IMPLICIT SUB-GRID SCALE MODELING WITHIN THE ENTROPIC LATTICE BOLTZMANN METHOD IN HOMOGENEOUS ISOTROPIC TURBULENCE

Zur Erlangung des akademischen Grades eines

DOKTORS DER NATURWISSENSCHAFTEN (DR. RER. NAT.)

an der Fakultät Mathematik und Naturwissenschaften der
Bergischen Universität Wuppertal vorgelegte und genehmigte

DISSERTATION

von
Guillaume Tauzin

Betreut durch Prof. Matthias Ehrhardt PD. Andreas Bartel,
Prof. Luca Biferale, und Prof. Mauro Sbragaglia

Dissertation eingereicht am: 21. Dezember 2018

Die Dissertation kann wie folgt zitiert werden:

urn:nbn:de:hbz:468-20190507-102445-6
[http://nbn-resolving.de/urn/resolver.pl?urn=urn%3Anbn%3Ade%3Ahbz%3A468-20190507-102445-6]

UNIVERSITÀ DEGLI STUDI DI ROMA
"TOR VERGATA"

FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E NATURALI

TESI DI DOTTORATO DI RICERCA IN FISICA

Implicit Sub-Grid Scale Modeling within the Entropic Lattice Boltzmann Method in Homogeneous Isotropic Turbulence

Guillaume Tauzin

A.A. 2018/2019

Relatore:

Prof. Luca Biferale
Prof. Matthias Ehrhardt

Thesis Committee:

Prof. Luca Biferale
Prof. Matthias Ehrhardt
Prof. Bálint Farkas
Prof. Francesco Knechtli
Prof. Dirk Pleiter
Prof. Mauro Sbragaglia

Coordinatore:

Prof. Roberto Benzi

Thesis Referees:

Prof. Luca Biferale
Prof. Matthias Ehrhardt
Prof. Federico Toschi

ABSTRACT

The Lattice Boltzmann Method (LBM) is a mesoscopic flow solver that has gained momentum due to its ability to deal with complex fluid dynamics. However, its application to the simulation of turbulent flows has been limited by instabilities arising when decreasing the viscosity. The Entropic LBM (ELBM) tackles this issue by equipping LBM with an H-theorem, achieving apparent unconditional stability. In practice, ELBM extends the Bhatnagar-Gross-Krook (BGK) [1] collision operator by allowing the relaxation time to fluctuate through the definition of an entropic parameter $\alpha(\boldsymbol{x}, t)$. ELBM has been put forward as an implicit Large-Eddy Simulation (LES) with an eddy viscosity Sub-Grid Scale (SGS) model, resulting from the assumption that the equation bridging the mesoscopic relaxation time with the macroscopic viscosity still holds when the relaxation time is fluctuating,

$$\nu_{\text{eff}}(\boldsymbol{x}, t) = c_s^2 \Delta_t \left(\frac{2\tau_0}{\alpha(\boldsymbol{x}, t)} - \frac{1}{2} \right) = \nu_0 + \delta\nu_e^M(\boldsymbol{x}, t), \tag{1}$$

with τ_0 the input relaxation time and ν_0 the input viscosity. The non-linear dependency of the entropic parameter on the distribution functions does not allow the effective viscosity to be expressed directly in terms of macroscopic quantities and therefore its physical meaning remains hidden. A hydrodynamical approximation of the eddy viscosity was proposed in Ref. [2] by expanding the entropic parameter using a Chapman-Enskog (C-E) procedure and leading to

$$\delta\nu_e^A \approx -c_s^2 \Delta t^2 \frac{1}{6\beta^2} \frac{\operatorname{Tr} \mathbf{S}^3}{\operatorname{Tr} \mathbf{S}^2}, \tag{2}$$

where $S_{ij} = \frac{1}{2}(\partial_i u_j + \partial_j u_i)$ is the strain-rate tensor. However, little has been done to numerically study the implicit SGS model implied by ELBM and the validity range of its macroscopic formulation (2). Therefore, it is still unclear whether it acts as a mere stabilizer or as an accurate representation of the unresolved physics of turbulence stemming from kinetic theory. In this thesis, we shed some light on this question in the context of two- and three-dimensional Homogeneous Isotropic Turbulent (HIT) flows.

A first step consists in quantifying the validity of ELBM as an implicit closure. We develop an analysis tool to assess the accuracy with which Navier-Stokes hydrodynamics is recovered by LBM compared to Pseudo-Spectral (PS) simulations of the Navier-Stokes Equations (NSE). We apply this tool to analyze two- and three-dimensional HIT ELBM simulations at different Reynolds numbers. On the one hand, the energy spectra reveals that ELBM is able to extend the inertial range up to 20 times the Reynolds of the last stable LBM simulation. On the other hand, the *a priori* assumption that ELBM can be macroscopically described as a LES with an eddy viscosity type SGS model (1) cannot be clearly confirmed. Furthermore, we study the validity of the approximated macroscopic

eddy viscosity formulation (2). We find that it captures the dynamics of the ELBM eddy viscosity only at low Reynolds numbers, while it fails in fully developed turbulent regimes.

To identify analytically the weak points of those approximations, we numerically check the assumptions made at every step of a C-E expansion of the entropic parameter $\alpha(\boldsymbol{x}, t)$. We explain why the macroscopic eddy viscosity formulation fails to be recovered at high Reynolds number by highlighting the presence of extra terms whose magnitude grows with the velocity gradients. We find that the implicit ELBM model is not only composed of an eddy viscous dissipation that depends on the effective relaxation time as assumed in the literature [3], but also of extra terms of the same order of magnitude. The latter result is of particular importance as it raises questions on the validity of many other eddy viscosity SGS models implemented in LBM.

The fact that the ELBM closure is more complex than a simple eddy viscosity model does not mean, in principle, that it is unable to model turbulence. To assess the capability of the ELBM closure, we compare the inertial range statistics of turbulent velocity fields obtained from an ELBM simulation, with those coming from a high-resolution Direct Numerical Simulation (DNS) of the NSE conducted with a PS code. First results show that ELBM is able to increase the inertial scaling range and partially captures the correct intermittent behaviors.

Moreover, the macroscopic approximated eddy viscosity (2) has an interesting feature, as it scales with the strain-rate tensor like the Smagorinsky eddy viscosity [4],

$$\delta\nu_e^S = (C_S\Delta)^2\sqrt{2\,\mathrm{Tr}\,\mathbf{S}^2}, \tag{3}$$

where C_S is a dimensionless coefficient and Δ the LES filter cut-off length. However, $\delta\nu_e^A$ in (2) is not positive-definite, meaning that it allows backscatter of energy from the unresolved to the resolved scales. Implementing the model (2) and the Smagrosinsky model (3) in a PS LES code, we observe that both closures have similar inertial range statistics.

A part of the present thesis is also dedicated to more practical numerical issues. We have written an open-source high-performance implementation of LBM [5] that support multiple 2D and 3D lattice stencils and can be used to conduct high-resolution flow simulations using different turbulence models and spectral HIT forcings. In particular, the code was optimized on multi-GPUs architectures by making use of NVSHMEM, a novel GPU-centric communication library. This implementation is being benchmarked against the standard CPU-centric communication implementation based on the Message-Passing

Interface (MPI) library. We introduce this on-going work and present preliminary results of the reference MPI implementation in terms of both strong and weak scaling properties.

THESIS OUTLINE

This thesis is organized as follows:

Chapter 1: Introduction

We presented the background and the numerical techniques on which our work is based on. We briefly summarize the main features of the physics of two- and three-dimensional HIT and introduce LES-based models of turbulence. We present the fundamentals of the mesoscopic LBM flow solver and of its unconditionally stable extension, the ELBM. We close this chapter by providing a brief overview of the state-of-the-art regarding the question of the interpretation of ELBM as an implicit physical SGS model. Chapters 2 to 5 will focus on further discussing this matter.

Chapter 2: A numerical tool for the assessment of the hydrodynamic recovery of LBM simulations

We will investigate the hydrodynamic recovery of LBM by analyzing exact balance relations for energy and enstrophy derived from averaging the equations of motion on sub-volumes of different sizes. In the context of 2D HIT, we first validate this approach on decaying turbulence by comparing the analysis of an ensemble of LBM simulations against one of an ensemble of PS simulations. We then conduct a benchmark of LBM simulations of forced turbulence with increasing Reynolds number by varying the input relaxation time of LBM. The results presented in this chapter are published in Ref. [6].

Chapter 3: A-priori study of ELBM hydrodynamics recovery and implicit SGS model

We investigate the validity of the *a priori* assumption that ELBM can be macroscopically described as a LES with an eddy viscosity model (1). For a set of two- and three-dimensional HIT flows at increasing Reynolds number, we conduct a statistical analysis of the hydrodynamic recovery of ELBM simulations. In order to do that, we extend the tool introduced in chapter 1 by adding to the balance equations the mesoscopic eddy viscous dissipation term stemming from Eq. (1). In parallel, we study numerically the approximated macroscopic formulation Eq. (2) to validate it against the measured mesoscopic expression Eq. (1).

Chapter 4: Study of the ELBM implicit SGS model at the macroscale

We perform a numerical check of the assumptions made at every step of the C-E expansion of the entropic parameter $\alpha(\boldsymbol{x}, t)$ that leads to the hydrodynamic eddy viscosity (2). We reveal that the ELBM SGS model does not consist only in an eddy viscosity term and obtain its full expression. The results presented in this chapter will be published in [7].

Chapter 5: Inertial range statistics of the Entropic Lattice Boltzmann and Large-Eddy Simulations in 3D turbulence

We conduct an analysis of the inertial range statistics of turbulent velocity fields comparing ELBM to a high-resolution Direct Numerical Simulation (DNS) of the NSE conducted with a PS code. Moreover, we also add the analysis from two PS LES: one equipped with the macroscopic approximation eddy viscosity Eq. (2) and one using a Smagorinsky model Eq. (3). These results will be published in [8].

Chapter 6: Accelerating Lattice Boltzmann flows simulation using NVSHMEM model for GPU-initiated communications

We discuss the implementation of the LBM algorithm on GPU-accelerated architectures and present the NVSHMEM programming interface for GPU-initiated communications. We show a benchmark of our code on a single GPU and present preliminary results of the performance scaling on multi-GPUs using MPI with the aim of using them as a reference to evaluate NVSHMEM implementations based on in-kernel communications. The results presented in this chapter are part of a paper in preparation [9].

Chapter 7: Conclusion

In this concluding chapter, we summarize the findings of this thesis work and highlight possibilities for future works.

ZUSAMMENFASSUNG

Die Lattice Boltzmann Methode (LBM) ist ein mesoskopischer Strömungslöser, der aufgrund seiner Fähigkeit, mit komplexen Strömungssituationen umzugehen, an Bedeutung gewonnen hat. Ihre Anwendung auf die Simulation turbulenter Strömungen wurde jedoch durch Instabilitäten begrenzt, die bei der Verringerung der Viskosität auftreten. Die Entropische LBM (ELBM) löst dieses Problem, indem sie die LBM mit einem H-Theorem ausstattet und so eine scheinbar bedingungslose Stabilität erreicht. In der Praxis erweitert ELBM den Bhatnagar-Gross-Krook (BGK) Kollisionsoperator [1], indem es die Relaxationszeit durch die Definition eines entropischen Parameters $\alpha(\boldsymbol{x}, t)$ schwanken lässt. ELBM wurde als implizite Large-Eddy Simulation (LES) mit einem Wirbelviskositäts-Sub-Grid-Skala (SGS)-Modell vorgeschlagen, das sich aus der Annahme ergibt, dass die Gleichung, die die mesoskopische Relaxationszeit mit der makroskopischen Viskosität überbrückt, auch bei schwankender Relaxationszeit gilt

$$\nu_{\text{eff}}(\boldsymbol{x}, t) = c_s^2 \Delta_t \left(\frac{2\tau_0}{\alpha(\boldsymbol{x}, t)} - \frac{1}{2} \right) = \nu_0 + \delta\nu_e^M(\boldsymbol{x}, t), \tag{4}$$

Hierbei ist τ_0 die Eingangsrelaxationszeit und ν 0 die Eingangsviskosität. Die nichtlineare Abhängigkeit des entropischen Parameters von den Verteilungsfunktionen erlaubt es nicht, die effektive Viskosität direkt in Form von makroskopischen Größen auszudrücken, so dass ihre physikalische Bedeutung verborgen bleibt. Eine hydrodynamische Approximation der Wirbelviskosität wurde in Ref. [2] durch Erweitern des entropischen Parameters mit einem Chapman-Enskog (C-E)-Verfahren durchgeführt und ergab

$$\delta\nu_e^A \approx -c_s^2 \Delta t^2 \frac{1}{6\beta^2} \frac{\mathrm{Tr}\,\mathbf{S}^3}{\mathrm{Tr}\,\mathbf{S}^2}, \tag{5}$$

wobei $S_{ij} = \frac{1}{2}(\partial_i u_j + \partial_j u_i)$ der Dehnungstrend-Tensor ist. Es wurde jedoch bisher wenig getan, um das implizite SGS-Modell (das durch ELBM impliziert wird), und den Gültigkeitsbereich seiner makroskopischen Formulierung (5) numerisch zu untersuchen. Daher ist es noch unklar, ob es als reiner Stabilisator oder als genaue Darstellung der ungelösten Physik der Turbulenz aus der kinetischen Theorie wirkt. In dieser Doktorarbeit werfen wir etwas Licht auf diese Frage im Zusammenhang mit zwei- und dreidimensionalen homogenen, isotropischen, turbulenten (HIT) Strömungen. Ein erster Schritt besteht darin, die Validität von ELBM als impliziten Abschluss zu quantifizieren.

Wir entwickeln ein Analysetool, um die Genauigkeit zu beurteilen, mit der die Navier-Stokes-Hydrodynamik durch LBM im Vergleich zu Pseudospektral-(PS)-Simulationen der Navier-Stokes-Gleichungen (NSG) gewonnen wird. Wir wenden dieses Tool an, um zwei- und dreidimensionale HIT-ELBM-Simulationen bei verschiedenen Reynoldszahlen zu analysieren. Einerseits zeigten die Energiespektren, dass ELBM in der Lage ist,

den Trägheitsbereich bis zum 20-fachen der Reynoldszahl der letzten stabilen LBM-Simulation zu erweitern. Andererseits kann die a priori angenommene Annahme, dass ELBM makroskopisch als LES mit einem Wirbelviskositätstyp SGS-Modell (1) beschrieben werden kann, nicht eindeutig bestätigt werden. Darüber hinaus untersuchen wir die Validität der approximierten makroskopischen Wirbelviskosität Formulierung (2). Wir finden, dass es die Dynamik der ELBM Wirbelviskosität nur bei niedrigen Reynoldszahlen erfasst, während es bei voll entwickelten turbulenten Regimen versagt.

Um die Schwachstellen dieser Annäherungen analytisch zu identifizieren, überprüfen wir numerisch die Annahmen, die bei jedem Schritt einer C-E-Erweiterung des entropischen Parameters $\alpha(\boldsymbol{x}, t)$ getroffen wurden. Wir erklären, warum die makroskopische Wirbelviskositätsformulierung bei hoher Reynoldszahl nicht wiederhergestellt werden kann, indem wir das Vorhandensein zusätzlicher Begriffe hervorheben, deren Größe mit den Geschwindigkeitsgradienten wächst. Wir stellen fest, dass das implizite ELBM-Modell nicht nur aus einer wirbelviskosen Dissipation besteht, die von der in der Literatur angenommenen effektiven Relaxationszeit abhängt [3], sondern auch aus zusätzlichen Begriffen der gleichen Größenordnung. Letzteres Ergebnis ist von besonderer Bedeutung, da es Fragen über die Gültigkeit vieler anderer SGS-Modelle mit Wirbelviskosität aufwirft, die in LBM implementiert sind.

Die Tatsache, dass der ELBM-Abschluss komplexer ist als ein einfaches Wirbelviskositätsmodell, bedeutet im Prinzip nicht, dass er Turbulenzen nicht modellieren kann. Zur Beurteilung der Leistungsfähigkeit des ELBM-Abschlusses vergleichen wir die Trägheitsbereichsstatistik turbulenter Geschwindigkeitsfelder, die aus einer ELBM-Simulation gewonnen wurden, mit denen, die aus einer hochauflösenden Direkten Numerischen Simulation (DNS) der Navier-Stokes Gleichung (NSG), durchgeführt mit einem PS-Code. Vorläufige Ergebnisse zeigen, dass ELBM in der Lage ist, den Trägheitsskalierungsbereich zu erhöhen und teilweise das korrekte intermittierende Verhalten widerspiegelt.

Darüber hinaus hat die makroskopisch approximierte Wirbelviskosität (5) ein interessantes Merkmal, da sie mit dem Dehnungstrend-Tensor wie die Smagorinsky Wirbelviskosität [4] skaliert,

$$\delta\nu_e^S = (C_S\Delta)^2\sqrt{2\,\mathrm{Tr}\,\mathbf{S}^2}, \tag{6}$$

wobei CS ein dimensionsloser Koeffizient ist und Δ die LES Filter Cut-Off Länge bezeichnet. Aber A in (5) ist nicht positiv-definit, was bedeutet, dass es eine Rückstreuung der Energie aus den nicht aufgelösten auf die aufgelösten Skalen geben kann. Bei der Implementierung des Modells (5) und dem Smagrosinsky Modell (6) in einem PS LES-Code beobachten wir, dass beide Abschlüsse eine ähnliche Verteilung des Trägheitsbereich haben.

Ein Teil der vorliegenden Doktorarbeit ist auch eher praktischeren numerischen Fragen gewidmet. Wir haben eine Open-Source-Hochleistungsimplementierung von LBM [5] geschrieben, die mehrere 2D- und 3D-Gitterstrukturen unterstützt und zur Durchführung hochauflösender Strömungssimulationen mit verschiedenen Turbulenzmodellen und spektralen HIT-Verstärkungenverwendet werden kann. Insbesondere wurde der Code auf Multi-GPUs-Architekturen optimiert, indem NVSHMEM, eine neuartige GPU-zentrierte Kommunikationsbibliothek, verwendet wurde. Diese Implementierung wird mit der standardmäßigen CPU-zentrierten Kommunikationsimplementierung auf Basis der Message-Passing Interface (MPI)-Bibliothek verglichen. Wir stellen diese laufenden Arbeiten vor und präsentieren erste Ergebnisse der Referenz-MPI-Implementierung sowohl in Bezug auf starke als auch auf schwache Skalierungseigenschaften.

GLIEDERUNG DER DOKTORARBEIT

Diese Arbeit ist wie folgt gegliedert:

Kapitel 1: Einführung

Im ersten Kapitel stellten wir den Hintergrund und die numerischen Techniken vor, auf denen unsere Arbeit basiert. Wir fassen kurz die wichtigsten Merkmale der Physik des zwei- und dreidimensionalen HIT zusammen und stellen LES-basierte Modelle der Turbulenz vor. Wir stellen die Grundlagen des mesoskopischen LBM-Strömungslösers und seiner bedingungslosen stabilen Erweiterung, dem ELBM, vor. Wir schließen dieses Kapitel mit einem kurzen Überblick über den Stand der Technik bei der Frage der Interpretation von ELBM als implizites physikalisches SGS-Modell. Die Kapitel 2 bis 5 werden sich auf die weitere Diskussion dieser Frage konzentrieren.

Kapitel 2: Ein numerisches Werkzeug zur Beurteilung der hydrodynamischen Erholung von LBM-Simulationen

In Kapitel 2 werden wir die hydrodynamische Rückgewinnung von LBM untersuchen, indem wir genaue Gleichgewichtsverhältnisse für Energie und Enstrophie analysieren, die sich aus der Mittelwertbildung der Bewegungsgleichungen auf Teilvolumina mit unterschiedlichen Größen ergeben. Im Rahmen von 2D HIT validieren wir diesen Ansatz zunächst, indem wir die Analyse eines Ensembles von LBM-Simulationen mit einer aus einem Ensemble von PS-Simulationen vergleichen. Anschließend führen wir einen

Benchmark durch. von LBM-Simulationen von erzwungenen Turbulenzen mit steigender Reynoldszahl durch Variation der Eingangsrelaxationszeit von LBM. Die in diesem Kapitel vorgestellten Ergebnisse sind in Ref. [6] publiziert.

Kapitel 3: A-priori-Studie zur ELBM-Hydrodynamik-Rückgewinnung und implizites SGS-Modell

In Kapitel 3 untersuchen wir die Gültigkeit der a priori Annahme, dass ELBM makroskopisch als LES mit einem Wirbelviskositätsmodell (4) beschrieben werden kann. Für eine Reihe von zwei- und dreidimensionalen HIT-Strömen mit steigender Reynoldszahl führen wir eine statistische Analyse der hydrodynamischen Erholung von ELBM-Simulationen durch. Um dies zu erreichen, erweitern wir das in Kapitel 1 vorgestellte Werkzeug, indem wir zu den Bilanzgleichungen den mesoskopischen Wirbel viskose Dissipationsterm hinzufügen, der sich aus Gleichung (4) ergibt. Parallel dazu untersuchen wir numerisch die approximierte makroskopische Formulierung Eq. (5), um sie gegen den gemessenen mesoskopischen Ausdruck Eq. (4) zu validieren.

Kapitel 4: Studie des ELBM impliziten SGS-Modells auf der Makroskala

In Kapitel 4 führen wir eine numerische Überprüfung der Annahmen durch, die bei jedem Schritt der C-E-Erweiterung des entropischen Parameters $\alpha(\boldsymbol{x}, t)$ getroffen wurden und die zur hydrodynamischen Wirbelviskosität (5) führten. Wir zeigen, dass das ELBM SGS-Modell nicht nur aus einem Begriff der Wirbelviskosität besteht und erhalten seine volle Gleichung. Die in diesem Kapitel vorgestellten Ergebnisse werden in [7] veröffentlicht.

Kapitel 5: Trägheitsreichweitenstatistik der Entropischen Lattice Boltzmann und Large-Eddy Simulationen in 3D-Turbulenzen

In Kapitel 5 führen wir eine Analyse der Trägheitsstatistik von turbulenten Geschwindigkeitsfeldern durch, indem wir ELBM mit einer hochauflösenden Direkten Numerischen Simulation (DNS) der NSG, die mit einem PS-Code durchgeführt wird, vergleichen. Darüber hinaus präsentieren wir auch die Analyse von zwei PS LES: eine mit der makroskopischen Approximation Wirbelviskosität Eq. (5) und eine mit einem Smagorinsky-Modell Eq. (6). Diese Ergebnisse werden in [8] veröffentlicht.

Kapitel 6: Beschleunigung der Lattice Boltzmann Strömungssimulation mit dem NVSH-MEM-Modell für die GPU-initiierte Kommunikation

In Kapitel 6 wird die Implementierung des LBM-Algorithmus auf GPU-beschleunigten Architekturen diskutiert und die NVSHMEM-Programmierschnittstelle für die GPU-initiierte Kommunikation vorgestellt. Wir zeigen einen Benchmark unseres Codes auf einer einzelnen GPU und präsentieren vorläufige Ergebnisse der Leistungsskalierung auf Multi-GPUs mit MPI mit dem Ziel, diese als Referenz für die Bewertung von NVSHMEM-Implementierungen auf Basis von in-Kernel-Kommunikation zu verwenden. Die in diesem Kapitel vorgestellten Ergebnisse sind Teil eines Artikels in Vorbereitung [9].

Kapitel 7: Fazit

In diesem abschließenden Kapitel fassen wir die Ergebnisse dieser Arbeit zusammen und zeigen Möglichkeiten für zukünftige Forschungsarbeiten auf.

ESTRATTO

Il metodo Lattice Boltzmann (LBM) è un numerico per la simulazioni di flussi alle scale mesoscopiche che ha acquisito molta importanza a causa della sua capacità di gestire fluidi complessi. Tuttavia, le sue applicazioni in flussi turbolenti risultano ancora limitate dalle presenza delle instabilità che si presentano al ridurre della viscosità. Il Lattice Boltzmann Entropico (ELBM) affronta questo problema incorporando nel classico metodo LBM il teorema H, ottenendo in questo modo una stabilità apparentemente incondizionata. In pratica, nel metodo ELBM l'operatore di collisione di Bhatnagar-Gross-Krook (BGK)[1] viene modificato consentendo al tempo di rilassamento di fluttuare attraverso la definizione di un parametro entropico $\alpha(\boldsymbol{x}, t)$. Il metodo ELBM è visto come un modello per la dinamica delle Scale Sotto Griglia (SGS) riconducibile alla classe di metodi impliciti utilizzati nelle Large Eddy Simulations (LES), dove alla normale viscosità cinematica viene aggiunta una viscosità extra derivante dal presupposto che l'equazione che collega il tempo di rilassamento mesoscopico con la viscosità macroscopica è ancora valida quando il tempo di rilassamento è lasciato fluttuare,

$$\nu_{\text{eff}}(\boldsymbol{x}, t) = c_s^2 \Delta_t \left(\frac{2\tau_0}{\alpha(\boldsymbol{x}, t)} - \frac{1}{2} \right) = \nu_0 + \delta\nu_e^M(\boldsymbol{x}, t), \tag{7}$$

dove τ_0 e ν_0 sono rispettivamente il tempo di rilassamento e la viscosità di ingresso passati al sistema. La dipendenza non lineare del parametro entropico rispetto alle funzioni di distribuzione non consente di esprimere direttamente la viscosità effettiva in termini di quantità macroscopiche e quindi il suo significato fisico rimane nascosto. L'approssimazione idrodinamica della extra viscosità derivante dal metodo entropico è stata proposta in Ref. [2] tramite l'espansione Chapman-Enskog (C-E) del parametro entropico,

$$\delta\nu_e^A \approx -c_s^2 \Delta t^2 \frac{1}{6\beta^2} \frac{\operatorname{Tr} \mathbf{S}^3}{\operatorname{Tr} \mathbf{S}^2}, \tag{8}$$

dove $S_{ij} = \frac{1}{2}(\partial_i u_j + \partial_j u_i)$ è il tensore degli sforzi. Tuttavia, ancora manca uno studio numerico dettagliato del modello SGS implicito prodotto dal metodo ELBM dal quale sia possibile anche estrarre il range di validità della formulazione macroscopica (8). Pertanto, non è ancora chiaro se l'ELBM agisca come un semplice stabilizzatore numerico o fornisca una rappresentazione accurata della dinamica turbolenta delle scale non risolte. L'obiettivo della tesi è di fare luce su questa questione nel contesto dei flussi turbolenti in condizioni omogenee ed isotrope (HIT) sia in geometrie bidimensionali che tridimensionali.

Il primo passo consiste nel quantificare la validità del metodo ELBM come chiusura implicita. In questo lavoro abbiamo sviluppato uno strumento di analisi per valutare l'accuratezza con cui l'idrodinamica di Navier-Stokes viene riprodotta dal metodo LBM

prendendo in riferimento simulazioni Pseudo-Spettrali (PS) in cui vengono risolte esplicitamente le equazioni di Navier-Stokes (NSE). In seguito abbiamo applicato questo strumento nell'analisi di simulazioni HIT ELBM sia bidimensionali che tridimensionali al variare del numero di Reynolds. Da una parte, gli spettri di energia hanno rivelato che ELBM è in grado di estendere l'interavllo di scale inerziali fino a 20 volte in più rispetto all'ultima simulazione LBM stabile. Dall'altra parte, l'ipotesi *a priori* che il metodo ELBM possa essere macroscopicamente descritto come un modello LES con viscosità extra (7) non può essere confermata chiaramente. Inoltre, abbiamo studiato la validità della formulazione approssimata di viscosità macroscopica (8). Scoprendo che questa cattura le dinamiche della extra viscosità prodotta dal metodo ELBM solo a bassi numeri di Reynolds, mentre fallisce nel regime di turbolenza sviluppata.

Per identificare analiticamente i punti deboli di tali approssimazioni, abbiamo controllato numericamente le assunzioni fatte ad ogni passo dell'espansione C-E del parametro entropico $\alpha(\boldsymbol{x}, t)$. In questo modo abbiamo potuto spiegare come la formulazione della viscosità macroscopica non riesca ad essere recuperata ad alti numeri di Reynolds, evidenziando la presenza di termini aggiuntivi la cui importanza cresce all'aumentare dei gradienti di velocità. Così abbiamo scoperto che il modello implicito ELBM non è composto solo da un solo termine di dissipazione viscosa legato al tempo di rilassamento effettivo come ipotizzato in letteratura [3], ma dipende anche da altri termini che non possono essere trascurati. Quest'ultimo risultato è di particolare importanza in quanto solleva interrogativi sulla validità di molti altri modelli SGS di viscosità implementati nel metodo LBM.

Il fatto che la chiusura ELBM sia più complessa di un semplice modello di extra viscosità non significa, in linea di principio, che non sia un buon modello per la turbolenza. Per valutare la qualità della chiusura ELBM, abbiamo confrontato la statistica dei campi di velocità nelle scale inerziali ottenuti da simulazioni ELBM, con quella proveniente da simulazione numeriche dirette (DNS) ad alta risoluzione. Alcuni risultati preliminari mostrano che il metodo ELBM è in grado di estendere l'intervallo inerziale catturando i corretti comportamenti intermittenti.

Inoltre, va osservato che l'extra viscosità macroscopica approssimata (8) ha una caratteristica interessante, poiché è qualitativamente simile alla formulazione di extra viscosità proposta da Smagorinsky [4],

$$\delta\nu_e^S = (C_S\Delta)^2\sqrt{2\,\mathrm{Tr}\,\mathbf{S}^2}, \tag{9}$$

dove C_S è un coefficiente adimensionale e Δ è la lunghezza caratteristica del filtro LES. Tuttavia, $\delta\nu_e^A$ introdotta in (8) non è definita positiva, il che significa che consente di riprodurre eventi di diffusione di energia all'indietro, dalle scale sotto griglia alle scale risolte. Implementando il modello (8) e il modello di Smagrosinsky (9) in un codice LES PS, abbiamo osservato che entrambe le chiusure producono una statistica simile nell'intervallo inerziale.

Una parte della tesi è dedicata allo studio di problemi numerici di natura pratica. Abbiamo scritto un'implementazione open-source del metodo LBM [5] per alte prestazioni, che supporta diversi reticoli sia 2D che 3D, e che può essere utilizzata per condurre simulazioni di flusso ad alta risoluzione utilizzando diversi modelli e forcing di turbolenza HIT in spazio di Fourier. In particolare, il codice è stato ottimizzato su architetture multi-GPU facendo uso di NVSHMEM, una nuova libreria di comunicazione incentrata su GPU. Questa implementazione viene confrontata con l'implementazione di comunicazione basata sulla CPU standard basata sulla libreria MPI (Message-Passing Interface). Nella tesi sono presentati i primi risultati di scalabilità sia "forte" che "debole" provenienti dal confronto della nuova libreria con l'implementazione MPI.

TRACCIA DELLA TESI

La tesi è organizzata come segue:

Capitolo 1: Introduzione

Nel primo capitolo abbiamo presentato il background e le tecniche numeriche su cui si basa il nostro lavoro. Riassumiamo brevemente le caratteristiche principali della fisica della turbolenza isotropo ed omogenea bidimensionale e tridimensionale ed introduciamo modelli di turbolenza basati su LES. Presentiamo i fondamenti del metodo LBM e della sua estensione incondizionatamente stabile, l'ELBM. Chiudiamo questo capitolo fornendo una breve panoramica dello stato dell'arte riguardante la questione dell'interpretazione del metodo ELBM come un modello implicito per la dinamica SGS. I capitoli da 2 a 5 si concentreranno su ulteriori discussioni su questo argomento.

Capitolo 2: Uno strumento numerico per la valutazione del recupero idrodinamico delle simulazioni LBM

Nel capitolo 2, studieremo il recupero idrodinamico del metodo LBM analizzando le relazioni di equilibrio esatte per energia ed enstrofia derivate dalla media delle equazioni del moto in sub-volumi di diverse dimensioni. Nel contesto di HIT 2D, prima convalidiamo questo approccio nel caso di turbolenza in decadimento, confrontando l'analisi di un insieme di simulazioni LBM con quelle provenienti da simulazioni PS. Successivamente conduciamo una validazione di simulazioni LBM di turbolenza forzata all'aumentare del numero di Reynolds, variando il tempo di rilassamento in ingresso al modello LBM. I risultati presentati in questo capitolo sono pubblicati in Ref. [6].

Capitolo 3: Studio a-priori del recupero dell'idrodinamica dell'ELBM e del modello SGS implicito

Nel capitolo 3, indaghiamo sulla validità dell'ipotesi a-priori che il metodo ELBM possa essere descritto macroscopicamente come un modello LES con extra viscosità (7). Per un insieme di flussi HIT bidimensionali e tridimensionali all'aumentare del numero di Reynolds, conduciamo un'analisi statistica del recupero idrodinamico delle simulazioni ELBM. Per fare ciò, estendiamo lo strumento introdotto nel capitolo 1 aggiungendo alle equazioni di bilancio il termine di dissipazione mesoscopica derivante da Eq. (7). In parallelo, studiamo numericamente la formulazione macroscopica approssimata Eq. (8) per convalidarla contro l'espressione mesoscopica misurata Eq. (7).

Capitolo 4: Studio del modello SGS implicito del metodo ELBM alle macroscale

Nel capitolo 4, eseguiamo un controllo numerico delle assunzioni fatte in ogni fase dell'espansione C-E del parametro entropico $\alpha(\boldsymbol{x}, t)$ che porta alla definizione della extra viscosità (8). Rileviamo che il modello SGS ELBM non consiste solo nei termini contenuti nella definizione della extra viscosità, e ne otteniamo la sua completa espressione. I risultati presentati in questo capitolo saranno pubblicati in [7].

Capitolo 5: Statistica nell'intervallo di scale inerziali prodotta del modello LBM entropico e dalle simulazioni LES per turbolenza 3D

Nel capitolo 5, conduciamo un'analisi della statistica del campo di velocità nell'intervallo di scale inerziali, confrontando l'ELBM con una simulazione numerica diretta ad alta risoluzione (DNS) condotta con un codice PS. Inoltre, aggiungiamo l'analisi di due PS LES: una equipaggiata con il modello derivante dall'approssimazione macroscopica di extra viscosità Eq. (8) e una in cui viene utilizzato il modello di Smagorinsky Eq. (9). Questi risultati saranno pubblicati in [8].

Capitolo 6: Accelerazione delle simulazioni LBM utilizzando il modello NVSHMEM per comunicazioni avviate dalla GPU

Nel capitolo 6 discutiamo l'implementazione dell'algoritmo LBM su architetture accelerate dalla GPU e presentiamo l'interfaccia di programmazione NVSHMEM per le comunicazioni avviate dalla GPU. Mostriamo una validazione del nostro codice su una singola GPU e presentiamo i risultati preliminari del ridimensionamento delle prestazioni su multi-GPU utilizzando MPI con l'obiettivo di utilizzarli come riferimento per valutare le implementazioni NVSHMEM basate sulle comunicazioni nel kernel. I risultati presentati in questo capitolo fanno parte di un documento in stato di preparazione [9].

Capitolo 7: Conclusioni

In questo capitolo conclusivo, riassumiamo i risultati del lavoro di tesi ed evidenziamo i possibili sviluppi futuri.

ACKNOWLEDGEMENTS

This thesis is the result of the continuous support of a large number of people during the past three years, both on an academic and a personal level.

