. Nonetheless, both DSMA and WALE have failed to adapt correctly during the first phase pointing to the excess of dissipation under a laminar case or during the transition to turbulence.

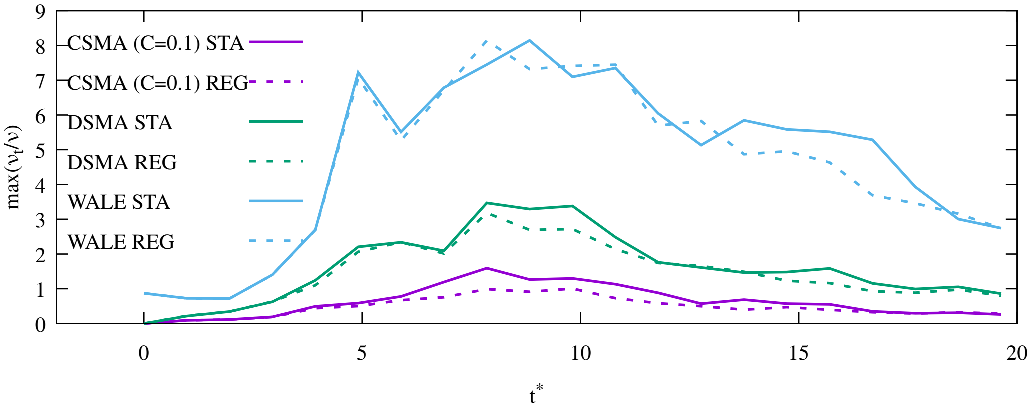


Figure 5.93: Evolution of the max value of eddy viscosity in the domain for the resolution of N 128 for CSMA ($C = 0.1$) (purple), DSMA (green) and WALE (blue). Solid lines shows the STA and dashed the REG model.

5.4.3 Resolution N 32

To close the section, Figs. 5.94 and 5.95 show the evolution of kinetic energy and dissipation rates for the $N32$ resolution. This resolution provides a significant challenge for the LES models due to their extreme coarseness. Indeed, examining the kinetic energy, all LES models have predicted significantly lower kinetic energy in the domain compared to the DNS reference, indicating notable damping of large vortices due to eddy viscosity during the initial stage. By investigating the dissipation rates, there is also a notable difference between STA and REG versions. In the case of REG, the previous two resolutions' familiar pattern is applied, where DNS has the more dissipation followed by CSMA and then the other two models. This time, WALE appears to perform better, and its curve is close to the DSMA one. However, in the situation of STA, DSMA has the worst performance. This indicates that resolution is a significant factor that needs to be considered for the comparison of LES models.

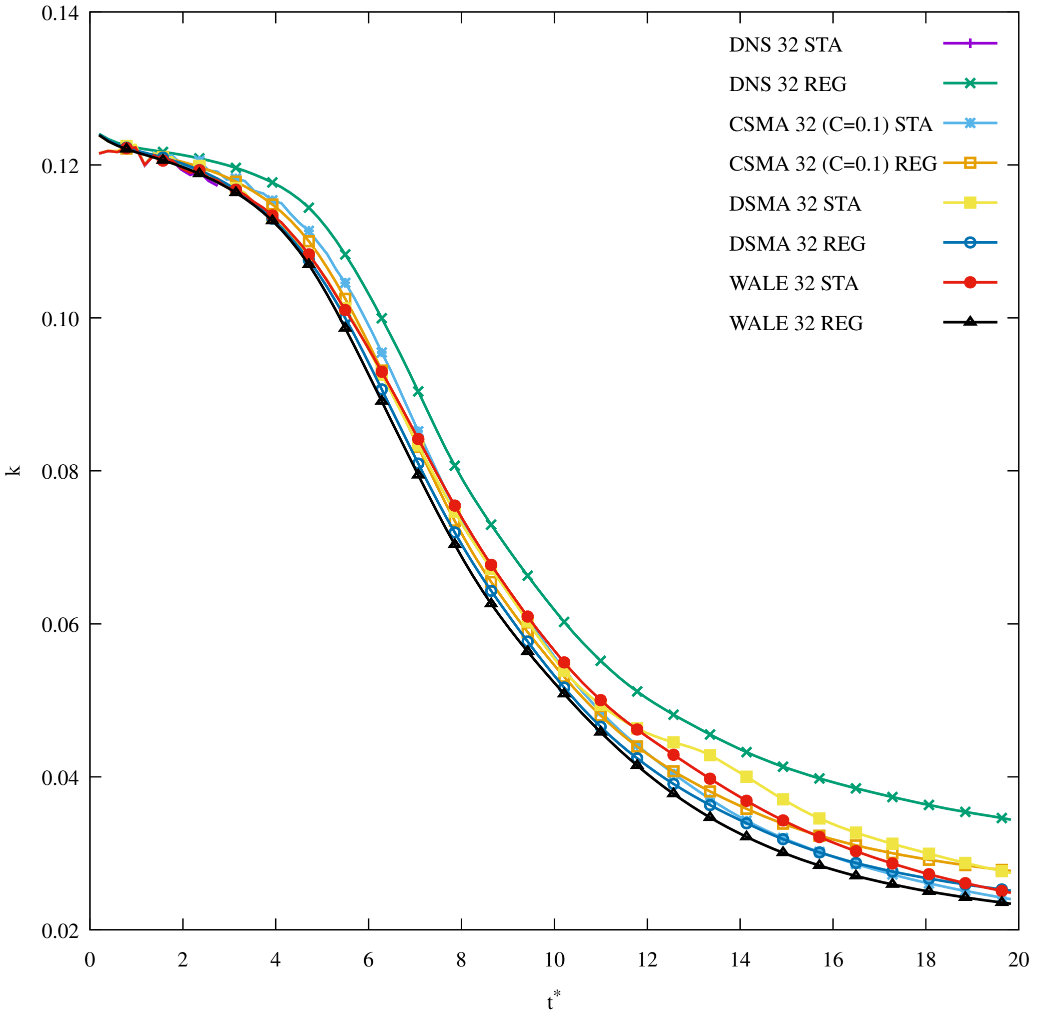


Figure 5.94: Evolution of kinetic energy for all LES models and DNS of both STA and REG versions for the resolution of $N32$.

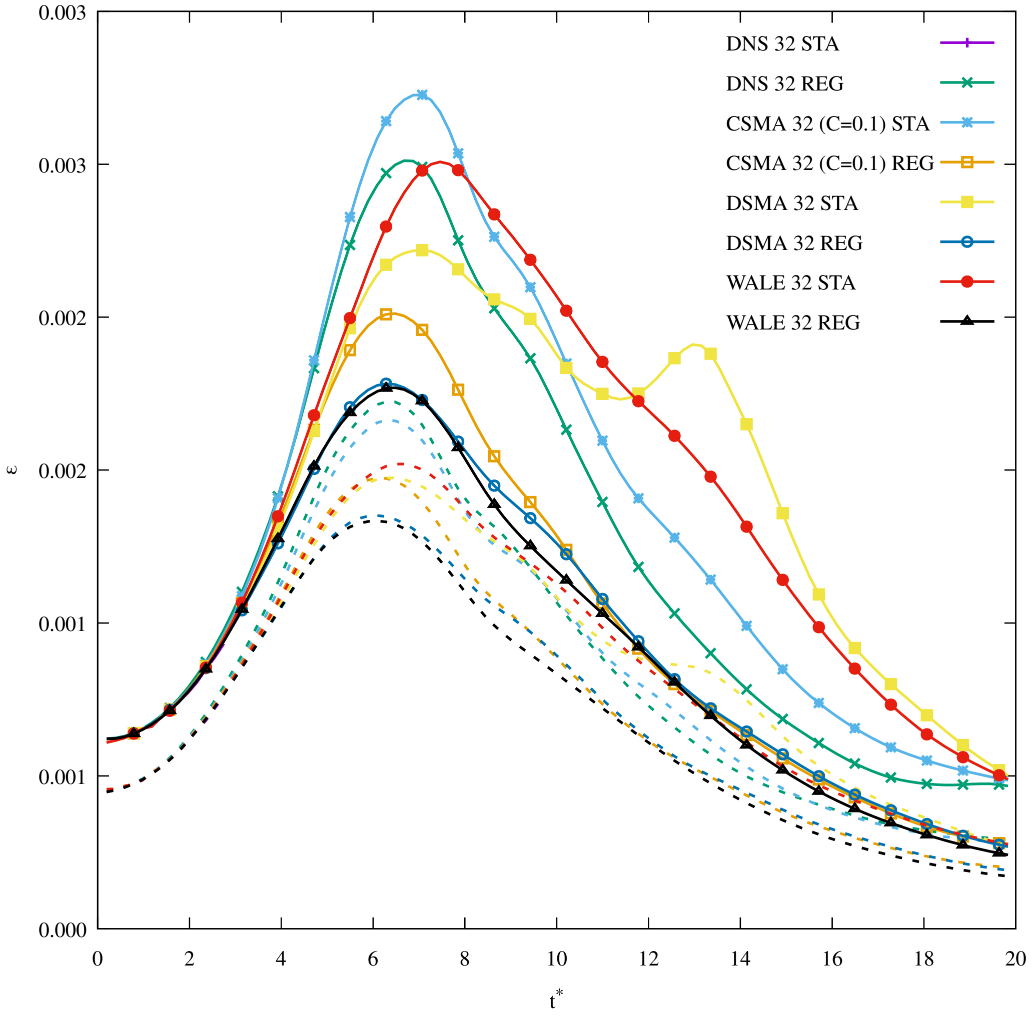


Figure 5.95: Evolution of dissipation rate (solid) and dissipation rate based on enstrophy (dashed) for all LES models and DNS of both STA and REG versions for the resolution of $N32$.

Chapter 6

Bi-periodic Channel

In the previous chapter, I have verified the newly implemented LES models by simulating homogeneous isotropic turbulence. The main idea was to prove that their energy spectra exhibit the expected shape based on Kolmogorov theory. Another situation where a universal behaviour can be identified and used for testing is the law of the wall appearing in a turbulent boundary layer.

A very convenient test case to simulate a turbulent boundary layer and examine it is the bi-periodic channel. The benefit of this canonical case is that plenty of data for comparison can be found in the literature. To simulate it, a body force needs to be applied to counterbalance the loss of momentum and to impose the specific Re_τ, estimated as

$$Re_\tau = \frac{u_\tau H}{\nu}. \tag{6.1}$$

In the above, u_τ is the friction velocity, H is the half-height of the channel and ν the viscosity of the fluid. I have set $H = 1$ for all simulations. Given Re_τ as input, Re_{bulk}, can be estimated using Dean's correlations

$$Re_{bulk} = (\frac{8}{0.073})^{4/7} Re_\tau^{8/7}. \tag{6.2}$$

The definition of Re_{bulk} is based on the mean bulk velocity u_b, which has been set equal to 1 for this case. Consequently, the viscosity can be estimated by using the equation of Re_{bulk} that reads

$$Re_{bulk} = \frac{2 u_b H}{\nu}. \tag{6.3}$$

The next step is the computation of u_τ by invoking Eq. (6.1). Finally, the formula of force reads

$$F_x = \rho\left(\frac{u_\tau^2}{H} + (u_b - \langle u \rangle)\frac{u_b}{H}\right), \tag{6.4}$$

where $\langle u \rangle$, is the averaged streamwise velocity estimated over all the domain every time step. Due to the solver interface, the calculation of $\langle u \rangle$ takes place before the beginning of the numerical step. This procedure leads to two important side-effects. Due to the order of the execution of the tasks during the numerical step, see Sec. 2.2.1, the application of the force lags by one Δt because of the intermediate streaming. Moreover, some boundary conditions, such as the wall function, are called to impose the correct macrovariables in the first fluid cell rather than the ghost cell. In this way, the first fluid cell may contain unknown values after the streaming step, affecting the calculation of the force. About the first issue, one can expect that the relatively small time step of the lattice Boltzmann is sufficient to restrict the error. To alleviate the second side effect, macrovariables are also set in ghost cells resulting in proper distribution functions propagating from them to the first layer of fluid cells.

A post-processing routine has also been coded to estimate the mean averaged values reported in this chapter. The spatial averaging has taken place over the whole domain. The idea is to estimate the profile of the variables in the normal to the wall direction. In the case of the streamline velocity, the formula reads

$$\langle u \rangle_i = \frac{1}{N} \sum_N u_i(\mathbf{x}, t), \quad \text{where} \quad i \in N_y, \tag{6.5}$$

where $N = N_x \cdot N_z$ is the number of all cells in a plain vertical to the normal direction. The normal direction consists of N_y points. One can also estimate the covariance of a variable as

$$\langle uu \rangle_i = \frac{1}{N} \sum_N (u_i(\mathbf{x}, t))^2, \quad \text{where} \quad i \in N_y. \tag{6.6}$$

To avoid storing a large amount of data, the temporal averaging is based on informing the current prediction with the new sample, and thus it takes place simultaneously with the spatial averaging process. For instance,

$$\langle u \rangle_i^{T+1} = \frac{\langle u \rangle_i + T \cdot \langle u \rangle_i^T}{T+1}, \tag{6.7}$$

where T is the current number of samples. For the estimation of the Reynolds stresses, it yields

$$\langle u'u' \rangle = \langle uu \rangle_i - (\langle u \rangle_i)^2. \tag{6.8}$$

The *rms* values can be calculated as $u_{rms} = \sqrt{\langle u'u' \rangle}$. The pressure has been estimated by multiplying the density with the square of the speed of sound.

This chapter consists of two parts. In the first part, I will examine the capability of the LES models to capture the law of the wall by running the bi-periodic channel in a WRLES mode. To achieve this target, a low Re_τ will be simulated, and the results will be compared with two reference sources from the literature. Simultaneously, I will report the performance of the new algorithm for imposing macrovariables in ghost cells described in Chap. 4.

In the second part of this chapter, I will utilize the bi-periodic channel to verify the wall function, also reported in Chap. 4. To do so, two Re_τ numbers will be simulated, and a convergence analysis will be carried out. The combination of the first Re_τ and the three resolutions will test the wall function under the expected conditions for its usability. On the other hand, the second Re_τ has been chosen large enough to test the new implementation under extreme coarse resolution. The data will be compared against reference results from the literature. It should be highlighted that none of these two cases could run without the wall treatment employment.

All the simulations in this chapter have been run with the STA collision model. I have tried to employ the REG model, but the simulations were crushed. The source of error was found to be in the vicinity of the boundaries. One can speculate that any inaccuracies in the distribution functions close to a boundary will be magnified by the regularisation process, Eq. (2.19). The application of regularisation affects all the distribution functions. Therefore, if some of them are inaccurate, this error will be distributed to all of them.

6.1 Wall Resolved LES ($Re_\tau = 183.6$)

To verify the LES model for wall-bounded flows, I will replicate the numerical experiment reported in (Liu et al., 2012). The idea is to simulate a turbulent open channel flow with dimensions ($6H \times H \times 3H$) and $Re_\tau = 183.6$. This corresponds to a Re_{bulk} of 5660. A no-slip condition is imposed for the bottom boundary, while at the top, a slip wall is applied. The algorithm of both boundaries is based on the standard bounce back method, where distribution functions are reflected following specific instructions depending on the type of boundary. Therefore, no macrovariables are estimated in the ghost cells and thus, the new algorithm for imposing macrovariables needs to be employed, particularly in the case of DSMA and WALE. A resolution of 45 cells per H has been used following the reference resulting in $y^+ \approx 2$ for the first fluid cell centre and $\Delta y^+ \approx 4$ for the spatial step.

I utilised the Musker profile of the wall function to initialise the flowfield and added turbulent fluctuations of $I = 5\%$ intensity. The density was set to $\rho = 1$ everywhere in the domain. Like the TGV case, the distribution functions were initialised to their equilibrium values, while the non-equilibrium part had a zero value. Consequently, some initial instabilities have been introduced, though it is expected that they have been dissipated by the time the statistical analysis took place. The domain and the velocity field's initialisation can be seen in Fig. 6.1. The simulations were run initially for $50T$, where $T = H/u_\tau$, to establish a turbulent field and then for other $16T$ when measurement and statistical analysis were carried out. The results will be compared with two references. The first is the NS DNS data reported in (Kim et al., 1987). The

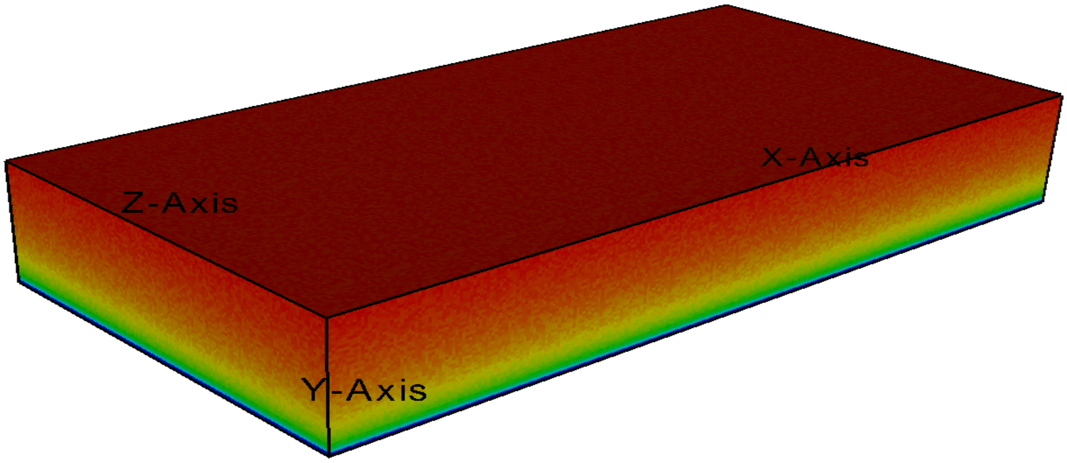


Figure 6.1: The domain and the initial velocity field for the case of $Re_\tau = 183.6$.

second reference is the data obtained from the MRT LBM solver presented in (Liu et al., 2012) with WALE as the turbulence model. In their case, they imposed $Ma = 0.28$, which is a relatively large value for LBM solvers. However, the MRT collision model's stability makes it possible, though with some loss of accuracy. In my case, I imposed $Ma = 0.2$ because STA SRT is not as stable.

LES	CSMA	DSMA	WALE
u_τ	0.0628	0.0609	0.0642
Rel. Err.	3.23%	6.14%	0.99%

Table 6.1: Estimated friction velocities for each LES models and their relative error.

Starting the discussion about the results, Table 6.1 reports the estimated friction velocities u_τ for all three models and their relative errors. The expected friction velocity is $u_\tau = 0.0649$. An interesting observation is that DSMA has the worst performance. One can speculate that the reason is that though one can impose the correct macrovariables at ghost cells, the algorithm of DSMA also needs accurate non-equilibrium parts to calculate the strain rate locally. However, this restriction cannot be satisfied, leading to an inaccurate result while calculating the constant at the first fluid cell.

On the other hand, WALE requires only accurate macrovariables at ghost cells to estimate its eddy viscosity. By imposing reasonable values with the proposed algorithm, it has the best performance with its relative error lower than 1%, as expected by a turbulent model designed to be consistently close to the wall. These two observations illustrate both the necessity and the efficiency of the newly proposed algorithm, see Sec. 4.3.

Figure 6.2 shows the mean streamwise velocity profile for the three LES models and the two reference solutions. Moreover, Fig. 6.3 shows the same information but with the more usual semi-log axes representation. All three models have captured the two first points accurately in the viscous sub-layer. Nonetheless, they have underestimated the values in the buffer layer. One can speculate that this inaccuracy is the effect of the relatively high Ma number with the combination of the STA SRT collision model. Finally, their accuracy has recovered in the log layer where the curve of DSMA fits the DNS data while the other two have underestimated the values slightly.

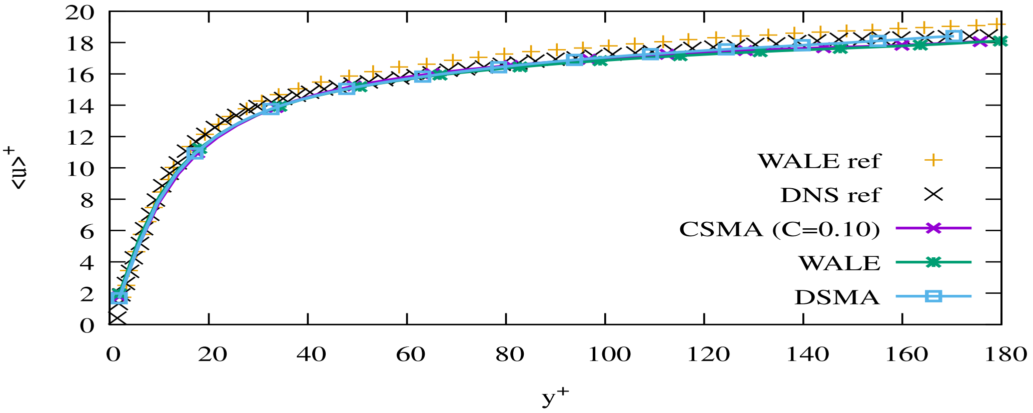


Figure 6.2: Mean streamwise velocity profile scaled at wall units.

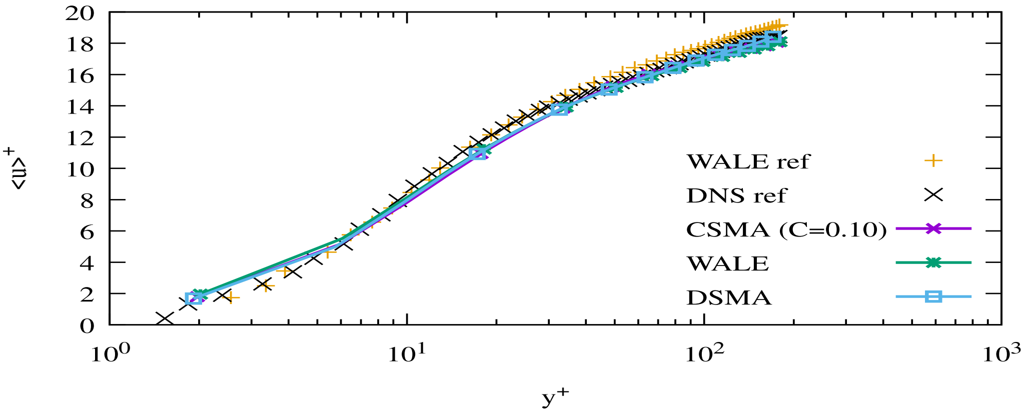


Figure 6.3: Mean streamwise velocity profile in semi-log axes scaled at wall units.

The following variable to be examined is the Reynolds shear stress $\langle uv \rangle$ appearing in Fig. 6.4. Like the streamwise velocity, all the models show some slight divergence from the reference data in the region of the buffer layer. However, they can recover in the log layer, where CSMA and DSMA fit DNS data accurately. On the other side, both STA and MRT WALE have under-predicted the values close to the peak, indicating that it is rather due to the turbulence model and not the collision one that affects this plot.

Continuing the discussion, Figs. 6.5, 6.6 and 6.7 plot the RMS streamwise, normal and spanwise velocity fluctuations, respectively. In the streamwise direction, DSMA and WALE converge to the WALE reference data, with slightly higher values than DNS. CSMA overpredicts the values for the majority of the channel height. In the normal, all the models have performed similarly with DSMA to return a curve somewhat closer to

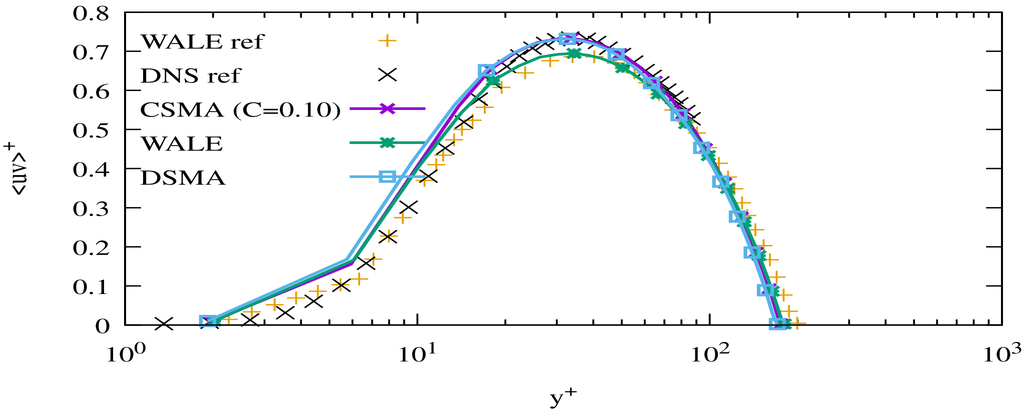


Figure 6.4: Reynolds shear stresses scaled at wall units.

the DNS data. Finally, in the spanwise direction, CSMA and WALE predict curves closer to the DNS data, while DSMA is closer to the WALE reference solution. Consequently, it appears that no model outperforms the others for this comparison though WALE has shown more consistency.

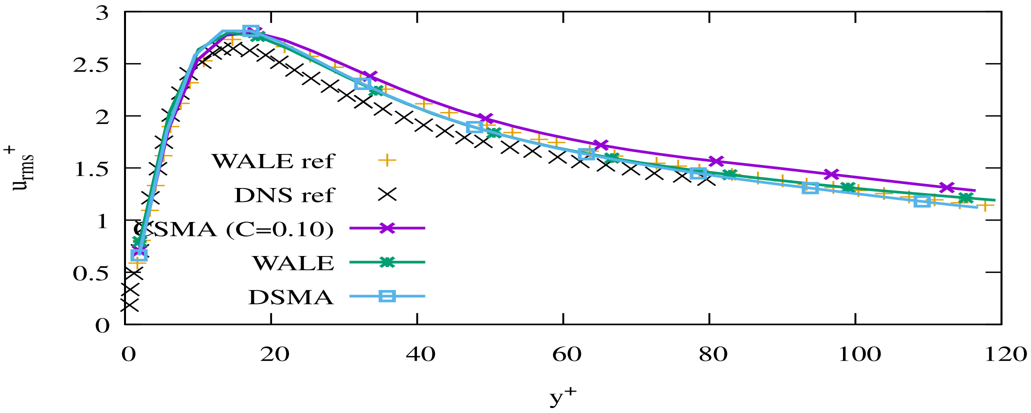


Figure 6.5: RMS streamwise velocity fluctuations scaled at wall units.

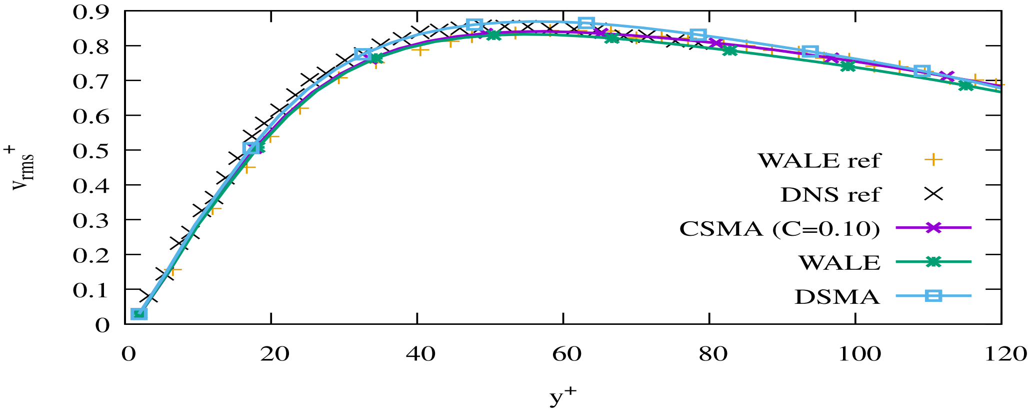


Figure 6.6: RMS normal velocity fluctuations scaled at wall units.

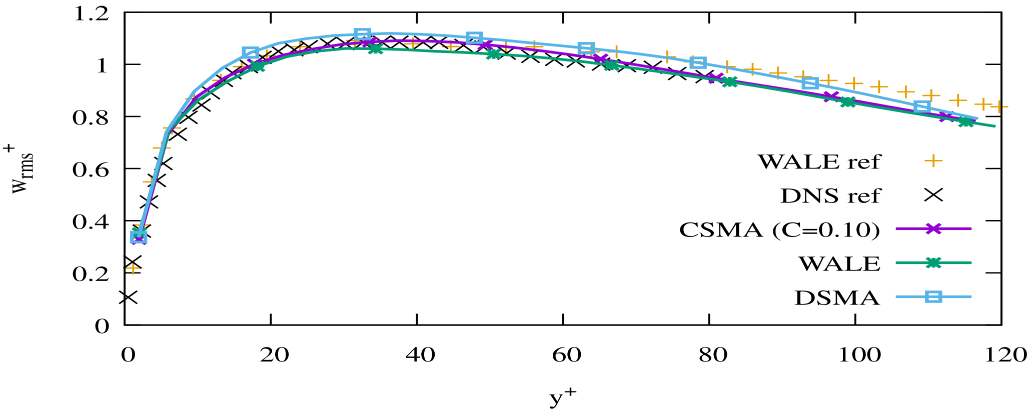


Figure 6.7: RMS spanwise velocity fluctuations scaled at wall units.

To further enrich the verification procedure, Figs. 6.8, 6.9 and 6.10 present the RMS streamwise, normal and spanwise vorticity fluctuations, respectively. In this case, there is no obvious discrepancy among the LES models. All the curves coincide very well with the WALE reference data, underpredicting the DNS results in the viscous sub-layer and buffer layer. Thus, one can speculate that the lack of resolution close to the wall is why this inaccuracy and not the effect of turbulence or collision model. For the estimation of vorticity, a stencil needs to be applied. Given that the output function is called after the streaming at the end of the step, the macrovariables in the first ghost cell have obtained random values polluting the calculation in the first fluid cell. Consequently, I have omitted the value of the first point as being inaccurate.

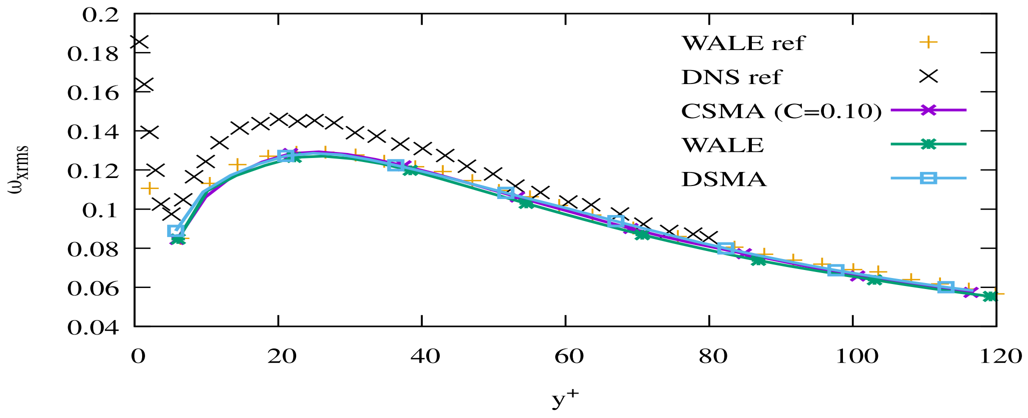


Figure 6.8: RMS streamwise vorticity fluctuations scaled at wall units.

The LBM is a weakly compressible method meaning that pressure fluctuations can also be estimated. Figure 6.11 shows the RMS pressure fluctuations estimated for the current test case. All models were able to capture the anticipated shape of the curve. CSMA and DSMA converge to the WALE reference data while the STA WALE of my simulation diverges from this reference but comes closer to the DNS one. One can speculate that the divergence of all the models, including the WALE reference, from the DNS data, occurs due to the high Ma number used in this case, (Liu et al., 2012).

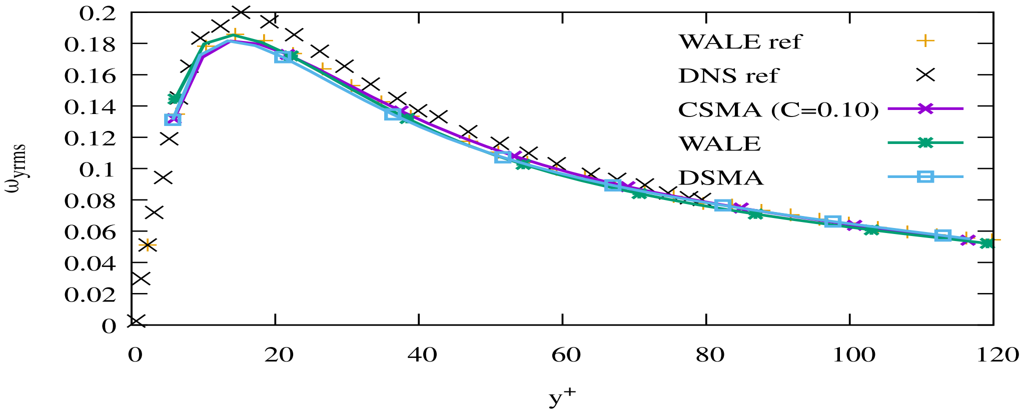


Figure 6.9: RMS normal vorticity fluctuations scaled at wall units.

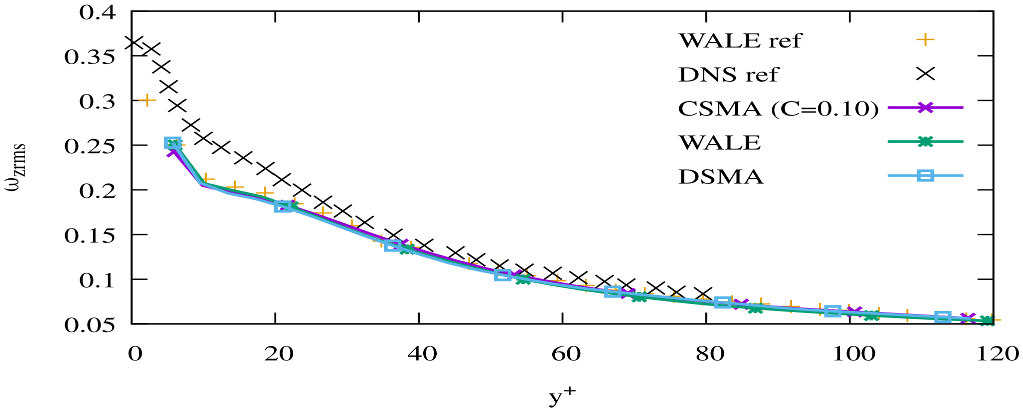


Figure 6.10: RMS spanwise vorticity fluctuations scaled at wall units.

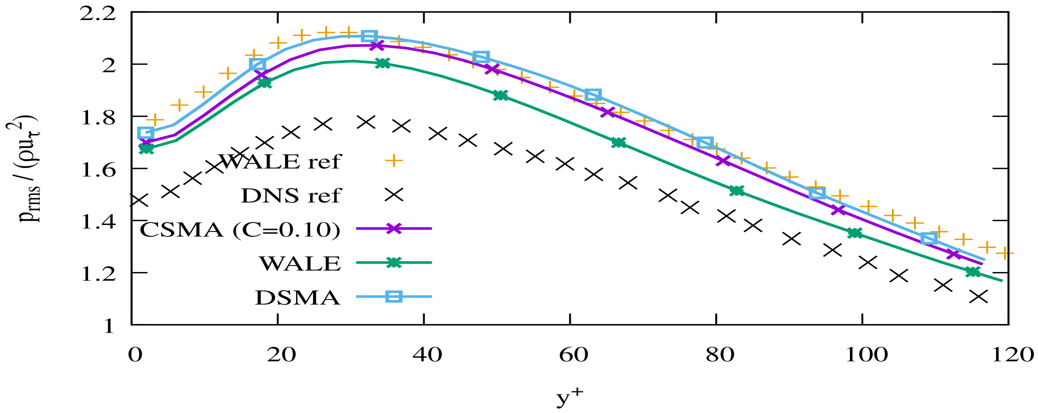


Figure 6.11: RMS pressure fluctuations scaled at wall units.

As for the final comparison, I present the eddy viscosity and ω_{eff} estimated by all three turbulence models for the current case. Figure 6.12 shows the eddy viscosity profile near the wall. The WALE model, by construction, tries to damp the eddy viscosity in the vicinity of the wall and follows the expected behaviour of $\nu_t \sim O(y^3)$. Consequently, this observation is another indication of the correctness of its implementation. In CSMA, it starts with a high value of eddy viscosity close to the wall and then decreases gradually.

This action tends to decrease its accuracy though it boosts its robustness. One can see that the addition of the dynamic behaviour for the DSMA has benefited the eddy viscosity. The eddy viscosity affects the simulation by altering the local collision frequency.

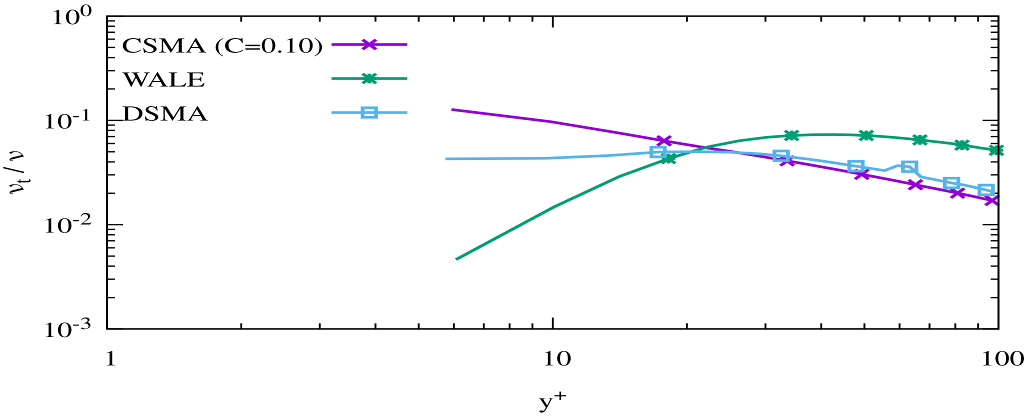


Figure 6.12: Eddy viscosity profile for the three LES models.

Figure 6.13 reports the effective collision frequency in the vicinity of the wall for the three models. As it is expected, its shape is opposite analogous to the one of eddy viscosity. Therefore, though the WALE model predicts the correct damping of turbulence close to the wall, this results in a higher value for the collision frequency at this area. A value closer to 2 in a region with large spatial derivatives and thus values of non-equilibrium part can lead to unstable results, particularly in the case of the STA model. Similarly to the vorticity calculation, the first cell has been omitted for both variables. Finally,

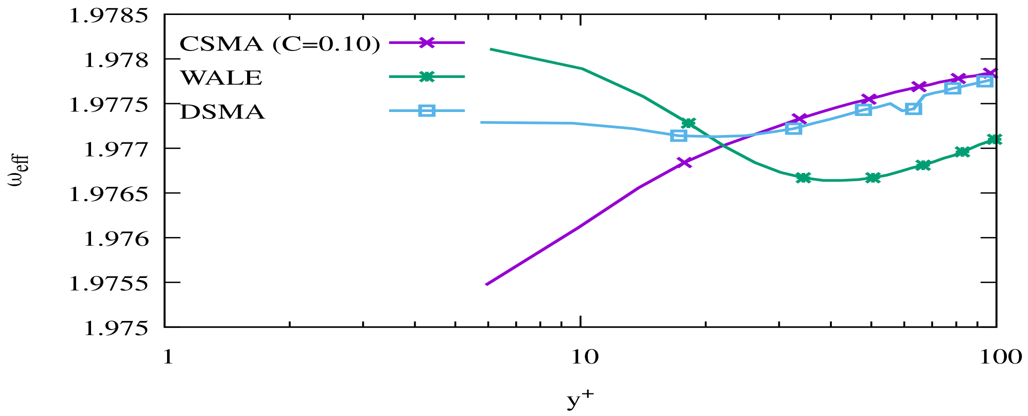


Figure 6.13: The profile of the effective collision frequency ω_{eff} close to the wall.

I will conclude this section by presenting in Fig. 6.14 the instantaneous values of the constant C calculated by the DSMA model. The locality of the algorithm has created a highly anisotropic behaviour with many regions experiencing either extrema.

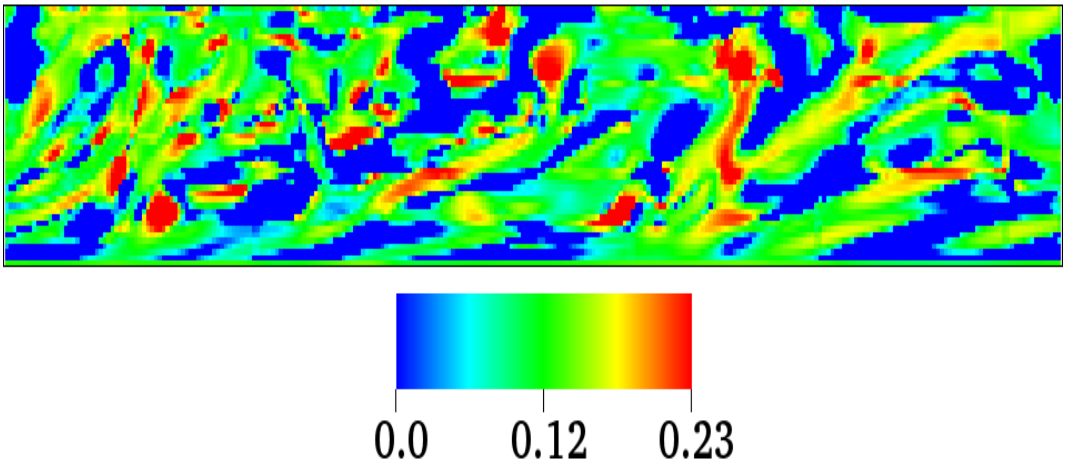


Figure 6.14: Instantaneous values of the constant C for the DSMA.

6.1.1 Evaluating the new algorithm for imposing macrovariables

In the previous section, I have illustrated the need for correct macrovariables in the ghost cells and how this can affect results. To further investigate the topic, I have also run the WALE model without applying the proposed algorithm. In Table 6.2, I present the comparison of the estimated friction velocities between the simulation with and without the application of the new algorithm. The absence of reasonable macrovariables in the ghost cells has deteriorated the accuracy of the simulation.

WALE	with algo	w/o algo
u_τ	0.0642	0.0632
Rel. Err.	0.99%	2.58%

Table 6.2: Estimated friction velocities for the WALE model with and without the proposed algorithm.

There were no evident discrepancies in the case of the streamwise velocity between the two versions. However, a variety of other variables have been altered. I will examine three of them. First of all, Fig. 6.15 shows the pressure fluctuations. It appears that the WALE version without the new algorithm has underestimated the variation of pressure in the vicinity of the wall compared to the more accurate version. The accurate estimation of pressure fluctuations, particularly around bodies, is of utmost importance for aeroacoustic applications.

The next variable to be discussed is the Reynolds shear stress $\langle uv \rangle^+$, presented in Fig. 6.16. Similarly to the pressure fluctuations, the version of WALE without the algorithm has underestimated the values. This quantity is a measurement of the intensity of the turbulence experienced by the flowfield. This discrepancy can also affect the turbulence model's behaviour away from the wall as, indeed, the eddy viscosity, plotted in Fig. 6.17, indicates. There is a discrepancy between the two versions.

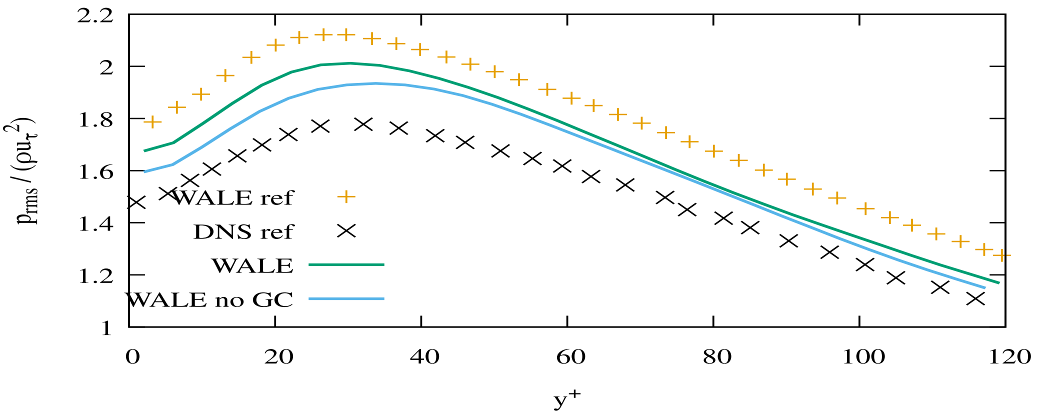


Figure 6.15: RMS pressure fluctuations for WALE scaled at wall units.

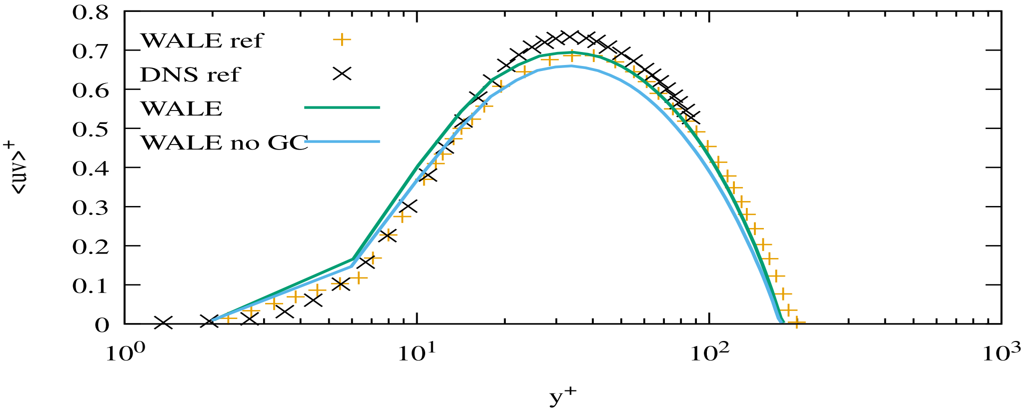


Figure 6.16: Reynolds shear stresses for WALE scaled at wall units.

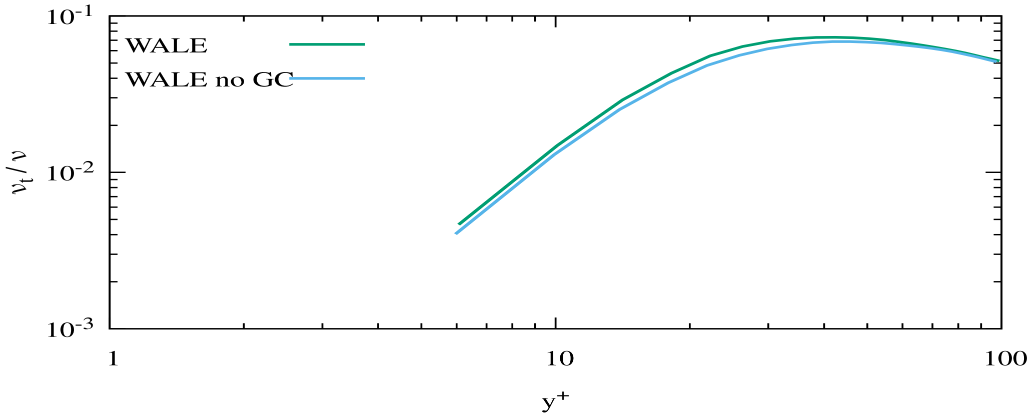


Figure 6.17: Eddy viscosity profile for WALE model.

In this specific testcase, the macrovariables at the ghost cells are affected only by a relatively small number of directions. Therefore, their values are not departed significantly from their initial ones, which are zeros. In the general case of the embedded boundaries, the streaming step may propagate distribution functions from fluid cells and other solid cells, resulting in more extreme values in the ghost cells. The latter can create an undefined behaviour for the turbulence model's response in the vicinity of the wall.

6.2 Wall Modelled LES

In the previous section, I simulated the bi-periodic channel using WRLES to verify the new models' implementation further. In this section, I will focus on examining and verifying the other new implementation, that is, the wall function.

Initially, I will replicate the numerical experiment presented in (Haussmann et al., 2019) of a bi-periodic channel with $Re_\tau = 1000$ or $Re_{bulk} \approx 39300$. I chose this reference because it provides data using an STA SRT LBM solver without calculating the non-equilibrium part in the first fluid cell, thus yielding a fair comparison for the current algorithm. Following their example, I have also carried out a convergence analysis by running three resolutions. This specific set-up will test the wall function under typically expected conditions.

As the next step, I will show data obtained using the same resolutions but with $Re_\tau = 20000$ or $Re_{bulk} \approx 1.2 \cdot 10^6$. The results will be compared against the ones in (Malaspinas and Sagaut, 2014). This set-up will let us examine the performance of the algorithm under very coarse resolution.