My PhD adventure started at the Bergische Universitát Wuppertal (BUW) and therefore, I would like to address my first thanks to my supervisor, Prof. Matthias Ehrhardt and to PD. Andreas Bartel for their continuous support, kindness, and committtment. From the very beginning they have shown strong appreciation of my efforts. Thank you for believing in me.

I grew a lot as a scientist under the supervision of Prof. Luca Biferale at the Università degli studi di Roma "Tor Vergata" (UTOV). With him, I have learned how a rigorous scientific method should be applied to research. I have also learned significantly having the chance to work with Prof. Mauro Sbragaglia. The clarity of his reasoning always impressed me and systematically lead to an acceleration of my progress.

During these three years, I also had the opportunity to work with people at NVIDIA and at the NVIDIA Application Lab at the Juelich Supercomputing Center (JSC). I have been very keen on learning about High Performance Computing after discovering it in a workshop there. But I would not have been able to go very deep into the topic without the generous support of Jiri Kraus, Prof. Dirk Pleiter, and Dr. Mathias Wagner. At the JSC and at NVIDIA, I found a very welcoming environment with passionate people that are always willing to share their extensive knowledge and that can debug my code just looking at profiling results within a 20 minutes video call.

This European Joint Doctorate was coordinated by Prof. Constantia Alexandrou. I would like to thank her along with her collaborators at the Cyprus Institute for their hard work in making HPC-LEAP happen. I would like to also thank Daniela Elbing, Prof. Andreas Frommer, Prof. Michael Guenther, and Prof. Francesco Knechtli at BUW and Lucia Cori and Lorena Gerosi at UTOV for their logisitic support.

A Marie Curie fellowship provides a significant amount of fundings but also comes with the opportunity to attend a number of events and do internships. Thanks to the European Union tax payers, I have both learned how to clone myself and teleport. Too bad I had no time to write about that!

I am grateful to Prof. Luca Biferale, Prof. Matthias Ehrhardt, and Prof. Federico Toschi for reviewing the present thesis. I would also like to thank Prof. Luca Biferale, Prof. Matthias Ehrhardt, Prof. Bálint Farkas, Prof. Francesco Knechtli, Prof. Dirk Pleiter, and Prof. Mauro Sbragaglia for accepting to join my PhD committee.

I have had the chance to be working among a very welcoming group of fellow students at BUW. I would like to thank for those good times Christian, Christoph, Dmitry, Igor, José, Kai, Long, Markus, Michele, and Zuzana.

In the group at UTOV, I have met a whole bunch of passionate and ambitious physicits from whom I have learned extensively. I would like to thank Daniele, Felix, Francesca, Giorgos, Massimo, Matteo, Michele, Patricio, Simona, Xiao. We have also recently been joined at UTOV by the STIMULATE program students. Daniele, Fabio, Guilherme, Lokahit, Maddy, your contagious motivation and energy was of great help in writting this thesis. Thank you. However, as fellows of the HPC-LEAP version 2.0 project, you have to do better than us, so stop asking me so many questions and get back to work! Good luck!

I also feel grateful to the group in JSC that adopted me during this PhD. I would like to thank Andreas, Markus, Salem, Stepan, and last but not least, Teodor whose benevolent efforts will not be forgotten. I am sorry you guys had to answer so many of my stupid questions!

This fellowship was an intense experience but I was lucky to find in the HPC-LEAP fellows group such talented guys. I enjoyed every minutes of the workshop and conferences with them and they taught me so much. Thank you Alessandro, Andrew, Aurora, Felix, Giorgia, Giorgos, Ibrahim, Salvatore, Simone, Slava, Srijit, Teodor, Thomas, Vanessa, Wenping, and Xiao. You really made this PhD program, an enjoyable one! A special mention to Felix, our student representative, should be made here. The administrative burdern of being part of a multi-institution project funded by the European Commission can be, at times, quite high. His continuous commitment over the years and his determination to ensure that the every need of his fellow students are met should be noted. Thank you for showing us the way Felix!

Friends are family you meet along the way and some of the people I met during those past three years had the most amazing influence on me. Those three years were full of ups and downs and I would like to espescially thank Alessandro, Dmitry, Ludwig, Michele, Patricio, and Teodor for being the reason befinh all the ups and for their support through the downs.

Last but not least, I would like to express my deepest gratitude to my parents for their patience and support along the years.

CONFERENCES AND PUBLICATIONS

Contributed talks

- 13th of October, 2016, HPC Applications to Turbulence and Complex Flows, Rome, Italy, Title: ”Entropic Lattice Boltzmann Method: An implicit Large-Eddy Simulation?”

- 24th of May 2017, FSIM 2017, Naples, Italy, Title: ”Entropic Lattice Boltzmann Method: An implicit Large-Eddy Simulation?”

- 13th of July 2017, 26th International Conference on Discrete Simulation of Fluid Dynamic, Erlangen, Germany Title: ”Entropic Lattice Boltzmann Method. An implicit Large-Eddy Simulation?”

- 20th of November 2017, 70th Annual Meeting of the APS Division of Fluid Dynamics, Denver, USA, Title: ”Entropic Lattice Boltzmann Method: an implicit Large-Eddy Simulation?”

- 9th of February, 2018, COST Flowing Matter meeting, Lisbon, Portugal, Title: ”Assessing Entropic LBM as an implicit Large Eddy Simulation: Lesson from the 2D case”

- 27th of June 2018, 27th International Conference on Discrete Simulation of Fluid Dynamic, Worcester, USA, Title: ”Entropic Lattice Boltzmann: Study of the Implicit Subgrid scale model”

- 11th of July, 2018, HPC-LEAP closing conference, Cambridge, UK, Title: ”Entropic Lattice Boltzmann: Study of the Implicit Subgrid scale model”

- 10th of September, 2018, 12th European Fluid Mechanics Conference, Vienna, Austria, Title: ”Entropic Lattice Boltzmann: Study of the Implicit Subgrid scale model”

Publications

- **G. Tauzin, L. Biferale, M. Sbragaglia, A. Gupta, F. Toschi, A. Bartel and M. Ehrhardt**. A numerical tool for the study of the hydrodynamic recovery of the Lattice Boltzmann Method. **Computers & Fluids**, 172:241-250, 2018

- **In preparation.** Title: "Study of the implicit sub-grid scale modeling within the Entropic Lattice Boltzmann".

- **In preparation.** Title: "Inertial range statistics of the Entropic Lattice Boltzmann and Large-Eddy Simulations in 3D turbulence".

- **In preparation.** Title: "Accelerating Lattice Boltzmann flows simulation using GPU-initiated communications with NVSHMEM".

TABLE OF CONTENTS

CHAPTER

APPENDICES

CHAPTER 1

INTRODUCTION

This chapter presents a description of both the theoretical and numerical tools used and developed within this thesis. In Section 1.1, we provide a brief summary of the theory of Homogeneous Isotropic Turbulence (HIT) both for two- and three-dimensional flows. Then, in Section 1.2, we present the technical difficulties in simulating such flows along with the Large-Eddy Simulation (LES) techniques commonly used to model turbulence. In Section 1.3, we introduce the Lattice Boltzmann Method (LBM), a mesoscale flow solver, while in Section 1.4 we detail an unconditionally stable LBM, the Entropic Lattice Boltzmann Method (ELBM), which is of main importance in this thesis.

1.1 Physics of turbulence

The simulation of turbulent flows pertains to a vast diversity of applications [10]. We focus here on theoretical turbulent flows that are incompressible, homogeneous, and isotropic in both two and three dimensions and that lives in a periodic box of volume $V_0 = L_0 \times L_0 \times L_0$. In this section, we summarize the basic concepts of the theory of turbulence in such systems.

1.1.1 Navier-Stokes Equations

The evolution of the turbulent flows of interest are modeled by the incompressible Navier-Stokes equations (NSE):

$$\begin{aligned} \nabla \cdot \boldsymbol{u} &= 0 \\ \frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} &= -\frac{1}{\rho}\nabla p + \nu_0 \Delta \boldsymbol{u} - \gamma\, \boldsymbol{u} + \boldsymbol{F} \end{aligned} \tag{1.1}$$

The first equation stems from the conservation of mass for an incompressible fluid, while the second equation is obtained from the conservation of momentum. Here $\boldsymbol{u}$ is the velocity field, p is the pressure, ν_0 is the kinematic viscosity and $\boldsymbol{F}$ is an external forcing, here taken as a homogeneous and isotropic injection of energy at a scale ℓ_{in}. In this thesis, we put ourselves in the theoretical setting of a flow in a periodic box of size L_0. To be able to reach a statistically stationary state, in the case of two-dimensional flows, we have included to the momentum equation a large scale drag term, $-\gamma\,\boldsymbol{u}$.

1.1.2 Control parameters

The dynamics of a flow governed by the NSE depends only on a set of two dimensionless parameters representing the relative importance of the non-linear term, which contributes to destabilizing the flow, in comparisons with the two dissipation terms, which stabilize it. The first one is the Reynolds number Re, defined as the ratio of inertial forced to viscous forces,

$$Re = \frac{|(\boldsymbol{u}\cdot\nabla)\boldsymbol{u}|}{\nu_0|\nabla^2\boldsymbol{u}|}, \tag{1.2}$$

and the second one is its large-scale drag counterpart R_γ

$$R_\gamma = \frac{|(\boldsymbol{u}\cdot\nabla)\boldsymbol{u}|}{\gamma\,|\boldsymbol{u}|}, \tag{1.3}$$

Rewriting the NSE equations using the non-dimensional variables

$$\tilde{\boldsymbol{x}} = \frac{\boldsymbol{x}}{\ell_{in}}; \qquad \tilde{\boldsymbol{u}} = \frac{\boldsymbol{u}}{U}; \qquad \tilde{t} = \frac{t}{\ell_{in}/U}; \qquad \tilde{p} = \frac{p}{\rho U^2}; \qquad \tilde{\boldsymbol{F}} = \frac{\boldsymbol{F}}{\ell_{in}U^2}, \tag{1.4}$$

based on the root mean squared velocity U and the forcing length scale ℓ_{in}, we obtain

$$\frac{\partial\tilde{\boldsymbol{u}}}{\partial\tilde{t}} + (\tilde{\boldsymbol{u}}\cdot\tilde{\nabla})\tilde{\boldsymbol{u}} = -\tilde{\nabla}\tilde{p} + \frac{1}{Re}\tilde{\nabla}^2\tilde{\boldsymbol{u}} - \frac{1}{R_\gamma}\tilde{\boldsymbol{u}} + \tilde{\boldsymbol{F}}, \tag{1.5}$$

with $Re = \frac{U\ell_{in}}{\nu_0}$ and $R_\gamma = \frac{U}{\gamma\ell_{in}}$.

Eq. (1.5) highlights the similarity principle, which states is that any system with the same initial conditions, Re, and R_γ will evolve towards the same solution.

Whenever the Reynolds number is small, $Re \ll 1$, the non-linear term becomes negligible compared to the viscous dissipation. The flow is said to be in the laminar regime and the solution is linear and smooth. On the other hand, as we increase Re, a series of transitional stages occur and the flow becomes chaotic. The chaotic behavior of turbulent flows, *i.e.* high Reynolds number flows, implies that any small uncertainty in the initial conditions will lead to very different flow evolutions making impossible a deterministic prediction. Therefore, one can only hope to build a statistical theory for describing the physical system.

Besides, since Richardson [11], it is known that in 3D, the generated vortex structures called eddies split into smaller and smaller eddies until they reach a scale η at which the Reynolds number defined for this scale $Re_\eta = \frac{\eta U_\eta}{\nu_0}$ is small enough to allow their dissipation. Thus, turbulent flows are multi-scale systems, in which energy cascades down to small scales and an increase in the Reynolds number results in an increase of the range of scales involved in the dynamics.

1.1.3 Fourier space representation and energy spectrum

In order to study a multi-scale system where energy cascades across scales, one first need to introduce the concept of 'scale'. As commonly done, we use the Fourier space-based definition. A velocity field $\boldsymbol{u}(\boldsymbol{x}, t)$ can be reconstructed through its Fourier series;

$$\boldsymbol{u}(\boldsymbol{x}, t) = \sum_{\boldsymbol{k}} \hat{\boldsymbol{u}}(\boldsymbol{k}, t) e^{i\boldsymbol{k}\cdot\boldsymbol{x}} \qquad \text{with} \qquad \hat{\boldsymbol{u}}(\boldsymbol{k}, t) = \frac{(2\pi)^d}{L_0^d} \int_0^{L_0} \boldsymbol{u}(\boldsymbol{x}, t) e^{-i\boldsymbol{k}\cdot\boldsymbol{x}} dx^d,$$

with d the dimension of the system, $\boldsymbol{k} = k_0\mathbf{n}$ the wavenumber, $\mathbf{n} \in \mathbb{Z}^d$ the mode number and $k_0 = \frac{2\pi}{L_0}$ the smallest wavenumber in the system.

We define the scale $l = \frac{2\pi}{k}$ as all modes of wavenumber amplitude $k = |\boldsymbol{k}|$ and the energy at that scale as the sum of all corresponding energies. The energy spectrum $E(k, t)$ is then defined as the sum of the energy at all wavenumbers contained in a spherical shell defined by $k \leq |\boldsymbol{k}| < k + \Delta k$, with $\Delta k = \frac{2\pi}{L_0}$:

$$E(k, t) = \frac{1}{\Delta k} \int_{k \leq |\boldsymbol{k}| < k+\Delta k} |\hat{\boldsymbol{u}}(\boldsymbol{k}, t)|^2 d\boldsymbol{k}. \tag{1.6}$$

We can recover the total energy of the system by summing over all shells the spectrum:

$$E(t) = \Delta k \sum_{k} E(k, t). \tag{1.7}$$

1.1.4 Isotropic 3D turbulence

Turbulence is often rightfully designated as the last open problem of classical mechanics. Indeed, no theory is able to deduce from the NSE results on the statistics of fluid quantities. However, by formulating a set of reasonable hypotheses, in 1941, Kolmogorov derived a phenomenological scaling theory (K41) [12] leading to dimensional predictions of the kinetic-energy spectrum [13] and the dissipative scales.

Conservation laws

In 3D turbulence, there are two global quantities that are of significant interest, the mean energy, $E = \langle \frac{1}{2}\boldsymbol{u} \cdot \boldsymbol{u} \rangle$, and the mean helicity, $H = \langle \frac{1}{2}\boldsymbol{u} \cdot \boldsymbol{\omega} \rangle$, where the angular brackets denotes averages over the periodic domain V_0, and $\boldsymbol{\omega} = \boldsymbol{\nabla} \times \boldsymbol{u}$ is the vorticity.

- Conservation of mean energy:

$$\frac{d}{dt}E = -\nu_0 \langle |\nabla \boldsymbol{u}|^2 \rangle - \gamma \langle |\boldsymbol{u}|^2 \rangle + \langle \boldsymbol{F} \cdot \boldsymbol{u} \rangle = -\varepsilon_{\nu_0} - \varepsilon_\gamma + \varepsilon_{in}, \tag{1.8}$$

  where ε_{ν_0} is the rate of viscous energy dissipation, ε_γ is the rate of energy dissipation due to the drag, and ε_{in} is the rate of energy injection.

- Conservation of mean helicity:

$$\frac{d}{dt}H = -\nu_0 \left\langle (\nabla \wedge \omega) \cdot \omega \right\rangle - \gamma \langle \boldsymbol{u} \cdot \omega \rangle + \langle \boldsymbol{F} \cdot \omega \rangle. \tag{1.9}$$

In the absence of drag ($\gamma = 0$), and in the inviscid ($\nu_0 = 0$), unforced ($\boldsymbol{F} = 0$) case, both energy and helicity are globally conserved. These conservation laws, along with the conservation of mass and momentum, are essential characteristics of the flow and its evolution. They are also important to consider whenever deriving a turbulence model. Indeed, if the model does not conserve those quantities, the resulting simulated solution will obviously be non-physical.

Phenomenological Kolmogorov 1941 theory

In the wake of Richardson's cascade of energy [11] for turbulent flow and in the limit of infinite Reynolds numbers, at small scales $\ell < \ell_{in}$, and far away from boundaries, Kolmogorov [12] made the following hypotheses:

1. *All the possible symmetries of Navier-Stokes equations, usually broken by the mechanism producing turbulence, are restored in a statistical sense.*

2. *Turbulent flows are self-similar at small scales.*

3. *All the small-scale statistical properties are uniquely and universally determined by the scale ℓ and the mean energy dissipation rate ε.*

He therefore hypothesised that there is a significant scale separation between the large scales ℓ_{in} and the small dissipative scales ℓ_{ν_0}, inferring the existence of an intermediate range of scales ℓ of self-similar eddies that are neither depending on the large-scale eddies nor affected by viscosity ($\ell_{in} \ll \ell \ll \ell_{\nu_0}$).

In order to understand the physical meaning of those hypotheses, we look at the statistics of the longitudinal velocity increments [14]

$$\delta_r \boldsymbol{u}_\parallel \equiv (\boldsymbol{u}(\boldsymbol{x}+\boldsymbol{r}) - \boldsymbol{u}(\boldsymbol{x})) \cdot \frac{\boldsymbol{r}}{\hat{\boldsymbol{r}}}, \text{ with } \hat{\boldsymbol{r}} = \frac{1}{r}\boldsymbol{r} \text{ and } r = \|\boldsymbol{r}\|$$

The self-similar behaviour hypothesis means that there exists a scaling exponent $h \in \mathbb{R}$ such that

$$\delta_{\lambda r} \boldsymbol{u}_\parallel \equiv \lambda^h \delta_r \boldsymbol{u}_\parallel \quad \forall \lambda \in \mathbb{R}_+,$$

.

From the universality hypothesis and using dimensional analysis we can obtain that the second order longitudinal structure function, $S_2^\parallel = \langle (\delta_r \boldsymbol{u}_\parallel)^2 \rangle$ follows the scaling equation

$$\langle (\delta_r \boldsymbol{u}_\parallel)^2 \rangle \sim C(\varepsilon_{\nu_0} r)^{2/3}, \tag{1.10}$$

where the angular brackets $\langle \cdot \rangle$ denote the spatial average (assuming a statistically homogeneous and isotropic system) and C is a non-dimensional universal constant.

Further assuming that the energy dissipation ε has a finite limit as $Re \to \infty$ (see section 1.1.4), Komogorov [12] obtained the four-fifth law

$$\langle (\delta_r \boldsymbol{u}_\parallel)^3 \rangle = -\frac{4}{5} \varepsilon_{\nu_0} r. \tag{1.11}$$

This implies that the p-th order longitudinal structure function has the following scaling behavior:

$$S_p^\parallel \langle (\delta_r \boldsymbol{u}_\parallel)^p \rangle \sim C_p (\varepsilon_{\nu_0} r)^{p/3}. \tag{1.12}$$

As a result of the formulation for the scaling behavior for the second order longitudinal structure function Eq. (1.10), Kolmogorov obtained the celebrated predictions for which the energy spectrum follow a $k^{-5/3}$ power-law:

$$E(k) = C_K \varepsilon_{\nu_0}^{2/3} k^{-5/3} \tag{1.13}$$

with C_K the dimensionless Kolmogorov constant. On Figure 1.1, we show such the sketch of an energy spectrum highlighting the predicted $-5/3$ slope and the cascading of energy accross scales.

Besides, Kolmogorov conjectured [12] that, a high, but not infinite Reynolds numbers, statistical properties at the small-scale ($l \ll l_{in}$) are solely and universally dependent on the scale l, the energy dissipation rate ε_{ν_0}, and the kinematic viscosity ν_0. Therefore,

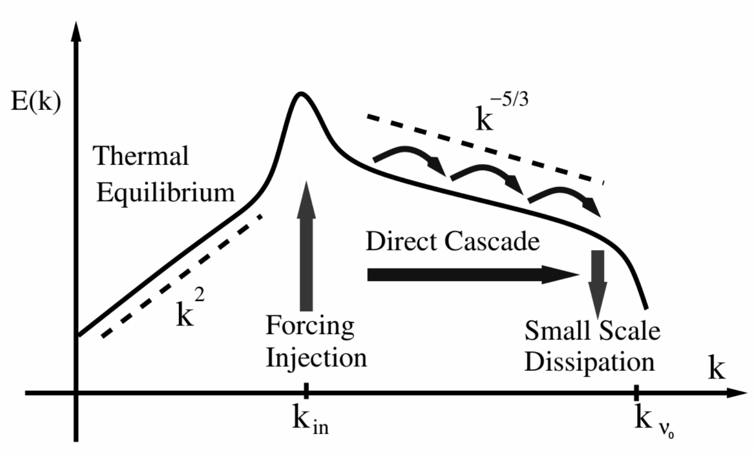


FIGURE 1.1. Sketch of the energy spectrum in log-log scale $E(k)$ versus k highlighting the direct energy cascade and Kolmogrov's -5/3 slope prediction. Energy is injected at wavenumber $k_{in} = 2\pi/\ell_{in}$ and dissipated at $k_{\nu_0} = 2\pi/\ell_{\nu_0}$ ($\gamma = 0$ is assumed). Taken from Ref. [15] with permissions.

following a dimensional analysis, one can express the length-, velocity-, and time-scales at which the viscous forces becomes effective;

$$\eta \propto \left(\frac{{\nu_0}^3}{\varepsilon_{\nu_0}}\right)^{1/4}, \quad \delta_\eta u \propto (\nu_0 \varepsilon_{\nu_0})^{1/4}, \quad \tau_\eta \propto \left(\frac{\nu_0}{\varepsilon_{\nu_0}}\right)^{1/2}, \tag{1.14}$$

which are respectively the Kolmogorov's scale η ($\propto \ell_{\nu_0}$), the velocity fluctuations at scale η and the 'eddy turnover time' associated with the scale η.

Departure from K41

In practice, on experimentally observed or numerically simulated turbulent flow, we observe the presence of disorderly strong fluctuations over a range of scales in both space and time. This phenomenon is known as small scales intermittency and its origin is one of the most significant open question of turbulence. Intermittency is usually highlighted by the departure from the K41-predicted scaling exponent of structure functions of order p, $S_p^{\|}(r)$ as a function of the distance r of the points

$$S_p^{\|}(r) \equiv \langle(\delta_r \boldsymbol{u}_{\|})^p\rangle = \langle[(\boldsymbol{u}(\boldsymbol{x}+\boldsymbol{r}) - \boldsymbol{u}(\boldsymbol{x}))\cdot\hat{\boldsymbol{r}}]^p\rangle \sim C_p(\varepsilon_{\nu_0} r)^{\zeta(p)}, \tag{1.15}$$

where $\zeta(p)$ is different from the predicted linear behaviour, $p/3$.

In practice to measure intermittency we calculate the kurtosis

$$K(r) \equiv \frac{\left\langle((\delta_r \boldsymbol{u}_{\|})^4\right\rangle}{\left\langle((\delta_r \boldsymbol{u}_{\|})^2\right\rangle^2} = \frac{S_4^{\|}(r)}{\left(S_2^{\|}(r)\right)^2}. \tag{1.16}$$

Assuming self-similarity, $K(r)$ is constant in the inertial range, therefore the deviation of $K(r)$ as $r \to \eta$ gives us a measure of the degree of intermittent behavior.

Moreover, She-Lêveque showed in Ref. [16] that the exponent of $S^{\parallel}_{(p)}$ can be modelled as

$$\xi_p = \frac{p}{9} + 2\left(1 - \left(\frac{2}{3}\right)^{p/3}\right). \tag{1.17}$$

It is also worth pointing out that in 1986, Parisi and Frisch [14] suggested that turbulent flows have a superposition of different scale-invariance, *i.e.*, there exist different values of the scaling exponent h occurring with a probability. The outcome model is known as the multifractal model and allow an excellent recovery of the experimentally and numerically measured structure-function exponents.

Dissipative anomaly

As mentioned above, Kolmogorov's four-fifth law Eq. (1.11) has been obtained assuming the finiteness of the energy dissipation (repeated indices are meant summed upon)

$$\varepsilon_{\nu_0} = \nu_0 \langle (\partial_j u_i + \partial_i u_j)^2 \rangle \xrightarrow[Re\to\infty]{} C, \tag{1.18}$$

where $\langle \cdot \rangle$ denotes volume average and C is a positive constant.

While there is no mathematical proof explaining this phenomenon, it has been systematically observed in numerical simulations and experiments and is referred to as the *dissipative anomaly.* Because taking the limit of $Re \to \infty$ is equivalent to taking $\nu \to 0$, one could conjecture that $\varepsilon_{\nu_0} \xrightarrow[\nu_0 \to 0]{} 0$ [17, 18].

Inspecting Eq. (1.18), we notice that, as the Reynolds number increases, the velocity gradients become increasingly singular in order to compensate for the viscosity reduction. Indeed, peaks of dissipation are localized intermittently, solely in small areas and the intensity and probability of extreme events in the velocity gradients sharply increase with Re [19].

1.1.5 Isotropic 2D turbulence

While 2D flows do not exist in the real world, their theoretical study provides an understanding of quasi-2D flows that evolves in a domain for which the depth L_z is very small compared to the other dimensions L_x, L_y such as geostrophic flows. Indeed, 2D turbulence is very different in substance from 3D turbulence, as it is characterized by two positive-definite inviscid quadratic invariants which cascade in opposite directions. In this section, we briefly summarize those results.

Conservation laws

In 2D, helicity is not of interest as it is identically zero. However, besides the energy, we have a second globally conserved quantity, the enstrophy ($\Omega = \frac{1}{2}\omega \cdot \omega$).

- Conservation of mean energy:
  As in 2D, the mean energy is conserved and its evolution in time is given Eq. (1.8).

- Conservation of mean enstrophy:
  $$\frac{d}{dt}\Omega = -\nu_0\langle|\nabla\omega|^2\rangle - \gamma\langle|\omega|^2\rangle + \langle\omega\cdot(\nabla\wedge\boldsymbol{F})\rangle = -\zeta_{\nu_0} - \zeta_\gamma + \zeta_{in}, \tag{1.19}$$
  where ζ_{ν_0} is the rate of viscous enstrophy dissipation, ζ_γ is the rate of enstrophy dissipation due to the drag, and ζ_{in} is the rate of enstrophy injection.

These two quantities are invariants in the inviscid ($\nu_0 = 0$), unforced ($\boldsymbol{F} = 0$) cases and in the absence of drag ($\gamma = 0$). The fact that they are both positive-definite implies that there is a dual split cascade, with the energy going to large scales and the enstrophy going to small scales.

Phenomenological Kraichnan-Leith-Batchelor theory

In 2D turbulence, the Richardson cascade of energy towards small scales does not hold. It is known since Fjortoft [20] (and more quantitatively since Kraichnan, Leith and Batchelor [21, 22, 23]) that energy and enstrophy cannot both cascade to small scales and that the presence of a second sign-definite inviscid quadratic invariant leads to the presence of a dual cascade: enstrophy goes to small scales ℓ_{ν_0} where it is dissipated by viscous effects, while energy goes to large scales ℓ_γ where it is dissipated by the large-scale friction. Following Kolmogorov hypotheses, the two intervals of scales $\ell_\gamma \ll \ell \ll \ell_{in}$ and $\ell_{in} \ll \ell \ll \ell_{\nu_0}$ are the inertial ranges over which statistics can be assumed universal.

In the indirect energy cascade inertial sub-range, $\ell_\gamma \ll \ell \ll \ell_{in}$, using the same arguments than in the K41 theory, we obtain the same prediction for the spectrum slope:

$$E(k) = C'_K \varepsilon_{\nu_0}^{2/3} k^{-5/3} \tag{1.20}$$

with C'_K another dimensionless Kolmogorov constant.

In the direct enstrophy cascade inertial sub-range, $\ell_{in} \ll \ell \ll \ell_{\nu_0}$, we can adapt K41 reasoning on a sub-range dominated by energy transfer to a sub-range dominated by enstrophy transfer at a rate ζ_{ν_0}. We obtain the Batchelor-Kraichnan prediction for the energy spectrum slope:

$$E(k) = C_B \zeta_{\nu_0}^{2/3} k^{-3} \tag{1.21}$$

with C_B a dimensionless constant. The KLB prediction for the energy spectrum of forced 2D HIT are summarized on the sketch Fig 1.2

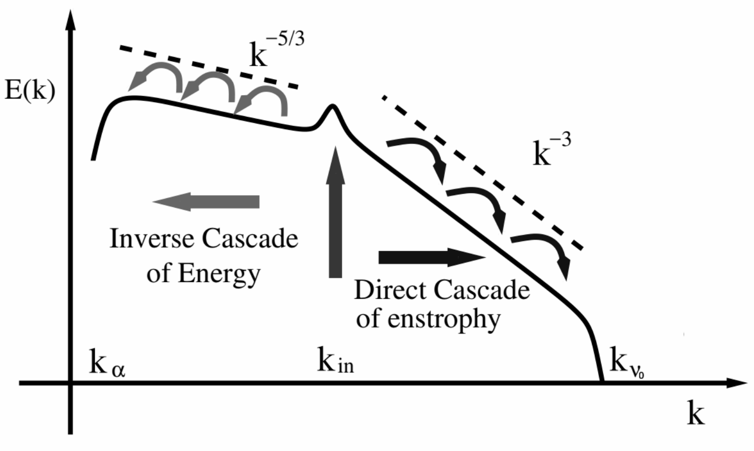


FIGURE 1.2. Sketch of the energy spectrum in log-log scale $E(k)$ versus k highlighting the indirect energy cascade's -5/3 slope prediction and the direct enstrophy cascade's -3 slope prediction. Energy is injected at wavenumber $k_{in} = 2\pi/\ell_{in}$, dissipated by viscosity at $k_{\nu_0} = 2\pi/\ell_{\nu_0}$ and by drag at $k_\gamma = 2\pi/\ell_\gamma$. Taken from Ref. [15] with permissions.

1.2 Turbulence modeling

As pointed out in the previous section, the range scales involved in a turbulent flow expand as the Reynolds number increases, making the Direct Numerical Simulation (DNS) of such flows computationally prohibitive. Therefore, we present in this section the Large-Eddy Simulation (LES) technique to reduce the number of degree of freedoms involved in the simulation of turbulent flows and introduce some of the available closure Sub-Grid Scale (SGS) models.

1.2.1 Direct Numerical Simulation

From section 1.1, we saw that in both 2D and 3D turbulence, the increase of the Reynolds numbers leads to an extension of the inertial sub-range of the direct cascade. This means that the smallest scale of motion of the flow, η, gets smaller with increasing the Reynolds number. From the phenomenological theories KLB in 2D and K41 in 3D, we can get the corresponding dissipative length-scales:

$$\eta^{2D} \propto \ell_{in} Re^{-1/2} \qquad \text{and} \qquad \eta^{3D} \propto \ell_{in} Re^{-3/4}. \tag{1.22}$$

DNS requires solving the NSE on a grid with a sufficient resolution to capture the smallest scales of motion of the flow. Therefore, the number of collocation points $N \sim \ell_{in}/L_0$ is

$$N^{2D} \sim Re \qquad \text{and} \qquad N^{3D} \sim Re^{9/4}, \tag{1.23}$$

in 2D and 3D respectively.

Therefore, turbulence at very high Reynolds numbers quickly becomes challenging even on the latest supercomputers. For this reason, there has been a considerable interest in the development of models that can be used to simulate real-world relevant quantities.

1.2.2 Large-eddy simulations

Large-Eddy Simulation (LES) is a workaround which allows a reduction of the number of degrees of freedom. It is acknowledged in the engineering community as a cost-effective alternative to DNS [24, 25, 26]. The principle of LES is to solve flow scales up to a cut-off and to filter the small scales out. As large scales and smaller scales are coupled, unresolved small scales have to be modeled using a so-called subgrid-scale (SGS) model.

Filtered NSE

In LES, the small scales are filtered using a filter kernel G^Δ associated to the filter width Δ. This coarse-graining operation applied to the velocity gives:

$$\overline{\boldsymbol{u}}^\Delta(\boldsymbol{x},t) \equiv \int_{L_0^3} d\boldsymbol{x}'\ G^\Delta(|\boldsymbol{x}-\mathbf{x}'|)\ \boldsymbol{u}(\boldsymbol{x}',t) = \sum_{\boldsymbol{k}\in\mathbb{Z}^3} \hat{G}^\Delta(\boldsymbol{k})\ \hat{\boldsymbol{u}}(\boldsymbol{k},t)e^{i\boldsymbol{k}\boldsymbol{x}}. \tag{1.24}$$

The most common filter is the spectral cutoff easily defined in Fourier space as;

$$G^\Delta(|\boldsymbol{k}|) = \begin{cases} 1, & \text{if } |\boldsymbol{k}| < 2\pi/\Delta \\ 0, & \text{otherwise} \end{cases} \tag{1.25}$$

which filters out all wavenumbers k above a given threshold $2\pi/\Delta$.

Another popular filter is the Gaussian smooth filter, which writes

$$G^\Delta(|\boldsymbol{k}|) = \exp{(-\frac{|\boldsymbol{k}|^2\Delta^2}{24})}. \tag{1.26}$$

Applying a filter kernel to the NSE (1.1), we obtain the governing equation for LES:

$$\partial_t \overline{\boldsymbol{u}} + \overline{(\boldsymbol{u} \cdot \nabla)\boldsymbol{u}} = -\nabla \overline{p} + \nu_0 \Delta \overline{\boldsymbol{u}} - \gamma\, \overline{\boldsymbol{u}}. \tag{1.27}$$

The filtered Navier-Stokes equations, eq. (1.27), cannot be simulated as the filtered non-linear term $\overline{(\boldsymbol{u} \cdot \nabla)\boldsymbol{u}}$ involves the product of two unfiltered fields, and thus cannot be recovered from the filtered velocity field $\overline{\boldsymbol{u}}$. Therefore, we introduce the subgrid-scale (SGS) tensor

$$\tau_{ij}(\boldsymbol{u}, \boldsymbol{u}) = \overline{u_i u_j} - \overline{u}_i \overline{u}_j \ , \tag{1.28}$$

and rewrite Eq. (1.27) as

$$\partial_t \overline{\boldsymbol{u}} + (\overline{\boldsymbol{v}} \cdot \nabla)\overline{\boldsymbol{u}} = -\nabla \overline{p} - \nabla \cdot \tau(\boldsymbol{u}, \boldsymbol{u}) + \nu_0 \Delta \overline{\boldsymbol{u}} - \gamma\, \overline{\boldsymbol{u}}. \tag{1.29}$$

Eq. (1.29) is the NSE (1.1) written for the filtered velocity field $\overline{\boldsymbol{u}}$ with an additional term involving the SGS-tensor. This extra-term is not closed in terms of filtered quantities and requires SGS modeling. Hence any LES implementation is based on the choice of (i) the filter G^{Δ} and (ii) the SGS model. For a review on the existing filters and models, the reader is referred to Refs. [27, 28, 29].

Eddy viscosity models

Here we derive the class of the so-called eddy viscosity models which lead to a NSE equation for the filtered velocity $\overline{\boldsymbol{u}}$ with an effective viscosity ν_{eff} fluctuating in space and time:

$$\partial_t \overline{\boldsymbol{u}} + (\overline{\boldsymbol{v}} \cdot \nabla)\overline{\boldsymbol{u}} = -\nabla \overline{p} + \nu_{\text{eff}}(\boldsymbol{x}, t) \Delta \overline{\boldsymbol{u}} - \gamma\, \overline{\boldsymbol{u}}, \tag{1.30}$$

where $\nu(\boldsymbol{x}, t) = \nu_0 + \delta\nu_e(\boldsymbol{x}, t)$, with ν_0 the fixed input viscosity, and $\delta\nu_e$ the fluctuating eddy viscosity.