6.2.1 WMLES ($Re_\tau = 1000$)

This test case's numerical domain will be the full bi-periodic channel, contrary to the half channel employed in the previous section. This is to increase the number of cells and thus improve the accuracy of the averaging procedure, particularly for the coarsest resolution. In this case, the dimensions will be $(6H \times 2H \times 6H)$. Three resolutions will be tested with $N = 10$, $N = 20$ and $N = 40$ per H, respectively.

Following the reference, I also ran the simulations for 550 channel passages (cp) with $cp = 6H/u_m$ and $u_m = 1\ m/s$ and I did statistics only for the last 150 cp. I kept the ensemble frequency 25 Hz. One important difference between my set-up and the one in the reference is that they used diffusive scaling $\Delta t \sim \Delta x^2$ to propagate from one resolution to the other. This means that by doubling the resolution, the Ma number is halved. They chose this scaling due to their AMR implementation. In this case, I used advection scaling $\Delta t \sim \Delta x$ because it is in agreement with the AMR provided by AMROC. In this way, the Ma number is the same for all resolutions. In other words, for the coarsest resolution $N = 10$ my time step is approximately the same as theirs $\Delta t_{\text{AMROC}} \approx \Delta t_{Ref}$, but for the other two resolution is a multiplication, for $N = 20$, $\Delta t_{\text{AMROC}} \approx 2\Delta t_{Ref}$ and for $N = 40$, $N = 20$ $\Delta t_{AMROC} \approx 4\Delta t_{Ref}$. I ran all the simulations with $Ma = 0.1$.

The initialisation of the domain was based on the wall function similar to the WRLES case. I will compare my results with the STA SRT LBM data with CSMA as turbulence model reporting in (Haussmann et al., 2019). They used $C = 0.12$, while in this case, I set the constant equal to 0.1. They have reported data with and without the calculation of the non-equilibrium part in the first fluid cell in this reference. The data that I will plot in this section are the ones without the inclusion of the non-equilibrium part. This will yield a fair comparison with my data, given that my implementation also assumes zero values for the non-equilibrium part, see Sec. 4.4. Moreover, the NS DNS data of (Lee and Moser, 2015) will also be provided.

Before discussing the results, it is essential to mention that, as in the case of the WR-LES, the estimation of the force and particularly the averaged streamwise velocity $\langle u \rangle$ evaluated before the call of the step function. This means that this calculation lags one Δt. Moreover, in this case, the boundary condition is applied in the first fluid cell resulting in random macrovariables after the streaming for this location. These unknown values would result in inaccurate calculation of $\langle u \rangle$ and thus the force. To alleviate the problem, I have imposed a zero velocity in the ghost cells.

Starting with the lowest resolution of $N = 10$, Fig. 6.18 presents the mean averaged streamline velocity scaled in wall units. In this scenario, y^+ at the first cell is around 50 in the log-layer. It appears that there is no significant discrepancy among the LES models, while their prediction is almost identical to the reference CSMA. I also plot the Reynolds normal stress in Fig. 6.19. In this case, the LES models have underestimated the quantities, particularly in the vicinity of the wall, while they seem to catch up with the reference solutions close to the centreline.

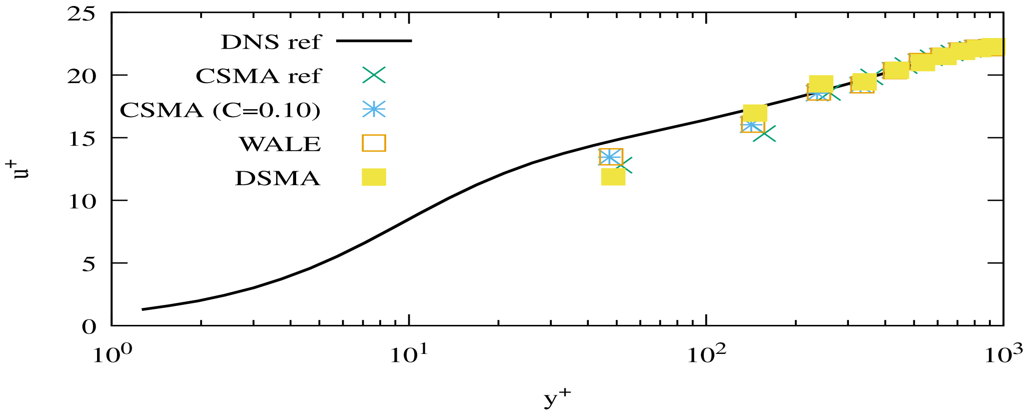


Figure 6.18: Comparison of mean averaged streamline velocity for $N = 10$.

Figures 6.20 and 6.21 show the corresponding plots for the resolution of $N = 20$. The y^+ at the first cell is around 20, inside the buffer layer. Observing the velocity profile obtained by the three LES models coincides very accurately with the reference data, with slightly overestimated values around $y^+ \approx 200$. On the other hand, WALE and DSMA can damp the oscillations appearing in the Reynolds normal stresses for both CSMA simulations.

Finally, Figs. 6.22 and 6.23 report the data for the finest resolution. The predicted y^+ at the first cell is less than ten and seems lower than the reference. Furthermore,

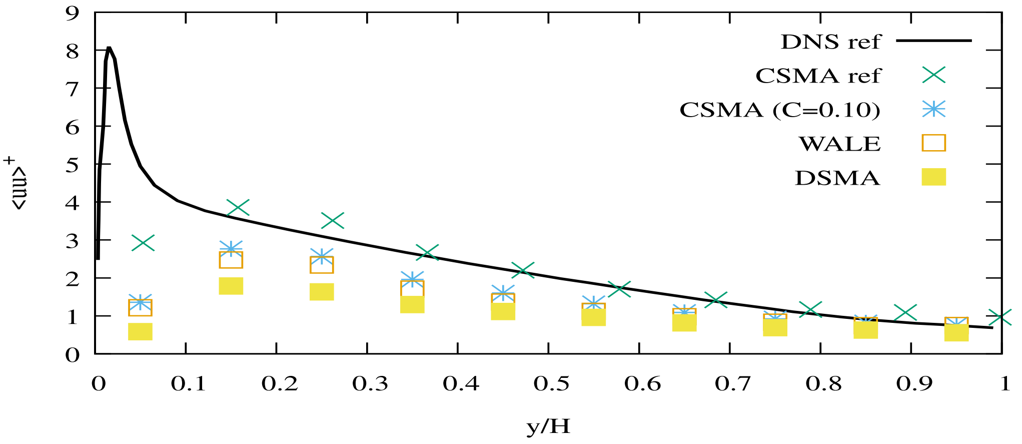


Figure 6.19: Comparison of mean averaged Reynolds normal stresses for $N = 10$.

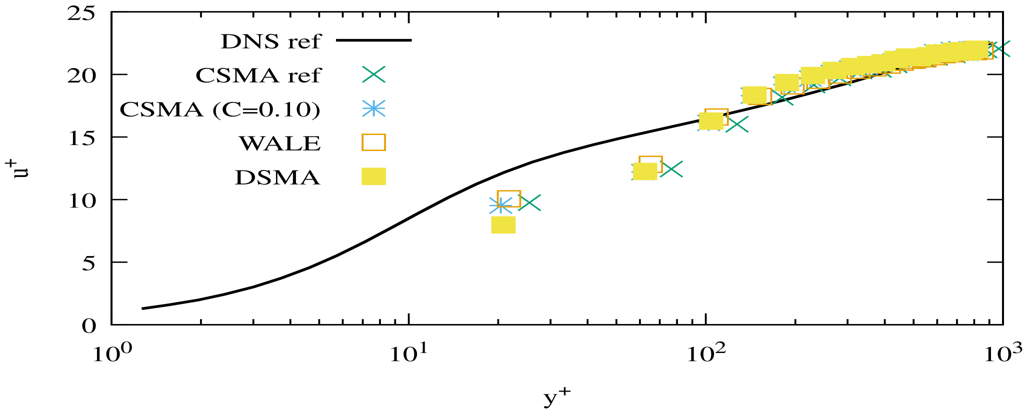


Figure 6.20: Comparison of mean averaged streamline velocity for $N = 20$.

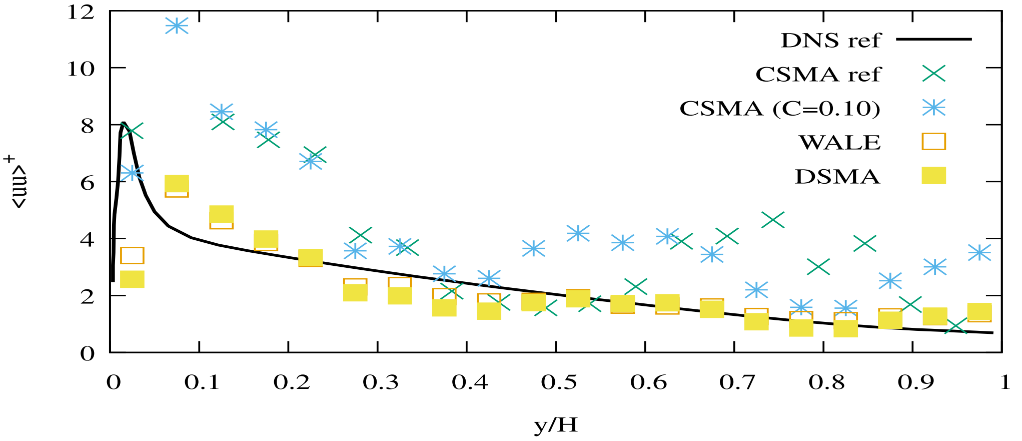


Figure 6.21: Comparison of mean averaged Reynolds normal stresses for $N = 20$.

none of the LES models achieved to capture the profile correctly, with WALE to show the largest divergence. The situation is much better for the Reynold normal stresses with some discrepancy close to the wall for the DSMA and WALE. It is well known that the wall function has increased error in the region of the buffer layer (Haussmann et al., 2019). Additionally, I have noticed that the prediction of the friction velocity u_τ by all three LES models were getting worse and worse with increasing resolution, with actually 40% error for the finest one. One can speculate that the main reason for this divergence is the higher temporal step, four times in the finest resolution, and Ma number compared to the reference solution in combination with the first cell lying in the buffer layer.

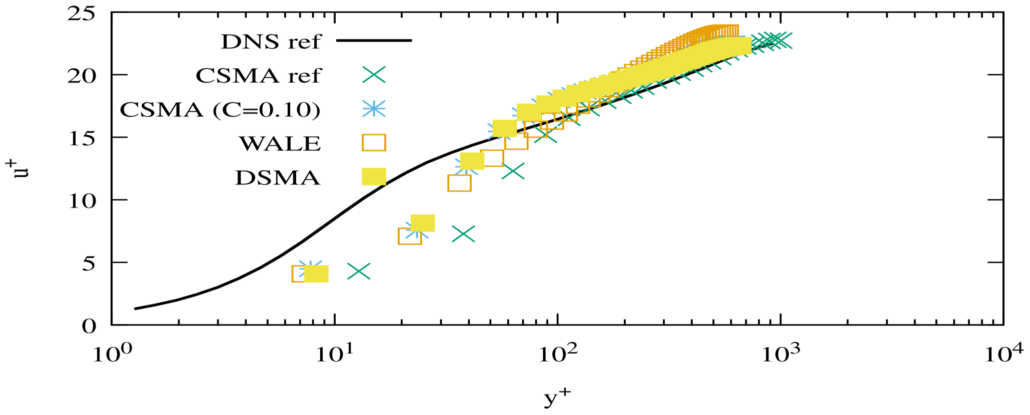


Figure 6.22: Comparison of mean averaged streamline velocity for $N = 40$.

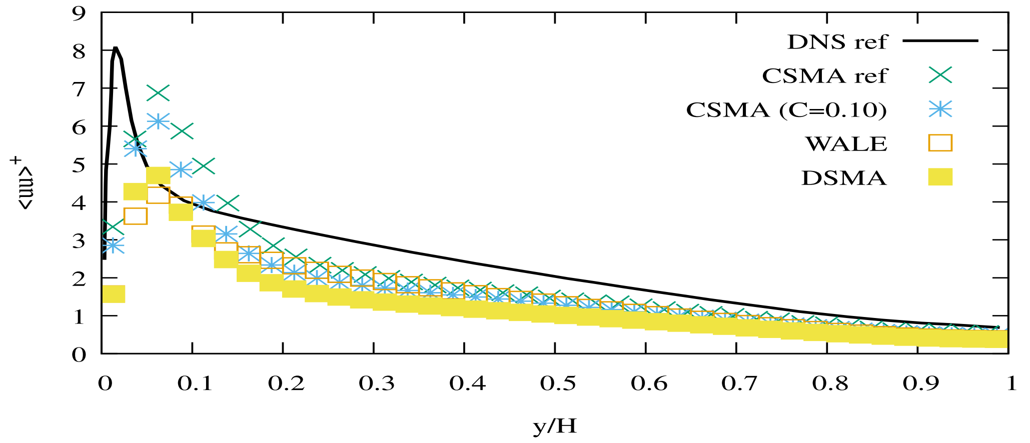


Figure 6.23: Comparison of mean averaged Reynolds normal stresses for $N = 40$.

An important observation is that this boundary does not conserve mass. This is in agreement with the reference (Haussmann et al., 2019). They have stated that this inaccuracy is the extrapolation of the density from the next fluid cell instead of using the more accurate algorithm proposed in (Zou and He, 1997) of estimating the density. However, I used the less accurate method due to the more straightforward extension to non-Cartesian boundaries. Increasing the resolution has shown to alleviate this issue, with a loss around $\sim$ 20% in the case of $N = 10$ reducing to $\sim$ 5% for $N = 40$ at the end of the simulations.

Another interesting observation is that none of the models under any resolution could accurately predict the Reynolds shear stresses $\langle uv \rangle$, miscalculating them by orders of magnitude. In the previous section, I have shown that all LES models managed to predict this quantity accurately. This result indicates that both their implementations and the set-up of this testcase coupling with the post-processing algorithm are sound. This conclusion points the blame to the direction of the algorithm of the wall function. However, the model imposes a zero normal velocity at the first cell and, in its current implementation, zero non-equilibrium part that could directly affect the calculation. Therefore, one can speculate that by not applying the collision step, since $f_i^{neq} = 0$, and thus applying the force in the first fluid cell, this has a negative effect on shear stresses. Another difference with the reference that might have affected the outcome is the application of the force without using the proposed way of Guo et al Guo et al. (2002a).

Finally, similarly to the first section, Fig. 6.24 shows the instantaneous values of the constant C calculated by DSMA for $N = 40$. In comparison with Fig. 6.14, it appears that the number of small scales has increased in the flowfield, an expected outcome due to the higher Re_τ. Again, it is clear that the locality of the algorithm leads to high levels of disparity because of not averaging in homogeneous directions.

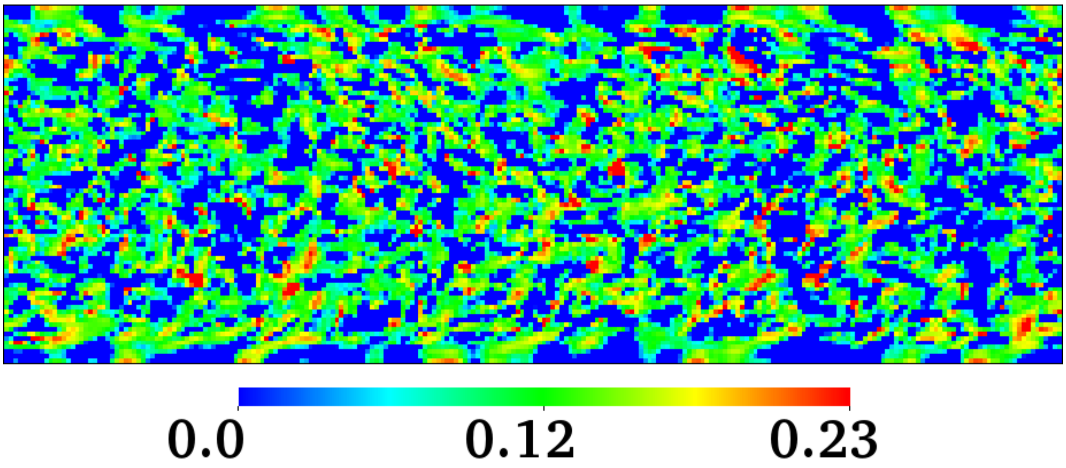


Figure 6.24: Instantaneous field of the constant for the DSMA model at Re 1000 and N40 resolution.

6.3 WMLES ($Re_\tau = 20000$)

In the previous section, I investigated the newly implemented wall function's performance to deal with flows where the first cell lies either in the buffer layer or at the beginning of the log-layer. The later condition that is a y^+ around 50 is the expected scenario for simulations.

This section will test the wall function under more extreme conditions, where $y^+ > 100$ in the first cell. To do so, I will replicate the numerical experiment of the bi-periodic channel at $Re_\tau = 20000$ reported in (Malaspinas and Sagaut, 2014) and compared with the given data. The numerical domain and the three resolutions are the same as in the previous case. The initial condition is again based on the wall function imposed in the whole domain. Due to the relatively high Re_τ, the constant in the case of CSMA has been set to $C = 0.25$ and for the WALE model $C_w = 0.7$. These values are significantly higher than the expected ones, but I employed them due to the coarseness of the mesh. The reference CSMA model also uses the former value of CSMA. Another difference with the reference (Malaspinas and Sagaut, 2014) is the use of the MRT collision model. Finally, since there are no DNS data for this case, I will compare it with the Musker profile.

Figures 6.25, 6.26 and 6.27 report the results for the three resolutions. There is no significant discrepancy between the LES models and the reference data, with the WALE model having delivered the closer results and CSMA the largest deviation.

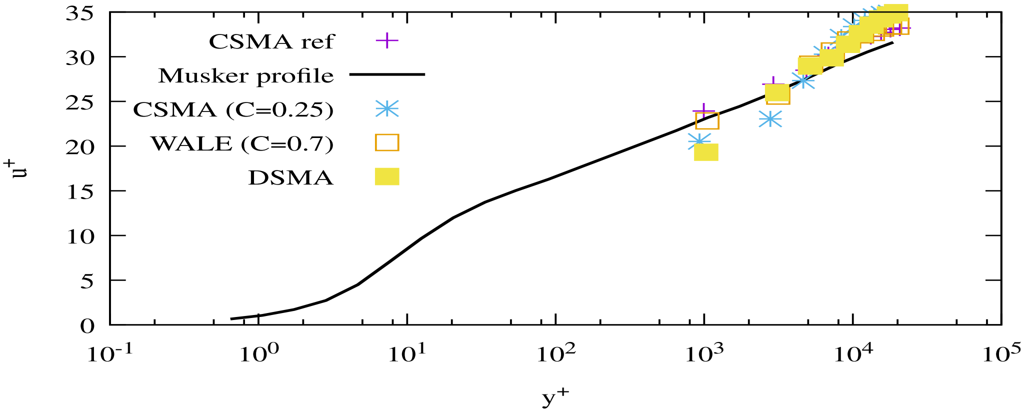


Figure 6.25: Comparison of mean averaged streamline velocity for $N = 10$ for $Re_\tau = 20000$.

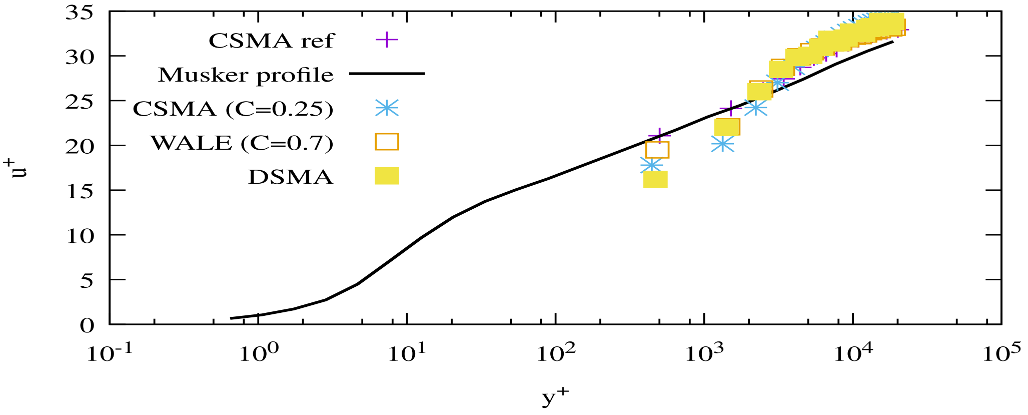


Figure 6.26: Comparison of mean averaged streamline velocity for $N = 20$ for $Re_\tau = 20000$.

Taking everything into consideration, I have illustrated that the current implementation of the wall function can capture the law of the wall accurately, at least when the first

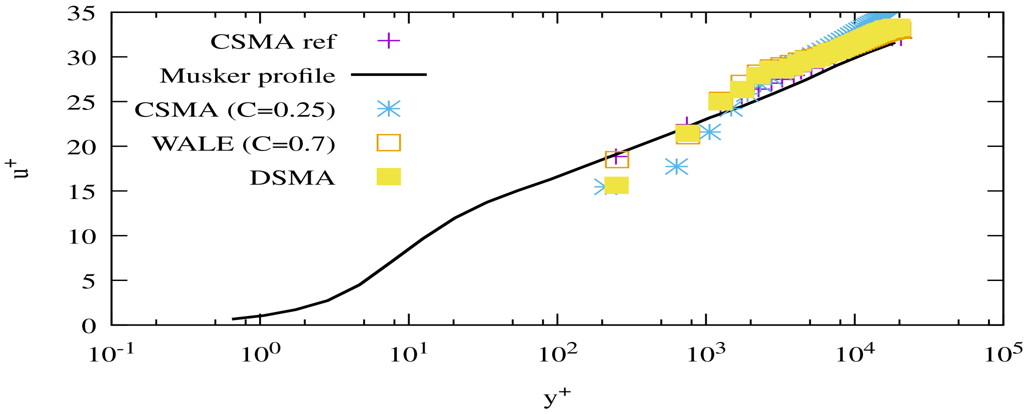


Figure 6.27: Comparison of mean averaged streamline velocity for $N = 40$ for $Re_\tau = 20000$.

cell lies in the log-layer. It is also reasonably accurate under a coarse resolution. This applies to all LES models though WALE has returned the most accurate data. Though it returns a reasonable prediction for the normal stresses, I have also noticed that it has failed to capture the Reynolds shear stresses. I speculate that this issue is a side-effect of the current set-up. However, the absence of the non-equilibrium part means that the implemented algorithm is only first order accurate.

Chapter 7

Cylinder on Plate

In the previous chapter, I have verified the newly implemented wall-function algorithm under the condition of a mesh aligning with the body. This chapter will enrich the verification by running a scenario with a body non-aligned to the Cartesian mesh.

We chose to simulate one of the experiments conducted in the subsonic aerodynamic wind tunnel (TA-3) at the Instituto de Aeronáutica e Espaço (IAE) in Brazil. An internal report describing the facility and the results were provided to us by the collaborators (Faria and Francisco, 2018; Faria et al., 2019). The motivation of the experiments was to test the interaction of an atmospheric boundary layer with a building and particularly to examine the wake. To do so, the experimental set-up consists of square cylinders mounted on a flat plate. The method of PIV was used to extract the data. A variety of configurations have been tested with different size of cylinders and angles of attack.

The input parameters that describe the case I have simulated can be found in Table 7.1. The body has an angle of 45^o with the Cartesian grid. Due to this non-alignment, the length used to estimate the Re number is the diagonal length $D^2 = 2L^2$. The apparent Ma number is $Ma = 5/340 \approx 0.015$. However, I have accelerated the flow internally by employing the speed-up factor with a value of 6. In this way, the Mach number is $Ma = 6 \cdot 5/340 \approx 0.09$, and the simulation is six times faster.

Height	240 (mm)
Length (L)	40 (mm)
AoA	45^o
U_∞	5 (m/s)
ν	$1.54 \cdot 10^{-5}$ (m^2/s)
Re	18367
c_s	340 (m/s)
Speed-up	6

Table 7.1: Input parameters for the Cylinder case.

To ensure that the domain boundaries would no affect the simulation, a relatively large domain of ($127D \times 70D \times 35D$) has been utilised. As for the boundaries themselves, I imposed inlet on the left face and outlet to the right, front, back and top, see Sec. 4.2. The bottom face, where the cylinder is mounted, was modelled by a no-slip wall to represent better the experiment. The position of the cylinder was $42D$ away from the inlet or at one-third of the length of the domain. As for initial conditions, I imposed U_∞ everywhere.

The resolution was ($720 \times 400 \times 200$) or $57.6 \cdot 10^6$ cells, and besides, I have also used four levels of AMR with a ratio of 2 between the levels. If I had used a uniform grid based on the finest level, the simulation would end up with $\sim 3 \cdot 10^{10}$ cells. In the beginning, the simulation was run for 2 sec or 177 cylinder passes to establish the flow. Then, I restarted and ran the simulation for the successive seven passes and saved data with a frequency of three savings per pass. Following the experimental procedure, an averaged flowfield has been estimated based on these instances. Figure 7.1 shows the evolution of the adaptive mesh after the first 2 sec in a plain horizontal to the cross-section of the cylinder. This plain is 220 mm from the base of the cylinder and is the plain where the PIV data were measured. In Fig. 7.2, one can see the mesh's evolution in the other direction. The expansion of the finer levels emulates the wake beside the cylinder. The resulting mesh pictured in these two figures consists of $\sim 96.4 \cdot 10^6$ cells.

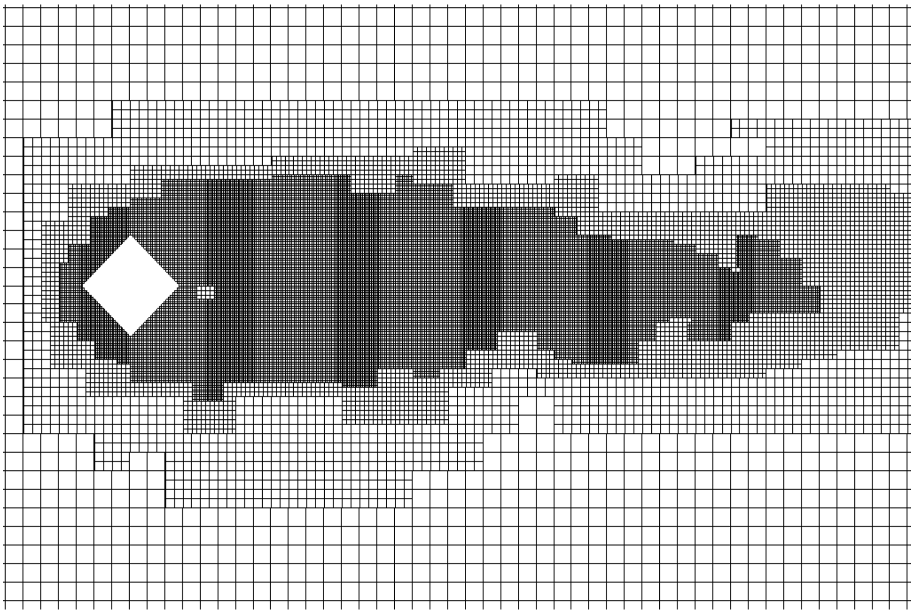


Figure 7.1: The development of the adaptive mesh after 2 sec in a horizontal plain.

The final number of cells seems to be relatively large for the given Re number. Moreover, the resulting value of the y^+ at the first cell appears to be ≈ 25 based on the calculation of the on-line tool provided in the site (CFD ONLINE). As I have demonstrated in the previous chapter, a more accurate value would be above 50. However, running the

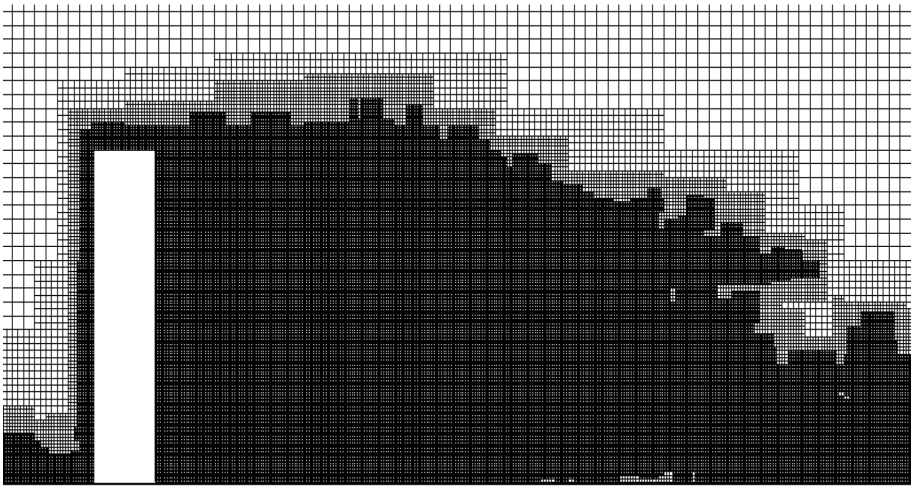


Figure 7.2: The development of the adaptive mesh after 2 sec in a vertical plain.

test case with such a resolution has ended up with instabilities throughout the whole domain, and no valid data could be extracted. Therefore, to stabilise the simulation, I increased the resolution. Simultaneously, only CSMA has been used with a relatively large constant of $C = 0.25$ as the most dissipative and stable turbulence model. One source of instabilities was inaccuracies in the interfaces between different levels leading to a possibly non-conservative behaviour. Nonetheless, one can speculate that using a more robust collision model rather than the two available ones, STA and REG, will improve the situation.

The first variable to be examined is the vorticity ω_z plotted on the PIV measurement's horizontal plain. Figure 7.3 shows the averaged vorticity measured during the experiment. Figure 7.4 plots the estimated averaged vorticity of the LBM solver. The simulation seems to have captured quite accurately the expected shape, though the size appears to be significantly smaller. One can speculate that the main reason for this difference is the exaggerated eddy viscosity provided by the CSMA model due to the relatively high value of the constant.

In Fig. 7.3, there are five lines vertical to the flow indicating the five positions where the streamline and parallel to these lines velocities have been measured. The labels describing these positions are CASE 1 to CASE 5, with CASE 1 being closer to the body. Figure 7.5 shows the comparison of streamline velocities between the experiment and the simulation. It is evident that the simulation accurately captured the velocity profiles particularly close to the body. In the case of 4 and 5, the simulation predicted recovery of the deficit faster than the experiment. As in the vorticity case, I attribute this discrepancy to the turbulence model's excessive dissipation.

In the case of the longitudinal velocity, the simulation diverges visibly from the experimental data. For the first two cases, it predicts a strong velocity field exactly behind the

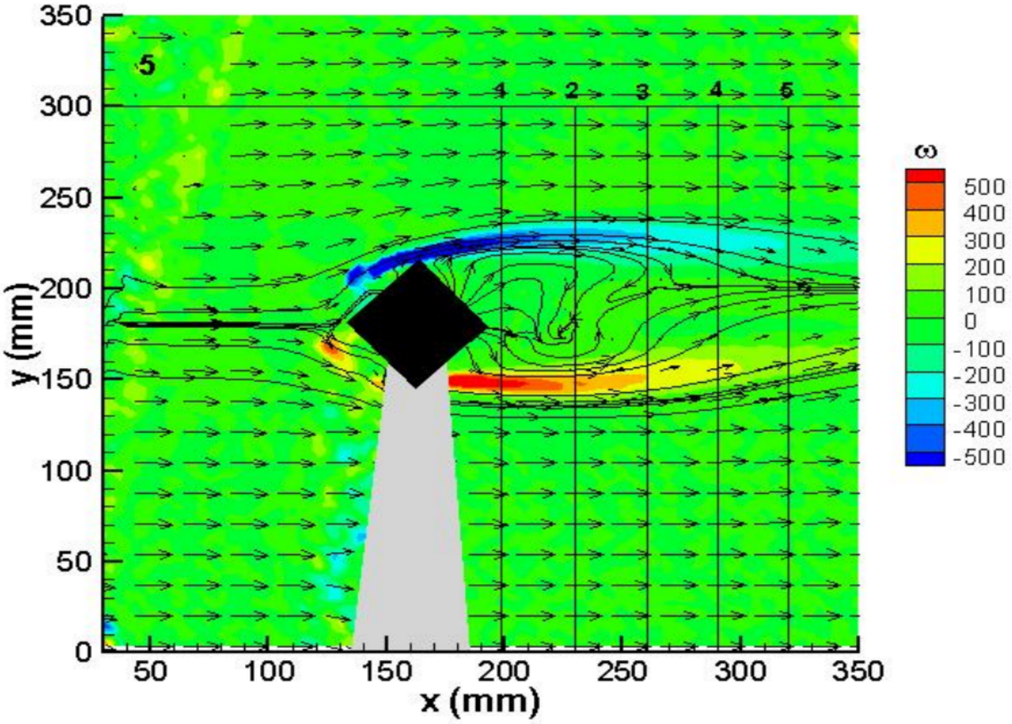


Figure 7.3: Vorticity maps created based on the experimental data (Faria and Francisco, 2018).

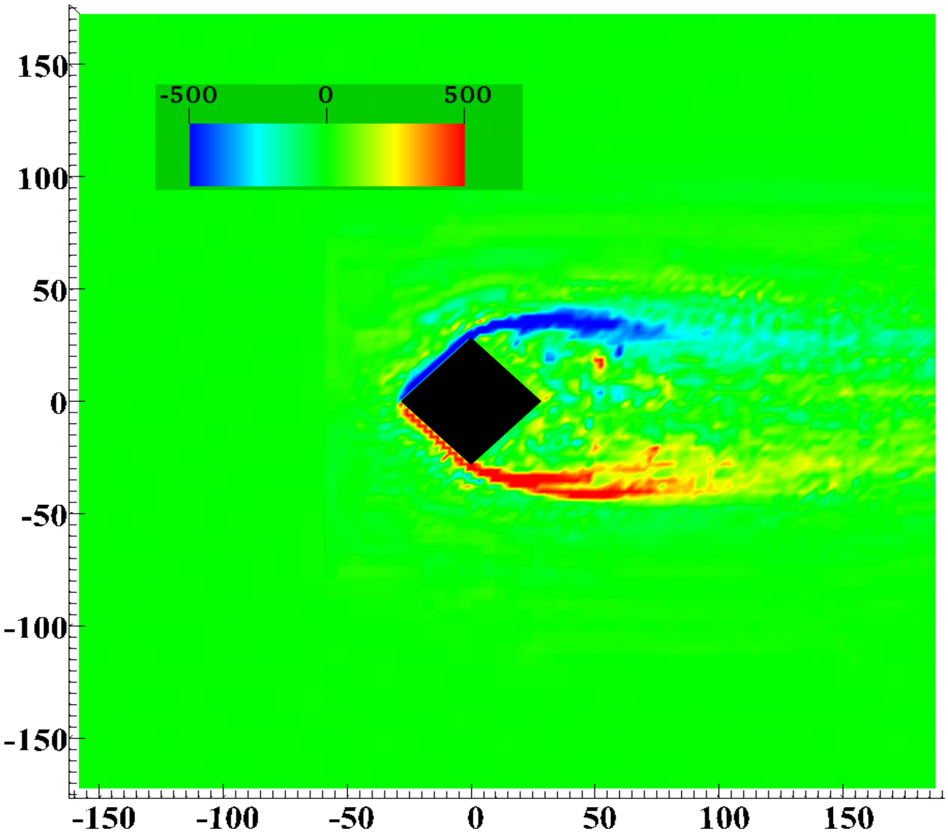


Figure 7.4: Vorticity maps estimated by AMROC LBM solver with CSMA $C = 0.25$.

cylinder, while according to the experimental data, the values there should be around zero. As the positions move away from the body, the flow appears to become more energetic, and the experimental data approach the results from the simulation at least on one side of the cylinder. However, contrary to the smooth transition between the two sides of the cylinder observed in the experiment, the simulation has returned a more abrupt one.

To close this chapter I also report the averaged streamline longitudinal and parallel to the cylinder velocities in Figs. 7.7, 7.8 and 7.9, respectively. I also present the vorticity's instantaneous magnitude in Fig. 7.10 estimated after the first 2 sec. The simulation has captured the expected features in front of the cylinder and the wake. Nonetheless, it is apparent that even averaging the data, the instabilities can be seen in some of these plots, particularly in the regions with the accelerated flow.

The main focus of this chapter was to verify the wall function for non-Cartesian geometry. The currently available collision models, AMR interface, and the first order accuracy of the wall function seem to have prevented a smooth flowfield. However, the wall function was the only boundary condition that stabilised the simulation and provided some useful data under the current set-up. For other solid boundary conditions to be employable, one should increase the resolution close to ten times, based on the bi-periodic channel experiment. Such a simulation will be non-available due to high computational cost. The other option is to increase further the constant of the turbulence model. Nonetheless, this will reduce the method's accuracy, further shrinking the wake behind the cylinder. Therefore, one can conclude that currently, only the wall function has provided a way to run a relatively high Re number simulation involving complex geometry with the available collision models.

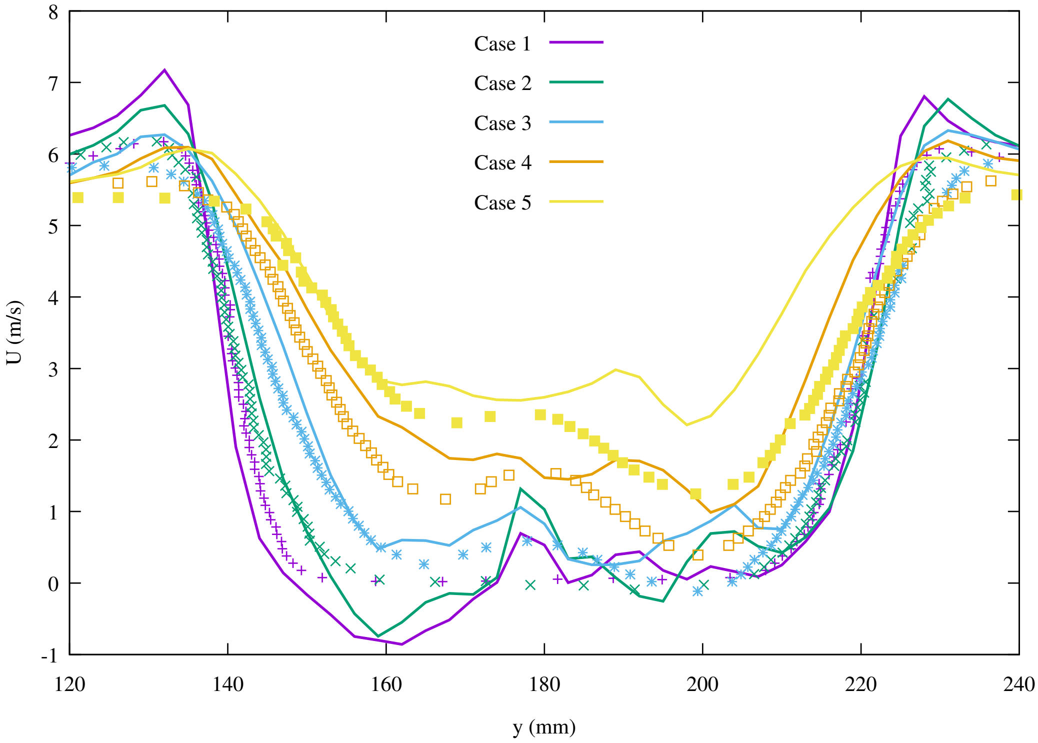


Figure 7.5: Comparison of mean averaged streamline velocity. The points belongs to the experimental data (Faria and Francisco, 2018) and the solid lines are predictions of the current solver.

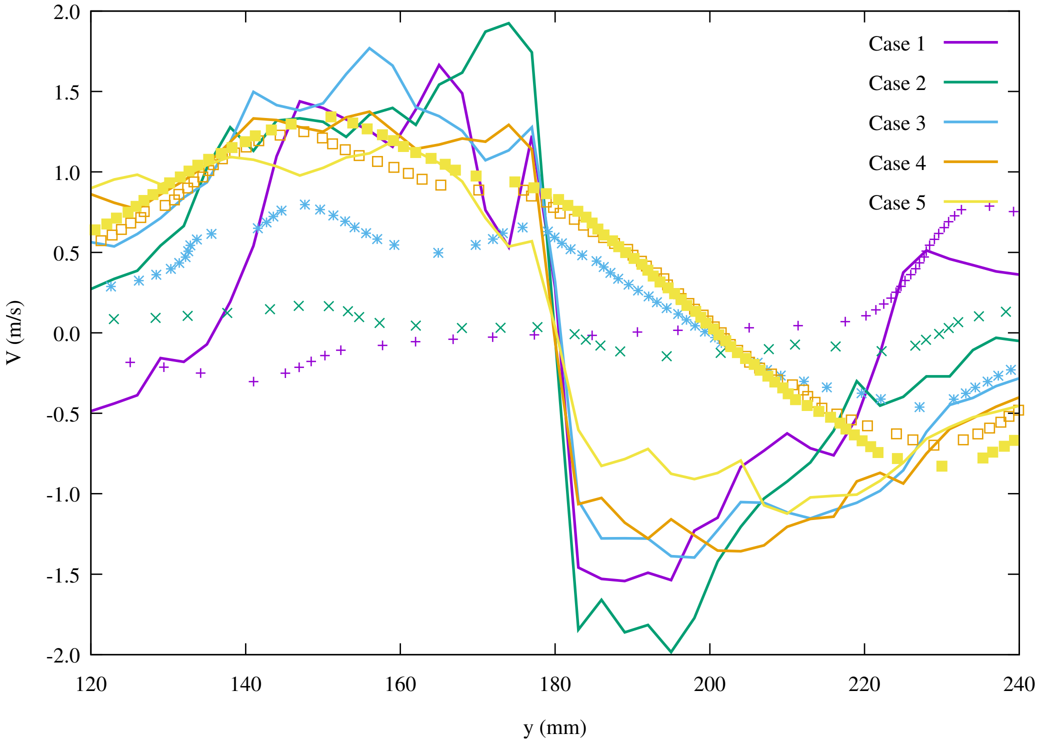


Figure 7.6: Comparison of mean averaged longitudinal velocity. The points belongs to the experimental data (Faria and Francisco, 2018) and the solid lines are predictions of the current solver.

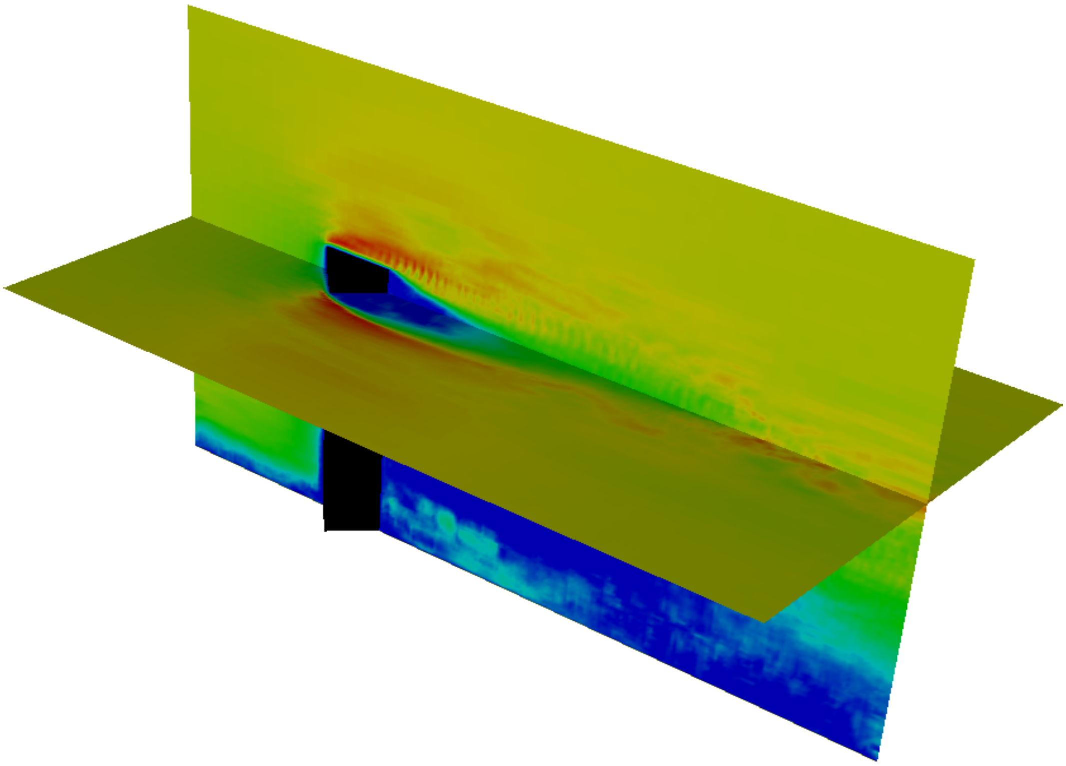


Figure 7.7: The averaged streamline velocity.

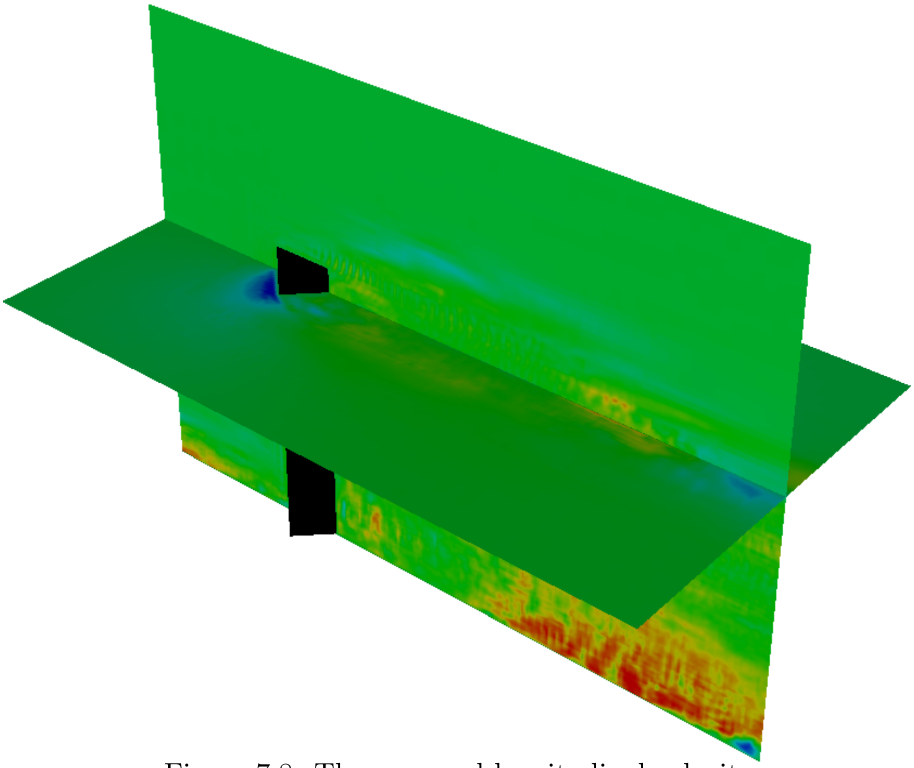


Figure 7.8: The averaged longitudinal velocity.

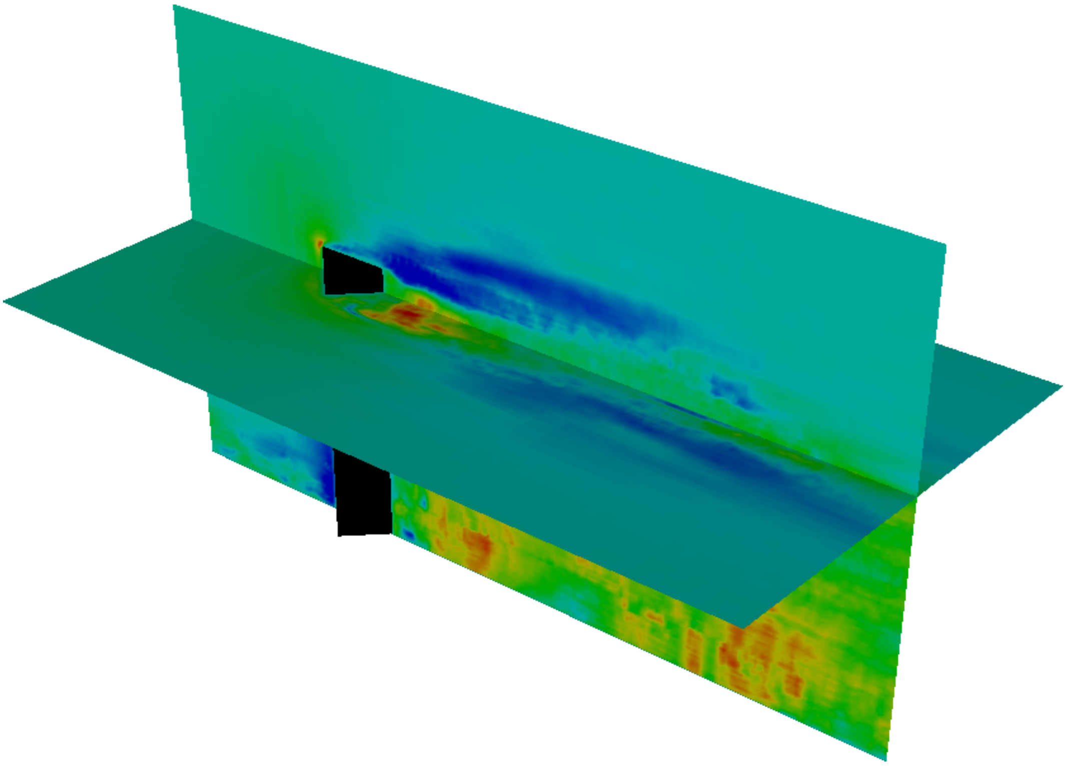


Figure 7.9: The averaged velocity parallel to the height of the cylinder.