Indeed, this can be obtained by setting the deviatoric part of the SGS-stress tensor τ_{ij} as

$$\tau^r_{ij} - \frac{1}{3}\tau_{kk} = -2\delta\nu_e \overline{S}_{ij}, \tag{1.31}$$

with $\overline{S}_{ij} = \frac{1}{2}(\partial_i \overline{u}_j + \partial_j \overline{u}_i)$, the resolved strain-rate tensor.

Assuming that the energy production and dissipation of the small scales are in equilibrium, Smagorinsky [4] proposed the first SGS model. The Smagorinsky eddy viscosity writes

$$\delta\nu^S_e = (C_S \Delta)^2 \sqrt{2 \overline{S_{ij}}\, \overline{S_{ij}}} \propto |\overline{\mathbf{S}}| \tag{1.32}$$

The Smagorinsky model has attracted a large interest from both the scientific and industrial communities and remains one of the most robust SGS models. However, it fails to model the transfer of energy from unresolved to resolved scales (the so-called energy backscatter) is not properly modeled as the eddy viscosity $\delta\nu_e^S$ is positive-definite and thus the model remains purely dissipative.

1.3 Lattice Boltzmann Method

The Lattice Boltzmann Method (LBM) is a mesoscale flow solver that has been gaining popularity because of it is highly scalable and suitable for parallel computing, as well as its ability to deal with multiple physics and complex boundary conditions. In this section, we give an introduction of LBM and highlight the stability challenge in using it as a turbulent flow solver and discuss techniques used to combine LES and LBM.

1.3.1 Boltzmann Equation

Considering a d-dimensional fluid, the probability of finding a particle at position $\boldsymbol{x}$ and velocity $\boldsymbol{v}$ at a given time t is given by the particle distribution function $f(\boldsymbol{x}, \boldsymbol{v}, t)$. Its evolution is given by the Boltzmann equation which, in the abscence of external forces reads:

$$\frac{\partial f}{\partial t} + \boldsymbol{v} \cdot \nabla f = \Omega(f) \tag{1.33}$$

where Ω is the collision operator. The collision operator is popularly modeled by the Bhatnagar-Gross-Krook (BGK) [1] relaxation towards a local equilibrium with a relaxation time τ. The BGK operator writes

$$\Omega(f) = -\frac{1}{\tau}\left[f - f^{eq}\right], \tag{1.34}$$

where τ is the relaxation time and f^{eq} the local equilibrium distribution function, described by a Maxwell-Boltzmann distribution:

$$f^{eq} = \rho \left(\frac{m}{2k_BT}\right)^{d/2} \exp\left(-\frac{m}{2k_BT}(\boldsymbol{v} - \boldsymbol{u})^2\right), \tag{1.35}$$

with k_B is the Boltzmann constant and m the particle mass. We will use a natural unit system for which $m = k_B = 1$ in this section. Macroscopic quantities such as mass density ρ, macroscopic velocity $\boldsymbol{u}$ and thermodyncamic temperature T are obtained by taking the moments of the particle distribution function:

$$\begin{aligned}\rho(\boldsymbol{x},t) &= \int f(\boldsymbol{x},\boldsymbol{v},t)d\boldsymbol{v},\\ \rho(\boldsymbol{x},t)\boldsymbol{u}(\boldsymbol{x},t) &= \int f(\boldsymbol{x},\boldsymbol{v},t)\boldsymbol{v}d\boldsymbol{v}, \text{ and}\\ \rho(\boldsymbol{x},t)T(\boldsymbol{x},t)\, d &= \int f(\boldsymbol{x},\boldsymbol{v},t)\|\boldsymbol{v}-\boldsymbol{u}(\boldsymbol{x},t)\|^2 d\boldsymbol{v}.\end{aligned} \tag{1.36}$$

1.3.2 Lattice Bhatnagar-Gross-Krook

To obtain a discrete Boltzmann equation, one starts with the discretization of the velocity space. First, we approximate f^{eq} by writing it on a truncated basis of Hermite polynomials [30], noticing that the first Hermite expansion coefficients correspond exactly to the moments Eq. (1.36) of the distribution [31].

To find a lattice stencil, *i.e.* a set of q poles $\{\boldsymbol{c}_\ell\}_{\ell=0}^{q-1}$ and weights $\{t_\ell\}_{\ell=0}^{q-1}$, we rely on Gauss-Hermite quadratures [32] so that we recover the appropriate moments of the distribution, while ensuring that all the weights are non-negative for numerical stability and that the poles belong to the nodes of a Cartesian grid for perfect streaming. As a result, for a given algebraic degree of precision $> 2M$, the knowledge of the discrete distributions $\mathrm{f} = \{f_\ell\}_{\ell=0}^{q-1}$ is enought to recover exactly the first M moments. In this thesis, we will be interested in the simulation of isothermal turbulence, therefore, we need $M > 2$ so that we have

$$\begin{aligned}\rho &= \sum_{\ell=0}^{q-1} f_\ell \text{ and}\\ \rho\boldsymbol{u} &= \sum_{\ell=0}^{q-1} f_\ell \boldsymbol{c}_\ell.\end{aligned} \tag{1.37}$$

The obtained lattice stencil is commonly named in the DdQq format. Usual lattice includes the D1Q3 and its tensorial products, the D2Q9, and the D3Q27. On such a lattice, we introduce the LBM equation discretized in time, which describes the streaming and collision of the discrete distribution functions $f_\ell(\boldsymbol{x},t)$ with a finite set of kinetic velocities $\boldsymbol{c}_\ell$ for $\ell = 0\ldots q-1$. Written for a BGK collision operator with a fixed *dimensionless* relaxation time τ_0, the LBM equation is called hereafter the LBGK equation:

$$f_\ell(\boldsymbol{x}+\boldsymbol{c}_\ell\Delta_t, t+\Delta_t) - f_\ell(\boldsymbol{x},t) = -\frac{1}{\tau_0}\left[f_\ell(\boldsymbol{x},t) - f_\ell^{eq}(\boldsymbol{x},t)\right] + S_\ell, \tag{1.38}$$

where Δ_t is the time step and S_ℓ is a suitable forcing term designed to reproduce a macroscopic external forcing [33, 34, 35].

The discrete equilibrium distribution f_ℓ^{eq} is assumed to have a finite Hermite polynomials expansion. We will use thereafter the third order ($M = 3$) expansion:

$$f_\ell^{\mathrm{eq}} = \rho t_\ell \left(1 + \frac{\boldsymbol{u}\cdot\boldsymbol{c}_\ell}{c_s^2} + \frac{\boldsymbol{uu} : \boldsymbol{c}_\ell\boldsymbol{c}_\ell - c_s^2|\boldsymbol{u}|^2}{2c_s^4} + \frac{\boldsymbol{uuu} :\cdot\, \boldsymbol{c}_\ell\boldsymbol{c}_\ell\boldsymbol{c}_\ell - 3c_s^2|\boldsymbol{u}|^2\boldsymbol{u}\cdot\boldsymbol{c}_\ell}{6c_s^6}\right), \quad (1.39)$$

where c_s the speed of sound in the lattice.

1.3.3 Algorithm

The LBGK algorithm is given in Listing 1. It should be noted that the popularity of LBGK as a flow solver mostly lies in the fact that its algorithm is fully local, and therefore can make use of highly parallel architectures such as General Purpose Graphical Processing Units (GPUs). This is explored in more details in Chapter 6.

Algorithm 1 Commonly used LBGK algorithm.

```
for each time step do
    for each lattice node do
        Calculate density and equilibrium velocity
        Calculate the equilibrium distribution $f^{eq}$
        Calculate the non-equilibrium part of the distribution $f^{neq}$
        Collide with a relaxation time of $\tau_0$
        Propagate
    end for
end for
```

1.3.4 Chapman-Enskog expansion

From a theoretical point of view, the use of a multi-scale Chapman-Enskog (CE) perturbative expansion allows to recover hydrodynamic equations [33, 34]. It is usually unrolled on the basis of a separation of the advective and dissipative time scales. The order of magnitude characterizing this scale separation is given by the Knudsen number $Kn = \frac{\lambda}{L_0} \ll 1$. Indeed, the kinetic system is characterized by a length scale, that is the mean free path λ, defined as the typical distance traveled by an atom between two collisions that is several orders of magnitude smaller than the macroscopic length scale L_0 of the fluid system. In C-E expansion, it is commonly hypothesized that the advective time scale is of $\mathcal{O}(Kn)$, while the dissipative time scale is of $\mathcal{O}(Kn^2)$. Here to describe the scale separation we expand in order of the time step Δ_t. For any order $N \geq 1$ we can write

$$f_\ell(\boldsymbol{x} + \boldsymbol{c}_\ell\Delta_t, t + \Delta_t) = f_\ell(\boldsymbol{x}, t) + \sum_{n=1}^{N} \frac{\Delta_t^n}{n!} D_\ell^n f_\ell(\boldsymbol{x}, t) + \mathcal{O}(\Delta_t^{N+1}), \quad (1.40)$$

where $D_\ell = \partial_t + \boldsymbol{c}_\ell \cdot \boldsymbol{\nabla}$ and Δ_t can be interpretated as the Knudsen number.

We then obtain

$$\sum_{n=1}^{N} \frac{\Delta_t^{n-1}}{n!} D_\ell^n f_\ell^{\mathrm{eq}} = -\frac{1}{\tau_0} f_\ell^{\mathrm{eq}} \phi_\ell - \sum_{n=1}^{N-1} \frac{\Delta_t^n}{n!} D_\ell^n (f_\ell^{\mathrm{eq}} \phi_\ell) + \mathcal{O}(\Delta_t^N), \tag{1.41}$$

where f_ℓ, which is a parametric function of Δ_t, has been decomposed as

$$f_\ell = f_\ell^{\mathrm{eq}} \left(1 + \Delta_t \phi_\ell\right), \tag{1.42}$$

with $\phi_\ell = \mathcal{O}(1)$ for $\Delta_t \to 0$. To recover hydrodynamic equation, we use the fact that we have the following the zeroth order moment relations

$$\rho = \sum_\ell f_\ell = \sum_\ell f_\ell^{\mathrm{eq}}, \qquad \sum_\ell f_\ell^{\mathrm{eq}} \phi_\ell = 0 \tag{1.43}$$

and the first order moment relations

$$\rho \boldsymbol{u} = \sum_\ell \boldsymbol{c}_\ell f_\ell = \sum_\ell \boldsymbol{c}_\ell f_\ell^{\mathrm{eq}}, \qquad \sum_\ell \boldsymbol{c}_\ell f_\ell^{\mathrm{eq}} \phi_\ell = \mathbf{0}. \tag{1.44}$$

Therefore taking the zeroth and first order moments of Eq. (1.41) for an equilibrium distribution chosen as in Eq. (1.39), one can recover the continuity and momentum equations corresponding to the athermal weakly compressible Navier-Stokes equations. The full derivation is provided in Appendix B and we give the outcome in Eq. (1.45).

$$\begin{aligned} \partial_t \rho + \partial_j (\rho u_j) &= 0 + \mathcal{O}(\Delta_t^2) \\ \partial_t \left(\rho u_i\right) + \partial_j \left(\rho u_i u_j\right) &= -\partial_i p + \partial_j \left(\rho \nu \left(\partial_j u_i + \partial_i u_j\right)\right) + F_i + \mathcal{O}(\Delta_t^2) + \mathcal{O}(Ma^3). \end{aligned} \tag{1.45}$$

Beyond the higher order corrections in the Knudsen number (here $\mathcal{O}(\Delta_t^2)$), in the recovery of the momentum equations one usually neglects terms which are cubic in the velocity [36], Hence we find the term $\mathcal{O}(Ma^3)$, with $Ma = \frac{U}{c_S}$ the Mach number, where U is the root mean squared velocity. The term $p = c_s^2 \rho$ is the fluid pressure and the viscosity ν is linearly dependent on the relaxation time τ in (1.46) and vanishes as $\tau \to 0.5$:

$$\nu_0 = c_s^2 \left(\tau_0 - \frac{1}{2}\right) \Delta_t. \tag{1.46}$$

1.3.5 LBM-based LES

An analysis of inertial range statistics was recently conducted for a high-resolution 3D HIT DNS using LBGK in Ref. [37], highlighting that it is possible to obtain a good recovery of both high- and low-order inertial range statistics with a LBM mesoscale flow solver.

However, the LBGK exhibits instabilities as the input relaxation time $\tau_0 \to 0.5$, *i.e.* for an input viscosity $\nu_0 \to 0$. This, along with the low Mach number, which is required to well recover the NSE, drastically limit the range of Reynolds number $Re = \frac{U\,L_0}{\nu_0}$ with L_0 the characteristic length of the flow, reachable at a fixed grid resolution [38].

The LBM community has been keenly proposing Navier-Stokes inspired SGS models to combine the intrinsic scalability of LBM with turbulence modelling. The majority of them are eddy viscosities models implemented by locally modifying the relaxation time τ, i.e. assuming that the bridge equation Eq. (1.46) holds and that an effective relaxation time $\tau_{\text{eff}}(\boldsymbol{x},\, t)$ results in an effective viscosity $\nu_{\text{eff}}(\boldsymbol{x},\, t)$ [39, 40, 41, 42]. This assumption is discussed in more details in Chapter 4.

In LBM simulations the filtering operation is implicit and the filter width is set by the choice of a grid resolution. But it is not obvious a priori that filtering (implicitly or not) the LBM equations is equivalent at the macroscopic scale to approximating filtered N-S equations. However, it was shown by Malaspinas & Sagaut that in the athermal weakly compressible limit [43], this is indeed the case. As we work with implicit filtering in this thesis, to simplify notations, we will drop the overline over the filtered fields.

To illustrate how LBM-LES is usually working, we take the example of the Smagorinsky SGS introduced Eq. (1.32). In that case, in order to obtain $\nu^S_{\text{eff}} = \nu_0 + \delta\nu^S_e$, we calculate $\tau_{\text{eff}} = \tau_0 + \delta\tau^S_e$ by rewriting Eq. (1.46) as

$$\delta\tau^S_e = \frac{\delta\nu^S_e}{c_s^2\Delta_t} \tag{1.47}$$

The eddy viscosity in the Smagorinsky model depends on the filtered strain-rate tensor. In LBM, this quantity can be obtained without calculating any derivative, directly from the non-equilibrium distribution:

$$\mathbf{S} \approx -\frac{1}{2\rho\tau_{\text{eff}}c_s^2\Delta_t}\mathbf{\Pi}^{(1)}, \tag{1.48}$$

with $\mathbf{\Pi}^{(1)} = \sum_\ell \boldsymbol{c}_\ell\boldsymbol{c}_\ell f^{neq}_\ell$ the non-equilibrium momentum tensor.

1.4 Entropic Lattice Boltzmann Method

Stabilization of LBGK has been linked to the existence of an underlying Lyapunov functional in the form of a discrete Boltzmann H-functional [44]. Indeed, it was previously shown that conventional LBGK schemes cannot obey an H-theorem [45]. However, by ensuring the local monotonicity of such a convex function provides in principle a simulated dynamical flow system with non-linear stability. Through the local calculation of an effective relaxation time that enforced a discrete local H-theorem, Karlin *et* *al.* [46] introduced a new sub-class of LBM, the Entropic Lattice Boltzmann Methods (ELBM).

The apparent unconditional stability of ELBM [47] has made it a popular choice for the simulation of low-dissipative flows. Naturally, it has been applied to the simulation of turbulent flows, both homogeneous isotropic [48] and wall-bounded [49]. Apart from the simulation of turbulent flows, the entropy feedback has been used to improve the stability of thermal flows [50, 51, 52], multiphase flows [53] and other fluid flows [54]. For a detailed review of ELBM, the reader is referred to Refs. [55, 56].

1.4.1 Enforcement of a discrete local H-theorem

Using the formalism of Karlin *et al.* [46], the ELBM eq. writes:

$$\begin{aligned} f_\ell(x + c_\ell \Delta_t, t + \Delta_t) &= f_\ell(\boldsymbol{x}, t) - \alpha(\boldsymbol{x}, t)\beta \left(f_\ell(\boldsymbol{x}, t) - f_\ell^{eq}(\boldsymbol{x}, t)\right) \\ &= (1-\beta)\ f_\ell^{pre}(\boldsymbol{x}, t) + \beta\, f_\ell^{mirror}(\boldsymbol{x}, t) \\ &= f_\ell^{post}. \end{aligned} \tag{1.49}$$

To equip a LBGK with a H-theorem, the ELBM collision operator adapts the BGK relaxation of the distribution functions f towards a local equilibrium f^{eq} with the use of an effective relaxation time $\tau_{\text{eff}}(\boldsymbol{x}, t) = \frac{1}{\alpha(\boldsymbol{x},t)\beta}$, with $\beta = \frac{1}{2\tau_0}$ and α a local parameter. As seen in the ELBM Eq. (1.49), the post-collision distribution $\mathrm{f}^{post}(\beta)$ can then be understood as a convex combination between the pre-collision distribution $\mathrm{f}^{pre} = \mathrm{f}$ and the so-called mirror distribution $\mathrm{f}^{mirror}(\alpha) = \mathrm{f}^{pre} - \alpha\, \mathrm{f}^{neq}$ with $\mathrm{f}^{neq} = \mathrm{f} - \mathrm{f}^{eq}$ the non-equilibrium part of f^{pre}. This convex combination is parametrized by the parameter β for which we have $0 < \beta < 1$ as $0.5 < \tau_0 < +\infty$.

For a convex H-functional commonly chosen [46, 57] as

$$H[\mathrm{f}] = \sum_{\ell=0}^{q-1} f_\ell \log\left(\frac{f_\ell}{t_\ell}\right),\ \mathrm{f} = \{f_\ell\}_{\ell=0}^{q-1}. \tag{1.50}$$

the discrete local H-theorem [45] can then be expressed as a the local decrease of the H-functional between the pre-collision and post-collision distributions,

$$\begin{aligned} \Delta H &= H[\mathrm{f}^{post}] - H[\mathrm{f}^{pre}] \\ &= H[(1-\beta)\mathrm{f}^{pre} + \beta \mathrm{f}^{mirror}(\alpha)] - H[\mathrm{f}] \leq 0, \end{aligned} \tag{1.51}$$

where the pre-collision distribution is $\mathrm{f}^{pre} = \mathrm{f}$ and the the post-collision distribution is $\mathrm{f}^{post} = (1-\beta)\mathrm{f}^{pre} + \beta\mathrm{f}^{mirror}(\alpha)$.

The equilibrium distribution $\mathbf{f^{eq}}$ is here defined as the extremum of a H-functionnal under the constraints of mass and momentum conservation [58]. This extremalization has an analytical solution for the D1Q3 lattice and its tensorial products, the D2Q9 and D3Q27,

$$f_\ell^{eq}(\boldsymbol{x},\, t) = t_\ell\, \rho \prod_{\gamma=1}^{d} \left\{ \left(2 - \sqrt{1 + \frac{u_\gamma^2}{c_s^2}}\right) \left[\frac{\frac{2u_\gamma}{\sqrt{3}c_s} + \sqrt{1 + \frac{u_\gamma^2}{c_s^2}}}{1 - \frac{u_\gamma}{\sqrt{3}c_s}} \right]^{\frac{c_{\ell,\,\gamma}}{\sqrt{3}c_s}} \right\}, \tag{1.52}$$

where d is the dimension of the DdQq lattice.

The first three moments of the equilibrium distribution Eq. (1.52) are exactly the same as the one coming from the third order truncated equilibrium distribution Eq. (1.39) [57]. Therefore, it allows the recovery of the same athermal weakly compressible Navier-Stokes given in Eq. (1.45).

Different approaches have been proposed to enforce the discrete local H-theorem Eq. (1.51). Recently, Atif *et al.* [59] expanded the left-hand side of Eq. (1.51) and highlighted the presence of negative-definite terms, providing an exact solution for α by solving a quadratic equation involving the remaining non-negative-definite terms. Bösch *et al.* [60] uncovered a new class of multi-relaxation time ELBM, the KBC family of methods, which minimize the H-value of the post-collision distributions defined for a decomposition of the pre-collision distribution in the kinematic part k, shear part s, and the remaining higher moment parts h: $\mathrm{f}^{pre} = \mathrm{k} + \mathrm{s} + \mathrm{h}$ and applying an entropic relaxation only to the higher moments part.

In this work, we use Karlin *et al.* [46] approach to ELBM, which calculates $\alpha(x, t)$ as the solution of the entropic step equation (1.53). Indeed, as the H function is convex and the post-collision distribution is a convex combination between two distributions $\mathrm{f}^{\mathrm{pre}}$ and $\mathrm{f}^{\mathrm{mirror}}$ of equal H-value, the monotonic decrease of the H-functional Eq. (1.51) value is ensured as illustrated in figure1.3.

$$H[\mathbf{f}] = H[\mathbf{f}^{\mathbf{mirror}}(\alpha)] \tag{1.53}$$

where $\mathbf{f}^{\mathbf{mirror}}(\alpha) = \mathbf{f} - \alpha \mathbf{f}^{\mathbf{neq}}$ with $\mathbf{f}^{\mathbf{neq}} = \mathbf{f} - \mathbf{f}^{\mathbf{eq}}$ the non-equilibrium part of the distribution $\mathbf{f}$.

ELBM, as a result of the enforcement of the H-theorem, is apparently unconditionally stable [47] LBM for $\tau_0 \to 0.5$ ($\nu_0 \to 0$).

1.4.2 Algorithm

The ELBM algorithm adds a single extra step to the LBGK algorithm (Listing 1), the entropic step, which returns the effective relaxation time $\tau_{\text{eff}} = \frac{1}{\alpha\beta}$. The entropic step

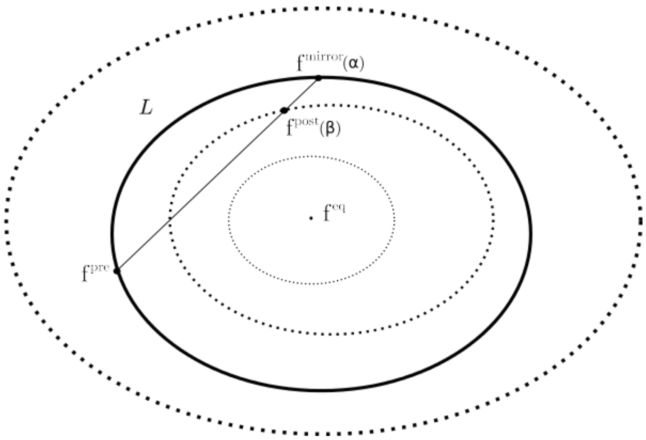


Figure 1.3. ELBM: Perspective from H-functional hypersurface [46]

equation (Eq. (1.53)) is popularly solved using a Newton-Raphson (N-R) algorithm. Typical values for the tolerance and the maximum number of iteration for the N-R solver are respectively 10^{-5} and 20, although in this thesis, to study the implicit closure, we have often used a tolerance of 10^{-8} and a maximum number of iteration of 1000. Without going to the extent of such high values, it should be noted that ELBM is a computationally expensive algorithm. Performing the simulations whose results are documented in this thesis, we have observed that whenever distributions are very close to their local equilibrium counterparts, the number of iterations required to solve the entropic step equation increases greatly. However, in that case, we know that if locally, the simulation is resolved ($\mathrm{f} \to \mathrm{f}^{eq}$), we have $\alpha \to \alpha_{LBGK} = 2$. As a result the ELBM community has been using an optimization that consists in calculating the relative deviation of the distribution to the equilibrium distribution $\Delta\left(\mathrm{f}, \mathrm{f}^{\mathrm{neq}}\right) = \max\limits_{0<\ell<q-1} |\frac{f_\ell^{neq}}{f_\ell}|$ and having the entropic step returning $\alpha = 2$ if this deviation is smaller than a threshold (usually taken equal to 10^{-3}). The ELBM algorithm including this optimization is shown in Listing 2.

1.4.3 Forcing the Entropic Lattice Boltzmann Method

In order to simulate a forced flow, one needs to add a suitable forcing source term S_ℓ to the ELBM Eq. (1.49). Therefore, the entropic step needs to be adapted to ensure that the H-functional decreases between the new pre- and post-collision states. In most of the literature, this adaptation is skipped and the algorithm for the calculation of the entropic parameter α neglects the presence of the forcing term in the LBM equation. Only one occurrence of the formal inclusion of this term, which was found in Ref. [53], suggests how to formally include it.

Indeed, by assuming that the force is applied before the collision takes place, $\mathrm{f}^{pre} = \mathrm{f} + \mathrm{S}$ and rewriting the mirror distribution as $\mathrm{f}^{mirror} = \mathrm{f}^{pre} - \alpha \mathrm{f}^{neq}$, we obtain again that the

Algorithm 2 Commonly used optmized ELBM algorithm.

```
for each time step do
    for each lattice node do
        Calculate density and equilibrium velocity
        Calculate the equilbrium distribution f$^{eq}$
        Calculate the non-equilibrium part of the distribution f$^{neq}$
        Check the deviation $\Delta\left(\mathrm{f},\mathrm{f}^{\mathrm{neq}}\right)$
        if $\Delta\left(\mathrm{f},\mathbf{f}^{\mathbf{neq}}\right) <= 10^{-3}$ then
            Set $\alpha = 2$
        else
            Calculate $\alpha_{max}$ corresponding to $\min\limits_{0<\ell<q-1;f_\ell^{neq}>0} |\frac{f_\ell}{f_\ell^{neq}}|$
            if $\alpha_{max}$ ¡ 2 then
                Set $\alpha = 0.9 \times \alpha_{max}$
            else
                Use Newton-Raphson method to solve $H\left(\mathbf{f}\right) = H\left(\mathbf{f} - \alpha\mathbf{f}^{\mathbf{neq}}\right)$ with $\alpha_{guess} = 2$, $\alpha_{min} = 1$ and previously calculated $\alpha_{max}$
            end if
        end if
        Collide with a relaxation time of $\frac{1}{\alpha\beta}$
        Propagate
    end for
end for
```

post-collision distribution is a convex combination between the pre-collision distribution and the mirror distribution (Eq. (1.54)), thus enabling a forced ELBM algorithm.

$$
\begin{aligned}
f_\ell(x + c_\ell\Delta_t, t + \Delta_t) &= f_\ell(\boldsymbol{x},t) - \alpha(\boldsymbol{x},t)\beta\left(f_\ell(\boldsymbol{x},t) - f_\ell^{eq}(\boldsymbol{x},t)\right) + S_\ell(\boldsymbol{x},t) \\
&= (1-\beta)\ \left(f_\ell(\boldsymbol{x},t) + S_\ell(\boldsymbol{x},t)\right) + \beta\left(f_\ell(\boldsymbol{x},t) + S_\ell(\boldsymbol{x},t) - \alpha f_\ell^{neq}(\boldsymbol{x},t)\right) \\
&= (1-\beta)\ f_\ell^{pre}(\boldsymbol{x},t) + \beta\, f_\ell^{mirror}(\boldsymbol{x},t) = f_\ell^{post}(\boldsymbol{x},t)
\end{aligned}
\tag{1.54}
$$

In our numerical experiments, we have always used small macroscopic forcing amplitudes and therefore, we did not notice that including the forcing source term in the entropic step had an impact on the physics of the flow. However, including it did increase the required number of Newton-Raphson solver.

In all the forced flow simulations documented in this thesis, we enforce the macroscopic forcing $\boldsymbol{F}$ using the exact-difference method forcing scheme [61] for which

$$
S_\ell = f_\ell^{eq}(\rho, \boldsymbol{u} + \frac{\boldsymbol{F}}{\rho}) - f_\ell^{eq}(\rho, \boldsymbol{u}). \tag{1.55}
$$

It is important to note, that not all forcing scheme can be used to force ELBM. Guo's forcing scheme [62] involves a source term that depends on the effective relaxation time τ_{eff} and therefore would require its knowledge before the entropic step is used to calculate it.

1.4.4 Beyond the stabilization, an implicit turbulence model?

As ELBM shows an apparent unconditional stability [47], it is possible to use ito simulate fluid flows with arbitrary small input viscosities. However, it is important to stress that stability does not mean accuracy, and that there is no garantees that the output flows physics is preserved by ELBM. In practice, a number of work have put forward that the ELBM is implicitly enforcing a SGS model of the eddy viscosity type [48, 3, 2]. Indeed, as it involves modifying the relaxation time in space and time and assuming that the bridge equation between viscosity and relaxation times Eq. (1.46) hold for fluctuating quantities, we have

$$\begin{aligned}\nu_{\text{eff}}\left(\alpha\right) =& c_s^2\left(\frac{1}{\alpha\beta}-\frac{1}{2}\right)\Delta_t \\ =& c_s^2\left(\frac{1}{2\beta}-\frac{1}{2}\right)\Delta_t + c_s^2\frac{2-\alpha}{2\alpha\beta}\Delta_t = \nu_0 + \delta\nu_e^M\left(\alpha\right),\end{aligned} \tag{1.56}$$

where the superscript M in $\delta\nu_e^M$, stands for measured, as it can be measured online directly from the α field.

The effective viscosity is larger than the input kinematic viscosity $\nu_0 = \nu_{\text{eff}}\left(2\right)$ if $\alpha > 2$ and smaller if $\alpha < 2$. For $\beta \to 1$ ($i.e. \tau \to 0.5$), the viscosity ν_0 vanishes and the effective viscosity $\nu_{\text{eff}}\left(\alpha\right)$ can become negative, without leading to instabilities. Thus, it seems to allow local backscatter of energy as a negative effective viscosity brings energy to the resolved scales. Moreover, it has been observed that if the simulation is resolved (*i. e.* $\mathbf{f} \to \mathbf{f^{eq}}$), the local parameter α becomes homogeneously equal to 2 [3] and the ELBM collision operator turns into a standard LBGK collision operator $\alpha \equiv \alpha_{LBGK} \equiv 2$.

As α has a complex non-linear dependencies on the distributions f_ℓ, the effective viscosity cannot *a priori* be understood in terms of macroscopic quantities. Therefore, ELBM physical meaning remains hidden. Unrolling a C-E perturbative expansion, it is possible to obtain a macroscopic approximation of the eddy viscosity Eq. (1.56). Initially due to Malaspinas *et al.* [2], we have corrected and clarified the range of validity of the derivation in Chapter 4. The resulting eddy viscosity approximation written reads

$$\delta\nu_e^A \approx -c_s^2\Delta t^2\frac{1}{6\beta^2}\frac{S_{\lambda\mu}S_{\mu\gamma}S_{\gamma\lambda}}{S_{\gamma\delta}S_{\gamma\delta}}, \tag{1.57}$$

where $S_{ij} = \frac{1}{2}(\partial_i u_j + \partial_j u_i)$ is the strain-rate tensor. This eddy viscosity possesses an interesting functional shape. As the Smagorinsky model [4], it scales as the strain-rate tensor, but is not positive-definite. This means, that it allows backscatter of energy from the unresolved scales to the resolved scales. Indeed, while energy transfer should in average be towards the small scales to model properly the small-scale dissipation, a realistic SGS should also intermittently transfer energy in the other direction.

The ELBM approach has been validated for a number of turbulent flows in terms of mean flow properties. [60, 63, 64]. However, little has been done to numerically study the implicit SGS model implied by ELBM and the validity range of its macroscopic formulation. Therefore, it is still unclear whether it acts as a mere stabilizer or as an accurate representation of the unresolved physics of turbulence stemming from kinetic theory.

CHAPTER 2

A NUMERICAL TOOL FOR THE ASSESSMENT OF THE HYDRODYNAMIC RECOVERY OF LBM SIMULATIONS

In this chapter we build a tool to take control of the hydrodynamic recovery and determine to which accuracy LBGK is able to recover the Navier-Stokes equations as a function of the analyzing sub-volume size. This chapter summarizes the results of Ref. [6] and is organized as follows: in section 2.1 we introduce the balance equations, their averaged counterparts over a sub-volume and we define balancing errors as a measure of the hydrodynamic recovery; in section 2.2 we present the numerical set-up for the simulations of 2D isotropic homogeneous turbulence and for the statistical analysis of the balancing errors; in section 2.3 we present a validation of the tool by comparing the hydrodynamic recovery of an ensemble of LBGK simulations to an ensemble of Pseudo-Spectral (PS) simulations in the case of decaying flows; in section 2.4 we benchmark the tool on LBGK simulations of forced turbulence for a range of increasing Reynolds numbers, while linking the results to the corresponding statistics of the Mach number; some concluding remarks follow in section 2.5.

2.1 Hydrodynamic recovery for energy and enstrophy balance in 2D

The hydrodynamic recovery of a simulation is studied from the perspective of kinetic energy and enstrophy balance equations, averaged over sub-volumes of the computational domain. The kinetic energy ($E = \frac{\rho u_i u_i}{2}$) and the enstrophy ($\Omega = \frac{\omega_i \omega_i}{2}$, with ω_i the component of the vorticity $\vec{\omega} = \vec{\nabla} \times \vec{u}$ along $\vec{e}_i$) balance equations are obtained from the macroscopic mass and momentum conservation equations recovered by LBM (see

Eq. (1.45)). The details of their derivation is given in appendix A and holds to

$$\begin{aligned}\partial_t\left(\frac{\rho u_i u_i}{2}\right) =& - u_i\partial_i p - \nu\rho\left(\partial_j u_i + \partial_i u_j\right)\partial_j u_i + u_i F_i \\ & - \partial_j\left(\frac{\rho u_i u_i}{2}u_j\right) + \partial_j\left(\nu\rho u_i\left(\partial_j u_i + \partial_i u_j\right)\right)\end{aligned} \tag{2.1}$$

$$\begin{aligned}\partial_t\left(\frac{\omega_i\omega_i}{2}\right) =& - \partial_j\left(\frac{\omega_i\omega_i}{2}u_j\right) + \omega_i\omega_j\partial_j u_i + H_i(\nu)\epsilon_{ijk}\partial_j\omega_k + \omega_i\epsilon_{ijk}\partial_j\left(\frac{1}{\rho}F_k\right) \\ & - \partial_j\left(\frac{\omega_i\omega_i}{2}u_j\right) + \partial_j\left(\epsilon_{ijk}\omega_i H_k(\nu)\right),\end{aligned} \tag{2.2}$$

where ϵ is the Levi-Civita symbol and $H_i(\nu) = \frac{1}{\rho}\partial_j\nu\rho\ (\partial_i u_j + \partial_j u_i)$. From the local equations (2.1) and (A.8) , we calculate their averaged counterparts over a sub-volume V

$$\begin{aligned}LHS_V^E =& \partial_t\langle\frac{\rho u_i u_i}{2}\rangle_V \\ =& - \langle\partial_j\left(\frac{\rho u_i u_i}{2}u_j\right)\rangle_V - \langle u_i\partial_i p\rangle_V + \langle u_i F_i\rangle_V \\ & - \langle\nu\rho\left(\partial_j u_i + \partial_i u_j\right)\partial_j u_i\rangle_V + \langle\partial_j\left(\nu\rho u_i\left(\partial_j u_i + \partial_i u_j\right)\right)\rangle_V \\ =& RHS_V^{E,1} + RHS_V^{E,2} + RHS_V^{E,3} + RHS_V^{E,4} + RHS_V^{E,5} \\ =& RHS_V^E\end{aligned} \tag{2.3}$$

$$\begin{aligned}LHS_V^\Omega =& \partial_t\langle\frac{\omega_i\omega_i}{2}\rangle_V \\ =& - \langle\partial_j\left(\frac{\omega_i\omega_i}{2}u_j\right)\rangle_V - \langle\frac{\omega_i\omega_i}{2}\partial_j u_j\rangle_V + \langle\omega_i\epsilon_{ijk}\partial_j\left(\frac{1}{\rho}F_k\right)\rangle_V \\ & + \langle H_i(\nu)\epsilon_{ijk}\partial_j\omega_k\rangle_V + \langle\partial_j\left(\epsilon_{ijk}\omega_i H_k(\nu)\right)\rangle_V + \langle\omega_i\omega_j\partial_j u_i\rangle_V \\ =& RHS_V^{\Omega,1} + RHS_V^{\Omega,2} + RHS_V^{\Omega,3} + RHS_V^{\Omega,4} + RHS_V^{\Omega,5} + RHS_V^{\Omega,6} \\ =& RHS_V^\Omega\end{aligned} \tag{2.4}$$

where $\langle\cdots\rangle_V$ stands for an average over an arbitrary sub-volume V. Each term of the right-hand side ($RHS_V^{E,\Omega}$) of equations (3.2) and (2.4) accounts for a contribution to the evolution of the averaged energy and enstrophy on the left-hand-side ($LHS_V^{E,\Omega}$). Equations (3.2) and (2.4) hold if the viscosity is allowed to flucte in space and time $\nu = \nu_{\text{eff}}(\vec{x}, t) = \nu_0 + \nu_{\text{t}}(\vec{x}, t)$. We note that, in 2D, the vortex streching term in the enstrophy balance is null [65].