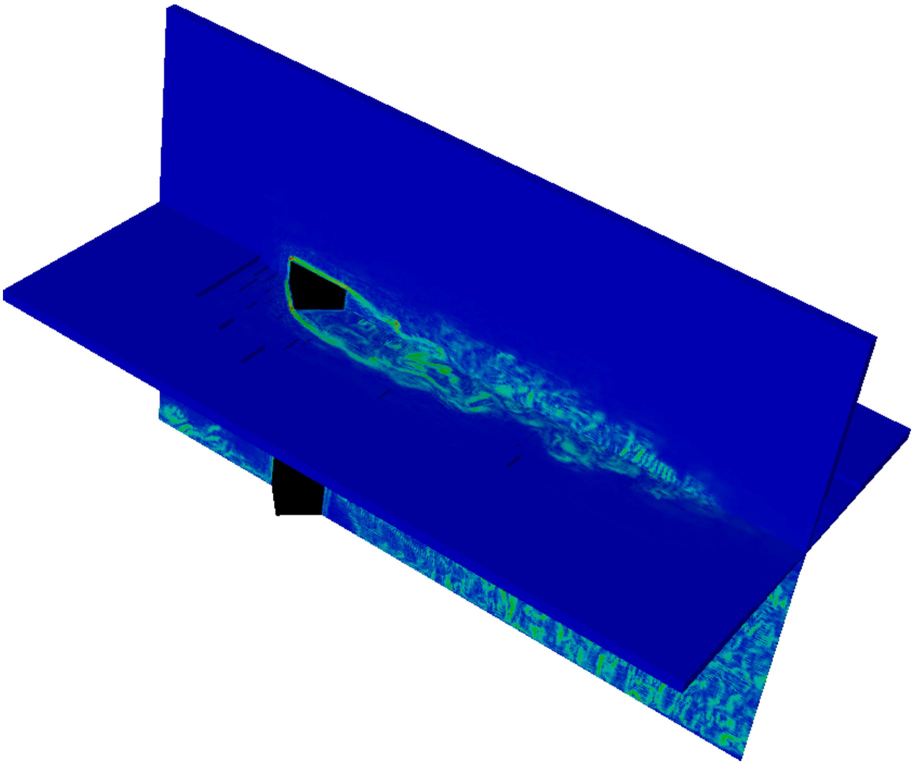


Figure 7.10: Instantaneous magnitude of the vorticity.

Chapter 8

Conclusions and Future Work

This project's target was to provide the essential tools for the AMROC LBM solver to deal with high Re turbulent flows. This goal has been achieved through the implementation of a variety of LES models and a wall treatment. Various test cases have been employed to verify the new implementations, resulting in valuable data and research findings. Simultaneously, the algorithmic environment provided by the solver's interface with the AMROC framework led to extra challenges, see Chap. 1.1. By dealing with these challenges, a new algorithm was devised. The author will discuss the contributions, findings, and innovation that this project has provided in the following section. Finally, a small section about future work will conclude this chapter.

8.1 Key Contributions

During this project, three LES models have been implemented. The first model is the constant Smagorinsky, and its implementation in the AMROC-LBM solver has been discussed in Sec. 3.2. Its implementation provides two versions based on the way the strain rate is evaluated. The first option is due to the non-equilibrium part of the distribution function, and the second option is based on a finite-difference stencil. The second turbulence model is the dynamic Smagorinsky detailed in Sec. 3.3. The third model is the wall-adapting local eddy-viscosity (WALE) discussed in Sec. 3.4.

To add further options for the solver, the author developed the REG SRT model for AMROC LBM, see Sec. 2.2.3. Moreover, two new approaches for inlet and outlet boundary conditions have been implemented and reported in Sec. 4.2. A significant addition for LES and high Reynolds turbulent flows was the development of a wall treatment based on the Musker velocity profile. Its implementation in the AMROV-LBM solver is discussed in Sec. 4.4. Furthermore, the addition of the force scheme for the FHIT test case, reported in Sec. 5.2.1, has enabled the solver to simulate homogeneous isotropic turbulence, a benchmark test case that can be used for the verification of other LES models. Similarly to homogeneous isotropic turbulence, the bi-periodic channel is another benchmark testcase.

Finally, for data collection in the cases of homogeneous isotropic turbulence and bi-periodic channel, two vital post-processing functions have been implemented. In the former case, the author coded a routine for evaluating power spectra and other turbulent statistics, see Sec. 5.1. For improved performance, the power spectra were calculated during simulations. MPI communication enabled the function to run in parallel. In the latter situation, a routine, see Chap. 6, with low storage requirements, was implemented to estimate the mean averaged values and other turbulent characteristics.

For the verification of the LES models and the wall treatment, various test cases have been simulated, and several observations have been collected. In this section, the most important research findings of this project will be reported.

In the situation of the FHIT test case, Sec. 5.2, an important outcome was the minimum resolution of $\kappa_{max}\eta \geq 5$ in order for an LBM solver to simulate the dissipation range adequately. This value was estimated using the model spectrum as a reference; see Sec. 5.1.1. Another important observation was the more energetic energy spectra of the STA model compared to REG. However, it was found that this behaviour was mainly due to instabilities appearing in the pressure fluctuation spectra. To the knowledge of the author, this is the first time that this finding is reported.

For the case of DHIT, Sec. 5.3, the author employed as an initial solution the results from the FHIT case. This was an efficient way to set up DHIT compared to the option of an initial fabricated solution that may need initialisation of the non-equilibrium part of the distribution function. Another observation was the LES models' consistency that converged to the DNS reference data for increased resolution. A significant finding was the discrepancy of the results between the local and stencil CSMA implementations for the STA model. The REG model did not experience this behaviour. The two essential outcomes are that the collision model may affect the LES model's accuracy, and depending on the collision model, the way that the strain rate is evaluated may yield different results.

Simulating TGV, Sec. 5.4, it was concluded that the STA collision model yields a less dissipative behaviour, though this observation is partially due to instabilities appearing in the pressure fluctuation spectra. An extensive analysis of the eddy viscosity field produced by the LES models and how it adapts to flow features has been carried out and reported in Sec. 5.4.2. It was also found that the two collision models estimated similar average values for the constant of DSMA in the domain during the simulation. Moreover, DSMA and WALE appeared to overact in the TGV case's initial laminar stage, resulting in a depressed flowfield in later stages compared to CSMA.

In the bi-periodic channel, all three LES models were successful in performing WRLES, see Sec. 6.1. The WALE model appeared to be the most accurate, returning the expected slope of the eddy viscosity in the vicinity of the wall. For the application of the WMLES cases, Sec. 6.2, the wall treatment was found to capture reasonable well the first order statistics and normal Reynolds stresses. However, it was unable to capture the Reynolds shear stresses for any resolution.

Finally, in the square cylinder, Chap. 7, the wall treatment with CSMA successfully coupled with the AMR algorithm. However, it was apparent that more advanced collision models are needed to simulate high Reynolds number flows.

The application of *bounce-back-type* boundaries, see Sec. 4.1, in combination with LES that requires a stencil operation was found to be problematic. The reason is that a *bounce-back-type* boundary will not impose a specific set of macrovariables in a ghost cell, resulting in unknown values. Consequently, the application of a stencil operation in a neighbour cell will yield questionable results. To fix this issue, the author has devised and proposed a new algorithm that utilises the unused lattice directions and imposes a set of known macrovariables. The algorithm is reported in 4.3 and it is evaluated in Sec. 6.1.1. This new proposal can be seen as an extension of the ghost-fluid-LBM approach proposed initially by (Tiwari and Vanka, 2012).

Future Work

As for future work, the author suggests three main directions:

1. **Collision Models.** During this project, it became apparent that the combination of STA or REG SRT collision models with LES was not a fully robust approach for high Reynolds turbulent flows. The author highly suggests implementing more advanced collision models, see Sec. 1.2. The three test cases of homogeneous isotropic turbulence and the bi-periodic channel provide a potent tool for testing and validating combinations of LES and collision models.

2. **Boundary Conditions.** The author has discussed extensively in the course of this thesis how the algorithmic environment of the AMROC framework affects the solver. However, the most affected aspect of it is applying the boundary conditions, see Chap. 4. Two examples are the non-equilibrium part in the wall treatment and the REG SRT model's interaction with boundaries. Consequently, the author strongly recommends more research into this direction. Given that the approach of boundary conditions in the AMROC-LBM solver is relatively uncommon in the LBM community, more investigation may yield significant contributions.

3. **LES modelling.** With a local estimation of the strain rate, CSMA appears to be the most robust LES model currently in the AMROC-LBM solver. Its robustness arises mainly due to the locality of its application and thus less exposure to inaccuracies in neighbour cells. In the AMROC-LBM solver, a source of such inaccuracies is the boundary conditions. The new algorithm for imposing macrovariables has enabled LES models based on a stencil, such as WALE, to perform in such an environment. However, DSMA is still affected by inaccurate boundaries mainly due to its extensive stencil. Therefore, the author suggests the investigation of further LES models that are based on a stencil of macrovariables such as the WALE model without the need for extensive stencils or accurate non-equilibrium parts in the neighbour cells.

Appendix A

More Results about HIT

The purpose of this appendix is to provide future researchers with more data for comparison. These data have been collected during FHIT, DHIT and TGV simulations, reported in Chap. 5.

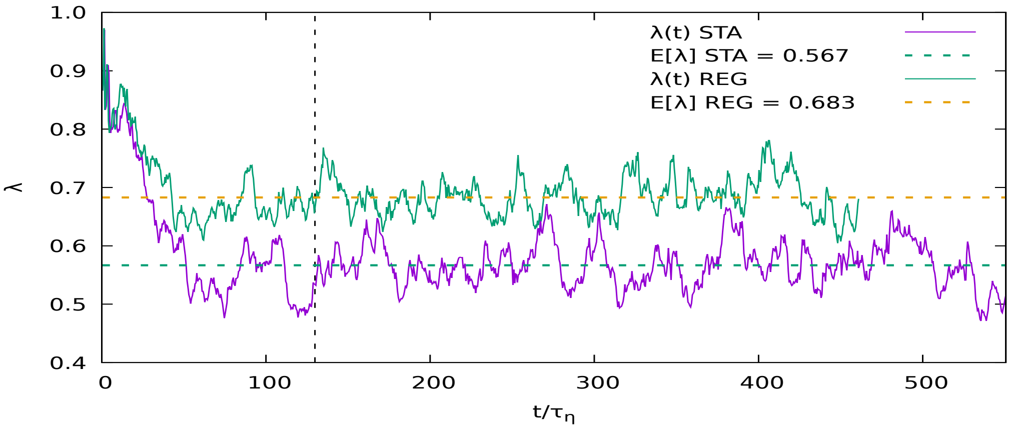


Figure A.1: Evolution of the Taylor length scale for STA and REG DNS of $N = 32$ resolution.

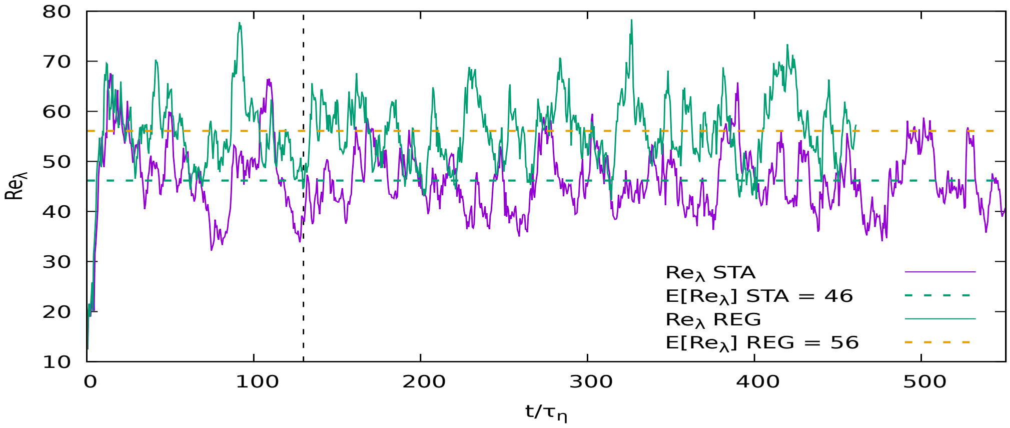


Figure A.2: Evolution of Re number for STA and REG DNS of $N = 32$ resolution.

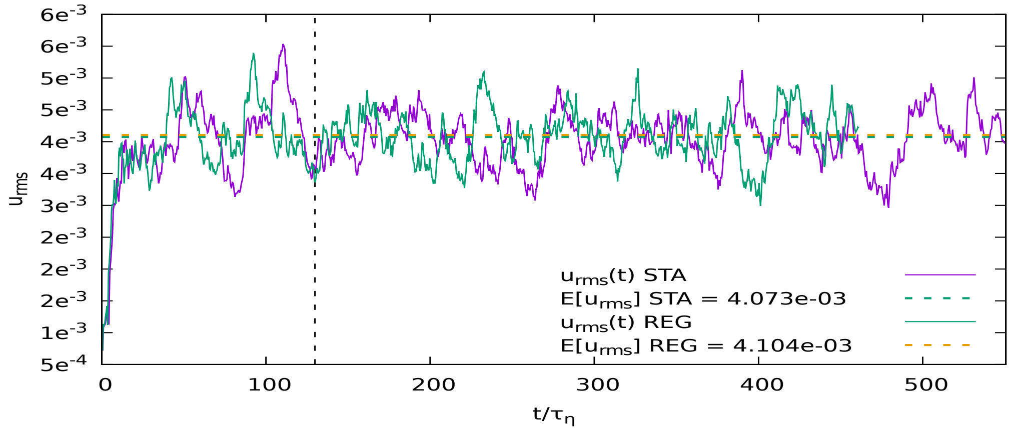


Figure A.3: Evolution of the standard deviation of fluctuations for STA and REG DNS of $N = 32$ resolution.

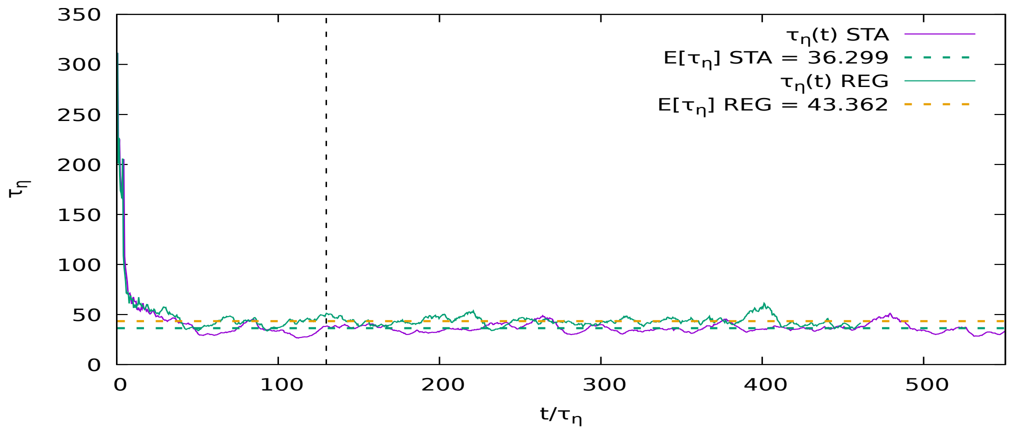


Figure A.4: Evolution of the Kolmogorov time scale for STA and REG DNS of $N = 32$ resolution.

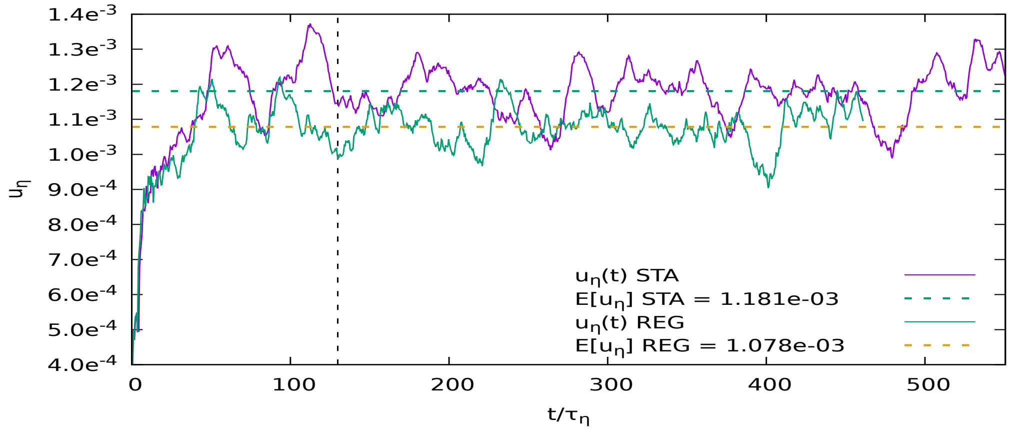


Figure A.5: Evolution of the Kolmogorov velocity scale for STA and REG DNS of $N = 32$ resolution.

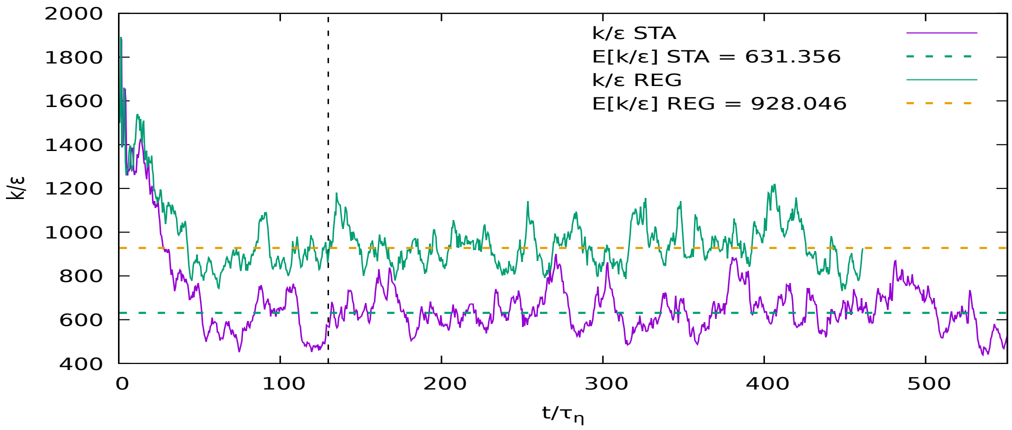


Figure A.6: Evolution of the ratio of turbulent kinetic energy to dissipation rate for STA and REG DNS of $N = 32$ resolution.

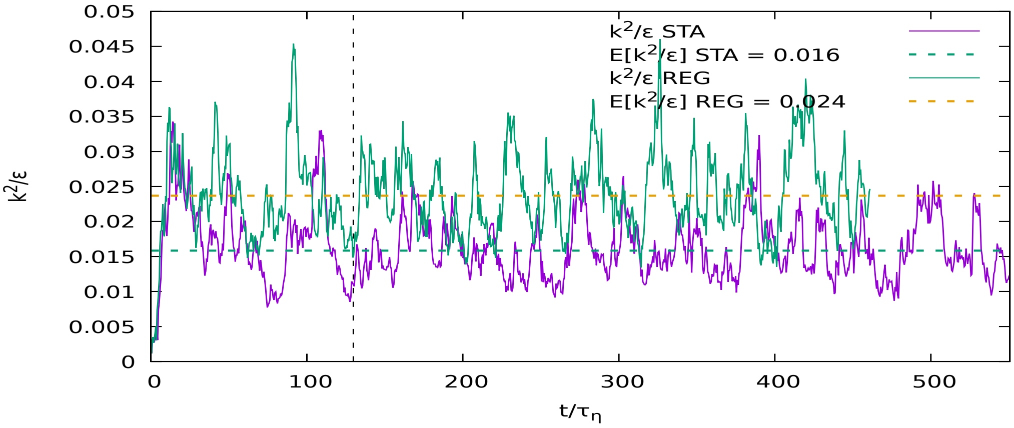


Figure A.7: Evolution of the ratio of turbulent kinetic energy squared to dissipation rate for STA and REG DNS of $N = 32$ resolution.

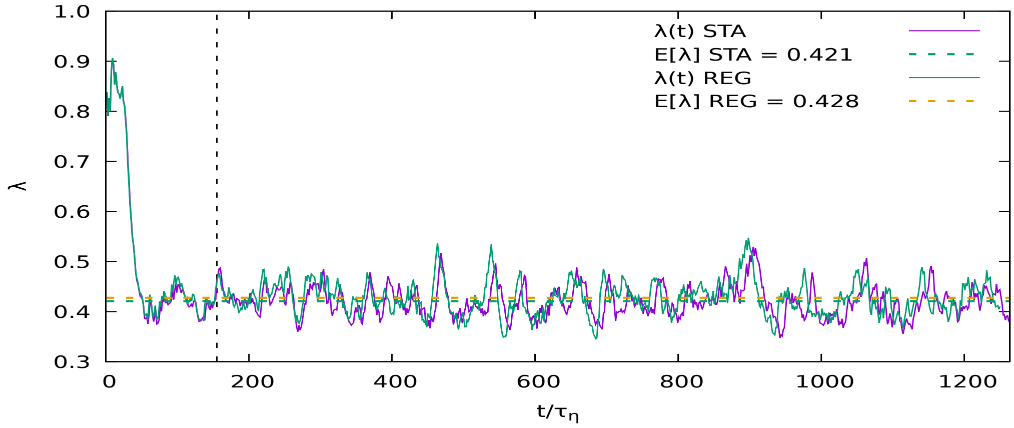


Figure A.8: Evolution of the Taylor length scale for STA and REG DNS of $N = 128$ resolution.

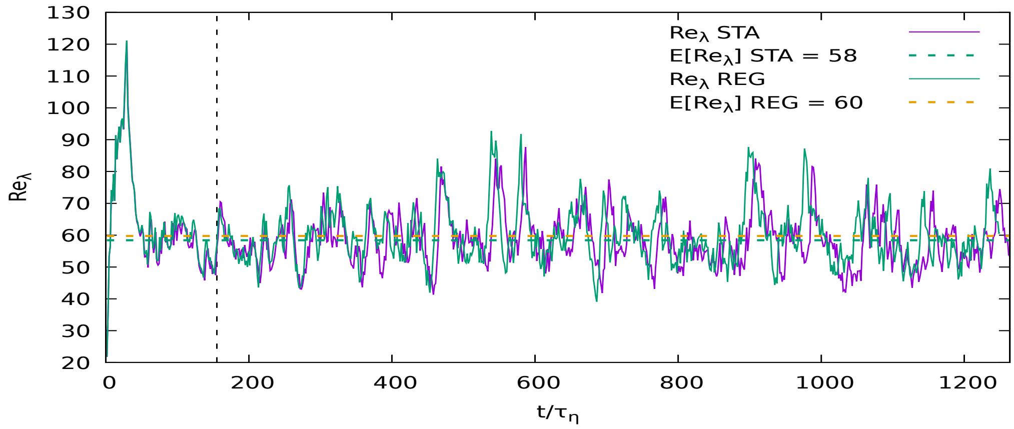


Figure A.9: Evolution of Re number for STA and REG DNS of $N = 128$ resolution.

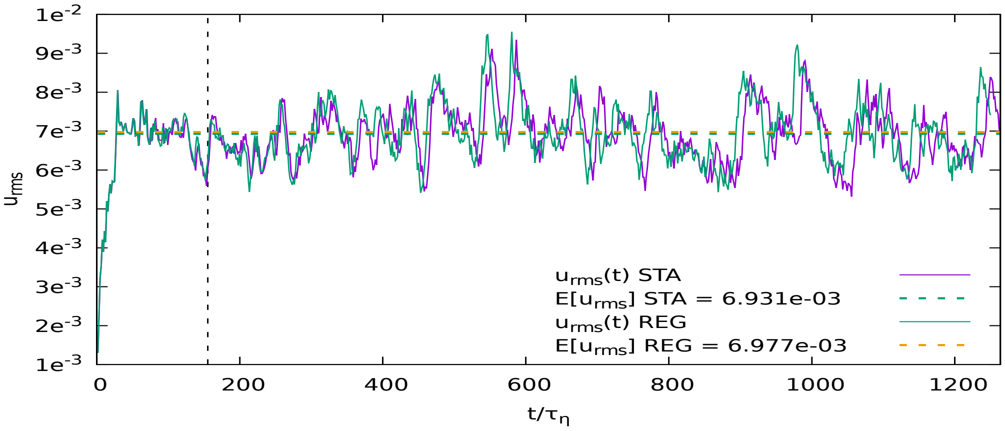


Figure A.10: Evolution of the standard deviation of fluctuations for STA and REG DNS of $N = 128$ resolution.

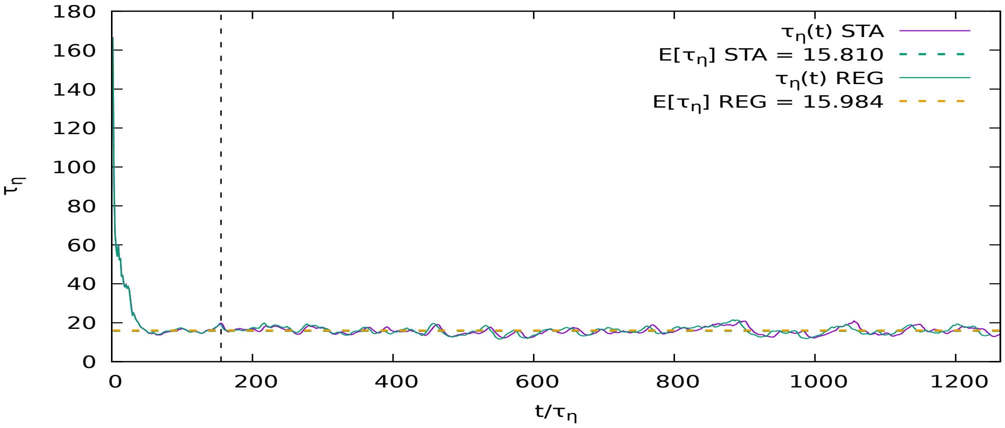


Figure A.11: Evolution of the Kolmogorov time scale for STA and REG DNS of $N = 128$ resolution.

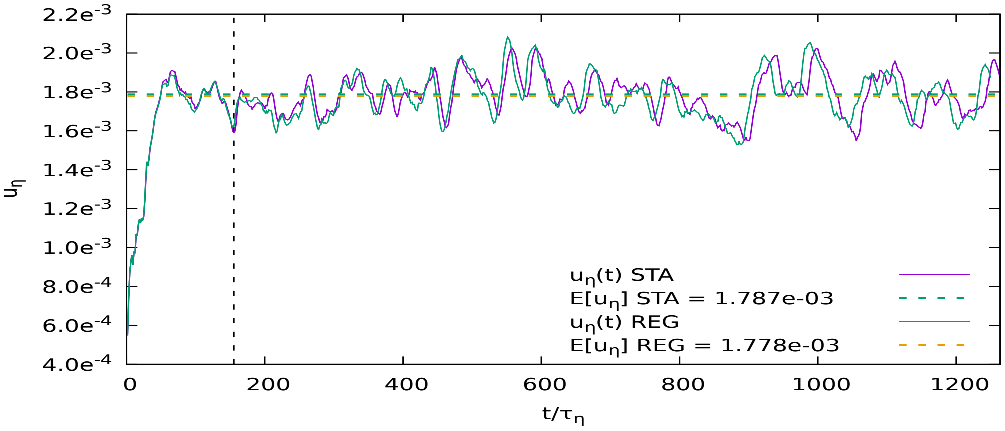


Figure A.12: Evolution of the Kolmogorov velocity scale for STA and REG DNS of $N = 128$ resolution.

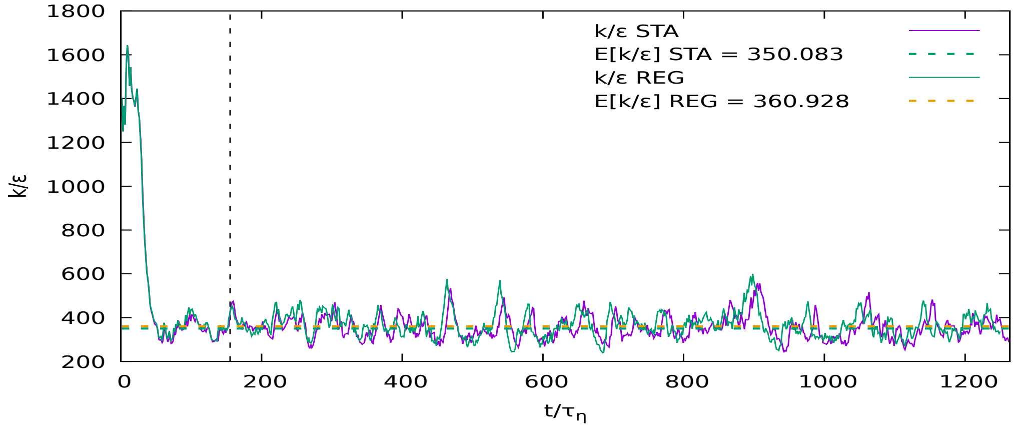


Figure A.13: Evolution of the ratio of turbulent kinetic energy to dissipation rate for STA and REG DNS of $N = 128$ resolution.

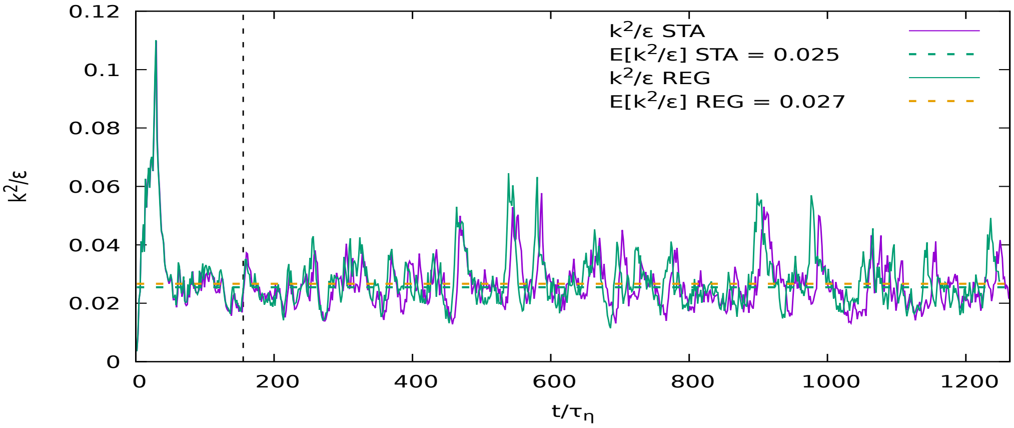


Figure A.14: Evolution of the ratio of turbulent kinetic energy squared to dissipation rate for STA and REG DNS of $N = 128$ resolution.

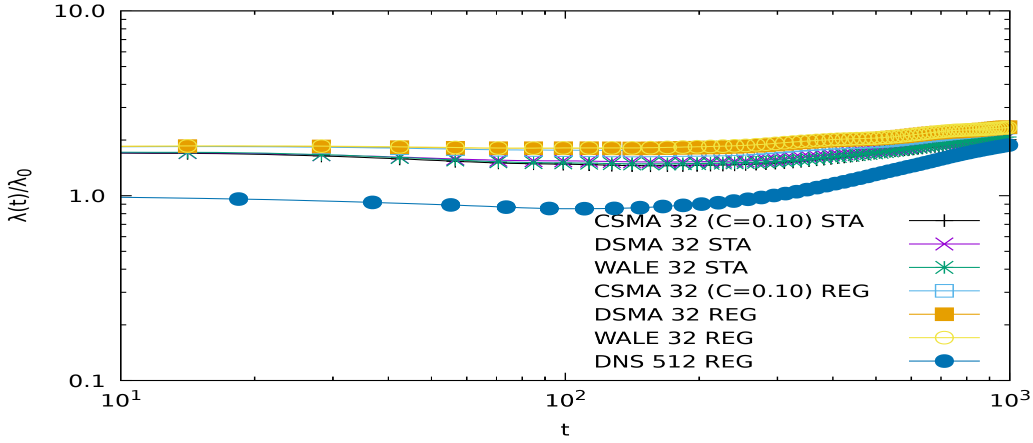


Figure A.15: Evolution of the Taylor length scale for STA and REG DNS of $N = 32$ resolution.

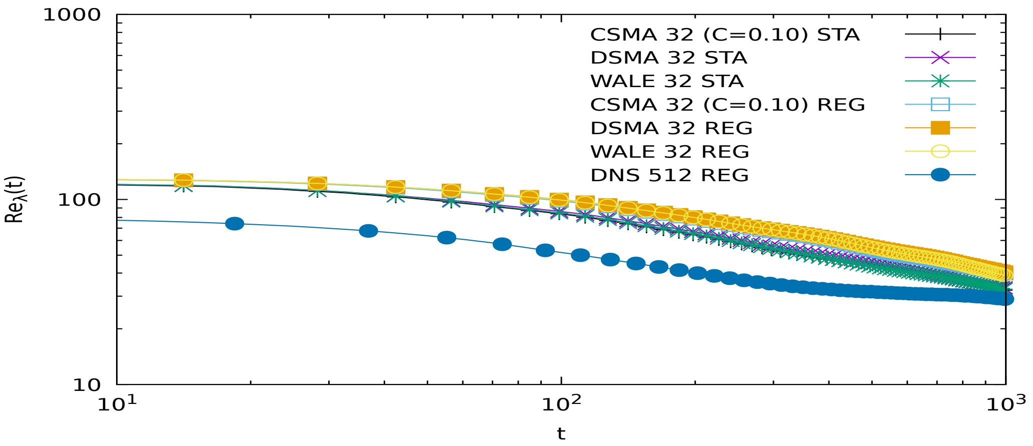


Figure A.16: Evolution of Re number for STA and REG DNS of $N = 32$ resolution.

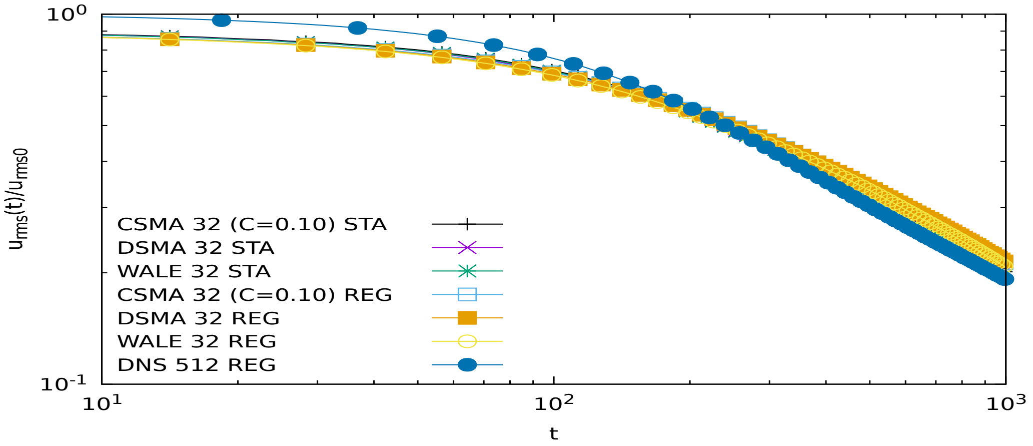


Figure A.17: Evolution of the standard deviation of fluctuations for STA and REG DNS of $N = 32$ resolution.

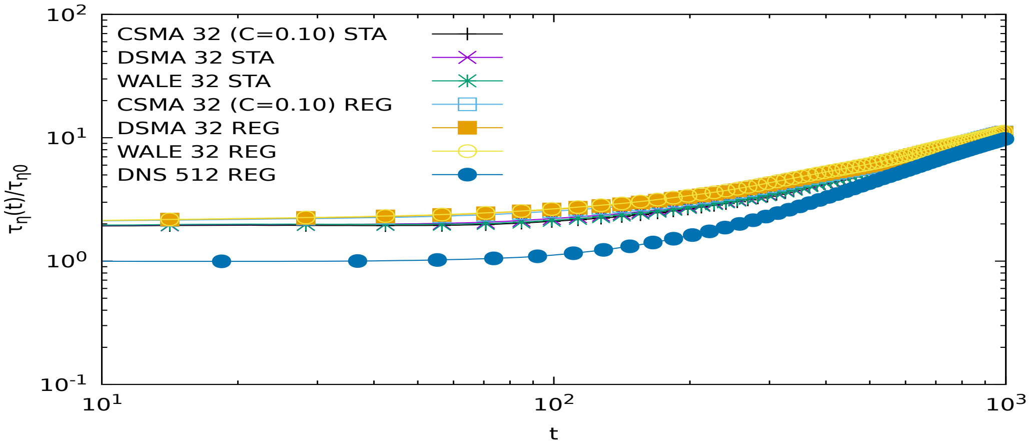


Figure A.18: Evolution of the Kolmogorov time scale for STA and REG DNS of $N = 32$ resolution.

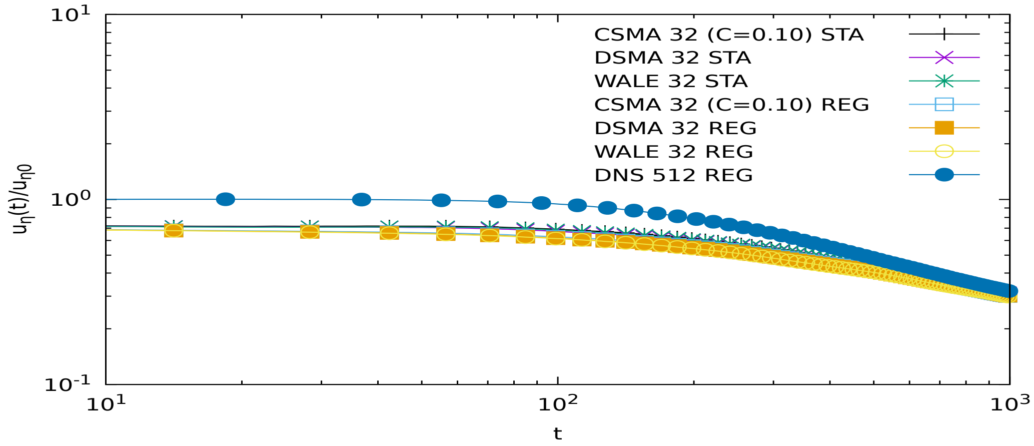


Figure A.19: Evolution of the Kolmogorov velocity scale for STA and REG DNS of $N = 32$ resolution.

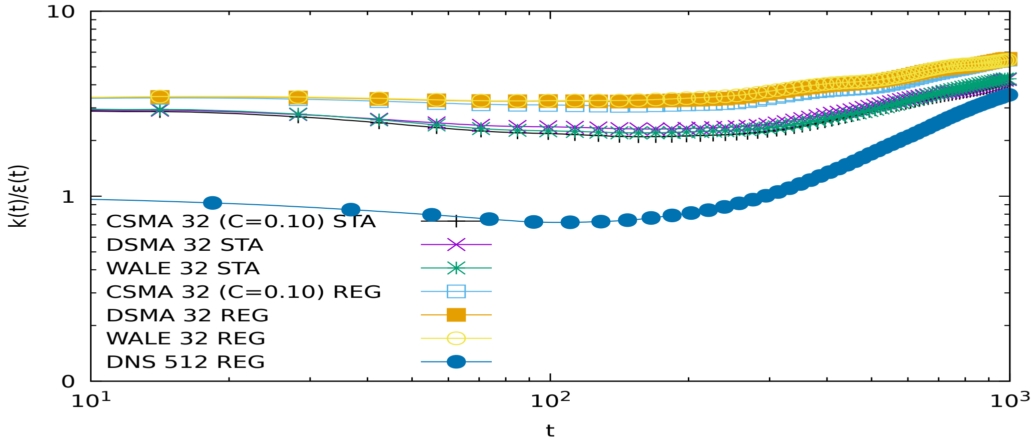


Figure A.20: Evolution of the ratio of turbulent kinetic energy to dissipation rate for STA and REG DNS of $N = 32$ resolution.

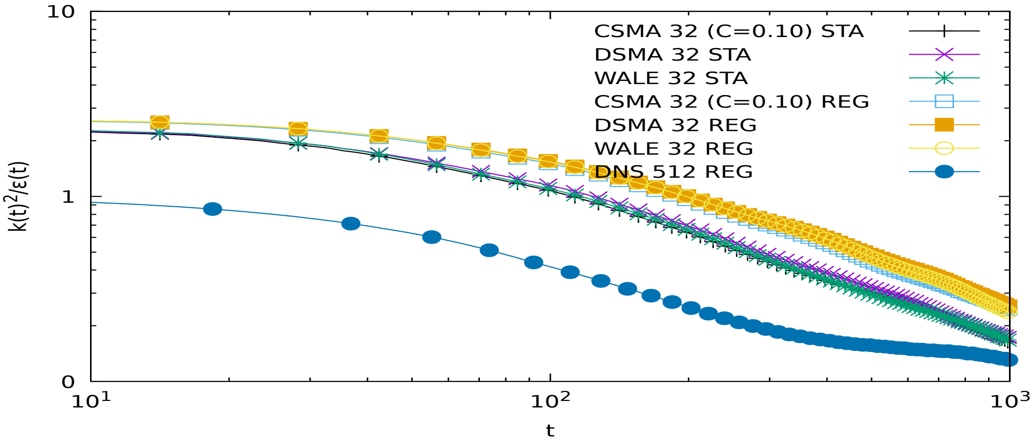


Figure A.21: Evolution of the ratio of turbulent kinetic energy squared to dissipation rate for STA and REG DNS of $N = 32$ resolution.

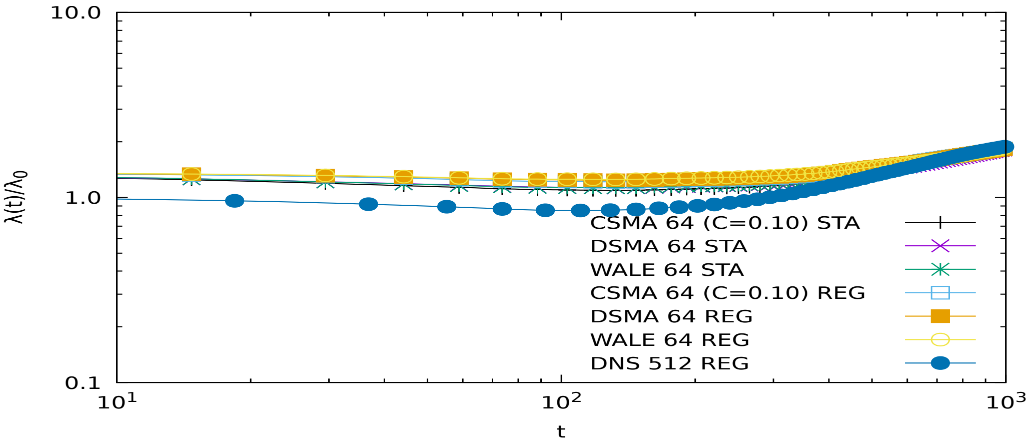


Figure A.22: Evolution of the Taylor length scale for STA and REG DNS of $N = 64$ resolution.

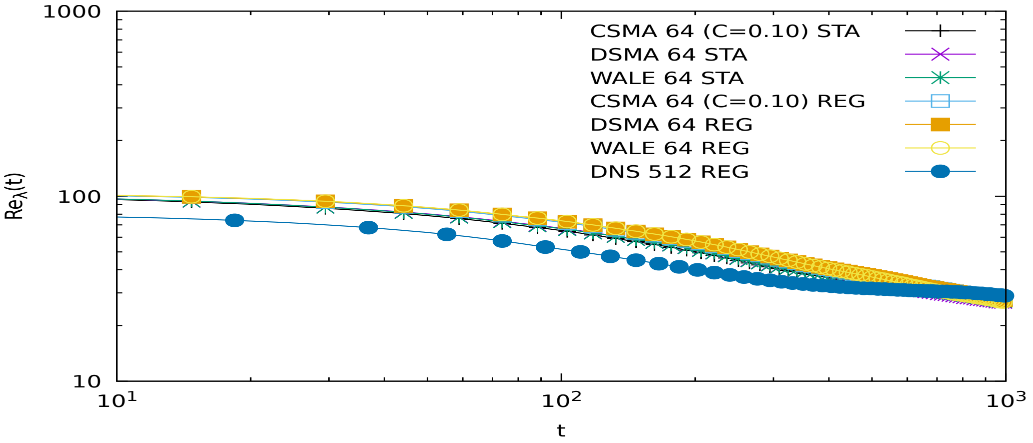


Figure A.23: Evolution of Re number for STA and REG DNS of $N = 64$ resolution.

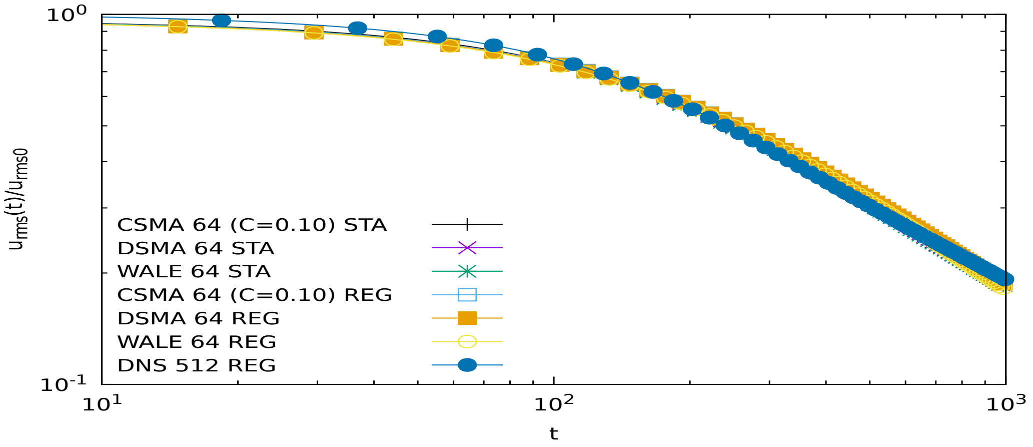


Figure A.24: Evolution of the standard deviation of fluctuations for STA and REG DNS of $N = 64$ resolution.

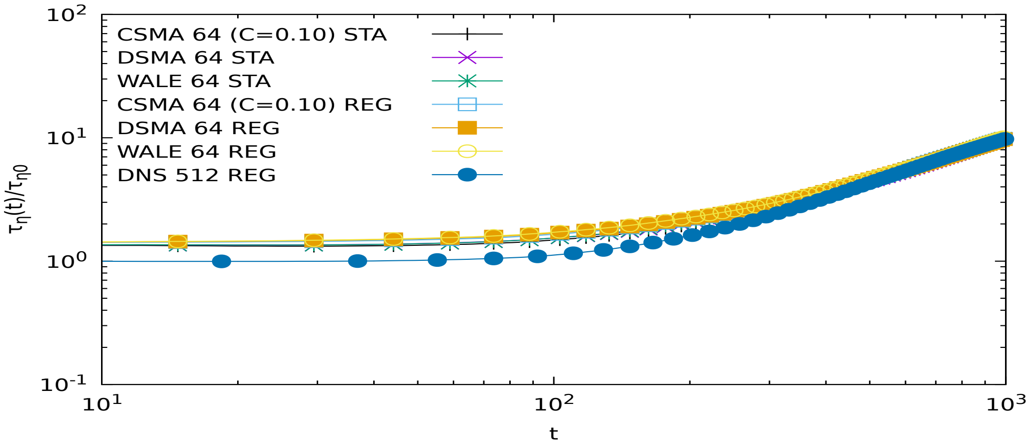


Figure A.25: Evolution of the Kolmogorov time scale for STA and REG DNS of $N = 64$ resolution.