The quality of the hydrodynamic recovery over a sub-volume $V$is defined through balancing errors for the kinetic energy and enstrophy balance, δ_V^E and δ_V^Ω. For a fixed time t, $\delta_V^{E,\Omega}(t)$ is calculated as the ratio of the absolute difference between the $RHS_V^{E,\Omega}(t)$ and the $LHS_V^{E,\Omega}(t)$ terms by the term of the right-hand side with the maximum absolute value *i.e.*

$$\delta_V^E(t) = \frac{\left|RHS_V^E(t) - LHS_V^E(t)\right|}{\max_i \left|RHS_V^{E,\,i}(t)\right|} \tag{2.5}$$

and

$$\delta_V^\Omega(t) = \frac{\left|RHS_V^\Omega(t) - LHS_V^\Omega(t)\right|}{\max_i \left|RHS_V^{\Omega,\,i}(t)\right|}. \tag{2.6}$$

Naturaly, a perfect matching of the averaged balance equations at a time t on a sub-volume V would lead to $\delta_V^E(t) \equiv \delta_V^\Omega(t) \equiv 0$.

2.2 Numerical set-up for the statistical analysis of 2D homogeneous isotropic turbulence hydrodynamics

We first apply the tool described in the previous section to simulations' output of a periodic two-dimensional 256 × 256 system. A homogeneous isotropic forcing with a constant phase ϕ on a shell of (dimensionless) wavenumbers $\vec{k}$ of magnitude from 5 to 7 is set in a stream-function formulation in order to preserve incompressibility in the system,

$$F_\Psi^T(\vec{x}) = F_0^T \sum_{5 \leq \|\vec{k}\| \leq 7} \cos\left(\frac{2\,\pi}{256}\vec{k}\cdot\vec{x} + \phi\right). \tag{2.7}$$

The force is then easily derived as

$$F_x^T = \partial_y F_\Psi^T \qquad \text{and} \qquad F_y^T = -\partial_x F_\Psi^T. \tag{2.8}$$

From this forcing, a time scale $T_f = \sqrt{\frac{2\pi}{k_f F_0^T}}$ can be defined with k_f taken equal to six. To maintain the Mach number under control by limitting the backward energy cascade, we apply a large-scale energy damping as a forcing

$$\vec{F}^R\,(\vec{x},\,t) = -F_0^R \sum_{1 \leq \|\vec{k}\| \leq 2} \vec{\hat{u}}(\vec{k},\,t)\,e^{\frac{2\,\pi}{256}\vec{k}\cdot\vec{x}}, \tag{2.9}$$

where $\vec{\hat{u}}(\vec{k},\,t)$ is the Fourier transform of $\vec{u}(\vec{x},\,t)$. The amplitudes of the forcings are taken equal to $F_0^T = 0.0008$ and $F_0^R = 0.00001$. The D2Q9 [33, 34, 35] is used for all LBGK simulations and the forcings are applied using the exact-difference method forcing scheme [61]. All terms of the averaged balance equations are calculated offline based on dumped configuration fields. Derivatives in those terms are obtained by using a 2$^{\text{nd}}$ order explicit Euler scheme for time derivatives and a 8$^{\text{th}}$ order centered scheme for the space derivatives, i.e.

$$\left.\frac{\partial \mathbf{A}}{\partial t}\right|_{i,j}^{n} \sim \frac{3A_{i,j}^n - 4A_{i,j}^{n-1} + A_{i,j}^{n-2}}{2\,\Delta t}, \text{ and} \tag{2.10}$$

$$
\left.\frac{\partial \mathbf{A}}{\partial x}\right|_{i,j}^{n} \sim \frac{-\frac{1}{56}A^n_{i+4,j} + \frac{4}{21}A^n_{i+3,j} - A^n_{i+2,j} + 4A^n_{i+1,j} - 4A^n_{i-1,j} + A^n_{i-2,j} - \frac{4}{21}A^n_{i-3,j} + \frac{1}{56}A^n_{i-4,j}}{5\,\Delta x}.
$$
$$
\left.\frac{\partial \mathbf{A}}{\partial y}\right|_{i,j}^{n} \sim \frac{-\frac{1}{56}A^n_{i,j+4} + \frac{4}{21}A^n_{i,j+3} - A^n_{i,j+2} + 4A^n_{i,j+1} - 4A^n_{i,j-1} + A^n_{i,j-2} - \frac{4}{21}A^n_{i,j-3} + \frac{1}{56}A^n_{i,j-4}}{5\,\Delta y}.
\tag{2.11}
$$

To illustrate the balancing of the terms of the energy and enstrophy equations, we plot on Figs. 2.1 and 2.2, the corresponding evolution in time of the left-hand side ($LHS_V^{E,\Omega}$) and the right-hand side ($RHS_V^{E,\Omega}$). As observed, the $RHS_V^{E,\Omega}$ terms is constructed from terms of much higher amplitudes. The balancing errors $\delta_V^{E,\Omega}$ are of the order of magnitude of $\mathcal{O}(10^{-3})$ thus highlighting an excellent hydrodynamic recovery.

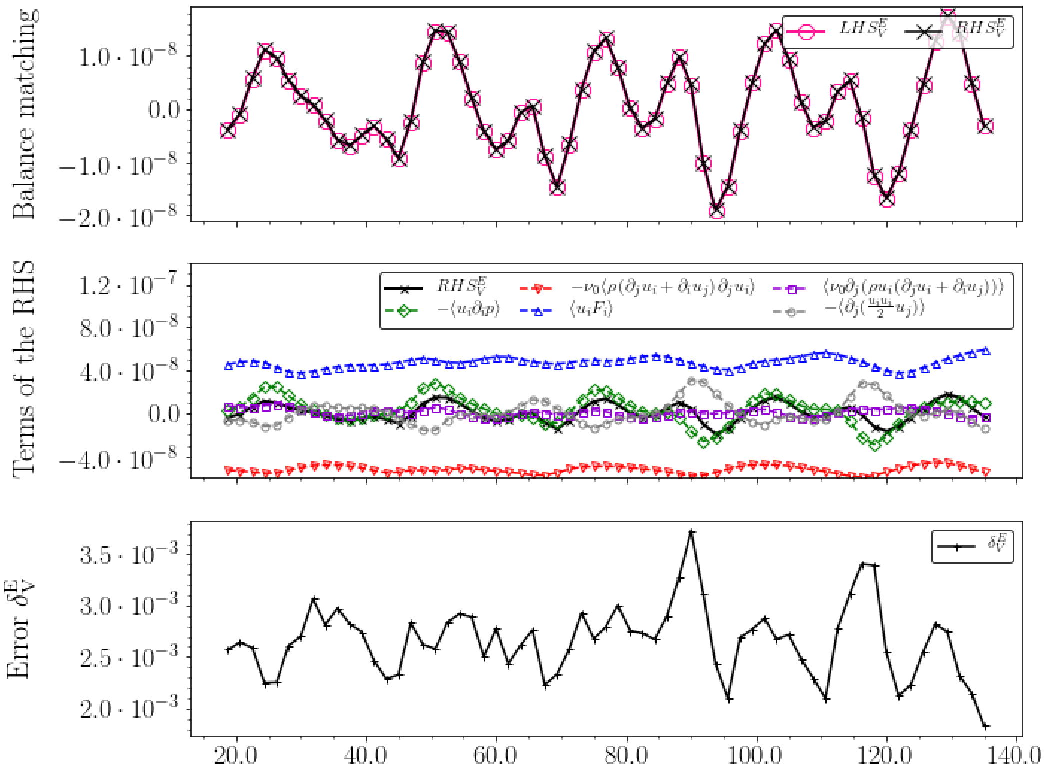


FIGURE 2.1. Typical time-evolution of the kinetic energy balancing over a single sub-volume of size 181 × 181 shown for a forced LBGK simulation with $\tau_0 = 0.60$ ($Re \approx 90$) on a 256×256 grid. The top figure shows the matching between the LHS_V^E and the RHS_V^E, the middle figure shows the contribution of each $RHS_V^{E,i}$ term and their sum RHS_V^E, and the bottom figure shows the balancing error δ_V^E. Taken from Ref [6] with permissions.

To conduct a statistical analysis for at a characteristic sub-volume size L at a time t, we calculate the balancing errors $\delta_V^{E,\Omega}(t)$ on randomly selected squared sub-volumes $V = L \times L$ as illustrated in Fig. 2.3.

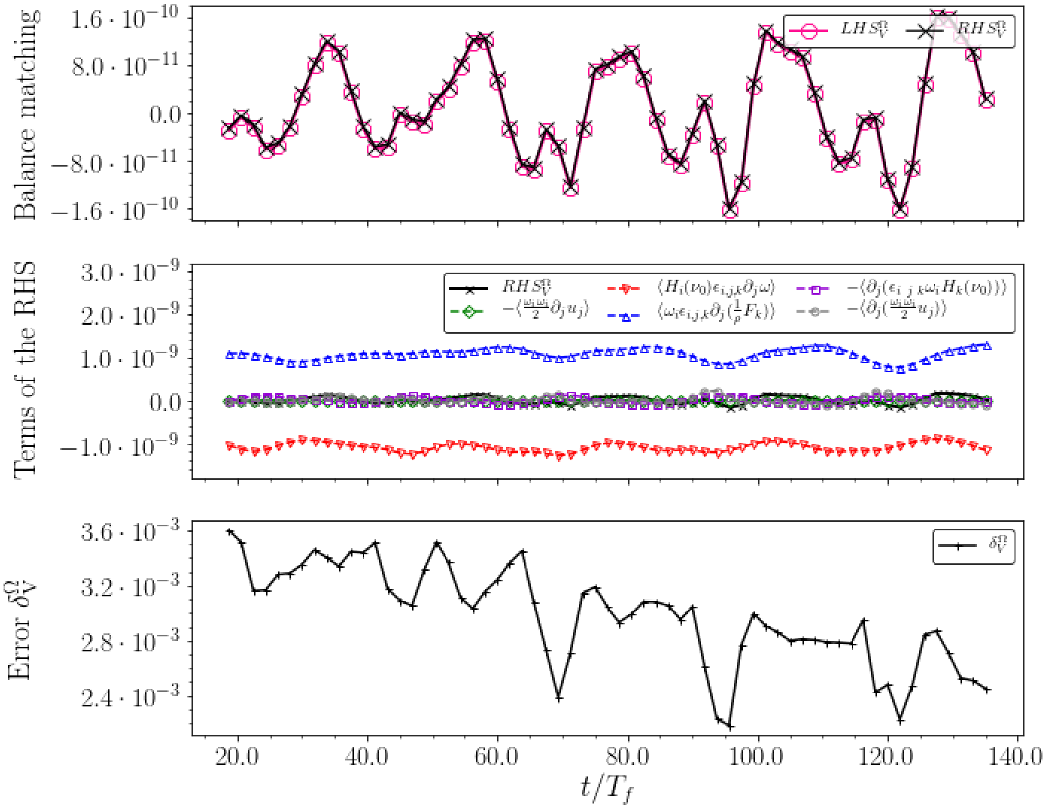


FIGURE 2.2. Typical time-evolution of the enstrophy balancing over a single sub-volume of size 181 × 181 shown for a forced LBGK simulation with $\tau_0 = 0.60$ ($Re \approx 90$) on a 256 × 256 grid. The top figure shows the matching between the LHS_V^Ω and the RHS_V^Ω, the middle figure shows the contribution of each $RHS_V^{\Omega,i}$ term and their sum RHS_V^Ω, and the bottom figure shows the balancing error δ_V^Ω. Taken from Ref [6] with permissions.

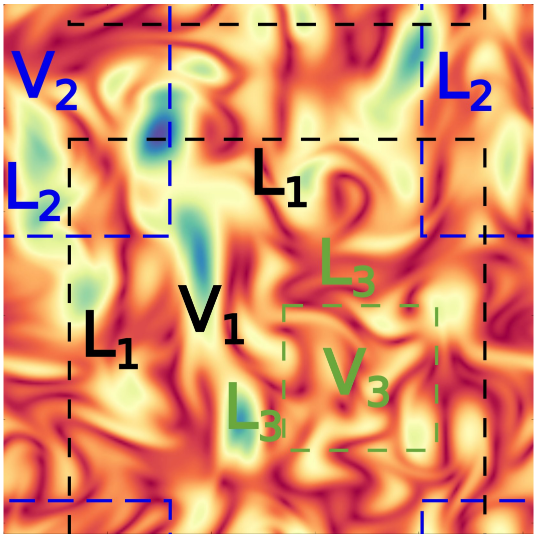


Figure 2.3. Illustration on a snapshot of the vorticity field of three random squared sub-volumes $V_1 = L_1 \times L_1$, $V_2 = L_2 \times L_2$, and $V_3 = L_3 \times L_3$ corresponding to the sub-volume size L_1, L_2, and L_3 respectively. Taken from Ref [6] with permissions.

We define a normalized sub-volume size $l = \frac{L}{L_0}$ with $L_0 = 256$ the of the computational grid, and we define the mean $\mu_l^{E,\Omega}(t)$ and the standard deviation $\sigma_l^{E,\Omega}(t)$ of the balancing errors $\delta_l^{E,\Omega}(t) = \delta_{V=L\times L}^{E,\Omega}(t)$ obtained for all sub-volumes of the same normalized sub-volume size l on a configuration at time t. For each normalized sub-volume size l, the number of sub-volumes selected is shown in Table 2.1.

Table 2.1. Number of sub-volumes processed per sub-volume size L.

Sub-volume size L	Normalized sub-volume size l	Number of sub-volumes
$L = 256$	$l = 1$	1
$100 \leq L < 256$	$0.4 \leq l < 1$	1000
$10 \leq L < 100$	$0.04 \leq l < 0.4$	5000
$L < 10$	$l < 0.04$	10000

2.3 Validation: LBGK against Pseudo-Spectral on an ensemble of decaying flow simulations

We first compare the hydrodynamics recovery of LBGK simulations with the one of PS simulations, taken as a reference for its renowned accuracy.

We study an ensemble of LBGK simulation against an ensemble of PS simulation: we perform a simulation of a statistically stationary turbulent flow at $Re \approx 1200$ ($\tau_0 = 0.52$) using LBGK and we take 25 configurations from it (Fig. 2.4) that we use as the initial configuration of both a LBGK simulation and a PS simulation. The vector potential $\vec{b}$ used to restart the PS simulations is obtained by inverting $\vec{u} = \vec{\nabla} \times \vec{b}$ and rescaled to ensure the same Reynolds number. More accurately, we set

$$Re = \frac{U^{LBGK} L^{LBGK}}{\nu_0^{LBGK}} = \frac{U^{PS} L^{PS}}{\nu_0^{PS}} \tag{2.12}$$

with

$$\begin{aligned} U^{PS} &= U^{LBGK} \frac{\Delta x^{LBGK}}{\Delta t^{LBGK}}, \\ L^{PS} &= 2\pi = L^{LBGK} \Delta x^{LBGK}, \\ \nu_0^{PS} &= \nu_0^{LBGK} \frac{(\Delta x^{LBGK})^2}{\Delta t^{LBGK}} \end{aligned} \tag{2.13}$$

and where $\nu_0^{LBGK} = c_s^2(\tau_0 - 0.5)$ with $\tau_0 = 0.52$. As $\Delta x^{LBGK} = \frac{2\pi}{256}$, $\tau_0 = 0.52$, and $\Delta t^{LBGK} = 0.001$ are fixed, we can get $\nu_0^{PS} \approx 0.004$. We fix $\Delta t^{PS} = 0.0005$ so that configurations of both PS and LBGK simulations are dumped at the same physical time ($\Delta t^{LBGK} \propto \Delta t^{PS}$)and the stability of the PS simulations is ensured. As a result, the initial LBGK simulation velocity field are rescaled by a factor $\frac{\Delta x^{LBGK}}{\Delta t^{LBGK}}$ before to restart the PS simulations. All simulations of both ensembles are then left to decay for a duration of $450\,T_f$. We show the resulting superposed ensemble-averaged energy spectrum of both the LBGK and PS ensembles at three chosen times $t_1 = 0$, $t_2 = 225T_f$, and $t_3 = 450T_f$in Fig. 2.5. Note that the pressure field of the PS simulations is calculated by solving the Poisson equation.

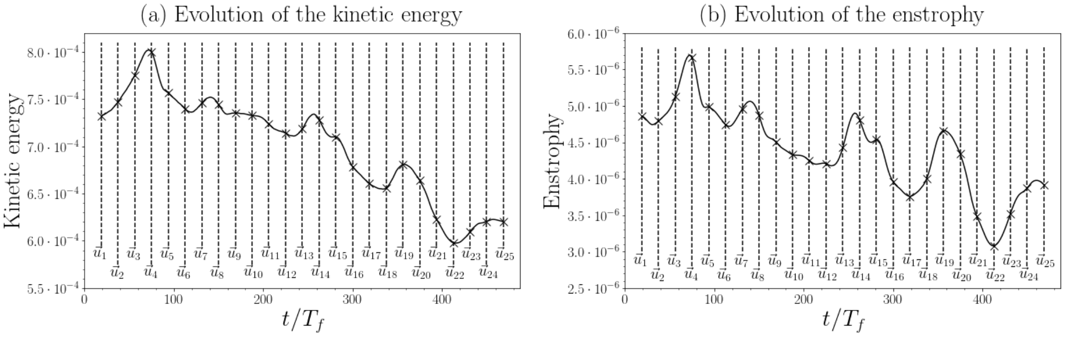


FIGURE 2.4. Evolution of the kinetic energy (a) and of the enstrophy (b) of the forced LBGK simulation. The 25 vertical lines highlight the sampled configurations used to initialize the 25 decaying flow simulations of the PS and the LBGK ensembles. Taken from Ref [6] with permissions.

The outcome of the statistical analysis of the kinetic energy balancing error δ_l^E and enstrophy balancing error δ_l^Ω is presented in Figs. 2.6 and 2.7. The reference PS ensemble's hydrodynamics recovery accuracy is significantly higher than the LBGK's. Besides, it

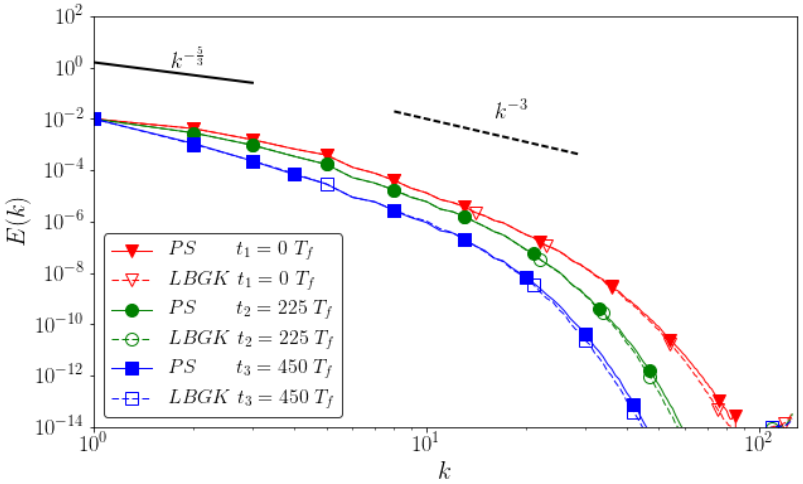


FIGURE 2.5. Superposed ensemble-averaged energy spectrum shown for three selected time instances for the PS and the LBGK simulations. The Kraichnan-Leith-Batchelor predictions for the backward energy cascade slope of $k^{-5/3}$ and forward enstrophy cascade slope of k^{-3} are given for reference. Taken from Ref [6] with permissions.

shows an improvement with time – and thus with increasingly lower Reynolds number – which cannot be observed from the results stemming from the LBGK ensemble in terms of both the energy balancing error statistics μ_l^E and σ_l^E (Fig. 2.6, Panels (c)-(d)) and the the enstrophy balancing error statistics μ_l^Ω and σ_l^Ω (Fig. 2.7, Panels (c)-(d)). All in all, both balancing errors highlight a quality hydrodynamic recovery, espescially on large sub-volumes with an accuracy two orders of magnitude higher than on small sub-volumes (see Figs. 2.6 and 2.7, Panels (a)-(b)), on which the errors still remain of order $\mathcal{O}(10^{-1})$.

In view of the previous result and on the dependence of the theoretical hydrodynamics recovery on the smallness of the Mach number, we plot, for the LBGK ensemble, a similar statistical analysis of its value on a normalized sub-volume size l, that is

$$Ma_l = \langle \frac{U}{c_s} \rangle_{V=L\times L},\ l = \frac{L}{L_0} \tag{2.14}$$

on Fig. 2.8. Its mean is observed to be independent of the sub-volume size (Fig. 2.8-(c)) with values around 0.55 to 0.4 and its standard deviation (Fig. 2.8-(d)) as well up to $L \approx 20$. As the flow decays, so does its Mach number number. This analysis provides an evaluation of the importance of the neglected $\mathcal{O}(Ma^3)$ in the momentum equation (see Eq. (1.45)). Looking at the statistics of the balancing errors in Figs. 2.6 and 2.7, we see that the value of our simulation's Mach number, $Ma \approx 0.05$ is low enough not to affect

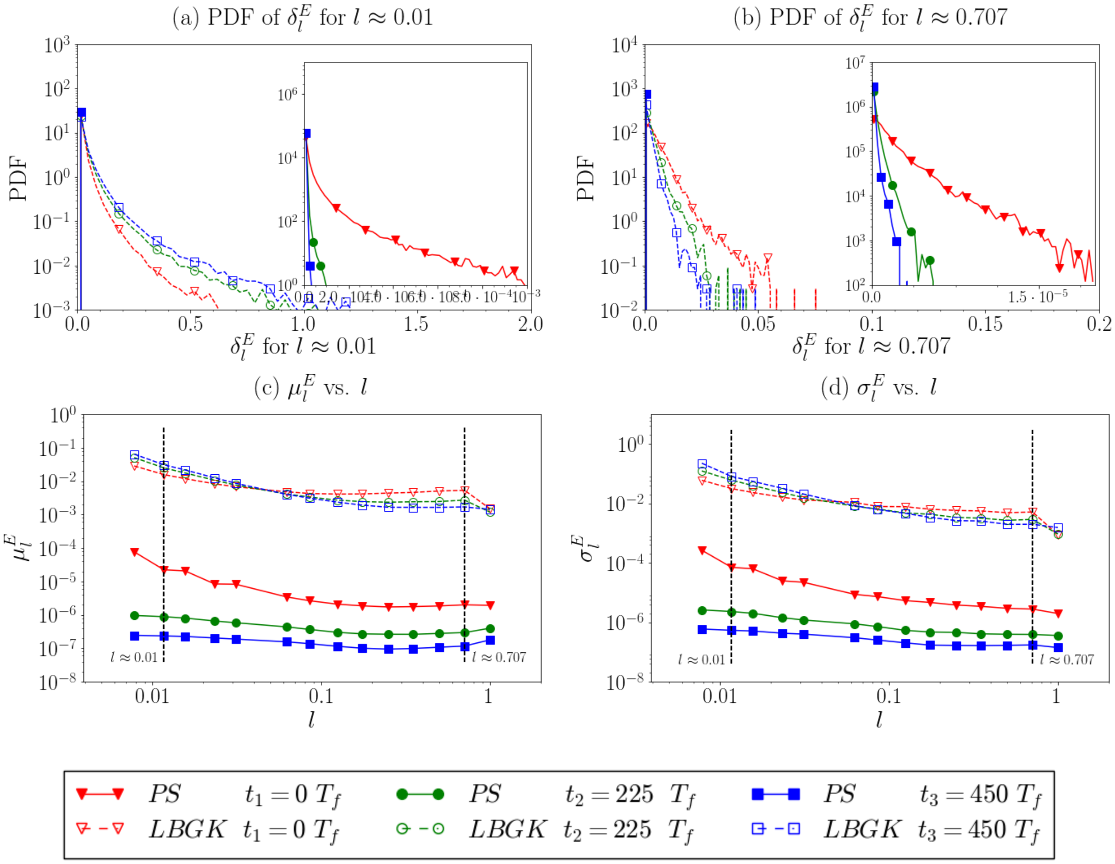


FIGURE 2.6. Statistics of the balancing error obtained from the kinetic energy balance δ_l^E (see Eq. (2.5)) against the normalized size of the sub-volume l shown for the PS and LBGK ensemble of 25 decaying simulations for three selected times. Top figures are PDF of the balancing error for sub-volumes corresponding to $l \approx 0.01$ (Panel (a)) and $l \approx 0.707$ (Panel (b)) and insets shows the PDFs of the balancing error for the PS ensemble alone. Bottom figures are the mean (Panel (c)) and the standard deviation (Panel (d)) of the balancing error. Taken from Ref [6] with permissions.

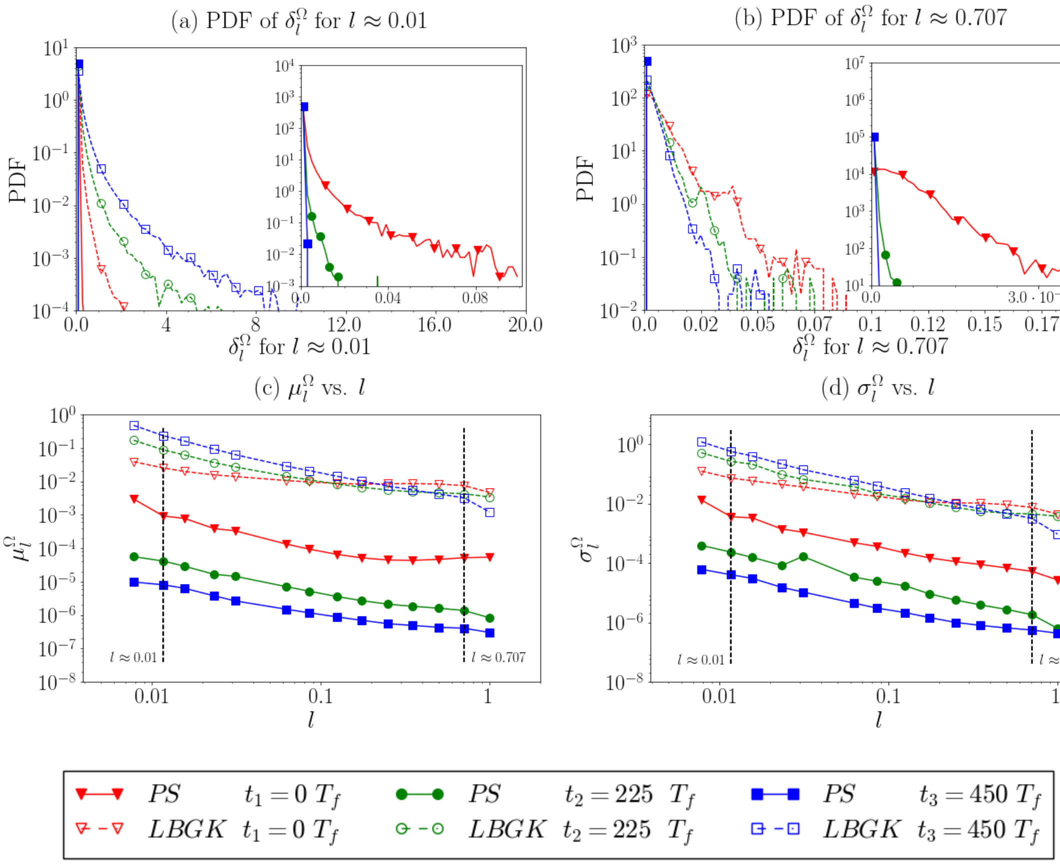


FIGURE 2.7. Statistics of the balancing error obtained from the enstrophy balance δ_l^Ω (see Eq. (2.6)) against the normalized size of the sub-volume l shown for the PS and LBGK ensemble of 25 decaying simulations for three selected times. Top figures are PDFs of the balancing error for sub-volumes corresponding to $l \approx 0.01$ (Panel (a)) and $l \approx 0.707$ (Panel (b)) and insets shows the PDFs of the balancing error for the PS ensemble alone. Bottom figures are the mean (Panel (c)) and the standard deviation (Panel (d)) of the balancing error. Taken from Ref [6] with permissions.

the accuracy of the hydrodynamic recovery as statistics do not vary in time as the Mach number decays and we therefore recover a reliable Navier-Stokes solver.

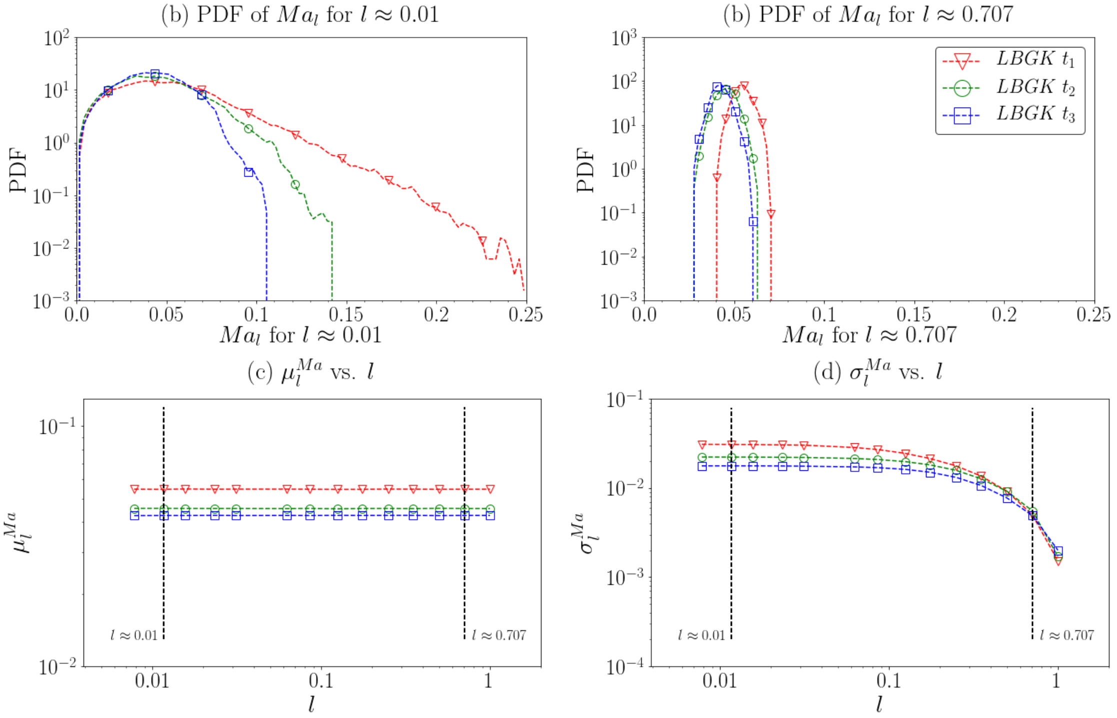


FIGURE 2.8. Statistics of the Mach number at normalized sub-volume size l (see Eq. (2.14)) Ma_l against the normalized size of the sub-volume l shown for the LBGK ensemble of 25 decaying simulations for three selected times. Top figures are PDFs of Ma_l for sub-volumes corresponding to $l \approx 0.01$ (Panel (a)) and $l \approx 0.707$ (Panel (b)). Bottom figures are the mean (Panel (c)) and the standard deviation (Panel (d)) of Ma_l. Taken from Ref [6] with permissions.

2.4 Forced LBGK hydrodynamics

Using the same computational set-up and the same forcing than in the previous section, we benchmark forced LBGK turbulent dynamics at changing Reynolds numbers. We process dumped configurations of simulations at five different Reynolds numbers $Re \approx$ 90, 390, 640, 1200 and 1800 (corresponding to relaxation times $\tau_0 = 0.60$, 0.54, 0.53, 0.52 and $\tau_0^{last} = 0.515$, beyond which LBGK is no longer stable) in the statistically stationnary regime. The energy spectrum of the simulations of the resulting simulations is plotted Fig. 2.10. The effect of the inverse cascade is successfully reduced thanks to the large-scale energy damping forcing. Moreover, as we decrease τ_0 (*i.e.* as we increase Re) the inertial-range of scales corresponding to the forward enstrophy cascade extends ang

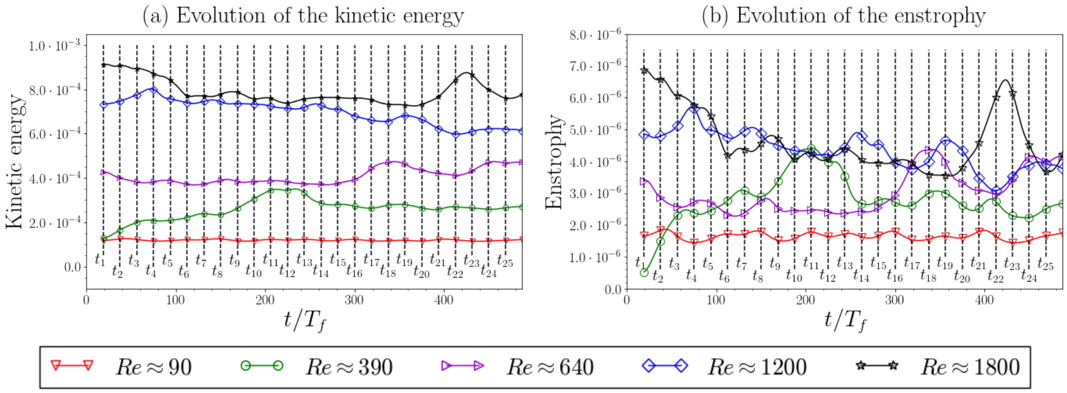


FIGURE 2.9. Evolution of the kinetic energy (a) and of the enstrophy (b) of LBGK simulations for five different relaxation times. The 25 vertical lines highlight the time when configurations were processed to gather statistics in space and time of the balancing errors. Taken from Ref [6] with permissions.

its corresponding slope approches the Kolmogorov-predict value of -3 [66, 14]. In order to evaluate the hydrodynamic recovery of those simulations, we calculate statistics of the balancing errors sampling sub-volume in space and also in time on 25 configurations (see Fig. 2.9).