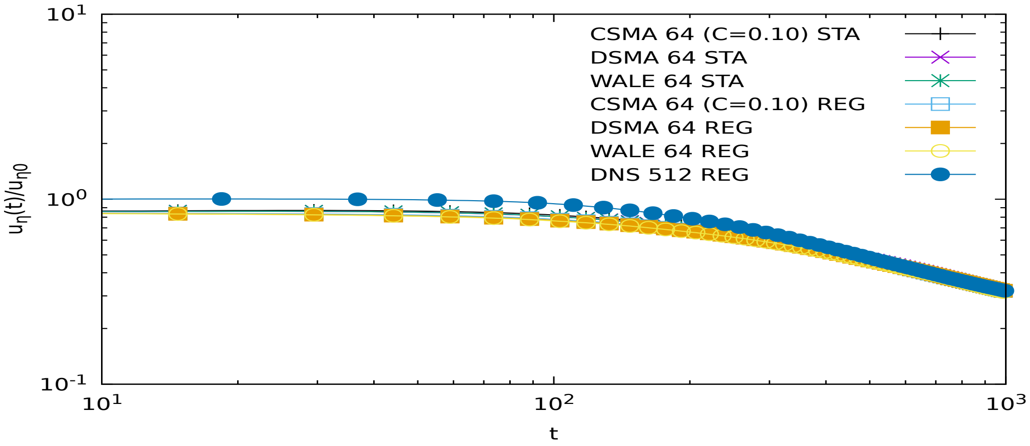


Figure A.26: Evolution of the Kolmogorov velocity scale for STA and REG DNS of $N = 64$ resolution.

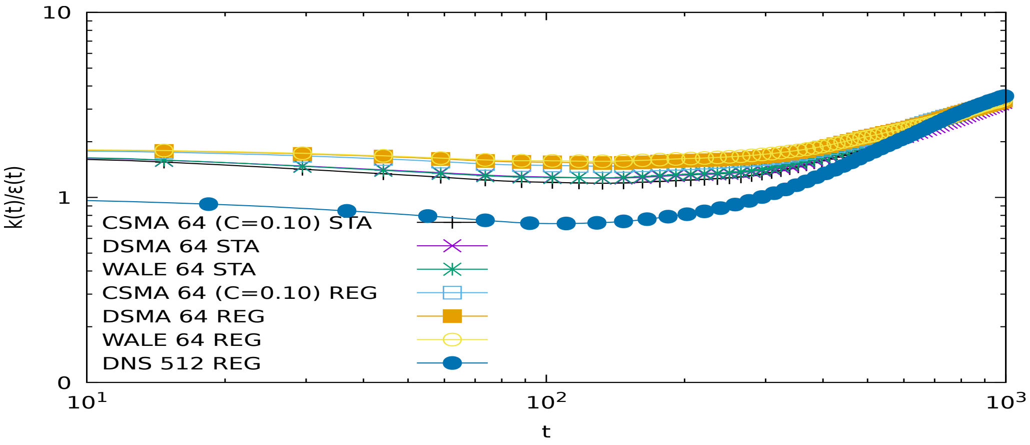


Figure A.27: Evolution of the ratio of turbulent kinetic energy to dissipation rate for STA and REG DNS of $N = 64$ resolution.

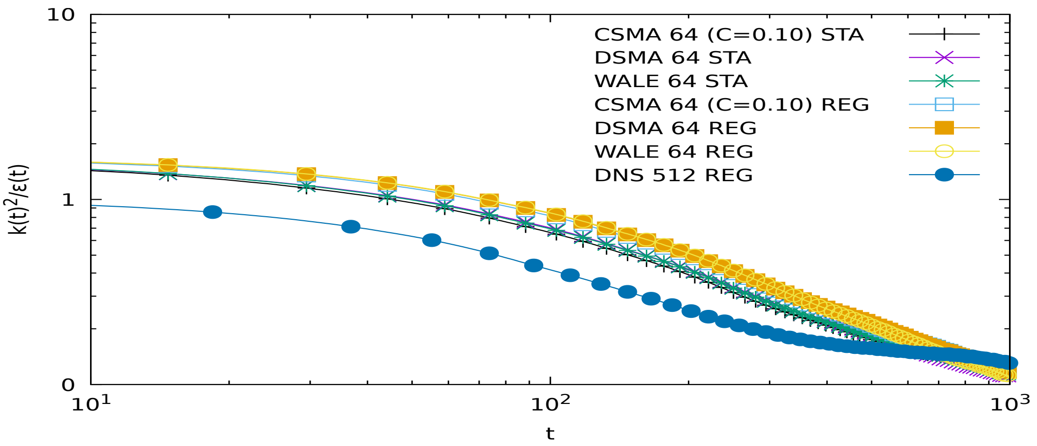


Figure A.28: Evolution of the ratio of turbulent kinetic energy squared to dissipation rate for STA and REG DNS of $N = 64$ resolution.

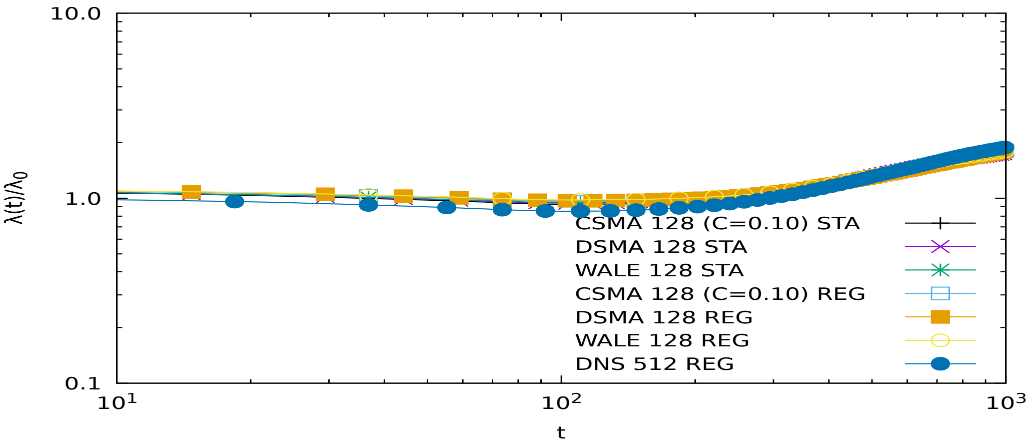


Figure A.29: Evolution of the Taylor length scale for STA and REG DNS of $N = 128$ resolution.

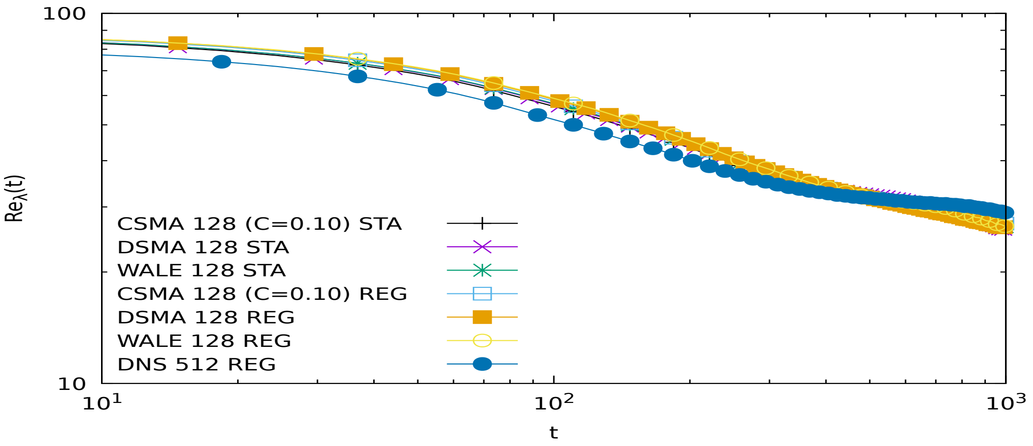


Figure A.30: Evolution of Re number for STA and REG DNS of $N = 128$ resolution.

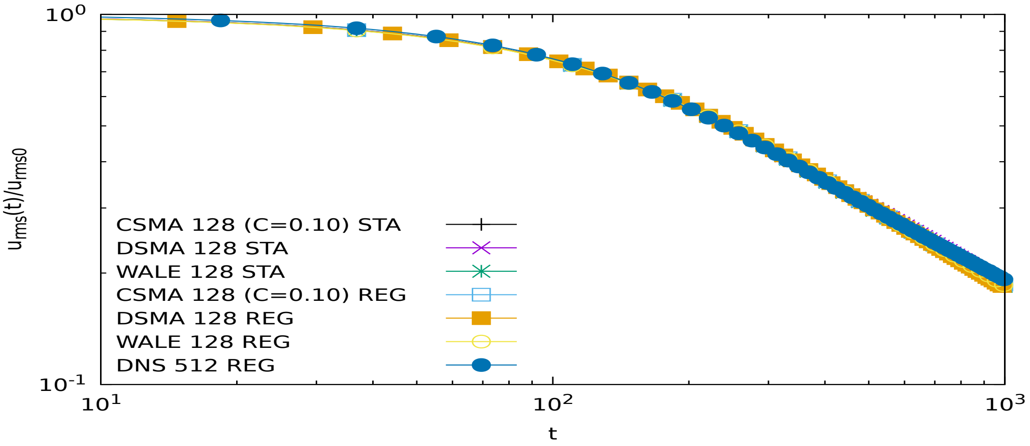


Figure A.31: Evolution of the standard deviation of fluctuations for STA and REG DNS of $N = 128$ resolution.

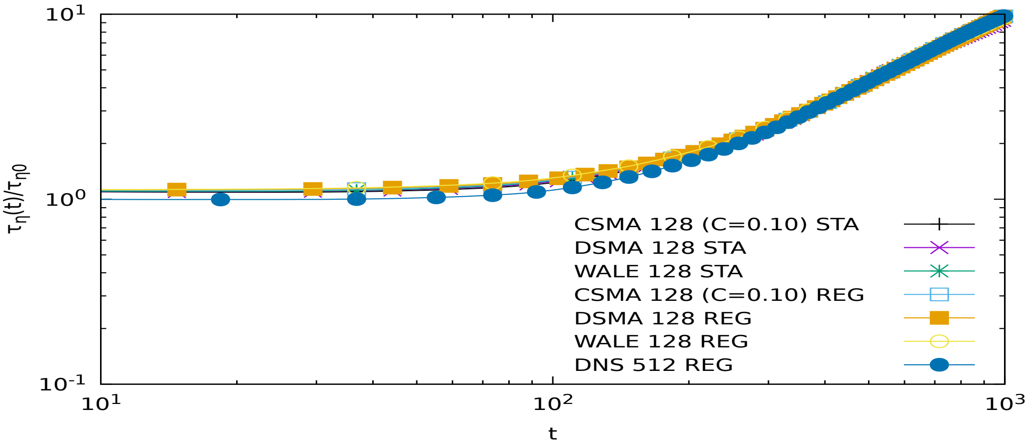


Figure A.32: Evolution of the Kolmogorov time scale for STA and REG DNS of $N = 128$ resolution.

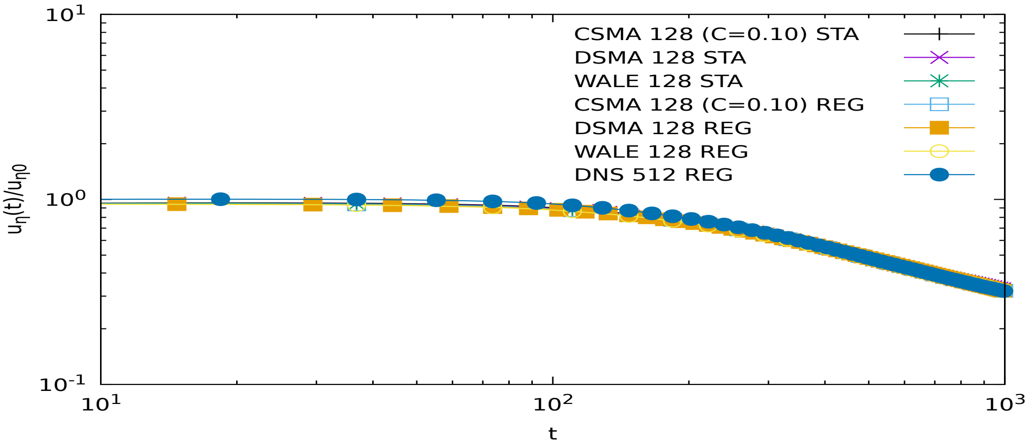


Figure A.33: Evolution of the Kolmogorov velocity scale for STA and REG DNS of $N = 128$ resolution.

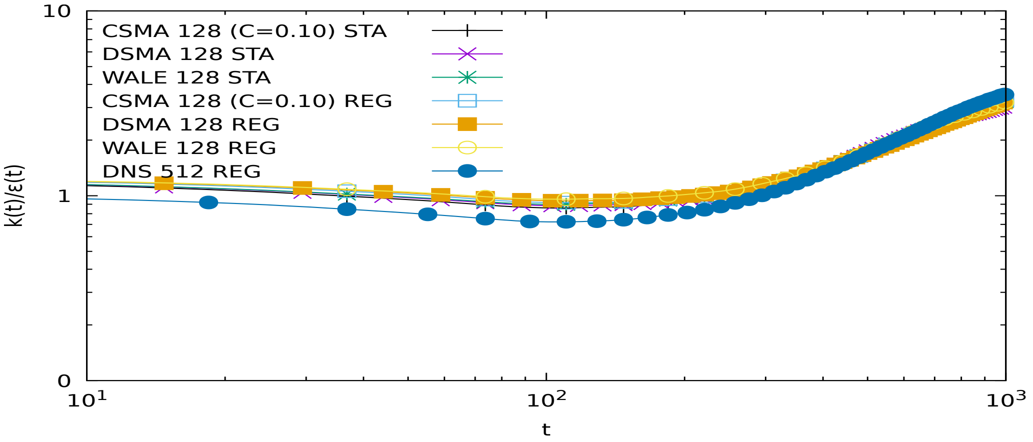


Figure A.34: Evolution of the ratio of turbulent kinetic energy to dissipation rate for STA and REG DNS of $N = 128$ resolution.

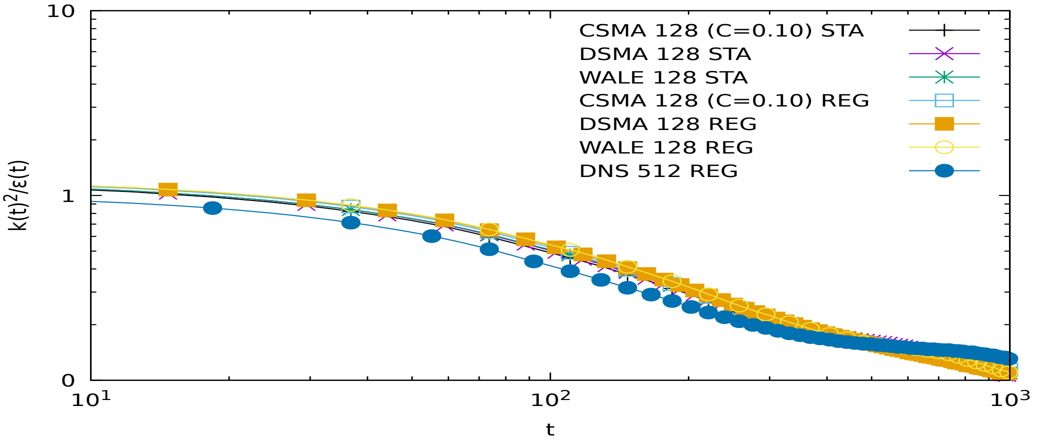


Figure A.35: Evolution of the ratio of turbulent kinetic energy squared to dissipation rate for STA and REG DNS of $N = 128$ resolution.

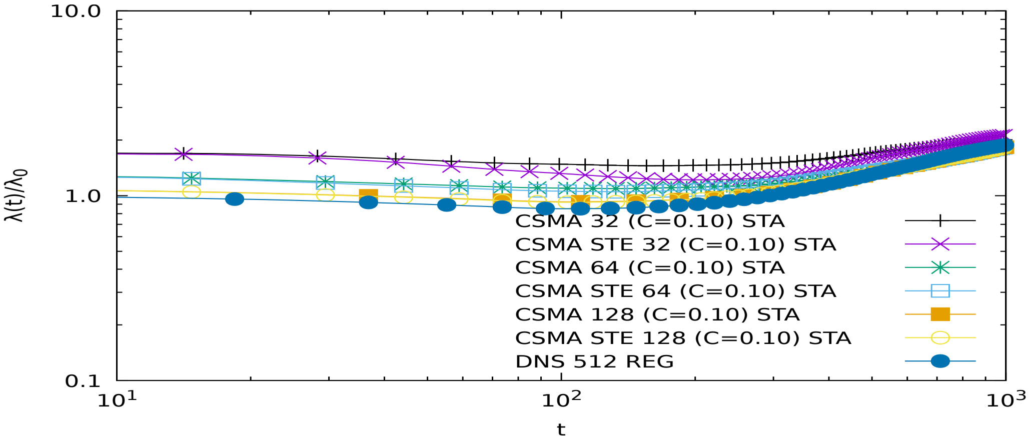


Figure A.36: Evolution of the Taylor length scale for CSMA with $C = 0.1$ STA SRT for both local and stencil-based estimation of strain rate.

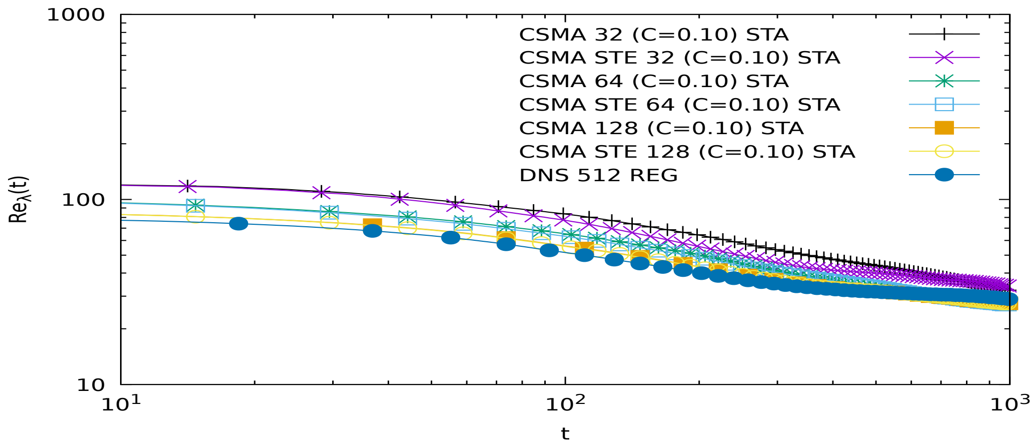


Figure A.37: Evolution of Re number for CSMA with $C = 0.1$ STA SRT for both local and stencil-based estimation of strain rate.

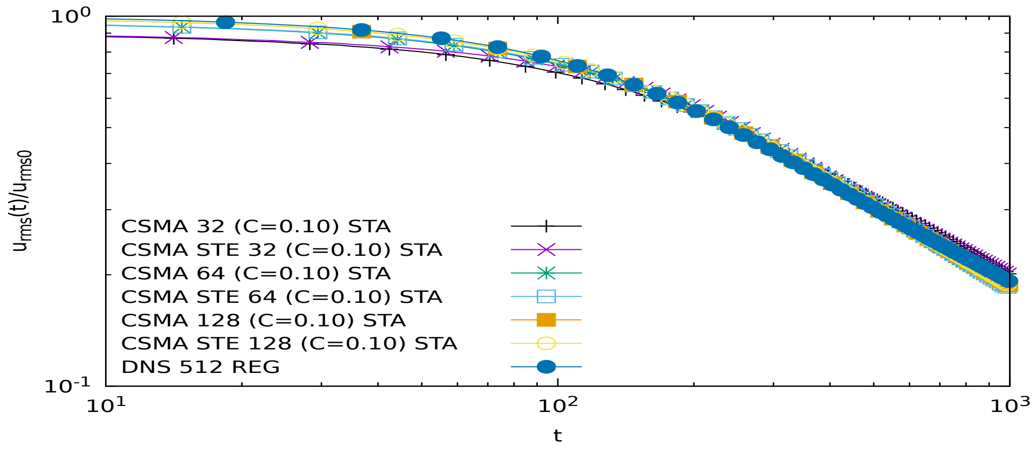


Figure A.38: Evolution of the standard deviation of fluctuations for CSMA with $C = 0.1$ STA SRT for both local and stencil-based estimation of strain rate.

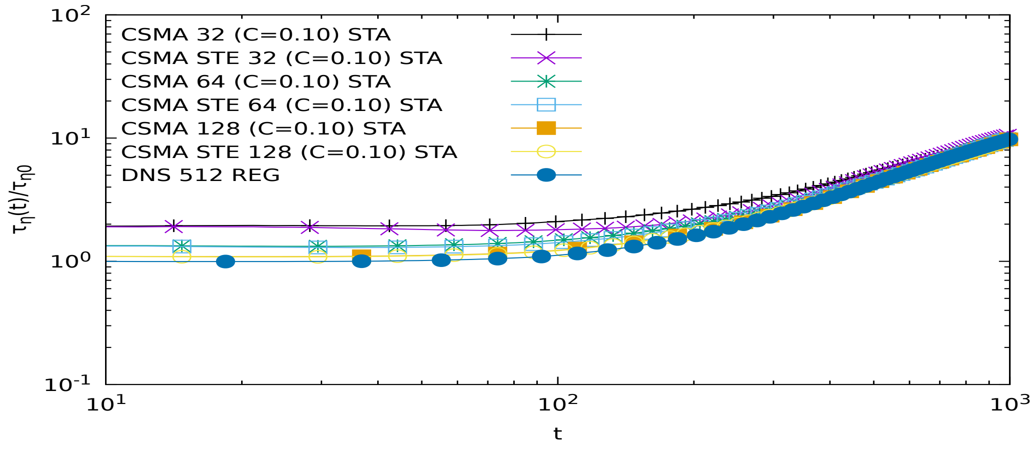


Figure A.39: Evolution of the Kolmogorov time scale for CSMA with $C = 0.1$ STA SRT for both local and stencil-based estimation of strain rate.

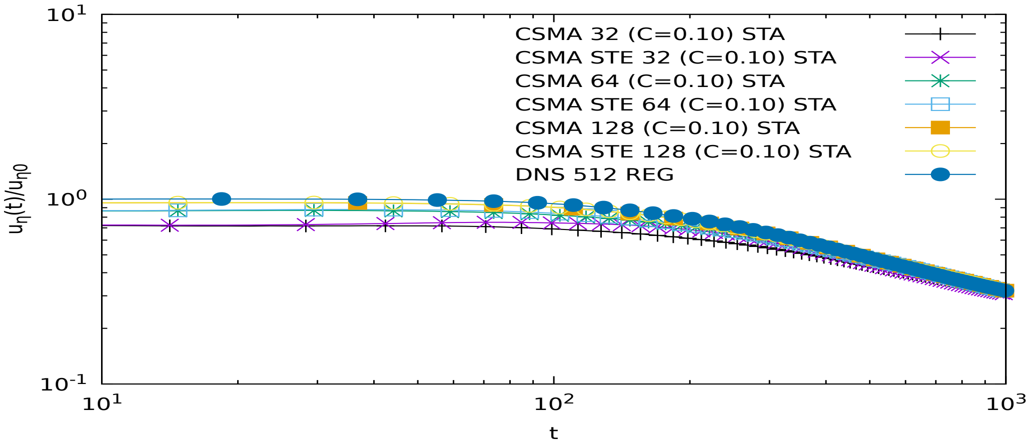


Figure A.40: Evolution of the Kolmogorov velocity scale for CSMA with $C = 0.1$ STA SRT for both local and stencil-based estimation of strain rate.

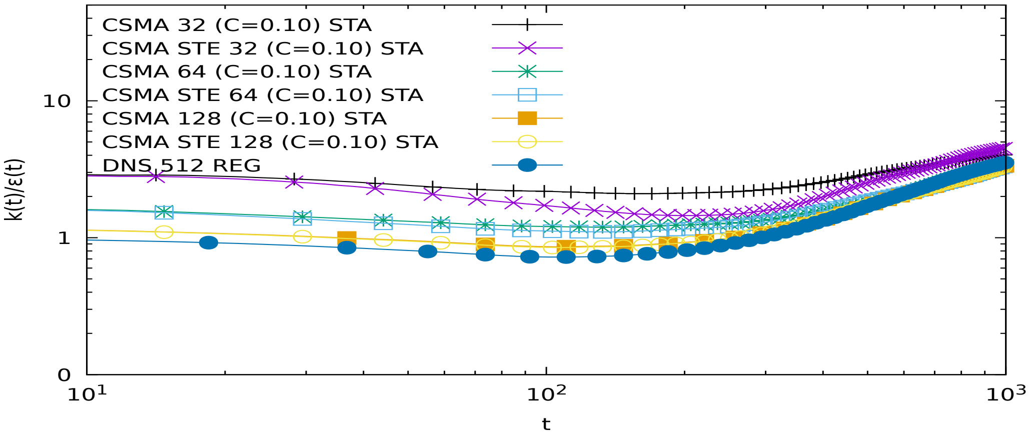


Figure A.41: Evolution of the ratio of turbulent kinetic energy to dissipation rate for CSMA with $C = 0.1$ STA SRT for both local and stencil-based estimation of strain rate.

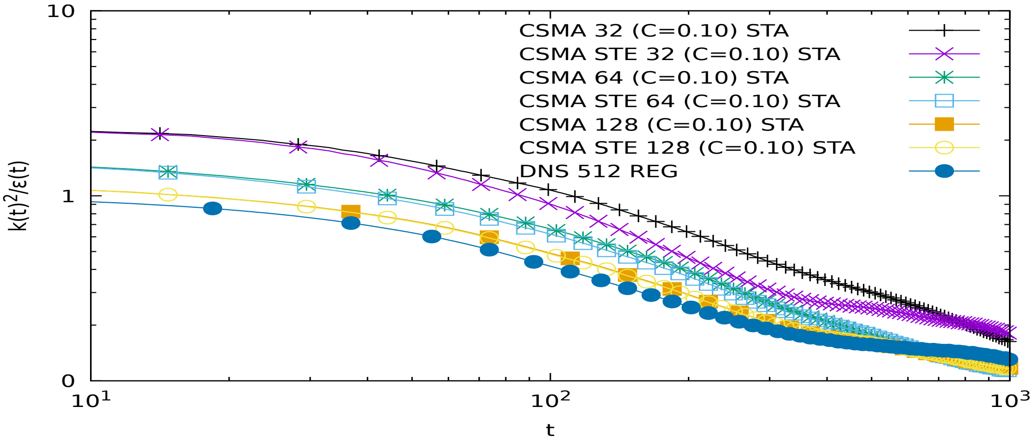


Figure A.42: Evolution of the ratio of turbulent kinetic energy squared to dissipation rate for CSMA with $C = 0.1$ STA SRT for both local and stencil-based estimation of strain rate.

References

(CFD Online). CFD Online, https://www.cfd-online.com/.

Abdel Kareem, W., Izawa, S., Xiong, A.-K., and Fukunishi, Y. (2009). Lattice Boltzmann simulations of homogeneous isotropic turbulence. *Computers & Mathematics with Applications*, 58(5):1055–1061.

Aidun, C. K. and Clausen, J. R. (2010). Lattice-Boltzmann Method for Complex Flows. *Annu. Rev. Fluid Mech.*, 42(1):439–472.

Balaras, E., Benocci, C., and Piomelli, U. (1996). Two-layer approximate boundary conditions for large-eddy simulations. *AIAA Journal*, 34(6):1111–1119.

Bhatnagar, P. L., Gross, E. P., and Krook, M. (1954). A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems. *Physical review*, 94(3):511.

Boghosian, B. M., Yepez, J., Coveney, P. V., and Wager, A. (2001). Entropic lattice Boltzmann methods. *Proc. R. Soc. Lond. A*, 457(2007):717–766.

Bouzidi, M., Firdaouss, M., and Lallemand, P. (2001). Momentum transfer of a Boltzmann-lattice fluid with boundaries. *Physics of Fluids*, 13(11):3452–3459.

Cate, A. t., Vliet, E. v., Derksen, J., and Akker, H. V. d. (2006). Application of spectral forcing in lattice-Boltzmann simulations of homogeneous turbulence. *Computers & Fluids*, 35(10):1239–1251.

Chapman, D. R. (1979). Computational aerodynamics development and outlook. *AIAA j.*, 17:1293–1313.

Chen, S. (2009). A large-eddy-based lattice Boltzmann model for turbulent flow simulation. *Applied Mathematics and Computation*, 215(2):591–598.

Choi, H. and Moin, P. (2012). Grid-point requirements for large eddy simulation: Chapmans estimates revisited. *Physics of Fluids*, 24(1):011702.

Coreixas, C., Chopard, B., and Latt, J. (2019). Comprehensive comparison of collision models in the lattice Boltzmann framework: Theoretical investigations. *Phys. Rev. E*, 100(3):033305.

DeBonis, J. (2013). Solutions of the Taylor-Green Vortex Problem Using High-Resolution Explicit Finite Difference Methods. In *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, Grapevine (Dallas/Ft. Worth Region), Texas. American Institute of Aeronautics and Astronautics.

Deiterding, R. (2011). Block-Structured Adaptive Mesh Refinement - Theory, Implementation and Application.

Deiterding, R. and Wood, S. L. (2016a). An adaptive lattice Boltzmann method for predicting wake fields behind wind turbines. In Dillmann, A., Heller, G., Krämer, E., Wagner, C., and Breitsamter, C., editors, *New Results in Numerical and Experimental Fluid Mechanics X*, volume 132 of *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, pages 845–857. Springer.

Deiterding, R. and Wood, S. L. (2016b). Predictive wind turbine simulation with an adaptive lattice Boltzmann method for moving boundaries. *J. Phys. Conf. Series*, 753:082005.

Dellar, P. J. (2013). An interpretation and derivation of the lattice Boltzmann method using Strang splitting. *Computers & Mathematics with Applications*, 65(2):129–141.

Ezzatneshan, E. (2019). Comparative study of the lattice Boltzmann collision models for simulation of incompressible fluid flows. *Mathematics and Computers in Simulation*, 156:158–177.

Faria, A. and Francisco, C. P. F. (2018). Experimental Transient Analysis of the Wake Around Bluff Bodies, an Analogy to the Alcântara Satellite Launching Center. Preliminary technical report, Instituto de Aeronáutica e Espaço (IAE), Sao Jose dos Campos, Brazil.

Faria, A. F., Avelar, A. C., and Fisch, G. (2019). Wind Tunnel Investigation of the Wind Patterns in the Launching Pad Area of the Brazilian Alcntara Launch Center. *J.Aerosp. Technol. Manag.*, 11.

Feaster, J., Battaglia, F., Deiterding, R., and Bayandor, J. (2016). Validation of an adaptive meshing implementation of the lattice Boltzmann method for insect flight. In *Proc. of the ASME 2016 Fluids Engineering Division Summer Meeting*, pages FEDSM2016–7782, V01AT12A007. ASME.

Feldhusen, K., Deiterding, R., and Wagner, C. (2016). A dynamically adaptive lattice Boltzmann method for thermal convection problems. *J. Applied Math. and Computer Science*, 26:735–747.

Feng, Y., Miranda, J., Guo, S., Jacob, J., and Sagaut, P. (2020). ProLB: A lattice Boltzmann solver of large-eddy simulation for atmospheric boundary layer flows. preprint, Meteorology.

Frigo, M. (1999). A fast Fourier transform compiler. In *Acm sigplan notices*, volume 34, pages 169–180. ACM.

Frisch, U., d'Humires, D., Hasslacher, B., Lallemand, P., Pomeau, Y., and Rivet, J.-P. (1987). Lattice Gas Hydrodynamics in Two and Three Dimensions. 1(4):649–707.

Geier, M., Greiner, A., and Korvink, J. G. (2006). Cascaded digital lattice Boltzmann automata for high Reynolds number flow. *Phys. Rev. E*, 73(6):066705.

Germano, M., Piomelli, U., Moin, P., and Cabot, W. H. (1991). A dynamic subgridscale eddy viscosity model. *Physics of Fluids A: Fluid Dynamics*, 3(7):1760–1765.

Gkoudesnes, C. and Deiterding, R. (2019a). Evaluating the Lattice Boltzmann Method for Large Eddy Simulation with Dynamic Sub-grid Scale Models. In *Eleventh International Symposium on Turbulence and Shear Flow Phenomena, TSFP11*, Southampton.

Gkoudesnes, C. and Deiterding, R. (2019b). Verification and Validation of Lattice Boltzmann Method Coupled with Complex Sub-grid Scale Turbulence Models. In *VI International Conference on Particle-based Methods - Fundamentals and Applications*, pages 510–521, Barcelona.

Gkoudesnes, C. and Deiterding, R. (2021). Verification of the WALE Large Eddy Simulation Model for Adaptive Lattice Boltzmann Methods Implemented in the AMROC Framework. In *Cartesian CFD Methods for Complex Applications*, number 3 in SEMA SIMAI, pages 107–127. Springer.

Guo, Z. and Shu, C. (2013). *Lattice Boltzmann method and its applications in engineering*. Number v. 3 in Advances in computational fluid dynamics. World Scientfic, [Hackensack,] New Jersey. OCLC: ocn840583182.

Guo, Z., Zheng, C., and Shi, B. (2002a). Discrete lattice effects on the forcing term in the lattice Boltzmann method. *Phys. Rev. E*, 65(4).

Guo, Z., Zheng, C., and Shi, B. (2002b). An extrapolation method for boundary conditions in lattice Boltzmann method. *Physics of Fluids*, 14(6):2007–2010.

Haussmann, M., Barreto, A. C., Kouyi, G. L., Rivière, N., Nirschl, H., and Krause, M. J. (2019). Large-eddy simulation coupled with wall models for turbulent channel flows at high Reynolds numbers with a lattice Boltzmann method Application to Coriolis mass flowmeter. *Computers & Mathematics with Applications*, 78(10):3285–3302.

Haussmann, M., Ries, F., Jeppener-Haltenhoff, J. B., Li, Y., Schmidt, M., Welch, C., Illmann, L., Bhm, B., Nirschl, H., Krause, M. J., and Sadiki, A. (2020). Evaluation of a Near-Wall-Modeled Large Eddy Lattice Boltzmann Method for the Analysis of Complex Flows Relevant to IC Engines. *Computation*, 8(2):43.

He, X. and Luo, L.-S. (1997). Theory of the lattice Boltzmann method: From the Boltzmann equation to the lattice Boltzmann equation. *Physical review*, 56(6):6811–6817.

He, Y.-L., Liu, Q., Li, Q., and Tao, W.-Q. (2019). Lattice Boltzmann methods for single-phase and solid-liquid phase-change heat transfer in porous media: A review. *International Journal of Heat and Mass Transfer*, 129:160–197.

Hénon, M. (1987). Viscosity of a Lattice Gas. *Complex Systems*, 1:763–789.

Hou, S., Sterling, J., Chen, S., and Doolen, G. D. (1994). A lattice Boltzmann subgrid model for high Reynolds number flows. *arXiv preprint comp-gas/9401004*.

Huang, M. and Leonard, A. (1994). Power-law decay of homogeneous turbulence at low Reynolds numbers. *Physics of Fluids*, 6(11):3765–3775.

Hudong, C., Shiyi, C., and William, H. M. (1992). Recovery of the Navier-Stokes equations using a lattice-gas Boltzmann method. *The Americal Physical Society*, 45(8):4.

Jacob, J., Malaspinas, O., and Sagaut, P. (2018). A new hybrid recursive regularised BhatnagarGrossKrook collision model for Lattice Boltzmann method-based large eddy simulation. *Journal of Turbulence*, 19(11-12):1051–1076.

Jafari, S. and Rahnama, M. (2011). Shear-improved Smagorinsky modeling of turbulent channel flow using generalized Lattice Boltzmann equation. *Int. J. Numer. Meth. Fluids*, 67(6):700–712.

Jahanshaloo, L., Pouryazdanpanah, E., and Che Sidik, N. A. (2013). A Review on the Application of the Lattice Boltzmann Method for Turbulent Flow Simulation. *Numerical Heat Transfer, Part A: Applications*, 64(11):938–953.

Junk, M. (2001). A finite difference interpretation of the lattice Boltzmann method. *Numer. Methods Partial Differential Eq.*, 17(4):383–402.

Junk, M., Klar, A., and Luo, L.-S. (2005). Asymptotic analysis of the lattice Boltzmann equation. *Journal of Computational Physics*, 210(2):676–704.

Kaneda, M., Haruna, T., and Suga, K. (2014). Ghost-fluid-based boundary treatment in lattice Boltzmann method and its extension to advancing boundary. *Applied Thermal Engineering*, 72(1):126–134.

Kang, S. K. and Hassan, Y. A. (2013). The effect of lattice models within the lattice Boltzmann method in the simulation of wall-bounded turbulent flows. *Journal of Computational Physics*, 232(1):100–117.

Kim, J., Moin, P., and Moser, R. (1987). Turbulence statistics in fully developed channel flow at low Reynolds number. *J. Fluid Mech.*, 177:133–166.

Kin, N., Deiterding, R., and Wagner, C. (2016). High-resolution simulation of side flow past a generic model of a high-speed train. In Dillmann, A., Heller, G., Krämer, E., Wagner, C., and Breitsamter, C., editors, *New Results in Numerical and Experimental Fluid Mechanics X*, volume 132 of *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, pages 421–431. Springer.

Krause, M. J., Kummerländer, A., Avis, S. J., Kusumaatmaja, H., Dapelo, D., Klemens, F., Gaedtke, M., Hafen, N., Mink, A., Trunk, R., Marquardt, J. E., Maier, M.-L., Haussmann, M., and Simonis, S. (2021). OpenLBOpen source lattice Boltzmann code. *Computers & Mathematics with Applications*, 81:258–288.

Krüger, T., Kusumaatmaja, H., Kuzmin, A., Shardt, O., Silva, G., and Viggen, E. M. (2016). *The lattice Boltzmann method: principles and practice.* Springer Berlin Heidelberg, New York, NY.

Krüger, T., Varnik, F., and Raabe, D. (2009). Shear stress in lattice Boltzmann simulations. *Phys. Rev. E*, 79(4):046704. arXiv: 0812.3242.

Lallemand, P. and Luo, L.-S. (2000). Theory of the lattice Boltzmann method: Dispersion, dissipation, isotropy, Galilean invariance, and stability. *Phys. Rev. E*, 61(6):6546–6562.

Laloglu, C. and Deiterding, R. (2017). Simulation of the flow around an oscillating cylinder with adaptive lattice Boltzmann methods. In Ivanyi, P. Topping, B. H. V. and Varady, G., editors, *Proc. 5th Int. Conf. on Parallel, Distributed, Grid and Cloud Computing for Engineering.* Civil-Comp Press.

Latt, J. (2007). *Hydrodynamic Limit of Lattice Boltzmann Equations.* PhD Thesis, Universit de Genve.

Latt, J. and Chopard, B. (2006). Lattice Boltzmann method with regularized precollision distribution functions. *Mathematics and Computers in Simulation*, 72(2-6):165–168.

Latt, J., Chopard, B., Malaspinas, O., Deville, M., and Michler, A. (2008). Straight velocity boundaries in the lattice Boltzmann method. *Phys. Rev. E*, 77(5):056703.

Latt, J., Malaspinas, O., Kontaxakis, D., Parmigiani, A., Lagrava, D., Brogi, F., Belgacem, M. B., Thorimbert, Y., Leclaire, S., Li, S., Marson, F., Lemus, J., Kotsalos, C., Conradin, R., Coreixas, C., Petkantchin, R., Raynaud, F., Beny, J., and Chopard, B. (2020). Palabos: Parallel lattice boltzmann solver. *Computers & Mathematics with Applications.*

Lee, M. and Moser, R. D. (2015). Direct numerical simulation of turbulent channel flow up to $Re_\tau \approx 5200$. *arXiv:1410.7809 [physics].* arXiv: 1410.7809.

Li, Q., Luo, K., Kang, Q., He, Y., Chen, Q., and Liu, Q. (2016). Lattice Boltzmann methods for multiphase flow and phase-change heat transfer. *Progress in Energy and Combustion Science*, 52:62–105.

Li, Y. and Jammalamadaka, A. (2015). Exploring an LBM-VLES Based CFD Approach for Predictions of Aero-Thermal Flows in Generic Turbo-machinery Devices. In *51st AIAA/SAE/ASEE Joint Propulsion Conference*, Orlando, FL. American Institute of Aeronautics and Astronautics.

Lilly, D. K. (1992). A proposed modification of the Germano subgridscale closure method. *Physics of Fluids A: Fluid Dynamics*, 4(3):633–635.

Liu, M., Chen, X.-P., and Premnath, K. N. (2012). Comparative Study of the Large Eddy Simulations with the Lattice Boltzmann Method Using the Wall-Adapting Local Eddy-Viscosity and Vreman Subgrid Scale Models. *Chinese Phys. Lett.*, 29(10):104706.

Malaspinas, O. (2015). Increasing stability and accuracy of the lattice Boltzmann scheme: recursivity and regularization. *arXiv:1505.06900 [physics]*. arXiv: 1505.06900.

Malaspinas, O. and Sagaut, P. (2011). Advanced large-eddy simulation for lattice Boltzmann methods: The approximate deconvolution model. *Physics of Fluids*, 23(10):105103.

Malaspinas, O. and Sagaut, P. (2012). Consistent subgrid scale modelling for lattice Boltzmann methods. *J. Fluid Mech.*, 700:514–542.

Malaspinas, O. and Sagaut, P. (2014). Wall model for large-eddy simulation based on the lattice Boltzmann method. *Journal of Computational Physics*, 275:25–40.

Mei, R., Shyy, W., Yu, D., and Luo, L.-S. (2000). Lattice Boltzmann Method for 3-D Flows with Curved Boundary. *Journal of Computational Physics*, 161:680–699.

Menon, S. and Soo, J.-H. (2004). Simulation of vortex dynamics in three-dimensional synthetic and free jets using the large-eddy lattice boltzmann method. *Journal of Turbulence*, 5:N32.

Mohamad, A. A. (2011). *Lattice Boltzmann method: fundamentals and engineering applications with computer codes / A. A. Mohamad.* Springer, London ; New York. OCLC: ocn706626387.

Mozafari-Shamsi, M., Sefid, M., and Imani, G. (2016). New formulation for the simulation of the conjugate heat transfer at the curved interfaces based on the ghost fluid lattice Boltzmann method. *Numerical Heat Transfer, Part B: Fundamentals*, 70(6):559–576.

Musker, A. J. (1979). Explicit Expression for the Smooth Wall Velocity Distribution in a Turbulent Boundary Layer. *AIAA Journal*, 17(6):655–657.

Nathen, P., Gaudlitz, D., Krause, M. J., and Adams, N. A. (2018). On the Stability and Accuracy of the BGK, MRT and RLB Boltzmann Schemes for the Simulation of Turbulent Flows. *CiCP*, 23(3).

Nicoud, F. and Ducros, F. (1999). Subgrid-Scale Stress Modelling Based on the Square of the Velocity Gradient Tensor. *Flow, Turbulence and Combustion*, 62:183–200.

Pasquali, A., Geier, M., and Krafczyk, M. (2020). Near-wall treatment for the simulation of turbulent flow by the cumulant lattice Boltzmann method. *Computers & Mathematics with Applications*, 79(1):195–212.

Peng, Y., Liao, W., Luo, L.-S., and Wang, L.-P. (2010). Comparison of the lattice Boltzmann and pseudo-spectral methods for decaying turbulence: Low-order statistics. *Computers & Fluids*, 39(4):568–591.

Pope, S. B. (2000). *Turbulent flows.* Cambridge University Press, Cambridge ; New York.

Premnath, K. N., Pattison, M. J., and Banerjee, S. (2009a). Dynamic subgrid scale modeling of turbulent flows using lattice-Boltzmann method. *Physica A: Statistical Mechanics and its Applications*, 388(13):2640–2658.

Premnath, K. N., Pattison, M. J., and Banerjee, S. (2009b). Steady State Convergence Acceleration of the Generalized Lattice Boltzmann Equation with Forcing Term through Preconditioning. *Journal of Computational Physics*, 228(3):746–769. arXiv: 0809.4730.

Qian, Y. H., D'Humires, D., and Lallemand, P. (1992). Lattice BGK Models for Navier-Stokes Equation. *Europhys. Lett.*, 17(6):479–484.

Schmitt, L. (1988). *Grobstruktursimulation turbulenter Grenzschicht-, Kanal- und Stufenströmungen.* PhD, München.

Schneider, A. (2015). *A Consistent Large Eddy Approach for Lattice Boltzmann Methods and its Application to Complex Flows.* PhD Thesis, Technischen Universität Kaiserslautern.

Shao, W. and Li, J. (2019). Review of Lattice Boltzmann Method Applied to Computational Aeroacoustics. *Archives of Acoustics*, 44(2):24.

Simulia (2019). PowerFlow.

Simulia (2021). XFlow.

Smagorinsky, J. (1963). General circulation experiments with the primitive equations, I The basic experiment. *Monthly Weather Review*, 91(3):99 – 164.

Spalart, P. R., Jou, W. H., Strelets, M., and Allmaras, S. R. (1997). Comments on the Feasibility of LES for Wings, and on a Hybrid RANS/LES Approach. In *Proceedings of the First AFOSR International Conference on DNS/LES*, pages 137–147, Louisiana Tech University, Ruston, USA.

Succi, S. (2001). *The Lattice Boltzmann Equation for Fluid Dynamics and Beyond.* OUP Oxford, Oxford : New York.

Tiwari, A. and Vanka, S. P. (2012). A ghost fluid Lattice Boltzmann method for complex geometries. *Int. J. Numer. Meth. Fluids*, 69(2):481–498.

Valio, L., Martn, J., and Hzi, G. (2010). Dynamics of Isotropic Homogeneous Turbulence with Linear Forcing Using a Lattice Boltzmann Method. *Flow, Turbulence and Combustion*, 84(2):219–237.

Van den Akker, H. E. (2018). Lattice Boltzmann simulations for multi-scale chemical engineering. *Current Opinion in Chemical Engineering*, 21:67–75.

Vreman, A. W. (2004). An eddy-viscosity subgrid-scale model for turbulent shear flow: Algebraic theory and applications. *Physics of Fluids*, 16(10):3670–3681.

Wang, J., Chen, L., Kang, Q., and Rahman, S. S. (2016). The lattice Boltzmann method for isothermal micro-gaseous flow and its application in shale gas flow: A review. *International Journal of Heat and Mass Transfer*, 95:94–108.

Weickert, M., Teike, G., Schmidt, O., and Sommerfeld, M. (2010). Investigation of the LES WALE turbulence model within the lattice Boltzmann framework. *Computers & Mathematics with Applications*, 59(7):2200–2214.

Werner, H. and Wengle, H. (1993). Large-Eddy Simulation of Turbulent Flow Over and Around a Cube in a Plate Channel. In Durst, F., Friedrich, R., Launder, B. E., Schmidt, F. W., Schumann, U., and Whitelaw, J. H., editors, *Turbulent Shear Flows 8*, pages 155–168. Springer Berlin Heidelberg, Berlin, Heidelberg.

Wilcox, D. C. (2006). *Turbulence modeling for CFD.* DCW Industries, La Cãnada, Calif, 3rd ed edition.

Wilhelm, S., Jacob, J., and Sagaut, P. (2018). An explicit power-law-based wall model for lattice Boltzmann methodReynolds-averaged numerical simulations of the flow around airfoils. *Physics of Fluids*, 30(6):065111.

Wilhelm, S., Jacob, J., and Sagaut, P. (2021). A New Explicit Algebraic Wall Model for LES of Turbulent Flows Under Adverse Pressure Gradient. *Flow Turbulence Combust*, 106(1):1–35.

Xia, Z., Shi, Y., Chen, Y., Wang, M., and Chen, S. (2015). Comparisons of different implementations of turbulence modelling in lattice Boltzmann method. *Journal of Turbulence*, 16(1):67–80.

Xu, A., Shyy, W., and Zhao, T. (2017). Lattice Boltzmann modeling of transport phenomena in fuel cells and flow batteries. *Acta Mech. Sin.*, 33(3):555–574.

Yu, H., Girimaji, S. S., and Luo, L.-S. (2005a). DNS and LES of decaying isotropic turbulence with and without frame rotation using lattice Boltzmann method. *Journal of Computational Physics*, 209(2):599–616.

Yu, H., Girimaji, S. S., and Luo, L.-S. (2005b). Lattice Boltzmann simulations of decaying homogeneous isotropic turbulence. *Phys. Rev. E*, 71(1):016708.

Zhao-Li, G., Chu-Guang, Z., and Bao-Chang, S. (2002). Non-equilibrium extrapolation method for velocity and pressure boundary conditions in the lattice Boltzmann method. *Chinese Phys.*, 11(4):366–374.

Zou, Q. and He, X. (1997). On pressure and velocity boundary conditions for the lattice Boltzmann BGK model. *Physics of Fluids*, 9(6):1591–1598.