Figs. 2.11 and 2.12 shows the results of the statistical analysis of the balancing errors δ_l^E and δ_l^Ω respectively. The forced LBGK results are very similar to the LBGK validation results with the accuracy of the hydrodynamic recovery depending significantly on the sub-volume size. On large sub-volumes, the balancing errors are of an order of magnitude of up to $\mathcal{O}(10^{-3})$ lower than on small sub-volumes, where it is of orders of magnitude $\mathcal{O}(10^{-1})$ (see dashed lines in Figs. 2.11 and 2.12, Panels (c)-(d)). On Fig. 2.11, we note a slight dependence of the energy balancing error on the Reynolds number. However, on Fig. 2.12, we see that the enstrophy balancing error improves as the Reynolds number decreases, as one could expect given that enstrophy is heavily impacted by the small-scale resolution.

As shown in Fig. 2.13, the Mach number of the forced LBGK changes with the Reynolds number. This is due to the fact that we forced all simulation with the same fixed forcinf amplitudes. Again, statistics of the Mach number at sub-volume size l, Ma_l (Eq. 2.14) do not affect the quality of the hydrodynamic recovery. Indeed, we are working with Mach numbers that have the same qualitative and quantitative statistics than the ones observed in the previous section (see Fig. 2.8).

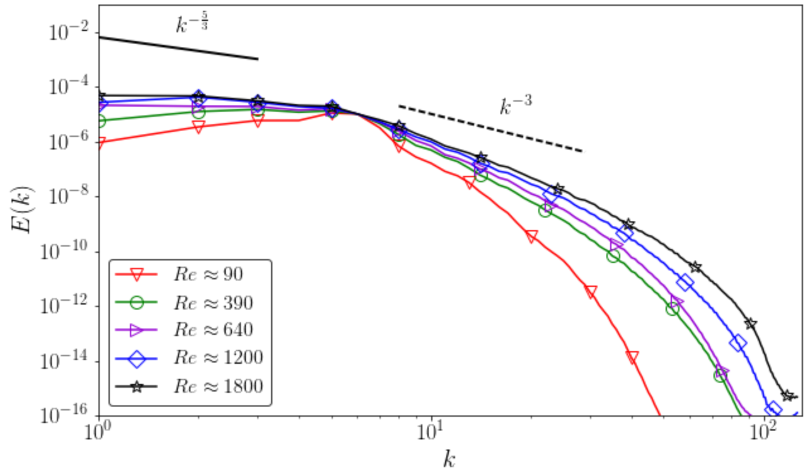


FIGURE 2.10. Superposed time-averaged spectrum of LBGK simulations for five different relaxation times. The Kraichnan-Leith-Batchelor predictions for the backward energy cascade slope of $k^{-5/3}$ and forward enstrophy cascade slope of k^{-3} are given for reference. Taken from Ref [6] with permissions.

2.5 Concluding remarks

In this work, we have developped a generic tool to assess the hydrodynamics of a fluid flow generated through numerical simulations. It relies on calculating averages of each term of the kinetic energy and enstrophy balance equations averaged on a randomly chosen sub-volumes of the computational domain in a systematic manner. To quantify the accuracy of the numerical simulation, we have defined balancing errors and performed a statistical analysis over squared sub-volumes of varying size.

We first validated this approach on ensembles of decaying 2D turbulence flows by and compared a D2Q9 LBGK to a reference PS code. The quality of the hyrodynamics recovered by the PS simulations was shown to be two to six orders of magnitudes higher than LBGK, which still recovered Navier-Stokes dynamics with high accuracy. In all cases, larger sub-volume size correlates with higher accuracy. We then applied the developed tool to benchmark LBGK hydrodynamics for forced turbulence at changing Reynolds number. In that case as well, statistics of the balancing errors were very close to the validation results. The forcing scheme additional approximate error was therefore not observed.

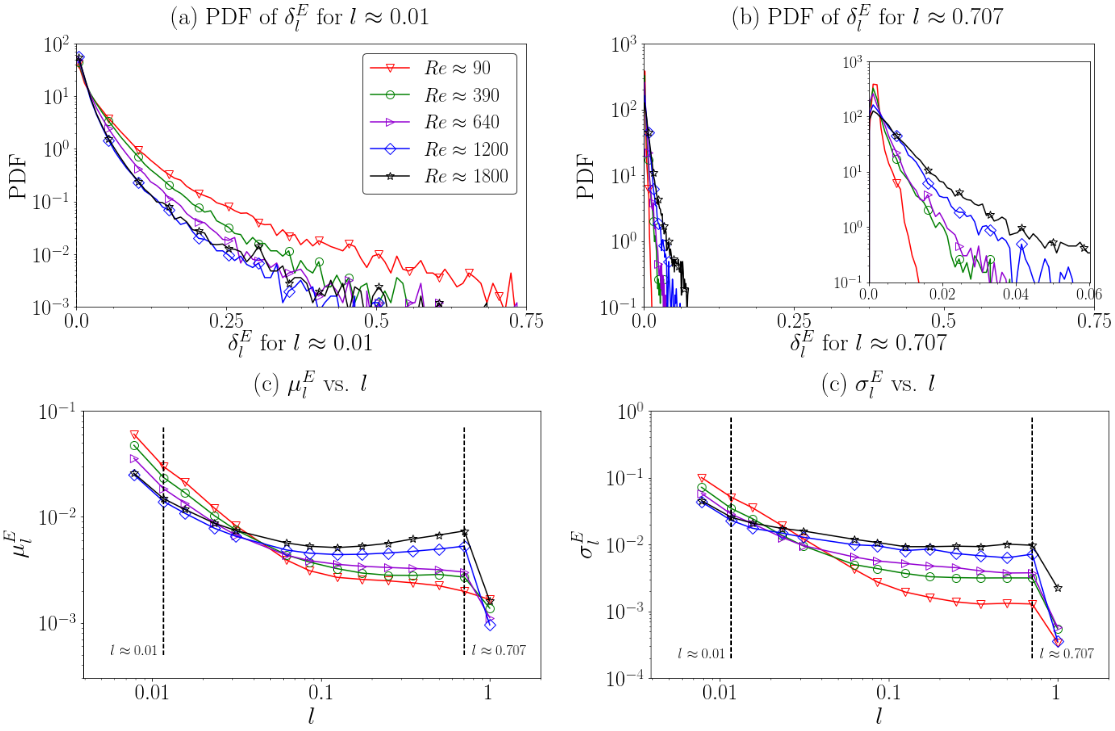
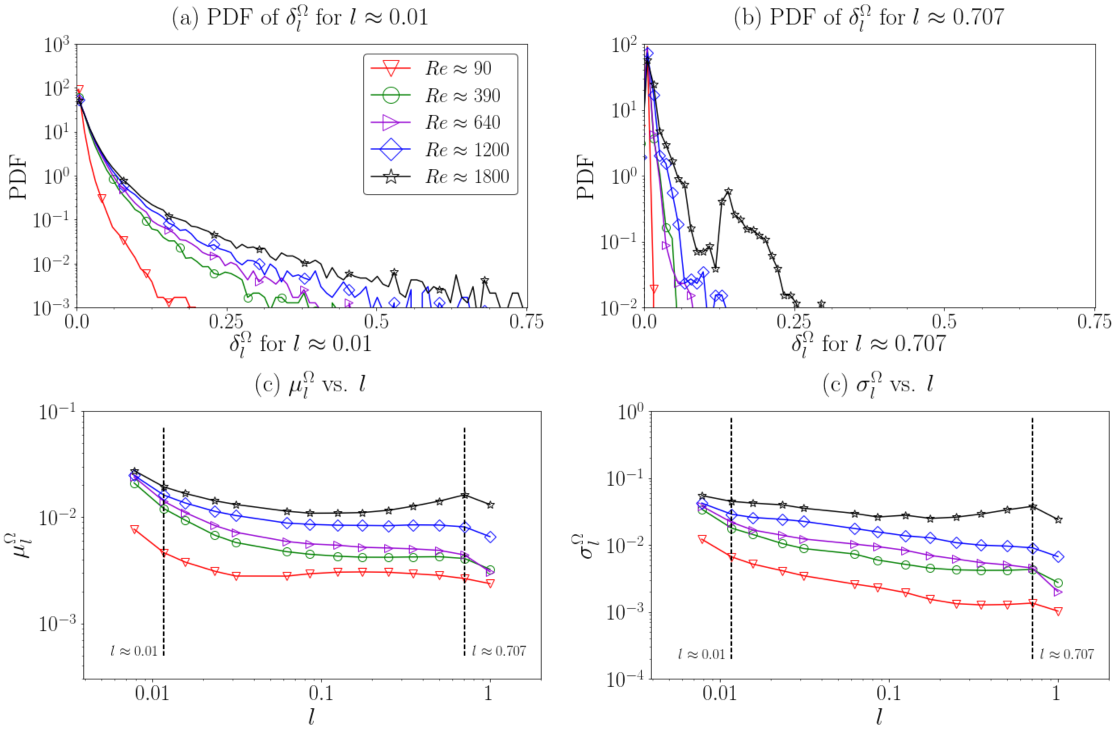


FIGURE 2.11. Kinetic energy balancing error δ_l^E (see Eq. (2.5)) in function of the size of the sub-volume l for 5 forced LBGK simulation of different Reynolds numbers. Top figures are PDF of the balancing error for sub-volumes corresponding to $l \approx 0.01$ (Panel (a)) and $l \approx 0.707$ (Panel (b)). Bottom figures are the mean (Panel (c)) and the standard deviation (Panel (d)) of the balancing error. Taken from Ref [6] with permissions.



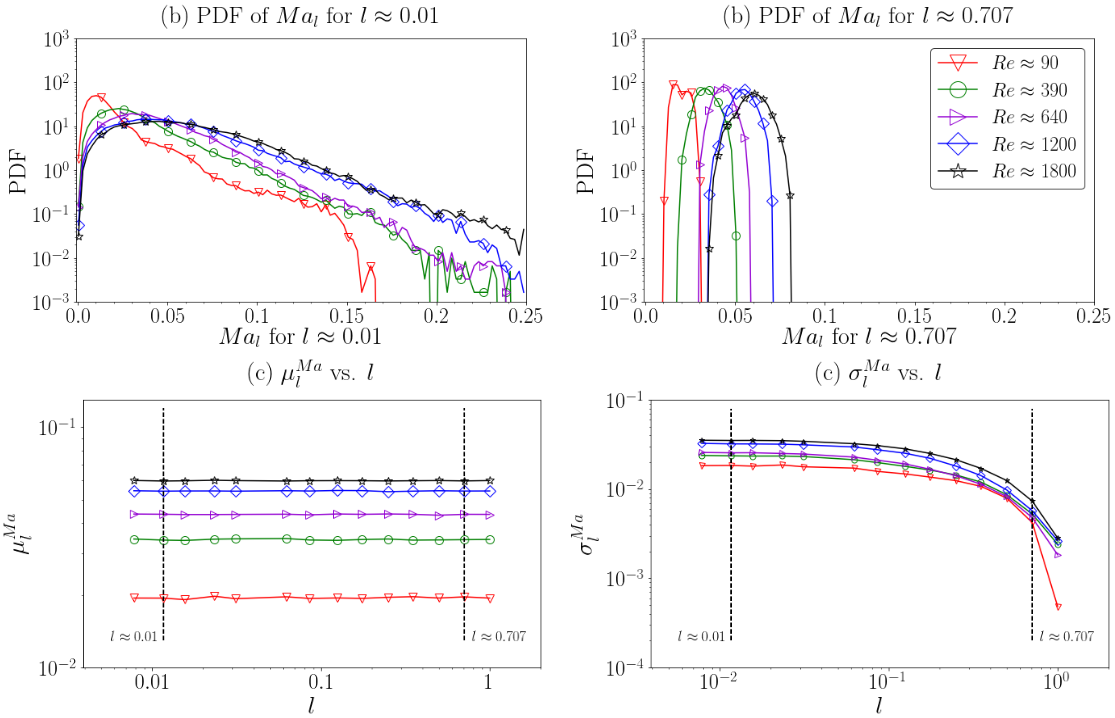
(a) PDF of δ_l^Ω for $l \approx 0.01$

(b) PDF of δ_l^Ω for $l \approx 0.707$

(c) μ_l^Ω vs. l

(c) σ_l^Ω vs. l

FIGURE 2.12. Enstrophy balancing error δ_l^Ω (see Eq. (2.6)) against the size of the sub-volume l shown for 5 forced LBGK simulation of different Reynolds numbers. Top figures are PDF of the balancing error for sub-volumes corresponding to $l \approx 0.01$ (Panel (a)) and $l \approx 0.707$ (Panel (b)). Bottom figures are the mean (Panel (c)) and the standard deviation (Panel (d)) of the balancing error. Taken from Ref [6] with permissions.



(b) PDF of Ma_l for $l \approx 0.01$

(b) PDF of Ma_l for $l \approx 0.707$

(c) μ_l^{Ma} vs. l

(c) σ_l^{Ma} vs. l

FIGURE 2.13. Local Mach number (see Eq. (2.14)) Ma_l in function of the normalized size of the sub-volume l shown for 5 forced LBGK simulation of different Reynolds numbers. Top figures are PDF of the balancing error for sub-volumes corresponding to $l \approx 0.01$ (Panel (a)) and $l \approx 0.707$ (Panel (b)). Bottom figures are the mean (Panel (c)) and the standard deviation (Panel (d)) of Ma_l. Taken from Ref [6] with permissions.

Overall, while the PS method offers a much higher level of accuracy than LBGK, LBGK proved to be able to recover the Navier-Stokes equation with a good agreement. Moreover, in both the validation and benchmark, a value of Mach number of 0.05 was shown to be low enough for its effect to be sub-leading in the hydrodynamic recovery. Therefore, we have obtained valuable insights on the LBGK recovery of hydrodynamics and we can now study the impact on this recovery of an additional SGS model term to the balancing equation while using the present results as reference.

CHAPTER 3

A-PRIORI STUDY OF ELBM HYDRODYNAMICS RECOVERY AND IMPLICIT SGS MODEL

In this chapter, we investigate the validity of the a priori assumption that ELBM can be macroscopically described as a LES with an eddy viscosity model. In section 3.1, we describe how the hydrodynamic recovery check tool was extended to include eddy viscous dissipation. In the context of 2D HIT (section 3.2), and 3D HIT (section 3.3), we first present the results of the extended hydrodynamic check tool and we numerically explore the range of validity of the macroscopic eddy viscosity; some concluding remarks follows in Section 3.4.

3.1 Extension of the hydrodynamic check tool

The hydrodynamic recovery check tool developed in Chapter 1 is based on calculating each term of the kinetic energy balance equations averaged over a suitable ensemble of sub-volumes of the computational grid and conducting a statistical analysis of an error to a perfect balancing. To extend this tool, we add the contributions related to the eddy viscous dissipation terms to the kinetic energy $E = \frac{\rho u_i u_i}{2}$) balance Eq (2.1). The full derivation is given in Appendix A and we give Eq. (3.2) the outcome. It describes the balance between the time derivative of the averaged (over a volume V denoted by $\langle \cdots \rangle_V$) kinetic energy(LHS_V^E) and the right-hand side (RHS_V^E) given by the contributions driving its evolution: the effect of compressibility, dissipation, input, and the transport and diffusive fluxes written for a viscosity changing in space and time $\nu = \nu_{\text{eff}}(\mathbf{x},\, t) = \nu_0 + \delta\nu_e(\mathbf{x},\, t)$. In this chapter, the eddy viscosity will be systematically taken as the measured eddy viscosity $\delta\nu_e = \delta\nu_e^M$, with

$$\delta\nu_e^M = c_s^2 \frac{2-\alpha}{2\alpha\beta}\Delta_t. \tag{3.1}$$

$$
\begin{aligned}
LHS_V^E =& \partial_t \langle \frac{\rho u_i u_i}{2} \rangle_V \\
=& - \left\langle \partial_j \left(\frac{\rho u_i u_i}{2} u_j \right) \right\rangle_V - \left\langle u_i \partial_i p \right\rangle_V + \left\langle u_i F_i \right\rangle_V \\
& - \left\langle \nu_0 \, \rho \left(\partial_j u_i + \partial_i u_j \right) \partial_j u_i \right\rangle_V + \left\langle \partial_j \left(\nu_0 \, \rho u_i \left(\partial_j u_i + \partial_i u_j \right) \right) \right\rangle_V \\
& - \left\langle \delta\nu_e \, \rho \left(\partial_j u_i + \partial_i u_j \right) \partial_j u_i \right\rangle_V + \left\langle \partial_j \left(\delta\nu_e \, \rho u_i \left(\partial_j u_i + \partial_i u_j \right) \right) \right\rangle_V \\
=& RHS_V^{E,1} + RHS_V^{E,2} + RHS_V^{E,3} + RHS_V^{E,4} + RHS_V^{E,5} + RHS_V^{E,6} + RHS_V^{E,7} \\
=& RHS_V^E
\end{aligned}
\tag{3.2}
$$

We redefine a balancing error δ_V^E in order to quantify the accuracy of the hydrodynamics recovery over a sub-volume V:

$$
\delta_V^E(t) = \frac{\left| RHS_V^E(t) - LHS_V^E(t) \right|}{L_0^{-1} \left(\max_t \left\langle E(t) \right\rangle_{V_0} \right)^{\frac{3}{2}}}. \tag{3.3}
$$

Note that the denominator of Eq. (3.3) used to normalize the error is arbitrary, and the one chosen here represents properly the order of magnitude and the dimension of the therms of the kinetical balance equation.

As in the previous chapter, the sub-volume averaged terms are calculated offline based on the output configuration fields. A 2$^{\text{nd}}$ order explicit Euler scheme is used (see Eq. (2.10)) to evaluate time derivatives. However, in this chapter, spectral derivatives are used to calculate spatial derivatives.

3.2 ELBM simulation of 2D Homogeneous Isotropic Turbulence, 2D HIT

In order to evaluate the implicit SGS of ELBM, we first conduct a set of simulations of forced 2D homogeneous isotropic turbulence (HIT) at different Reynolds numbers on a periodic two-dimensional 256 × 256 computational grid. We make use of a 2D lattice with 9 discrete velocities, the well-known D2Q9 [33, 34, 35]. The macroscopic forcing used to trigger homogeneous isotropic turbulence acts on a shell of (dimensionless) wavenumbers $\mathbf{k}$ of magnitude from 5 to 7 with a constant phase ϕ. We write it in a stream-function formulation to ensure that it does not generate any incompressibility:

$$
F_\Psi^T(\mathbf{x}) = F_0^T \sum_{5 \leq \|\mathbf{k}\| \leq 7} \cos\left(\frac{2\,\pi}{L_0} \mathbf{k} \cdot \mathbf{x} + \phi \right) \tag{3.4}
$$

with $L_0 = 256$, the size of the computational domain.

The reciprocal force $\mathbf{F^T}$ is then given by

$$F_x^T = \partial_y F_\Psi^T \qquad \text{and} \qquad F_y^T = -\partial_x F_\Psi^T, \tag{3.5}$$

in which the forcing amplitude F_0^T is kept constant for all simulations. Based on it, we define a time-scale $T_f = \sqrt{\frac{2\pi}{k_f F_0^T}}$, where k_f is taken equal to six, the average wavenumber forced in Eq, (3.4). As 2D turbulence is characterized by the presence of a backward energy cascade [66, 14], in order to recover the NSE using LBM, it is necessary to maintain the Mach number under control. Therefore, as in Eq. (1.1), we introduce a large scale drag term, but here in order not to spoil the small scale physics, we use spectral forcing to damp only energy in the shell of wavenumber of amplitude between 1 to 2:

$$\mathbf{F^R}(\mathbf{x},\, t) = -F_0^R \sum_{1 \le \|\mathbf{k}\| \le 2} \hat{\mathbf{u}}(\mathbf{k},\, t)\, e^{\frac{2\,\pi}{L_0}\mathbf{k}\cdot\mathbf{x}}, \tag{3.6}$$

where $\hat{\mathbf{u}}(\mathbf{k},\, t)$ is the Fourier transform of $\mathbf{u}(\mathbf{x},\, t)$.

We enforce the resulting forcing $\mathbf{F} = \mathbf{F^T} + \mathbf{F^R}$ using the exact-difference method forcing scheme [61] (See Chapter 1.4.3).

Table 3.1. Parameters of the conducted simulations

Re	τ_0	F_0^T	F_0^R
60	0.51	$5.0\ 10^{-4}$	$1.0\ 10^{-6}$
240	0.5025	$5.0\ 10^{-4}$	$1.0\ 10^{-5}$
1200	0.5005	$5.0\ 10^{-4}$	$1.5\ 10^{-5}$
6000	0.5001	$5.0\ 10^{-4}$	$2.0\ 10^{-5}$
12000	0.50005	$5.0\ 10^{-4}$	$5.0\ 10^{-4}$

We conduct a set of one LBGK and five ELBM statistically stationary simulations varying the input relaxation time τ_0 to gradually increase the Reynolds number while changing the large-scale energy damping forcing amplitude to maintain the Mach number under control. The conducted LBGK simulation uses the last stable relaxation time $\tau_0 = 0.51$. For the first ELBM simulation, we use the same relaxation time $\tau_0 = 0.51$ and we gradually decrease it down to $\tau_0 = 0.50005$. The parameters used in each simulation are summarized in Table 3.2. Fig. 3.9 shows that the Mach number is successfully maintained around a value of $5.0\ 10^{-2}$. This Mach number was shown to be sufficiently low to recover well the NSE in Chapter 2.

We show in Fig. 3.10 the superposed time-averaged spectrum for the conducted simulations. At large scale, the energy removal prevents the energy to accumulate. On the other hand, at small scales, we observe that, as we decrease τ_0 (increasing Re), the flow becomes more turbulent and the slope gets increasingly closer to the forward enstrophy

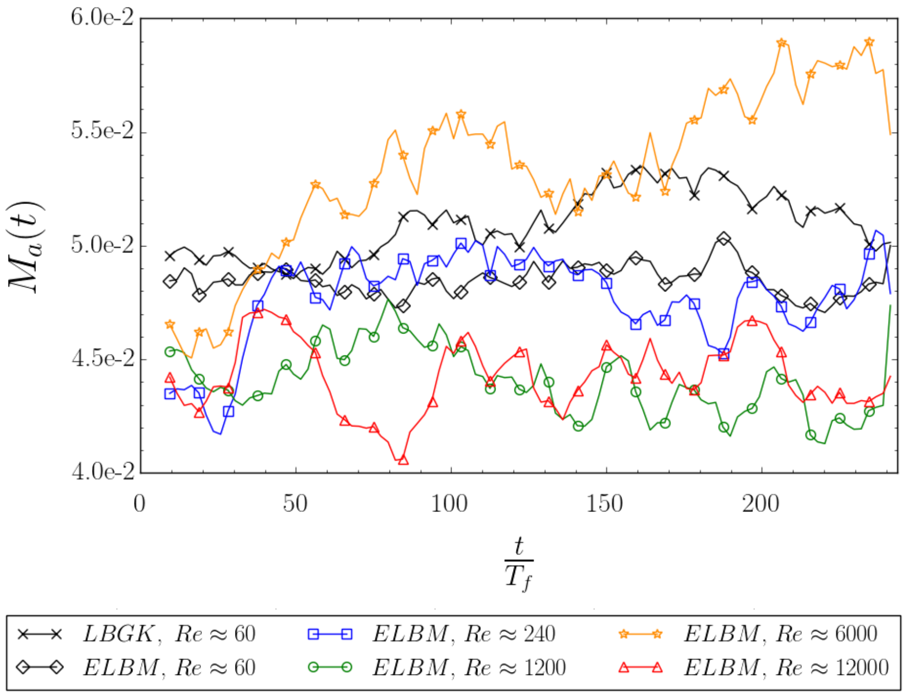


FIGURE 3.1. Evolution of the Mach number of the conducted LBGK and ELBM simulations.

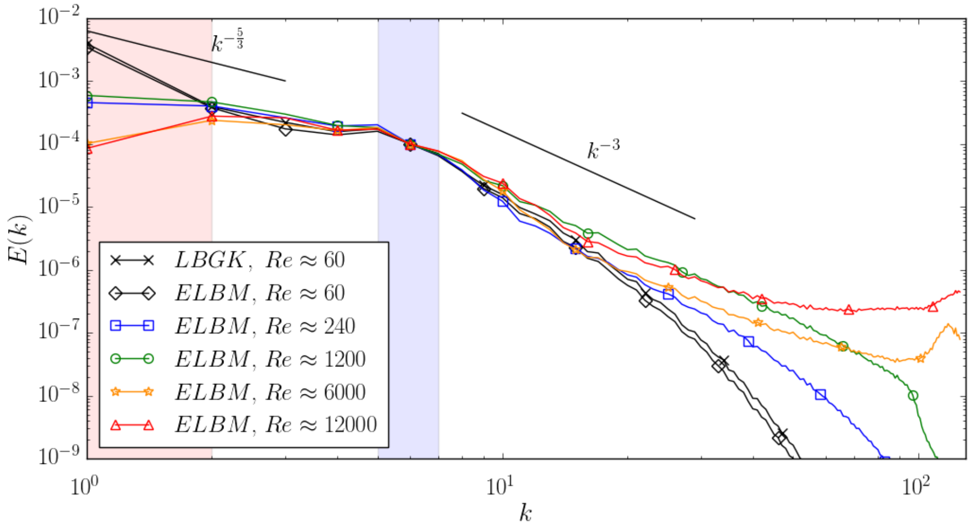


FIGURE 3.2. Superposed time-averaged spectrum of a LBGK simulation for the ELBM simulations for five different relaxation times. The red background highlights the range of wavenumbers where energy damping is active and the blue backgrounds show the range of wavenumbers where energy is injected. The Kraichnan-Leith-Batchelor predictions for the backward energy cascade slope of $k^{-5/3}$ and forward enstrophy cascade slope of k^{-3} are given for reference.

cascade slope of -3. For ELBM simulations of $Re \approx 1200$, the implicit ELBM SGS model nicely extends the inertial range of the direct cascade, while for simulations of $Re \approx 6000$ and $Re \approx 12000$, the energy accumulates at small scales, showing that the implicit SGS model is not dissipative enough.

From those observations, we identify three typical ELBM simulation cases. The first one, for $Re \approx 60$, exhibits an exponential decay at small scales and therefore, the resolution of the simulation is high enough to resolve all scales (*over-resolved* case). The second one for $Re \approx 1200$, shows a long inertial range and will be labeled as the *optimally-resolved* case. The third one, for $Re \approx 12000$, highlights a significant energy accumulation at small scales, showing that not enough scales are resolved and the model is not able to properly dissipate energy. It will be designated as the *under-resolved* case.

3.2.1 Validation of the ELBM hydrodynamics

To illustrate the typical hydrodynamic recovery of an ELBM simulation, we sho in Fig. 3.11 an example of the evolution in time of the kinetic energy balancing for the optimally-resolved simulation on a randomly chosen sub-volume of size 128×128. The matching between the left-hand side (LHS_V^E) and the right-hand side (RHS_V^E) shows very small discrepancies, whereas the total RHS_V^E terms are obtained by summing up terms of significantly higher amplitude. Overall, the balancing error δ_V^E is of the order $\mathcal{O}(10^{-2})$ to $\mathcal{O}(10^{-3})$ and underlines an excellent hydrodynamics recovery.

To describe the hydrodynamics recovery accuracy, we conduct a statistical analysis of the balancing error $\delta_V^E(t)$ over randomly chosen squared sub-volumes $V = L \times L$. We introduce the normalized sub-volume size $l = \frac{L}{L_0}$ with $L_0 = 256$ the size of the squared computational domain and gather the balancing errors $\delta_l^E = \delta_{V=L\times L}^E$ obtained for 10000 sub-volumes of the same normalized sub-volume size l on 25 configurations sampling the statistically stationary domain. We call their mean μ_l^E.

We show the results of the statistical analysis of the kinetic energy balancing error δ_l^E in Fig. 3.4. As expected from the hydrodynamics recovery of LBGK simulations in chapter 2, the size of the sub-volume strongly affects the hydrodynamics recovery accuracy. Indeed, hydrodynamics is much better recovered on large sub-volumes than on small sub-volumes with up to three orders of magnitudes difference (compare values at dashed lines in Fig. 3.4, Panels (c)).

On the statistics of the kinetic energy balancing error in Fig. 3.4, we can also observe a dependence on the Reynolds number on the mean μ_l^E (Panel (c)) with the balancing errors globally being higher for the simulations that are less resolved. For simulations

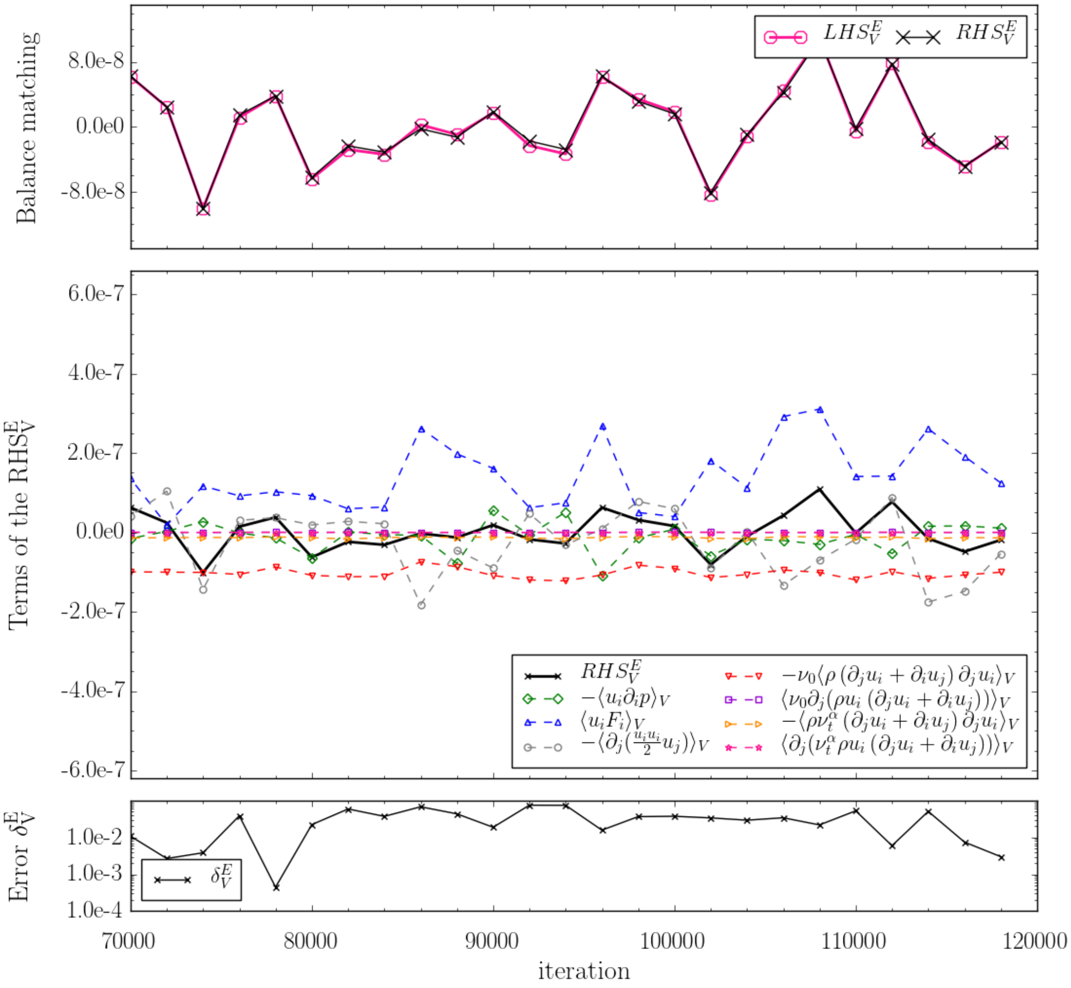


FIGURE 3.3. Typical time-evolution of the kinetic energy balancing over a single sub-volume of size 128 × 128 shown for the optimally-resolved ELBM simulation ($R_e \approx 1200$) on a 256 × 256 grid. The top figures shows the evolution of the matching between the LHS_V^E and the RHS_V^E, the middle figures shows the contribution of each $RHS_V^{E,i}$ term and their sum, RHS_V^E, and the bottom figure shows the evolution of the balancing error δ_V^E.

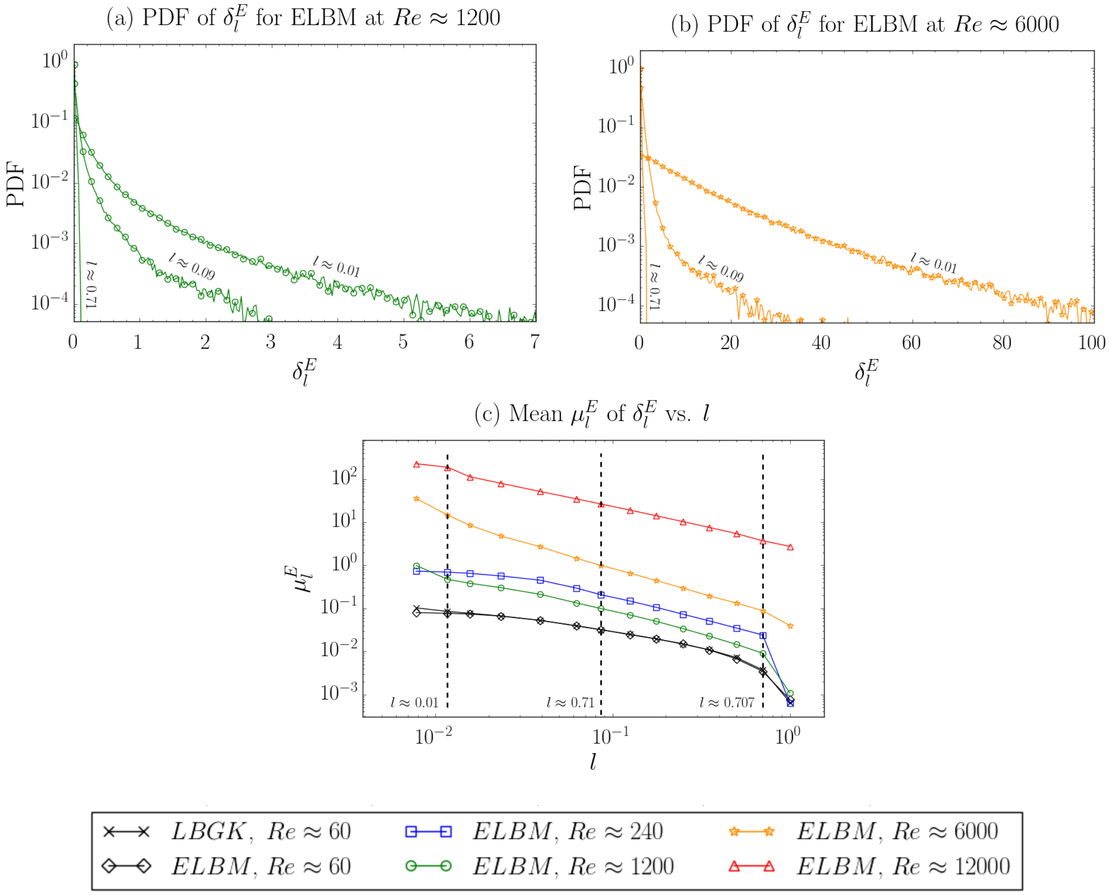


FIGURE 3.4. Statistics of the balancing error obtained from the kinetic energy balance δ_l^E (see Eq. (3.3)) against the size of the sub-volume l for 5 forced LBGK simulation of different Reynolds numbers. Top figures are PDF of the balancing error for ELBM simulations at $Re \approx 1200$ (Panel (a)) and at $Re \approx 6000$ (Panel (b)) for sub-volumes corresponding to $l \approx 0.01$, $l \approx 0.09$, and $l \approx 0.707$. Bottom figure is the mean of the balancing error (Panel (c)).

of $R_e \leq 1200$, up to the optimally-resolved case, this trend is less obvious and the accuracy of the hydrodynamics of simulation of $R_e \approx 240$ is slightly better than the one of $R_e \approx 1200$ (optimally-resolved case). Beyond this Reynolds number, the quality of hydrodynamics recovery appears to quickly decrease as can be seen on the probability distribution functions (PDF) of δ_l^E for the simulations $Re \approx 1200$ (Panel (a)) and $Re \approx 6000$ (Panel (b)). Indeed, they highlight that both at small sub-volumes and large sub-volumes, there is a jump of over an order of magnitude in the balancing error for simulations of Reynolds numbers beyond the optimally-resolved simulation's.

Overall, the hydrodynamics recovery accuracy remains good for simulations of Reynolds number from 60 (over-resolved case) to 1200 (optimally-resolved case) but is not maintained to the LBGK's order of magnitude. Higher Reynolds number simulations up to $R_e = 12000$ (under-resolved case) exhibit a higher balancing error, especially for small sub-volumes. As a result, it appears that ELBM's implicit SGS modeling is able to properly maintain a recovery of hydrodynamics for simulations with a Reynolds number up to 20 times the Reynolds number of the last stable LBGK simulation. For higher Reynolds number, we observed that hydrodynamics is not preserved and ELBM acts as a mere stabilizer.

3.2.2 Numerical assessment of the macroscopic behavior of ELBM

The derivation of the approximated eddy viscosity

$$\delta\nu_e^A \approx -c_s^2 \Delta t^2 \frac{1}{6\beta^2} \frac{S_{\lambda\mu} S_{\mu\gamma} S_{\gamma\lambda}}{S_{\gamma\delta} S_{\gamma\delta}}, \tag{3.7}$$

where $S_{ij} = \frac{1}{2}(\partial_i u_j + \partial_j u_i)$ is the strain-rate tensor (see Chapter 4 for the full derivation), is based on the assumption of small deviation of α around 2. As shown on the PDF of α (Fig. 3.15), this assumption remains valid for all of our ELBM simulations with a maximum deviation of 1% in the under-resolved case. We note that the smaller the Reynolds number, the smaller the fluctuations of α around 2 are.

To assess the validity of the approximated eddy viscosity $\delta\nu_e^A$, we measure the eddy viscosity stemming from the effective relaxation time $\delta\nu_e^M$ using Eq. (1.56). We show on Fig. 3.16, the joint PDF between $\delta\nu_e^M$ and $\delta\nu_e^A$ both expressed relatively to the input viscosity ν_0. Pearson's correlation coefficient r decreases as the Reynolds number increases and the amplitude of the fluctuations of α increases. It reaches $r = 0.89$ in the over-resolved case, showing a quite strong correlation, to $r = 0.07$ in the under-resolved case, where the approximate eddy viscosity no longer holds. However, even in the over-resolved case, the joint PDF highlights the presence of two branches, one of

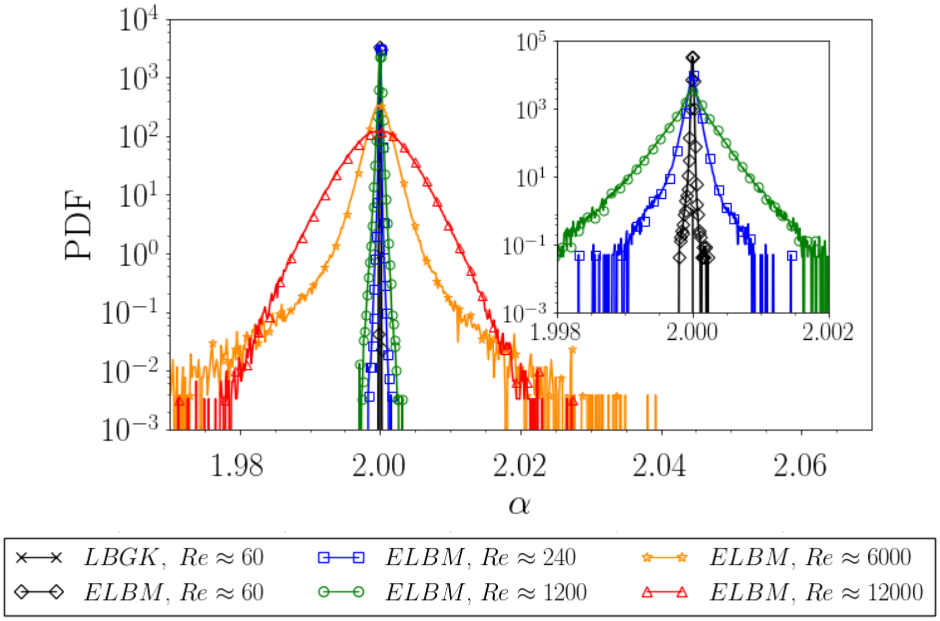


FIGURE 3.5. PDF of α for the 5 ELBM simulations. A line of $\alpha = 2$ has been added for the LBGK simulation for reference.

them showing a very good agreement between the measured and approximated viscosity while the other is not properly understood at this stage.

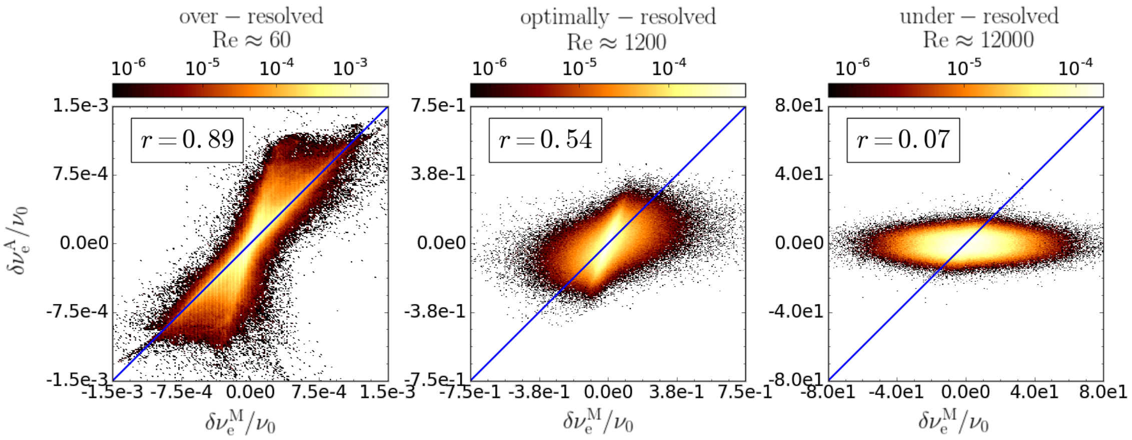


FIGURE 3.6. Joint PDF between the measured eddy viscosity $\delta\nu_e^M/\nu_0$ (see Eq. (3.1)) and the approximated eddy viscosity $\delta\nu_e^A/\nu_0$ (see Eq. (3.7)) expressed relatively to the input viscosity ν_0 for the three showcases: the over-resolved case, optimally-resolved case, and under-resolved case. The blue curve shows a perfect fit $\delta\nu_e^M = \delta\nu_e^A$, and r is Pearson's correlation coefficient.

The snapshot Fig. 3.18 gives a valuable insight on the spatial correlation between the measure eddy viscosity $\delta\nu_e^M$ and the approximated one $\delta\nu_e^A$. The vorticity ω is also plotted as a reference. We observe that the eddy viscosity is a small scale quantity and the spoiling of the spatial correlation between $\delta\nu_e^M$ and $\delta\nu_e^A$ with the increase of the Reynolds number is again highlighted.

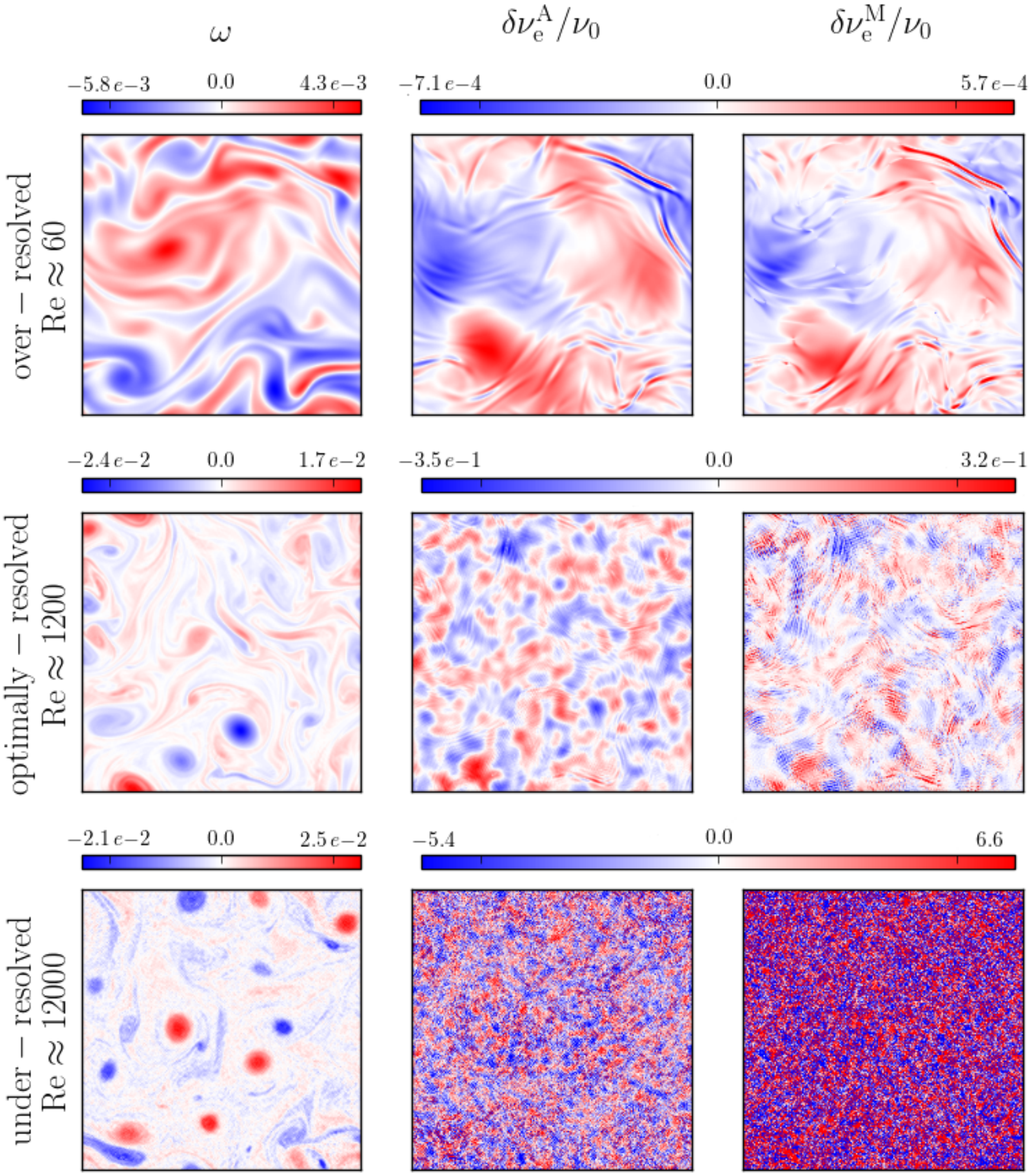


FIGURE 3.7. Snapshot of the vorticity $\omega = \partial_x u_y - \partial_y u_x$, of the measured eddy viscosity $\delta\nu_e^M/\nu_0$ (see Eq. (3.1)) and the approximated eddy viscosity $\delta\nu_e^A/\nu_0$ (see Eq. (3.7)) expressed relatively to the input viscosity ν_0 for the three showcases. From left to right, the over-resolved case, the optimally-resolved case, and under-resolved case.

Additionally, we can observe on the PDF of the measure and approximated eddy viscosity in Fig. 5.3 that for simulations of Reynolds number below the one of the optimally-resolved simulation, the measured turbulent viscosity $\delta\nu_e^M$ is slightly positively skewed, making the ELBM implicit SGS overall dissipative, while the approximated one, $\delta\nu_e^A$ SGS does not exhibit this characteristic.

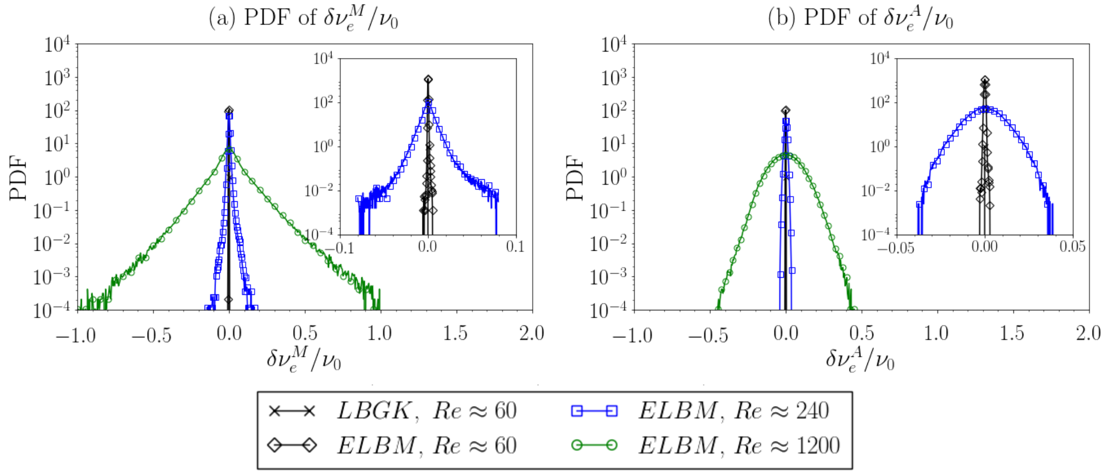


FIGURE 3.8. PDF of $\frac{\delta\nu_e^M}{\nu_0}$ (a) and $\frac{\delta\nu_e^A}{\nu_0}$ (b) for the LBGK and ELBM simulations at $Re \approx 60$ and the ELBM simulations at $Re \approx 240$ and $Re \approx 1200$. The For the LBGK simulation, $\frac{\delta\nu_e^M}{\nu_0} = 0$ and $\frac{\delta\nu_e^A}{\nu_0} = 0$ are plotted for reference. Insets contain zoom-ins of the PDF for LBGK and ELBM simulations at $Re \approx 60$ and the ELBM simulation at $Re \approx 240$.

3.3 ELBM simulation of 3D Homogeneous Isotropic Turbulence, 3D HIT

We investigate 3D HIT by conducting a set of forced simulations on a periodic three-dimensional $256 \times 256 \times 256$ computational grid on a lattice with 27 discrete velocities, the D3Q27 [33, 34, 35]. The macroscopic homogeneous isotropic turbulence forcing acts on a shell of (dimensionless) wavenumbers $\mathbf{k}$ of magnitude from 1 to 2 with a constant phase ϕ. We take its rotational formulation to ensure that it does not generate any incompressibility:

$$\mathbf{F} = F_0^T \nabla \times \mathbf{F}^{Shell} \text{ with } \hat{\mathbf{F}}^{Shell}(\mathbf{k}, t) = \begin{cases} \mathbf{1}, & \text{if } |\mathbf{k}| \leq 2 \\ \mathbf{0}, & \text{otherwise} \end{cases}, \tag{3.8}$$

where $\hat{\mathbf{F}}^{Shell}(\mathbf{k}, t)$ is the Fourier transform of $\mathbf{F}^{Shell}(\mathbf{x}, t)$ and the forcing amplitude F_0^T is kept constant for all simulations to $5.96\ 10^{-8}$. This value has been chosen in order to to maintain a low Mach number, which allows the recovery of the NSE using LBM. Based on this forcing amplitude, we can define a time-scale $T_f = \sqrt{\frac{2\pi}{k_f F_0^T}}$, where $k_f = 2$.

We enforce the resulting forcing $\mathbf{F}$ using, again, the exact-difference method forcing scheme [61] (see Chapter 1.4.3)

We conduct a set of four LBGK and five ELBM statistically stationary simulations varying the input relaxation time τ_0 to gradually increase the Reynolds number. The conducted LBGK simulations starts from $\tau_0 = 0.55$ to the last stable relaxation time $\tau_0 = 0.502$. For the first ELBM simulation, we start at the first with the smaller relaxation time of the conducted LBGK simulations for which the entropic scheme is active (α not homogeneously equal to 2), $\tau_0 = 0.51$, and we conduct simulations gradually decreasing it down to $\tau_0 = 0.50002$. The parameters used in each simulation are summarized in Table 3.2. Fig. 3.9 shows that the Mach number is maintained around a value of 5.0 10^{-2}. This Mach number was shown to be sufficiently low to recover well the NSE in 2D turbulent flows in Chapter 2.

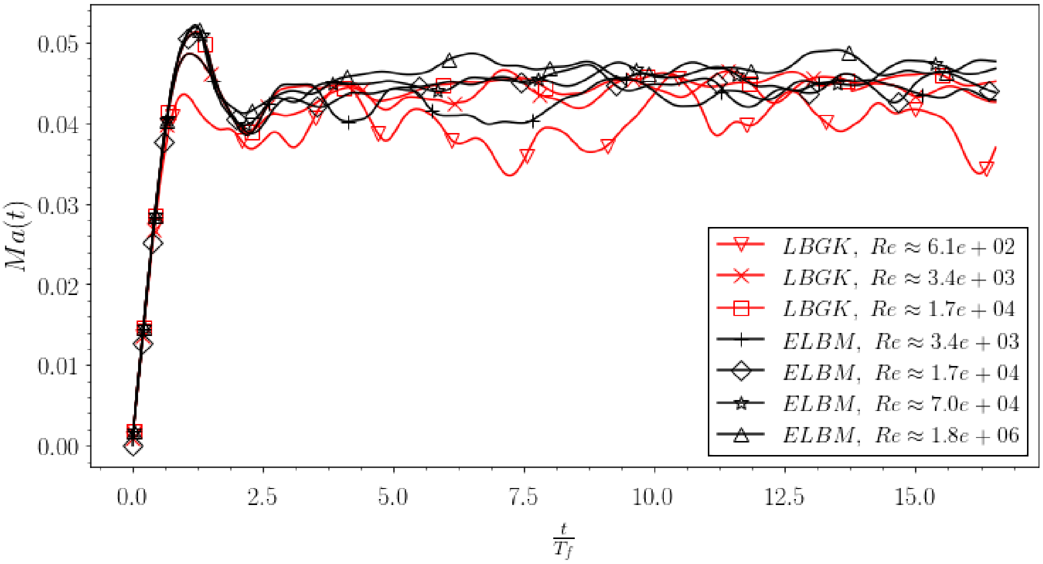


FIGURE 3.9. Evolution of the Mach number of the conducted LBGK (red lines) and ELBM (black lines) simulations.

TABLE 3.2. Parameters of the conducted simulations

Re	τ_0	$\mathbf{F_0^T}$
6.1 10^2	0.55	5.96 10^{-8}
3.4 10^3	0.51	5.96 10^{-8}
1.7 10^4	0.502	5.96 10^{-8}
7.0 10^4	0.5005	5.96 10^{-8}
1.8 10^6	0.50002	5.96 10^{-8}

We show in Fig. 3.10 the superposed time-averaged spectrum for the conducted simulations. At small scales, we observe as we decrease τ_0 (increasing Re) that the flow becomes more turbulent and the slope gets increasingly closer to the forward energy cascade slope of $-5/3$ [28, 14]. For the ELBM simulation at $Re \approx 7.0\ 10^4$, we observe that the implicit ELBM SGS model is able to nicely extends the inertial range of the direct cascade, while for the simulation at $Re \approx 1.8\ 10^6$ the energy accumulates at small scales, showing that the implicit SGS model is not dissipative enough.

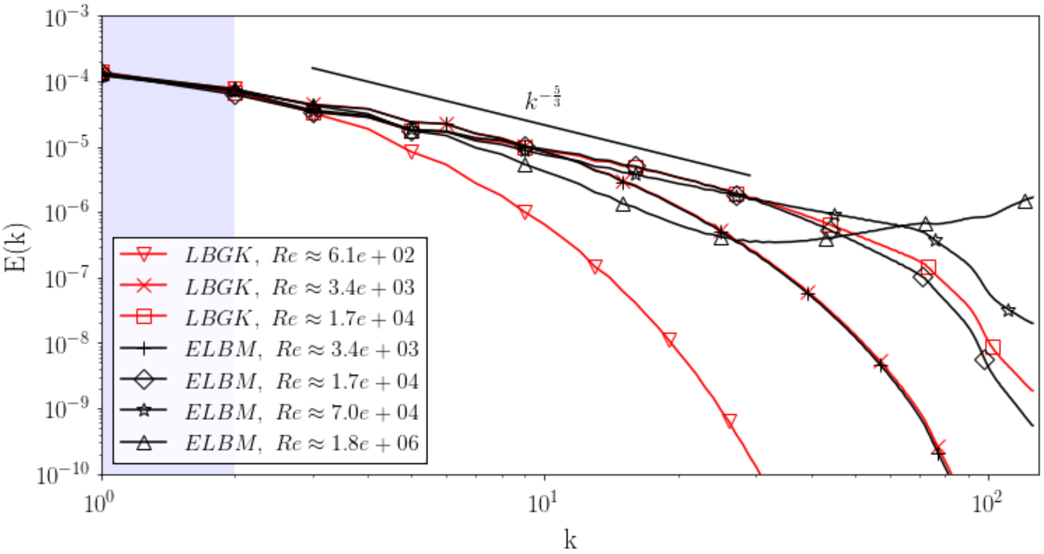


FIGURE 3.10. Superposed time-averaged spectrum of four LBGK simulations (red lines) and five ELBM simulations (black lines) varying the relaxation times. The blue background shows the the range of wavenumbers where energy is injected.

We identify four typical ELBM simulation cases. The first one, for $Re \approx 3.4\ 10^3$, exhibits an exponential decaying at small scales and therefore, the resolution of the simulation is high enough to resolve all scales (*over-resolved* case). The second one for $Re \approx 1.7\ 10^4$, is barely able to solve all scales of the flow and also corresponds to the smallest stable τ_0 for LBGK. It will be referred as the *resolved* case. The third one, $Re \approx 7.0\ 10^4$, shows an extended inertial range and will be labeled as the *optimally-resolved* case. The last one, for $Re \approx 1.8\ 10^6$, highlights a significant energy accumulation at small scales, showing that not enough scales are resolved and the model is not able to properly dissipate energy. It will be designated as the *under-resolved* case.

3.3.1 Validation of the ELBM hydrodynamics

Fig. 3.11 presents an example of the evolution in time of the kinetic energy balancing for the over-resolved simulation on a randomly chosen sub-volume of size 128×128×128. The matching between the left-hand side (LHS^E_V) and the right-hand side (RHS^E_V) shows very small discrepancies, whereas the total RHS^E_V terms are obtained by summing up terms of significantly higher amplitude. Overall, the balancing error δ^E_V is of the order $\mathcal{O}(10^{-1})$ and highlights a good hydrodynamics recovery.

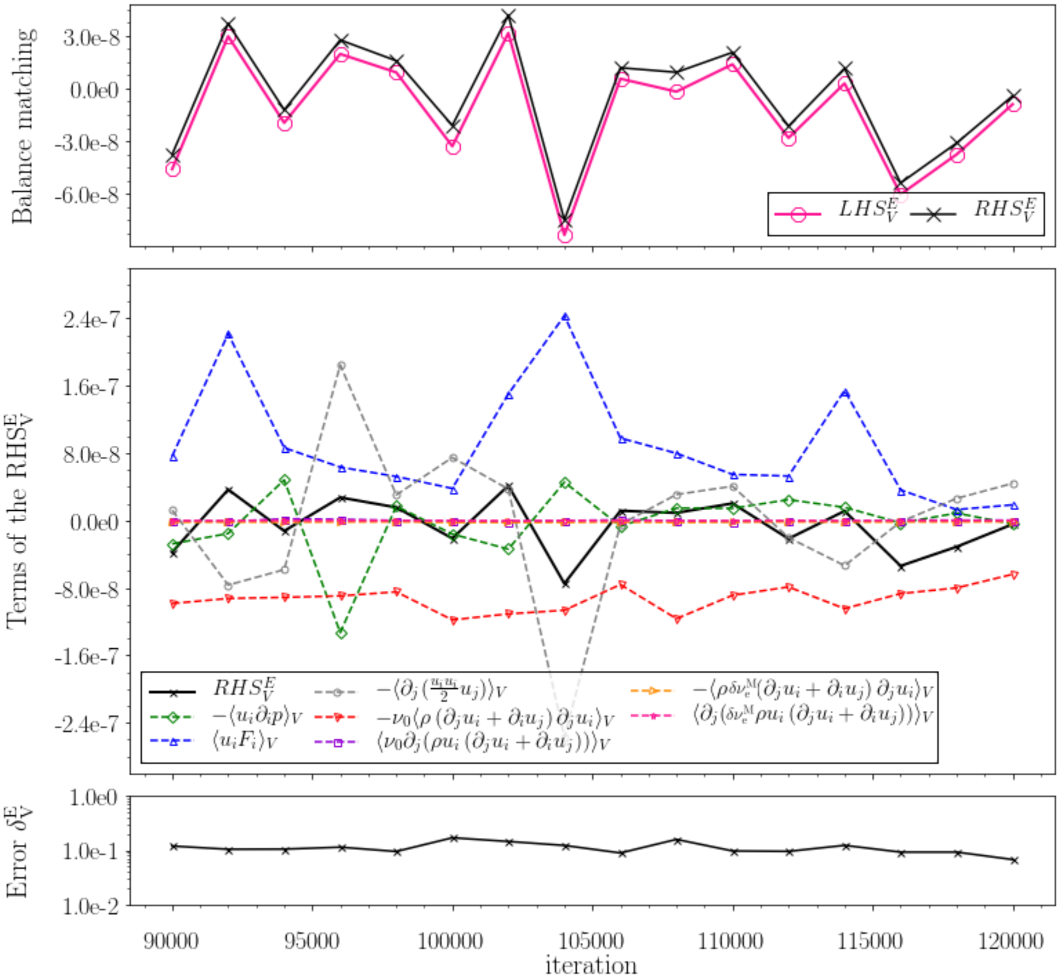


FIGURE 3.11. Typical time-evolution of the kinetic energy balancing over a single sub-volume of size 128 × 128 × 128 shown for the *over-resolved* ELBM simulation ($Re \approx 3.4\ 10^3$) on a 256 × 256 × 256 grid. The top figures shows the evolution of the matching between the LHS^E_V and the RHS^E_V, the middle figures shows the contribution of each $RHS^{E,i}_V$ term and their sum, RHS^E_V, and the bottom figure shows the evolution of the balancing error δ^E_V.

However, it is worth noticing that rescaling the input viscosity ν_0 can significantly improve the quality of the hydrodynamics recovery. Indeed, we show on Fig. 3.12, the improvement compared to Fig. 3.11 by rescaling ν_0 by a factor 1.06. The achieved balancing error improved by over an order of magnitude to $\mathcal{O}(10^{-2})$–$\mathcal{O}(10^{-4})$. This is the sign of the presence of so-called numerical dissipation, where the numerical scheme artificially increase dissipation, although it is usually seen with Finite-Difference-Method-based schemes (see for example Ref. [67]).

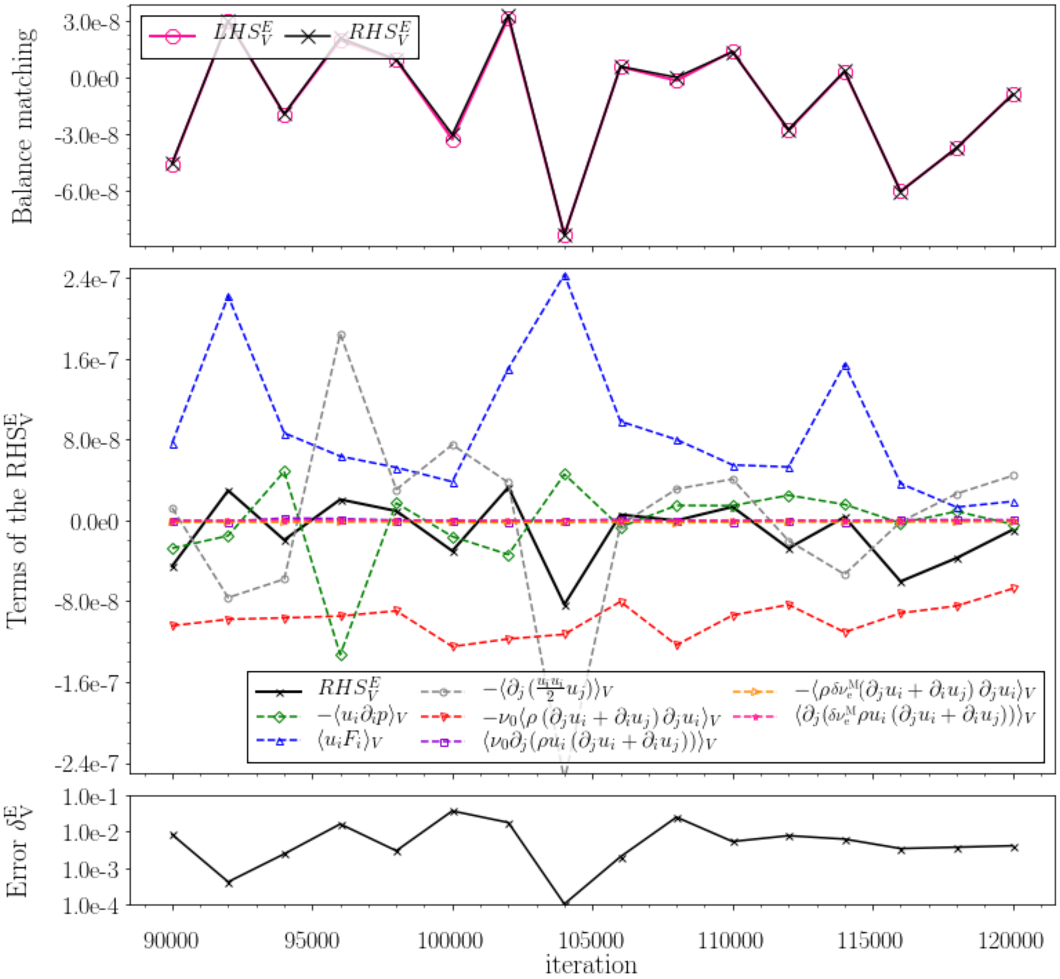


FIGURE 3.12. Typical time-evolution of the kinetic energy balancing over a single sub-volume of size 128 × 128 × 128 shown for the *over-resolved* ELBM simulation ($Re \approx 3.4\ 10^3$) on a 256 × 256 × 256 grid. The top figures shows the evolution of the matching between the LHS_V^E and the RHS_V^E, the middle figures shows the contribution of each $RHS_V^{E,i}$ term and their sum, RHS_V^E, and the bottom figure shows the evolution of the balancing error δ_V^E.

We show the results of the statistical analysis of the kinetic energy balancing error δ_l^E in Fig. 3.4. As expected from the hydrodynamics recovery of 2D HIT simulations presented in the previous section, the Reynolds number of the simulation, as well as the size of the sub-volume, strongly affects the hydrodynamics recovery accuracy. Indeed, as seen on the mean hydrodynamics is much better recovered on large sub-volumes (Fig. 3.13, Panels (b)) than on small sub-volumes (Fig. 3.13, Panels (a)) with up to three order of magnitudes difference (Fig. 3.13, Panels (c)). However, this trends is only valid for small subvolume sizes. From $\ell = 40$ on, we observe that we reach a plateau of accuracy.

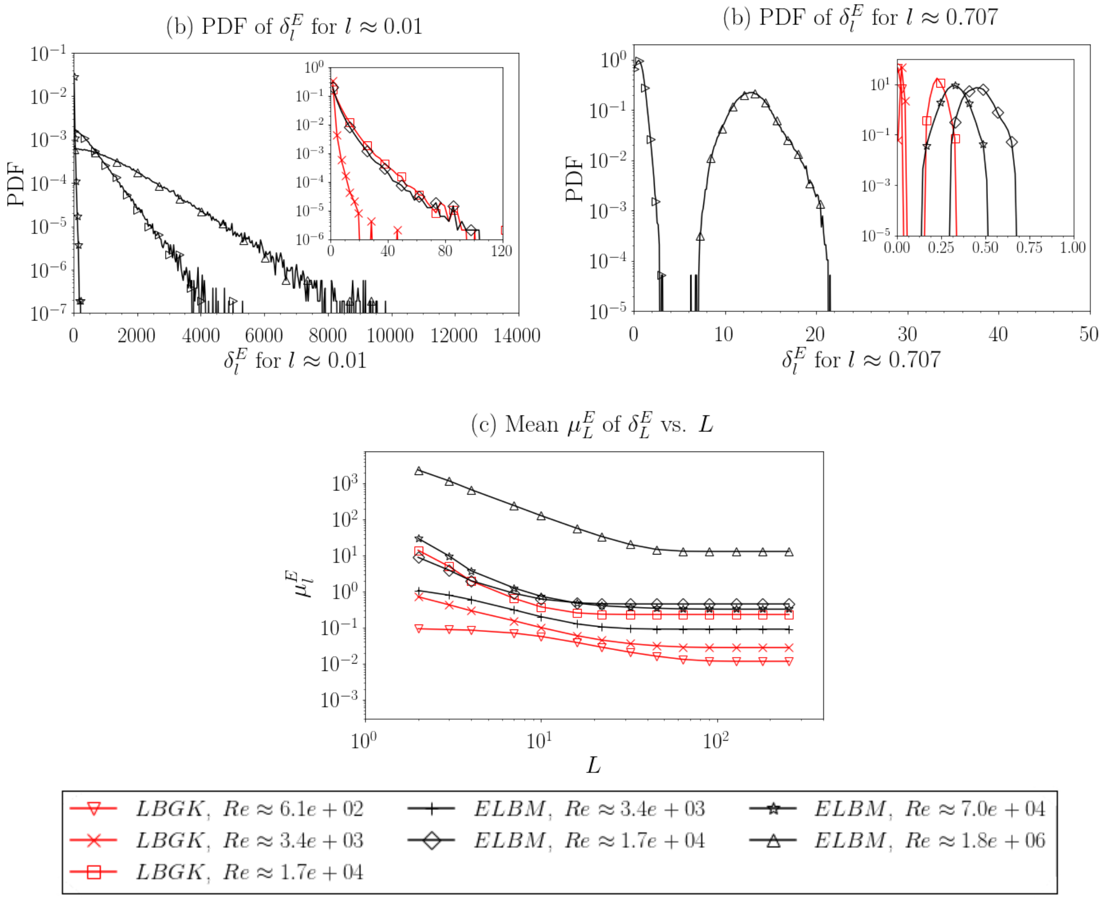


FIGURE 3.13. Statistics of the balancing error obtained from the kinetic energy balance δ_l^E (see Eq. (3.3)) against the size of the sub-volume l for all conducted simulations. Top figures are PDF of the balancing error at $\ell \approx 0.01$ (Panel (a)) and at $\ell \approx 0.707$ (Panel (b)) for sub-volumes corresponding to $l \approx 0.01$, $l \approx 0.09$, and $l \approx 0.707$. Bottom figure is the mean of the balancing error (Panel (c)).

After having observed the existence of a numerical dissipation, we have found the rescaling factors for the viscosity ν_0 that best improve best the hydrodynamic recovery accuracy (see Tab. 3.3). Taking this phenomenon into account, we present the kinetic energy balancing error for rescaled viscosities in Fig. 3.14. We observe a dependence on the Reynolds number on the mean μ_l^E (Panel (c)) with the balancing errors globally being higher for the simulations that are less resolved. For simulations of $Re \approx 3.4\ 10^3$ (*over-resolved* case), up to $Re \approx 7.0\ 10^4$ (*the optimally-resolved* case), we observe that the accuracy of the hydrodynamics of the simulations increases with their Reynolds number but are still contained within an order of magnitude. For Reynolds numbers beyond the *optimally-resolved* simulation's, the accuracy of the hydrodynamics recovery appears to quickly decrease as can be seen on the probability distribution functions (PDF) of δ_l^E both at small sub-volume sizes (Panel (a)) and large sub-volume sizes (Panel (b)). Indeed, they highlight a jump of several orders of magnitude in the balancing error for the *under-resolved* simulation.

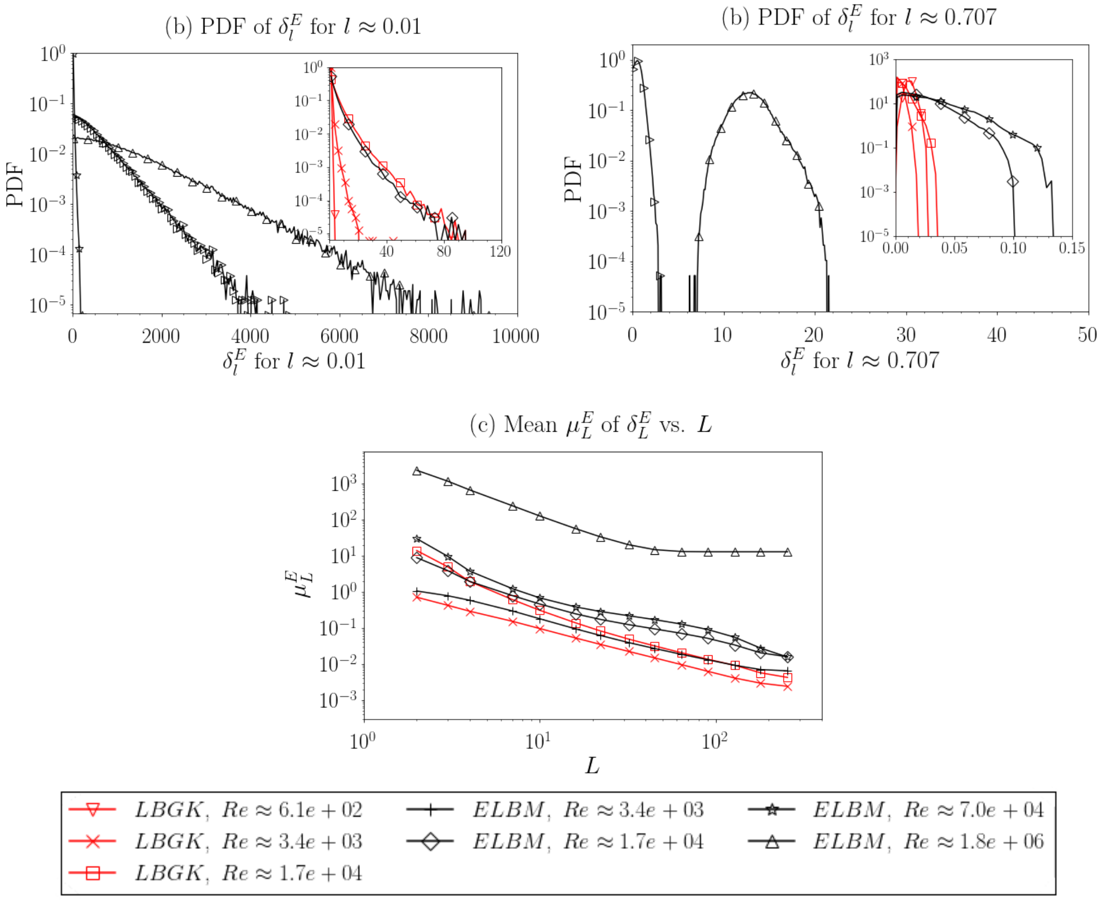


FIGURE 3.14. Statistics of the balancing error obtained from the viscosity-rescaled kinetic energy balance δ_l^E (see Eq. (3.3)) against the size of the sub-volume l for all conducted simulations. Top figures are PDF of the balancing error at $\ell \approx 0.01$ (Panel (a)) and at $\ell \approx 0.707$ (Panel (b)) for sub-volumes corresponding to $l \approx 0.01$, $l \approx 0.09$, and $l \approx 0.707$. Bottom figure is the mean of the balancing error (Panel (c)).

3.3.2 Numerical assessment of the macroscopic behavior of ELBM

The derivation of $\delta\nu_e^A$ Eq. (1.57) is based on the assumption of small deviation of α around 2. Again and as shown on the PDF of α (Fig. 3.15), this assumption remains

Table 3.3. Rescaling factors for the input viscosity ν_0 for each of the conducted simulations. For the ELBM simulation no rescaling factors were found to improve the hydrodynamic recovery.

Method	Re	Scaling factor for ν_0
LBGK	6.1 10^2	1.00
LBGK	1.7 10^4	1.02
ELBM	1.7 10^4	1.06
LBGK	3.4 10^3	1.19
ELBM	3.4 10^3	1.55
ELBM	7.0 10^4	1.60
ELBM	1.8 10^6	NA

valid for all of our ELBM simulations with a maximum deviation of 1% in the under-resolved case. We note, that as observed in the 2D HIT case in the previous section, the smaller the Reynolds number, the smaller the fluctuations of α around 2 are.

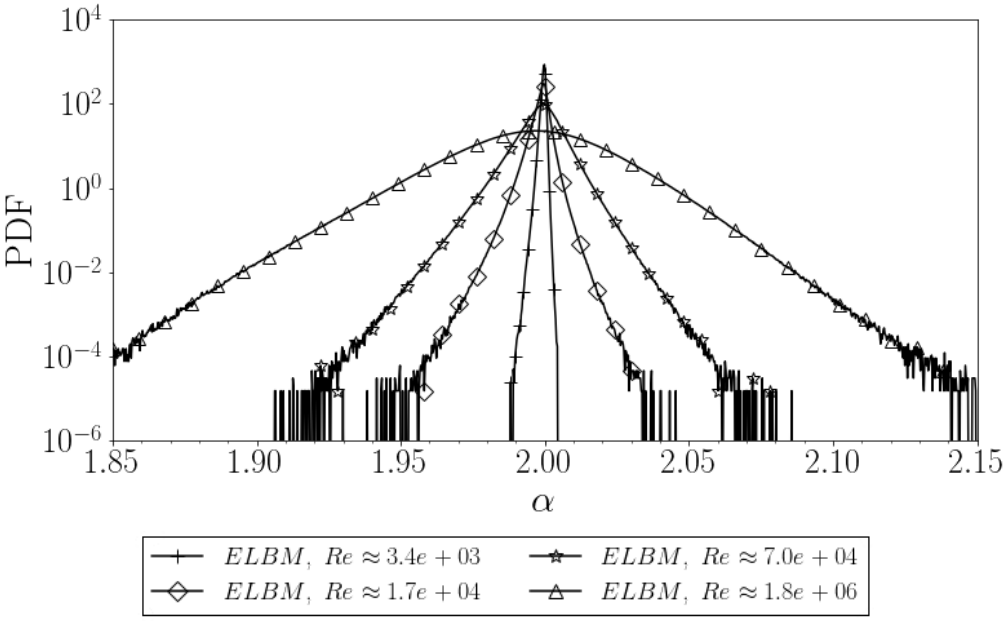


Figure 3.15. PDF of α for the 4 ELBM simulations. $\alpha = 2$ corresponds to the value for LBGK simulationa.

To assess the validity of the approximated eddy viscosity $\delta\nu_e^A$ Eq. (1.57), we measure the eddy viscosity stemming from the effective relaxation time $\delta\nu_e^M$ using Eq. (1.56). We show on Fig. 3.16, the joint PDF between $\delta\nu_e^M$ and $\delta\nu_e^A$ both expressed relatively to the input viscosity ν_0. Pearson's correlation coefficient r decreases as the Reynolds number increases and the amplitude of the fluctuations of α increases. It reaches $r = 0.90$ in the over-resolved case, showing a quite strong correlation, but quickly goes down to $r = 0.49$

for the optimally-resolved case and to $r = 0.1$ in the under-resolved case, where the approximate eddy viscosity no longer holds.

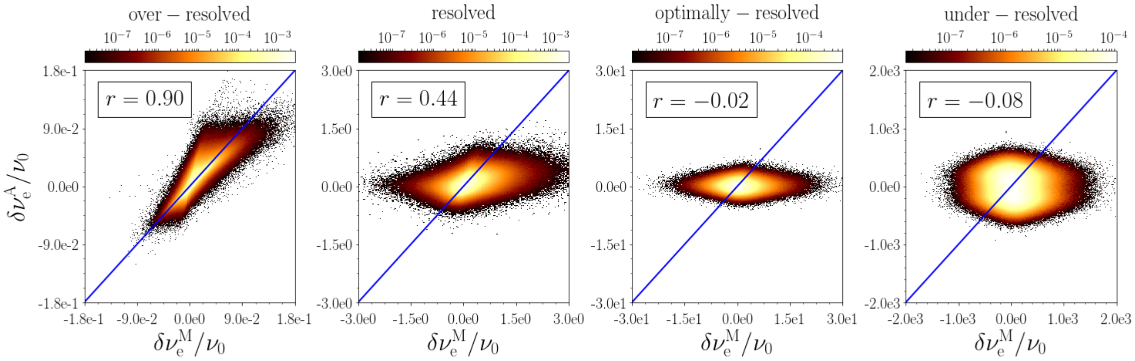


FIGURE 3.16. Joint PDF between the measured eddy viscosity $\delta\nu_e^M/\nu_0$ (see Eq. (3.1)) and the approximated eddy viscosity $\delta\nu_e^A/\nu_0$ (see Eq. (3.7)) expressed relatively to the input viscosity ν_0 for the three showcases: the over-resolved case, optimally-resolved case, and under-resolved case. The blue curve shows a perfect fit $\delta\nu_e^M = \delta\nu_e^A$, and r is Pearson's correlation coefficient.

As in the 2D HIT case, we can observe on the PDF of the measure and approximated eddy viscosity in Fig. 5.3 that for simulations of Reynolds number below the one of the optimally-resolved simulation, only the measured turbulent viscosity $\delta\nu_e^M$ is positively skewed, making the ELBM implicit effective viscosity higher than the input one. The approximated eddy viscosity $\delta\nu_e^A$ seem to not be able to approximate the dissipative properties of the measure eddy viscosity.

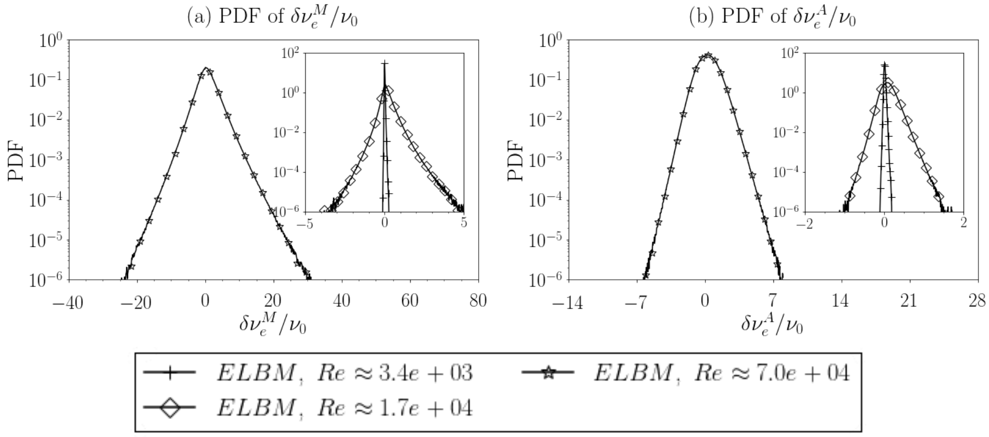


FIGURE 3.17. PDF of the measured eddy viscosity (pannel (a)) $\delta\nu_e^M/\nu_0$ (see Eq. (3.1)) and the approximated eddy viscosity (pannel (b)) $\delta\nu_e^A/\nu_0$ (see Eq. (3.7)) expressed relatively to the input viscosity ν_0 at different reynolds numbers

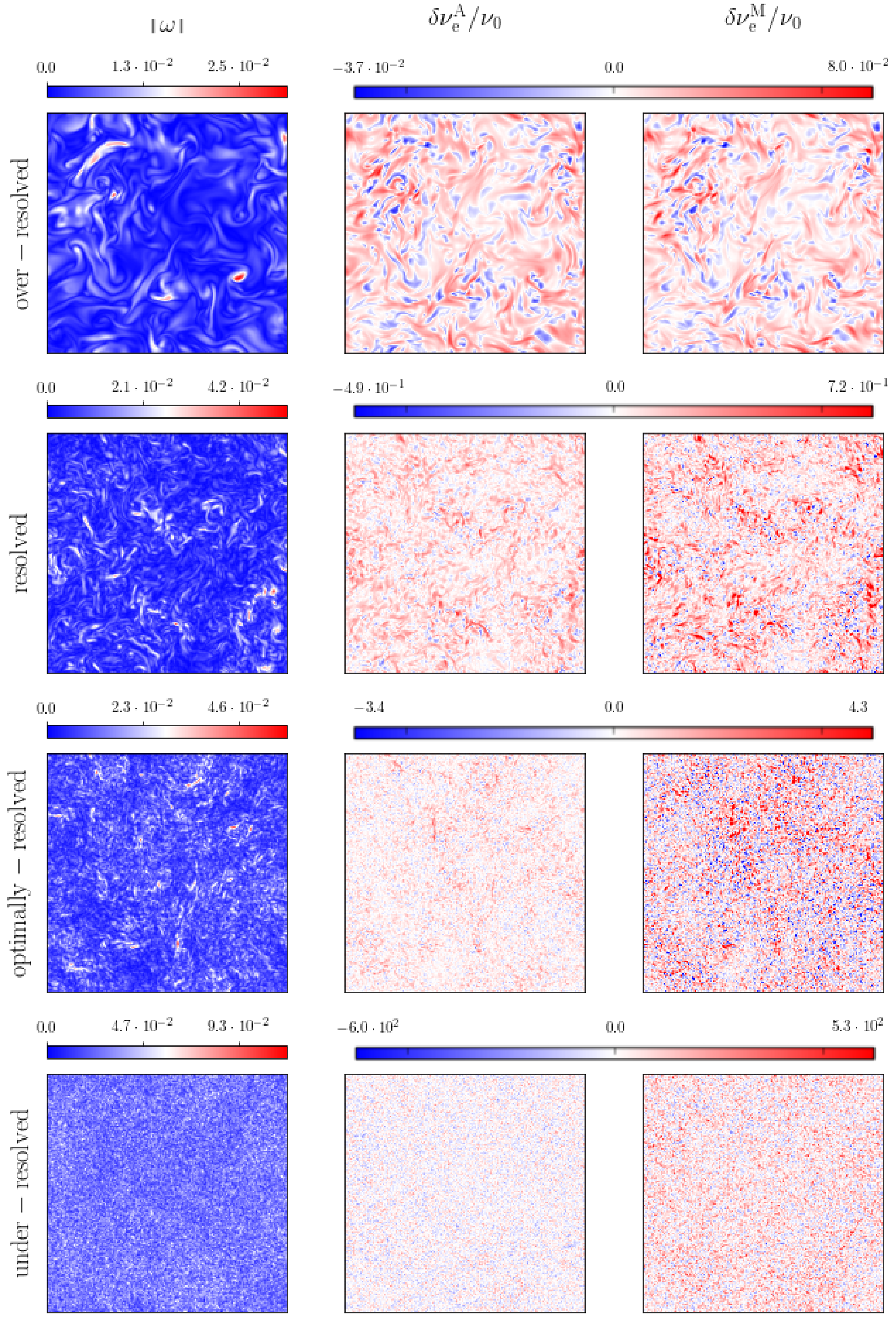


FIGURE 3.18. Snapshot of the vorticity $\|\omega| = |\nabla \times \mathbf{u}\|$, of the measured eddy viscosity $\delta\nu_e^M/\nu_0$ (see Eq. (3.1)) and the approximated eddy viscosity $\delta\nu_e^A/\nu_0$ (see Eq. (3.7)) expressed relatively to the input viscosity ν_0 for the four showcases: the over-resolved, resolved, ptimally-resolved, and under-resolved cases.

The snapshot Fig. 3.18 gives a valuable insight on the spatial correlation between the measure eddy viscosity $\delta\nu_e^M$ and the approximated one $\delta\nu_e^A$. The vorticity ω is also plotted as a reference. We observe that the eddy viscosity is a small scale quantity and the spoiling of the spatial correlation between $\delta\nu_e^M$ and $\delta\nu_e^A$ with the increase of the Reynolds number is again highlighted.

3.4 Concluding remarks

In this work, we have presented a detailed numerical study of the implicit SGS model stemming from ELBM in the context of both forced 2D and 3D HIT, assuming *a-priori* that ELBM can be macroscopically described as a LES eddy viscosity model. We conducted LBGK simulations at decreasing input relaxation time τ_0 until the lowest relaxation time ensuring stability and then ELBM simulations of further decreasing relaxation times (*i.e.* gradually increasing Reynolds numbers).

Firstly, we have applied a systematic statistical hydrodynamics recovery accuracy tool to the performed simulations. This tool rests on the measurement of each term of the kinetic energy balance averaged over random sub-volumes for different configurations in the statistically stationary domain. The generated hydrodynamics recovery is then assessed through the definition of the balancing errors and its calculations across a large number of sub-volumes of a wide range of sizes. In the case of 3D HIT simulations, we observed the presence of numerical dissipation. When we rescaled appropriately the input viscosity in the energy balance, we obtained results similar to the 2D HIT simulations case. Using this approach, we have been able to observe that the ELBM hydrodynamics recovery accuracy is not maintained at the order of magnitude of LBGK. Although, it seems that it is able to maintain a range of validity as a turbulence model going up to 20 times the Reynolds number of the last stable LBGK simulation, while the mean balancing error increase by one order of magnitude. Above this value, we have observed that ELBM still ensure stability without preserving the eddy viscosity-based hydrodynamics.

Secondly, we numerically checked the approximated eddy viscosity for the conducted simulations. The approximation appeared to be very good for low Reynolds numbers, with a Pearson's correlation coefficient of about 0.9 in an over-resolved setting both in 2D and 3D simulations of HIT. However, it quickly worsens as the Reynolds number of the simulation increases and, even in the over-resolved case, the joint PDF highlights the presence of two branches, one of them showing a very good agreement between the measured and approximated viscosity while the other is not properly understood at this stage. Besides, only the measured turbulent viscosity $\delta\nu_e^M$ and not its approximation

$\delta\nu_e^A$ have PDF that exhibits positive skewness. This means that $\delta\nu_e^A$ does not recover well the dissipative characteristics of $\delta\nu_e^M$.

CHAPTER 4

STUDY OF THE ELBM IMPLICIT SGS MODEL AT THE MACROSCALE

In this chapter, we focus on unrolling and numerically checking the derivation of the implicit sub-grid scale model implied by ELBM at the macroscopic scale. We first present a set of decaying 2D Homogeneous Isotropic Turbulence (HIT) simulations conducted using lattices of different levels of isotropy in section 4.1. Then, in Section 4.2, we numerically check step-by-step each assumption made in the derivation of the macroscopic approximation of the entropic parameter α. In section 4.3, we discuss the macroscopic momentum equation recovered by ELBM and the validity of eddy viscosities SGS model within LBM. Some concluding remarks follows in Section 4.4.

4.1 Simulations of 2D decaying homogeneous isotropic turbulence with ELBM

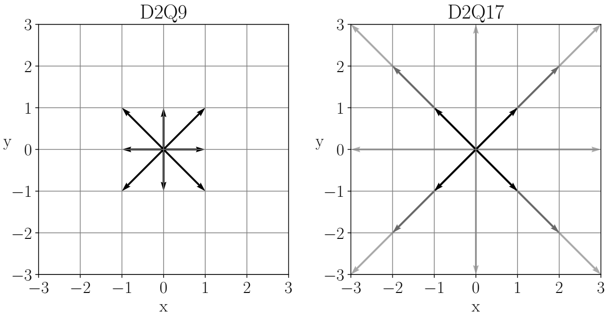


FIGURE 4.1. Lattices used to simulate 2D HIT. D2Q9 and D2Q13 have 4th order isotropy while D2Q17 and D2Q21 have 6th order isotropy.

The derivation of the macroscopic eddy viscosity is not valid in general for forced flows (unless one can further assume $\|\boldsymbol{F}\| \ll 1$). To numerically validate it, we, therefore, work with decaying HIT flows. We consider decaying flows from previously forced 2D HIT configuration using two lattices: a 4th order isotropic lattice, the D2Q9, and a 6th order isotropic lattice, the D2Q17 (See stencils on Fig. 6.1). We first conduct simulations of forced HIT flows using a spectral forcing with an amplitude F_0^T to trigger 2D HIT. Energy is injected in a shell of (dimensionless) wavenumbers $\boldsymbol{k}$ of magnitude from 5 to 7 with a constant phase ϕ. The forcing is written in a stream-function formulation to ensure that it does not generate any incompressibility. As 2D turbulence is characterized by the presence of a backward energy cascade [66, 14], we introduce a spectral forcing with an amplitude F_0^R to damp large-scale energy ($|\boldsymbol{k}| \leq 2$) to maintain the Mach Number under control and ensure a quality recovery of the NSE (see Chapter 2). We enforce the resulting forcings using the exact-difference method forcing scheme [61]. The parameters on Table 4.1 of the two forced simulations have been chosen iteratively to obtain, in the statistically stationary regime an inertial range exhibiting an energy spectrum with a k^{-3} slope.

Table 4.1. Parameters of the conducted forced 2D HIT simulations.

Lattice	τ_0	$\boldsymbol{F}_0^T$	$\boldsymbol{F}_0^R$
D2Q9	0.50001	$1.0e-06$	$5.0e-04$
D2Q17	0.500001	$2.2e-07$	$1.1e-04$

At $t = t_0$, we turn off the forcings and let the turbulent flow simulations decay. For each simulation we will follow the decay around three times t_1, t_2, and t_3. On Fig. 4.2 and 4.3, we show the energy spectra and the Mach number evolution for the D2Q9 and the D2Q17 simulations respectively. On the energy spectra of the D2Q17 simulation, we observe nonphysical bumps at the smaller scales, especially for the D2Q17 simulation. This is most likely linked to spurious velocities effects due to the multi-speed characteristic of the D2Q17 stencil (see stencil on Fig. 6.1), as described in [68]. Moreover, for both D2Q9 and D2Q17, it seems that the smallest scales are not properly dissipated during the decaying, which poses the question of the differentiability of the flow. For both simulations, we have $Ma = \frac{U}{c_s} \ll 1$, with U the Root-Mean Squared velocity, which ensures that we fall within the low Mach number approximation required during the C-E expansion to recover N-S equations.

4.2 Derivation and numerical check of the macroscopic expression of α

In this section, we follow the steps of Malaspinas *et. al* [2] to conduct a C-E expansion of the entropic parameter α, but we use our own C-E formalism (see Chapter 1.3.4). The parameter α is calculated at each lattice cell, at each time step as the solution of the entropic step equation Eq. (1.53), which fully developed yields:

$$\sum_{\ell=0}^{q-1}(f_\ell - \alpha f_\ell^{neq}) \ln\left(\frac{f_\ell - \alpha f_\ell^{neq}}{t_\ell}\right) - \sum_{\ell=0}^{q-1} f_\ell \ln\left(\frac{f_\ell}{t_\ell}\right) = 0 \tag{4.1}$$

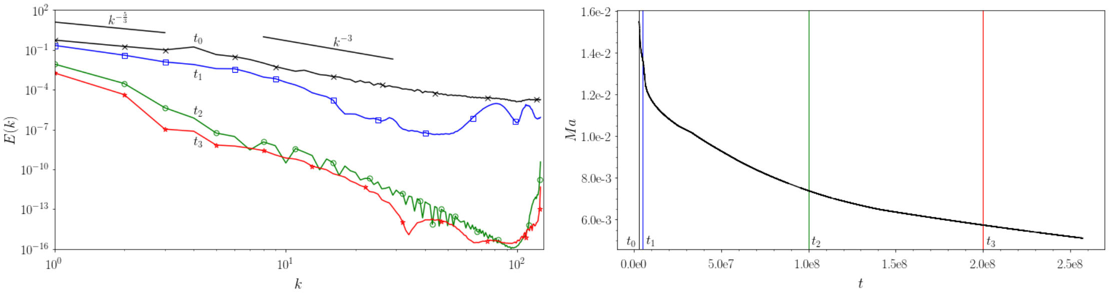


FIGURE 4.2. D2Q9: On the left panel, energy spectrum and Mach number evolution of the decaying ELBM simulation. The three spectrum corresponds to the studied times t_1, t_2, and t_3 highlighted in the corresponding colored area of the mach number evolution shown on the right panel.

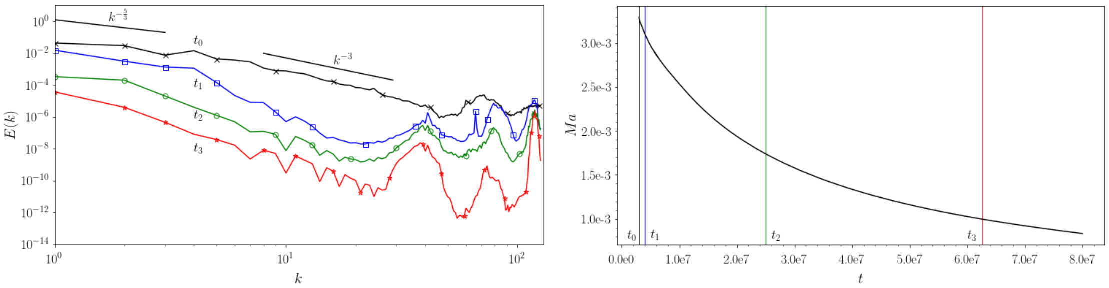


FIGURE 4.3. D2Q17: On the left panel, energy spectrum and Mach number evolution of the decaying ELBM simulation. The three spectrum corresponds to the studied times t_1, t_2, and t_3 highlighted in the corresponding colored area of the mach number evolution shown on the right panel.

This is solved using Newton-Raphson's (N-R) algorithm with a tolerance of 10^{-8}. If a solution is not found within 1000 iterations, then $\alpha = 2$ is returned. The ELBM community has been limiting the computational expense induced by the entropic step by returning $\alpha = 2$ whenever the deviation of f to f^{eq} is relatively small [3]. As we are interested in the solution of Eq. (4.1), we do not use this deviation criteria and we bear the full computational cost of the ELBM. Fig.4.4 and Fig. 4.5 show a joint PDF between the number of N-R iterations and the value of α for the D2Q9 and the D2Q17 simulation respectively.

Following the idea of [2] and our formalism for C-E expansions presented in chapter 1.3.4, the entropic parameter α itself admits an expansion in Δ_t as a function of f and f^{eq}, which themselves have expansions in Δ_t. Using an anologous formalism than for f Eq. (1.42), we write

$$\alpha = 2(1 + \Delta_t \phi_\alpha), \tag{4.2}$$

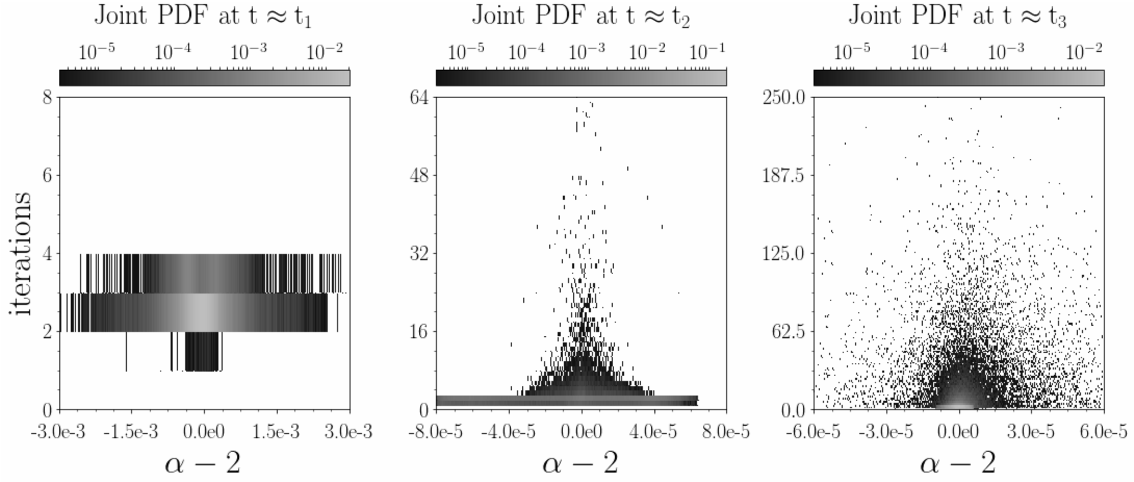


FIGURE 4.4. D2Q9: Joint PDF between α and the number of iteration of N-R algorithm. Each column corresponds to one of the three studied times of the decaying t_1, t_2, and t_3.

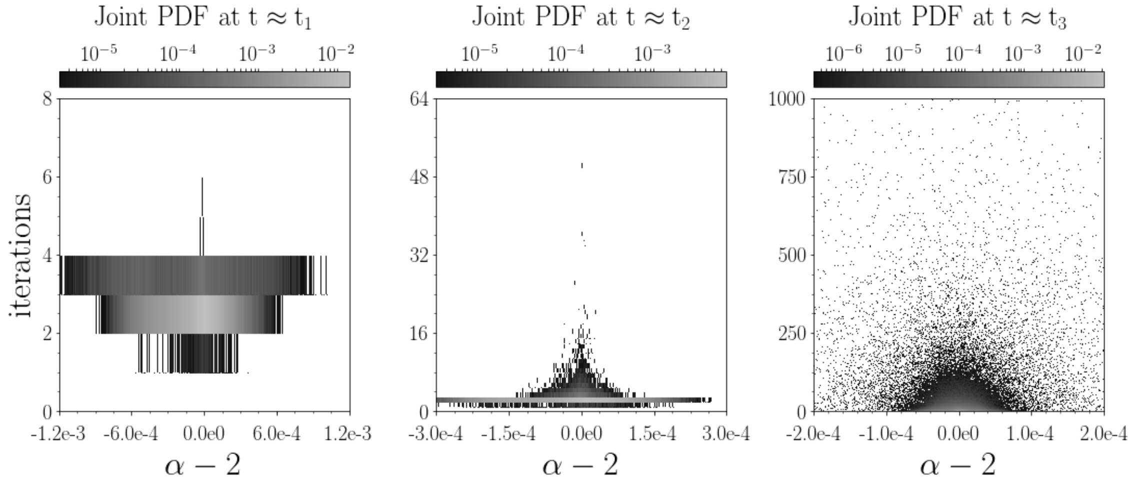


FIGURE 4.5. D2Q17: Joint PDF between α and the number of iteration of N-R algorithm. Each column corresponds to one of the three studied times of the decaying t_1, t_2, and t_3.

with $\phi_\alpha = \mathcal{O}(1)$ for $\Delta_t \to 0$. This implies that in the limit of fine lattice resolution, we have $\alpha \to 2$. Indeed, Fig. 4.6 as well as previous simulations confirms that α remains close to its LBGK value $\alpha_{LBGK} = 2$.

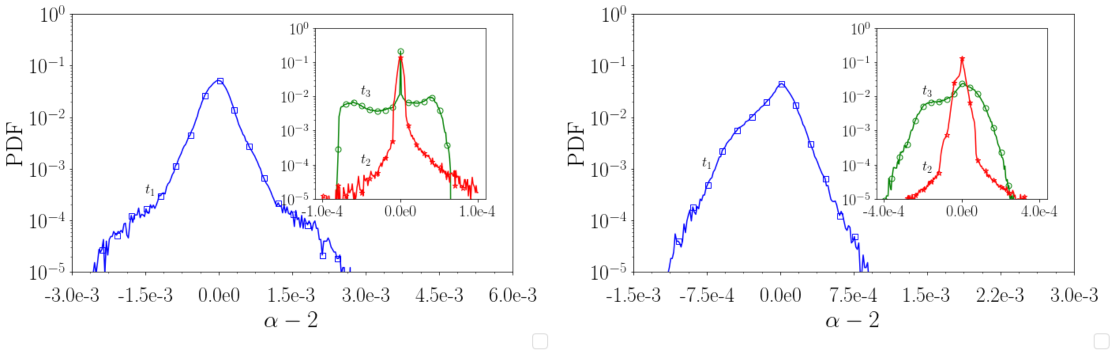


FIGURE 4.6. PDF of α for the D2Q9 (left panel) and the D2Q17 (right panel) simulations shown around the three different times t_1, t_2, and t_3.

4.2.1 Step 1: Solving the taylor expanded entropic step

In order to estimate the zeroth order of ϕ_α, we inject Eq. (4.2) in Eq. (4.1) and expand in Δ_t, to get

$$(2-\alpha)\sum_\ell f_\ell^{\mathrm{eq}}\phi_\ell^2 - \Delta_t\left(1-\alpha+\frac{\alpha^2}{3}\right)\sum_\ell f_\ell^{\mathrm{eq}}\phi_\ell^3 = \mathcal{O}(\Delta_t^2). \tag{4.3}$$

Replacing Eq. (4.2) in Eq. (4.3) and solving for ϕ_α at the zeroth order in Δ_t, we obtain

$$\phi_\alpha = -\frac{1}{6}\frac{\sum_\ell f_\ell^{\mathrm{eq}}\phi_\ell^3}{\sum_\ell f_\ell^{\mathrm{eq}}\phi_\ell^2} + \mathcal{O}(\Delta_t), \tag{4.4}$$

leading to the approximation of the value of α at the first order in Δ_t as an outcome of this first step

$$\alpha_1 = 2 - \frac{\Delta_t}{3}\frac{\sum_\ell f_\ell^{\mathrm{eq}}\phi_\ell^3}{\sum_\ell f_\ell^{\mathrm{eq}}\phi_\ell^2} + \mathcal{O}(\Delta_t^2) = 2 - \frac{1}{3}\frac{T_3}{T_2} + \mathcal{O}(\Delta_t^2), \tag{4.5}$$

where $T_2 = \sum_\ell f_\ell^{\mathrm{eq}}(\Delta_t\phi_\ell)^2$ and $T_3 = \sum_\ell f_\ell^{\mathrm{eq}}(\Delta_t\phi_\ell)^3$. If $T_2 = 0$, we have $\phi_\ell = 0$ and therefore $\mathrm{f} = \mathrm{f}^{eq}$. It is worth pointing out that this asymptotic expansion for α has long been used in the literature. In Ref. [69], the authors have derived an equivalent expression and suggest that it can be used instead of N-R to reduce the computational expense of ELBM.

We show on Fig. 4.7 and on Fig. 4.8, the summary of the numerical check for step 1 for the D2Q9 and the D2Q17 simulations respectively. On each figure, the first line corresponds to the joint PDF between α and α_1. We observe as the flow decays, that the perfect matching for time around t_1 and t_2 is spoiled by a vertical branch around $t = t_3$ corresponding to $\alpha_1 = 0$. This is due to the fact that, the approximated solution $\alpha_1 = 2 - \frac{T_3}{3T_2}$ will reach $\alpha_1 = 2$ when $T_3 = 0$ without the N-R algorithm returning $\alpha = 2$. Indeed if $T_3 = 0$, we don't necessarily have $T_2 = 0$ and thus we are not in the limit of a fluid at equilibium, $\mathrm{f} = \mathrm{f}^{eq}$.

On the second line of Figs 4.7 and 4.8, we plot the joint PDF between α and T_2. We observe that as $t \to t_3$, T_2 gets closer to 0, but α does not necesserally tend toward its equilibrium value $\alpha_{LBGK} = 2$. Therefore, we filter *a posteriori* data for which T_2 is inferior to a threshold of 10^{-10} for the D2Q9 simulation and 10^{-11} for the D2Q17 simulation and we plot again the joint PDF between α and α_1 on the last line of Figs 4.7 and 4.8. We observe that this filtering operation removes the vertical branch observed for $t \approx t_3$.

4.2.2 Step 2: Approximating the non-equilibrium distribution by the regularized distribution

From the full second order C-E expansion provided in Appendix B and Eq. (B.4) in particular, we can write

$$\phi_\ell = -\frac{1}{2\beta} D_\ell \ln f_\ell^{\mathrm{eq}} + \mathcal{O}(\Delta_t), \tag{4.6}$$

and hence

$$\phi_\alpha = \frac{1}{12\beta} \frac{\sum_\ell f_\ell^{\mathrm{eq}} \left(D_\ell \ln f_\ell^{\mathrm{eq}}\right)^3}{\sum_\ell f_\ell^{\mathrm{eq}} \left(D_\ell \ln f_\ell^{\mathrm{eq}}\right)^2} + \mathcal{O}(\Delta_t). \tag{4.7}$$

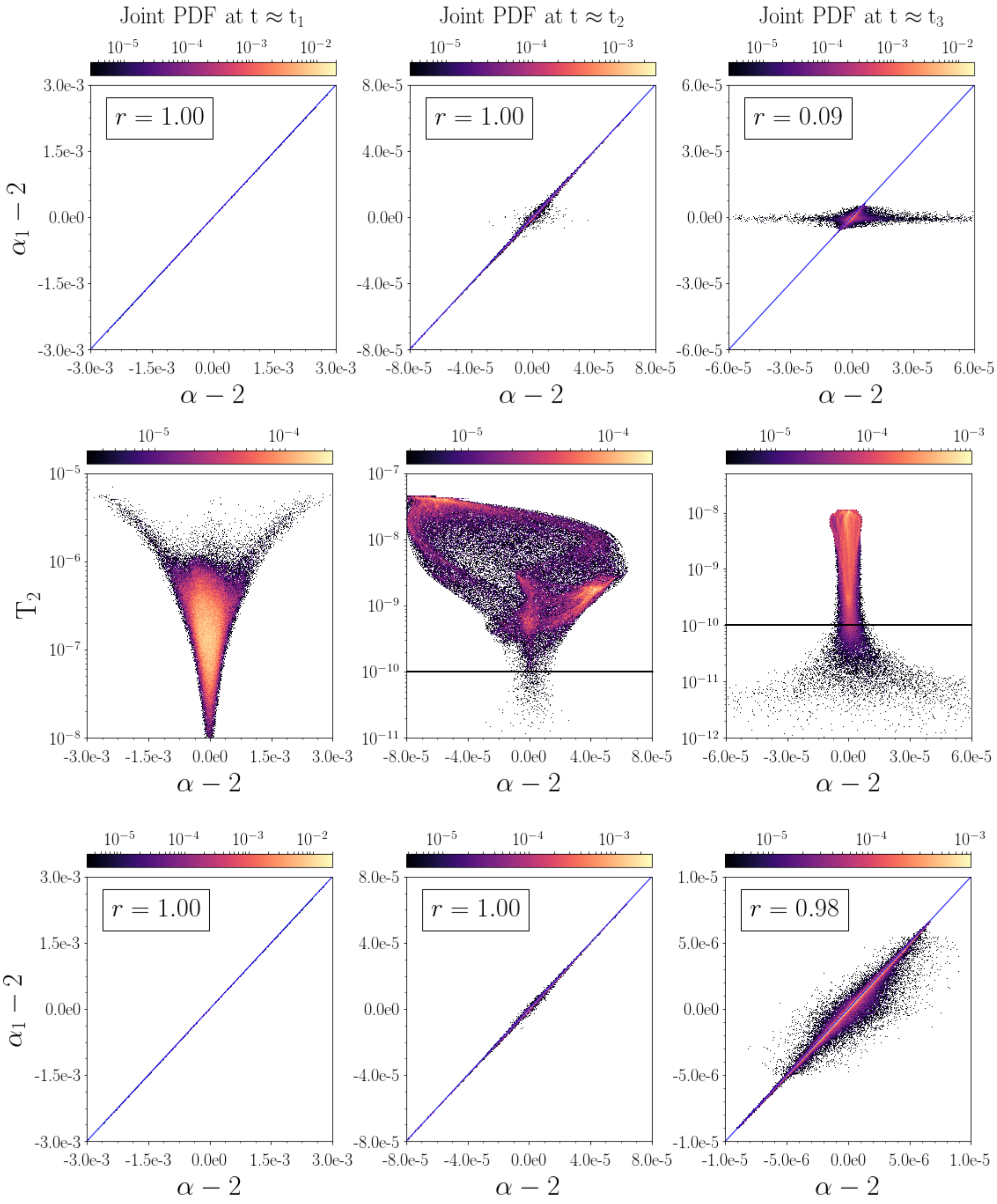


FIGURE 4.7. D2Q9: The first line shows the joint PDF between α_1 (Eq.(4.5)) and α (calculated using Newton Raphson). The second line shows the joint PDF between T_2 and α with the horizontal black line highlighting the thereshold $T_2 = 10^{-10}$. The last line shows the joint PDF between α and α_1 for data corresponding to $T_2 > 10^{-10}$. Each column corresponds to one of the three studied times of the decaying t_1, t_2, and t_3. The blue curve shows a perfect matching while r is the Pearson's correlation coefficient.

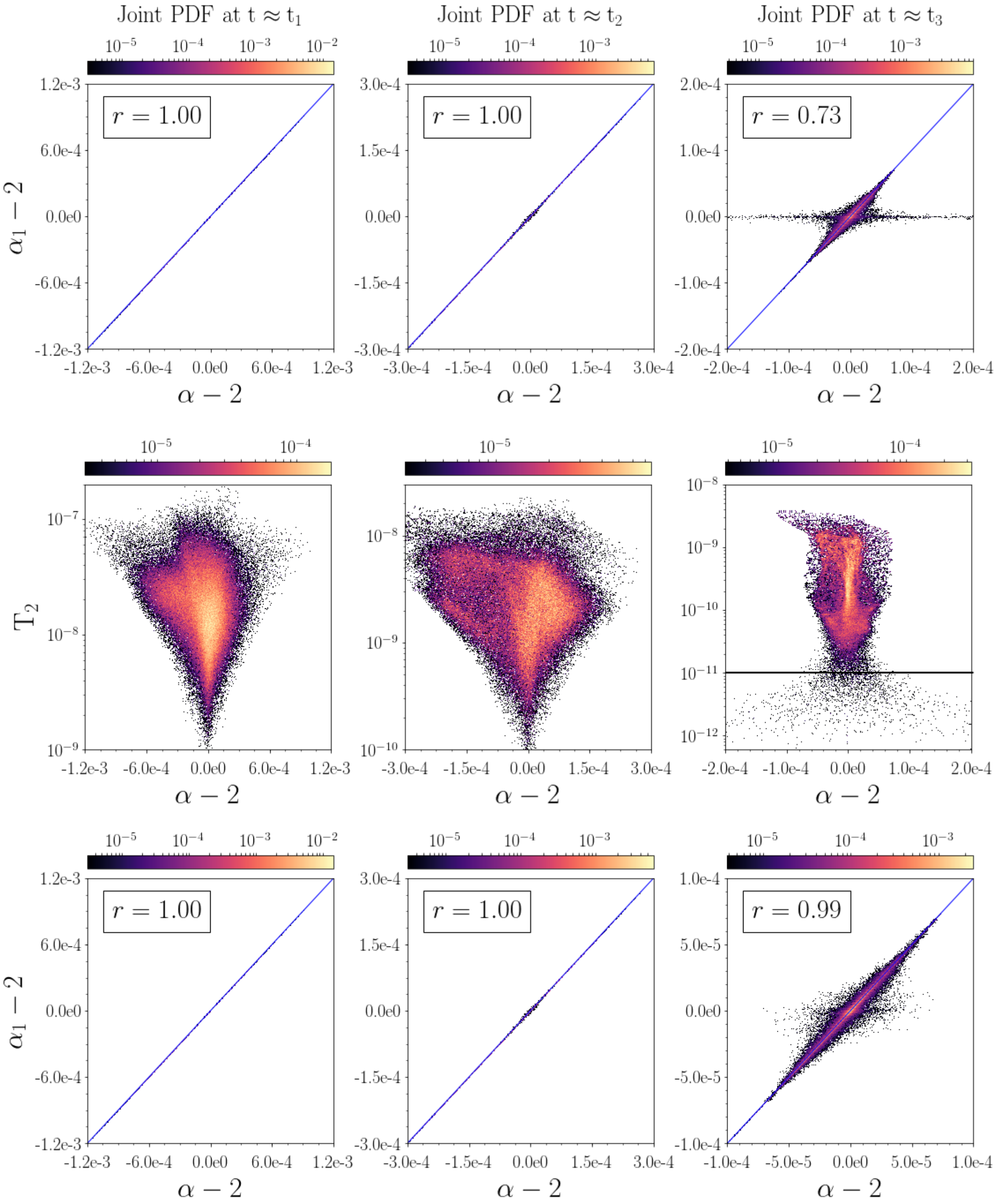


FIGURE 4.8. D2Q17: The first line shows the joint PDF between α_1 (Eq.(4.5)) and α (calculated using Newton Raphson). The second line shows the joint PDF between T_2 and α with the horizontal black line highlighting the thereshold $T_2 = 10^{-11}$. The last line shows the joint PDF between α and α_1 for data corresponding to $T_2 > 10^{-10}$. Each column corresponds to one of the three studied times of the decaying t_1, t_2, and t_3. The blue curve shows a perfect matching while r is the Pearson's correlation coefficient.

From the expression of the equilibrium distribution Eq. (1.39), we have

$$
\begin{aligned}
D_\ell f_\ell^{\mathrm{eq}} &= t_\ell \left[\partial_t \rho - \frac{\partial_t(\rho|\boldsymbol{u}|^2)}{2c_s^2}\right] \\
&\quad + t_\ell \left[\frac{\partial_t(\rho \boldsymbol{u})}{c_s^2} + \boldsymbol{\nabla}\rho - \frac{\boldsymbol{\nabla}(\rho|\boldsymbol{u}|^2)}{2c_s^2} - \frac{\partial_t(\rho|\boldsymbol{u}|^2\boldsymbol{u})}{2c_s^4}\right] \cdot \boldsymbol{c}_\ell \\
&\quad + t_\ell \left[\frac{\partial_t(\rho \boldsymbol{u}\boldsymbol{u})}{2c_s^4} + \frac{\boldsymbol{\nabla}(\rho\boldsymbol{u})}{c_s^2} - \frac{\boldsymbol{\nabla}(\rho|\boldsymbol{u}|^2\boldsymbol{u})}{2c_s^4}\right] : \boldsymbol{c}_\ell\boldsymbol{c}_\ell \\
&\quad + t_\ell \left[\frac{\boldsymbol{\nabla}(\rho\boldsymbol{u}\boldsymbol{u})}{2c_s^4} + \frac{\partial_t(\rho\boldsymbol{u}\boldsymbol{u}\boldsymbol{u})}{6c_s^6}\right] :\cdot\, \boldsymbol{c}_\ell\boldsymbol{c}_\ell\boldsymbol{c}_\ell \\
&\quad + t_\ell \frac{\boldsymbol{\nabla}(\rho\boldsymbol{u}\boldsymbol{u}\boldsymbol{u})}{6c_s^6} :: \boldsymbol{c}_\ell\boldsymbol{c}_\ell\boldsymbol{c}_\ell\boldsymbol{c}_\ell + \mathcal{O}(\Delta_t) \\
&= t_\ell \left[\frac{\boldsymbol{\nabla}\cdot\big(\rho|\boldsymbol{u}|^2\boldsymbol{u}\big)}{2c_s^2} - \rho\boldsymbol{\nabla}\cdot\boldsymbol{u}\right] \\
&\quad + t_\ell \left[-\frac{\rho\boldsymbol{u}\boldsymbol{\nabla}\cdot\boldsymbol{u}}{c_s^2} - \frac{2\rho\boldsymbol{u}\cdot\boldsymbol{\nabla}\boldsymbol{u}}{c_s^2} + \frac{\boldsymbol{\nabla}\cdot\big(\rho|\boldsymbol{u}|^2\boldsymbol{u}\boldsymbol{u}\big)}{2c_s^4}\right] \cdot \boldsymbol{c}_\ell \\
&\quad + t_\ell \left[\frac{\rho\boldsymbol{\nabla}\boldsymbol{u}}{c_s^2} - \frac{\boldsymbol{\nabla}\cdot(\rho\boldsymbol{u}\boldsymbol{u}\boldsymbol{u})}{2c_s^4} - \frac{\boldsymbol{\nabla}(\rho|\boldsymbol{u}|^2\boldsymbol{u})}{2c_s^4}\right] : \boldsymbol{c}_\ell\boldsymbol{c}_\ell \\
&\quad + t_\ell \left[\frac{\rho\boldsymbol{u}\boldsymbol{\nabla}\boldsymbol{u}}{c_s^4} - \frac{\boldsymbol{\nabla}\cdot(\rho\boldsymbol{u}\boldsymbol{u}\boldsymbol{u}\boldsymbol{u})}{6c_s^6}\right] :\cdot\, \boldsymbol{c}_\ell\boldsymbol{c}_\ell\boldsymbol{c}_\ell \\
&\quad + t_\ell \frac{\boldsymbol{\nabla}(\rho\boldsymbol{u}\boldsymbol{u}\boldsymbol{u})}{6c_s^6} :: \boldsymbol{c}_\ell\boldsymbol{c}_\ell\boldsymbol{c}_\ell\boldsymbol{c}_\ell + \mathcal{O}(\Delta_t) \\
&= t_\ell\rho\left(-\boldsymbol{\nabla}\cdot\boldsymbol{u} - \left[\frac{\boldsymbol{u}\boldsymbol{\nabla}\cdot\boldsymbol{u}}{c_s^2} + \frac{2\rho\boldsymbol{u}\cdot\boldsymbol{\nabla}\boldsymbol{u}}{c_s^2}\right]\cdot\boldsymbol{c}_\ell + \frac{\boldsymbol{\nabla}\boldsymbol{u}}{c_s^2} : \boldsymbol{c}_\ell\boldsymbol{c}_\ell + \frac{\boldsymbol{u}\boldsymbol{\nabla}\boldsymbol{u}}{c_s^4} :\cdot\, \boldsymbol{c}_\ell\boldsymbol{c}_\ell\boldsymbol{c}_\ell\right) \\
&\quad + \mathcal{O}(\nabla u^3) + \mathcal{O}(\Delta_t),
\end{aligned}
\tag{4.8}
$$

As in Ref. [2], we can introduce the dissipative part of the momentum tensor $\boldsymbol{\Pi}^{(1)}$ into Eq. (4.12). It is directly linked to the strain rate tensor [70], $\mathbf{S} = \frac{1}{2}[\boldsymbol{\nabla}\boldsymbol{u} + (\boldsymbol{\nabla}\boldsymbol{u})^T]$:

$$
\boldsymbol{\Pi}^{(1)} = -2\rho c_s^2 \frac{1}{2\beta}\mathbf{S} + \mathcal{O}(\Delta_t). \tag{4.9}
$$

We have

$$
\begin{aligned}
\mathbf{Q}_\ell : \boldsymbol{\Pi}^{(1)} &= -\frac{\rho c_s^2}{\beta}\mathbf{Q}_\ell : \mathbf{S} + \mathcal{O}(\Delta_t) \\
&= -\frac{\rho c_s^4}{\beta}(-\boldsymbol{\nabla}\cdot\boldsymbol{u} + \frac{\boldsymbol{\nabla}\boldsymbol{u}}{c_s^2} : \boldsymbol{c}_\ell\boldsymbol{c}_\ell) + \mathcal{O}(\Delta_t)
\end{aligned}
\tag{4.10}
$$

with $\mathbf{Q}_\ell = \boldsymbol{c}_\ell \boldsymbol{c}_\ell - c_s^2 \mathbf{1}$. As a result, Eq. (4.8) becomes

$$D_\ell f_\ell^{\mathrm{eq}} = t_\ell \rho \left(-\frac{\beta}{\rho c_s^4} \mathbf{Q}_\ell : \mathbf{\Pi}^{(1)} - \left[\frac{\boldsymbol{u}\boldsymbol{\nabla}\cdot\boldsymbol{u}}{c_s^2} + \frac{2\rho\boldsymbol{u}\cdot\boldsymbol{\nabla}\boldsymbol{u}}{c_s^2} \right] \cdot \boldsymbol{c}_\ell + \frac{\boldsymbol{u}\boldsymbol{\nabla}\boldsymbol{u}}{c_s^4} \because \boldsymbol{c}_\ell\boldsymbol{c}_\ell\boldsymbol{c}_\ell \right) + \mathcal{O}(\nabla u^3) + \mathcal{O}(\Delta_t), \tag{4.11}$$

and hence, as $D_\ell \ln f_\ell^{\mathrm{eq}} = D_\ell f_\ell^{\mathrm{eq}} / f_\ell^{\mathrm{eq}}$,

$$\begin{aligned}
D_\ell \ln f_\ell^{\mathrm{eq}} &= \frac{-\frac{\beta}{\rho c_s^4} \mathbf{Q}_\ell : \mathbf{\Pi}^{(1)} - \left[\frac{\boldsymbol{u}\boldsymbol{\nabla}\cdot\boldsymbol{u}}{c_s^2} + \frac{2\rho\boldsymbol{u}\cdot\boldsymbol{\nabla}\boldsymbol{u}}{c_s^2} \right] \cdot \boldsymbol{c}_\ell + \frac{\boldsymbol{u}\boldsymbol{\nabla}\boldsymbol{u}}{c_s^4} \because \boldsymbol{c}_\ell\boldsymbol{c}_\ell\boldsymbol{c}_\ell + \mathcal{O}(\Delta_t) + \mathcal{O}(\nabla u^3)}{1 + \frac{\boldsymbol{u}\cdot\boldsymbol{c}_\ell}{c_s^2} + \frac{\boldsymbol{u}\boldsymbol{u}:\boldsymbol{c}_\ell\boldsymbol{c}_\ell - c_s^2 |\boldsymbol{u}|^2}{2c_s^4} + \frac{\boldsymbol{u}\boldsymbol{u}\boldsymbol{u}\because \boldsymbol{c}_\ell\boldsymbol{c}_\ell\boldsymbol{c}_\ell - 3c_s^2|\boldsymbol{u}|^2\boldsymbol{u}\cdot\boldsymbol{c}_\ell}{6c_s^6}} \\
&= -\frac{\beta}{\rho c_s^4} \mathbf{Q}_\ell : \mathbf{\Pi}^{(1)} - \left[\frac{\boldsymbol{u}\boldsymbol{\nabla}\cdot\boldsymbol{u}}{c_s^2} + \frac{2\rho\boldsymbol{u}\cdot\boldsymbol{\nabla}\boldsymbol{u}}{c_s^2} \right] \cdot \boldsymbol{c}_\ell + \frac{\boldsymbol{u}\boldsymbol{\nabla}\boldsymbol{u}}{c_s^4} \because \boldsymbol{c}_\ell\boldsymbol{c}_\ell\boldsymbol{c}_\ell \\
&\quad - \frac{\boldsymbol{u}\cdot\boldsymbol{c}_\ell}{c_s^2} \left(-\frac{\beta}{\rho c_s^4} \mathbf{Q}_\ell : \mathbf{\Pi}^{(1)} \right) + \mathcal{O}(\nabla u^3) + \mathcal{O}(\Delta_t) \\
&= -\frac{\beta}{\rho c_s^4} \mathbf{Q}_\ell : \mathbf{\Pi}^{(1)} - \left[\frac{\boldsymbol{u}\boldsymbol{\nabla}\cdot\boldsymbol{u}}{c_s^2} + \frac{2\rho\boldsymbol{u}\cdot\boldsymbol{\nabla}\boldsymbol{u}}{c_s^2} \right] \cdot \boldsymbol{c}_\ell + \frac{\boldsymbol{u}\boldsymbol{\nabla}\boldsymbol{u}}{c_s^4} \because \boldsymbol{c}_\ell\boldsymbol{c}_\ell\boldsymbol{c}_\ell \\
&\quad - \frac{\boldsymbol{u}\cdot\boldsymbol{c}_\ell}{c_s^2} \left(-\boldsymbol{\nabla}\cdot\boldsymbol{u} + \frac{\boldsymbol{\nabla}\boldsymbol{u}}{c_s^2} : \boldsymbol{c}_\ell\boldsymbol{c}_\ell \right) + \mathcal{O}(\nabla u^3) + \mathcal{O}(\Delta_t) \\
&== -\frac{\beta}{\rho c_s^4} \mathbf{Q}_\ell : \mathbf{\Pi}^{(1)} - \frac{2\boldsymbol{u}\cdot\boldsymbol{\nabla}\boldsymbol{u}}{c_s^2} \cdot \boldsymbol{c}_\ell + \mathcal{O}(\nabla u^3) + \mathcal{O}(\Delta_t).
\end{aligned} \tag{4.12}$$

Consequently,

$$\begin{aligned}
(D_\ell \ln f_\ell^{\mathrm{eq}})^2 &= \left[-\frac{\beta}{\rho c_s^4} \mathbf{Q}_\ell : \mathbf{\Pi}^{(1)} - \frac{2\boldsymbol{u}\cdot\boldsymbol{\nabla}\boldsymbol{u}}{c_s^2} \cdot \boldsymbol{c}_\ell + \mathcal{O}(\nabla u^3) + \mathcal{O}(\Delta_t) \right]^2 \\
&= \frac{\beta^2}{\rho^2 c_s^8} (\mathbf{Q}_\ell : \mathbf{\Pi}^{(1)})^2 + \mathcal{O}(\nabla^2 u^3) + \mathcal{O}(\Delta_t)
\end{aligned} \tag{4.13}$$

and

$$\begin{aligned}
(D_\ell \ln f_\ell^{\mathrm{eq}})^3 &= -\left[-\frac{\beta}{\rho c_s^4} \mathbf{Q}_\ell : \mathbf{\Pi}^{(1)} - \frac{2\boldsymbol{u}\cdot\boldsymbol{\nabla}\boldsymbol{u}}{c_s^2} \cdot \boldsymbol{c}_\ell + \mathcal{O}(\nabla u^3) + \mathcal{O}(\Delta_t) \right]^3 \\
&= \frac{\beta^3}{\rho^3 c_s^{12}} (\mathbf{Q}_\ell : \mathbf{\Pi}^{(1)})^3 + \mathcal{O}(\nabla^3 u^4) + \mathcal{O}(\Delta_t),
\end{aligned} \tag{4.14}$$

as $\mathbf{Q}_\ell : \mathbf{\Pi}^{(1)} = \mathcal{O}(\nabla u)$. Notice that from all the terms in Eq. (4.11), the term in $\mathbf{Q}_\ell : \mathbf{\Pi}^{(1)}$, is the only one playing a role. This justifies the validity of the regularization approximation [70] made in Ref. [2], as we have

$$\mathrm{f}^{neq} = \mathrm{f}^{eq} \phi_\ell \Delta_t \approx -\frac{\Delta_t}{2\beta} D_\ell f_\ell^{eq} \approx \frac{t_\ell \Delta_t}{2c_s^4} \mathbf{Q}_\ell : \mathbf{\Pi}^{(1)} = \bar{\mathrm{f}}^{(1)} \tag{4.15}$$

It is important to keep in mind that this approximation is valid at $\mathcal{O}(\nabla u^3)$ (and $\mathcal{O}(\Delta_t)$). As the flow starts decaying from a highly turbulent configuration, which involves high velocity gradients, it is therefore expected that this approximation will get increasingly valid as turbulence decays. This will be discussed further in section 4.2.5.

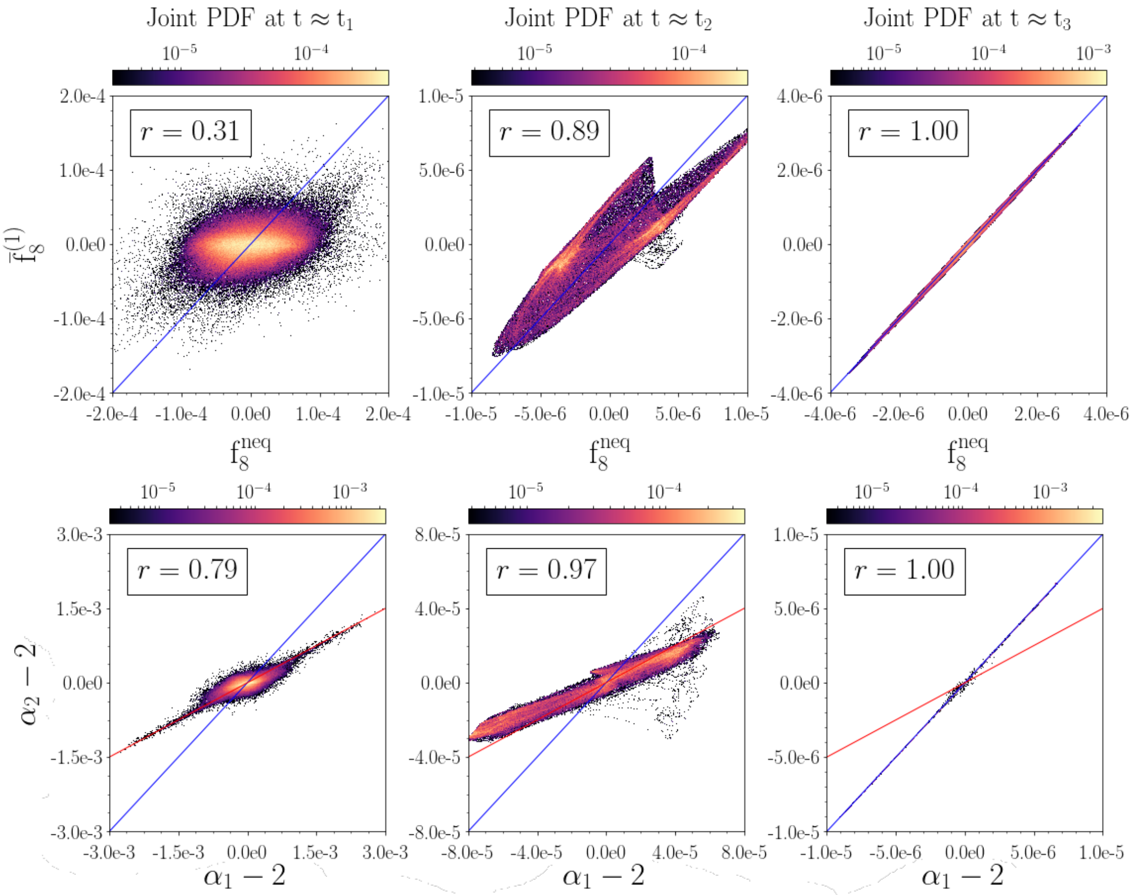


FIGURE 4.9. D2Q9: The first line shows the joint PDF between f_ℓ^{neq} and $\bar{f}_\ell^{(1)}$ (Eq. (4.15)) for $\ell = 8$. The second line shows the joint PDF between α_1 (Eq. (4.5)) and α_2 (Eq. (4.17)). Each column corresponds to one of the three studied times of the decaying t_1, t_2, and t_3. The blue curve shows a perfect matching, while the red curve shows the expected expression from ref [2]. r is the Pearson's correlation coefficient.

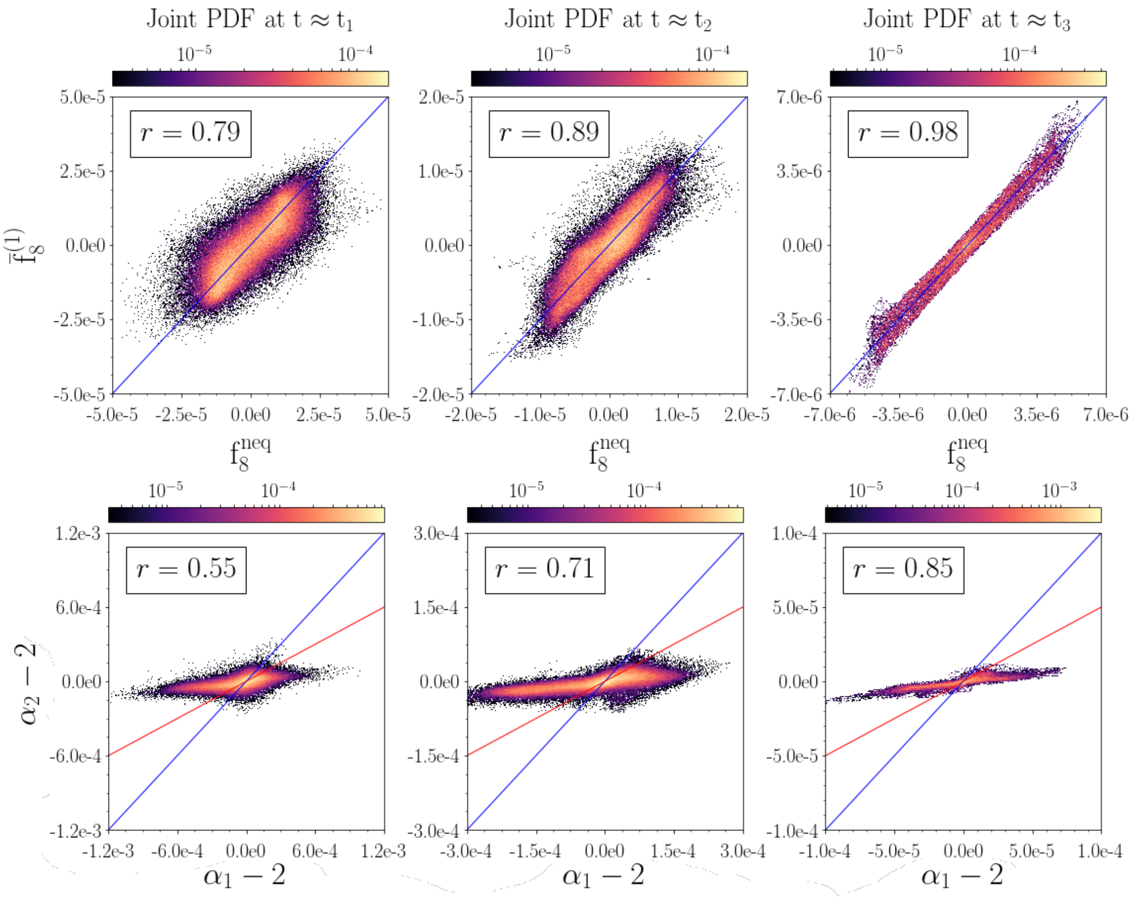


FIGURE 4.10. D2Q7: The first line shows the joint PDF between f_ℓ^{neq} and $\bar{f}_\ell^{(1)}$ (Eq. (4.15)) for $\ell = 8$. The second line shows the joint PDF between α_1 (Eq. (4.5)) and α_2 (Eq. (4.17)). Each column corresponds to one of the three studied times of the decaying t_1, t_2, and t_3. The blue curve shows a perfect matching, while the red curve shows the expected expression from ref [2]. r is the Pearson's correlation coefficient.

Finally, we obtain

$$\begin{aligned}\phi_\alpha &= -\frac{1}{12\rho c_s^4}\frac{\sum_\ell f_\ell^{\text{eq}}[(\mathbf{Q}_\ell : \mathbf{\Pi}^{(1)})^3 + \mathcal{O}(\nabla^3 u^4)]}{\sum_\ell f_\ell^{\text{eq}}[(\mathbf{Q}_\ell : \mathbf{\Pi}^{(1)})^2 + +\mathcal{O}(\nabla^2 u^3)]} + \mathcal{O}(\Delta_t) \\ &= -\frac{1}{12\rho c_s^4}\frac{\sum_\ell t_\ell(\mathbf{Q}_\ell : \mathbf{\Pi}^{(1)})^3}{\sum_\ell t_\ell(\mathbf{Q}_\ell : \mathbf{\Pi}^{(1)})^2} + \mathcal{O}(\nabla u^2) + \mathcal{O}(\Delta_t),\end{aligned} \tag{4.16}$$

giving as an outcome of step 2

$$\alpha_2 = -\frac{\Delta_t}{6\rho c_s^4}\frac{\sum_\ell t_\ell(\mathbf{Q}_\ell : \mathbf{\Pi}^{(1)})^3}{\sum_\ell t_\ell(\mathbf{Q}_\ell : \mathbf{\Pi}^{(1)})^2} + \mathcal{O}(\Delta_t \nabla u^2) + \mathcal{O}(\Delta_t^2). \tag{4.17}$$

This expression for α_2 differs from the one obtained by Malaspinas *et al.* by a factor 2 (Ref. [2] Eq. (18)).

In Figs. 4.9 and 4.10, we first show a check of the reguralized distribution approximation $\bar{f}_\ell^{(1)}$ is a good approximation of f_ℓ^{neq} at $\ell = 8$ for respectively the D2Q9 and D2Q17 simulations. In both cases, the agreement improves as time decays, to reach an excellent level at $t \approx t_3$, especially for the D2Q9 simulation. We see that due to the complex non-linear dependency of α_2 on $\mathrm{f}^{\mathrm{neq}}$, the agreement between $\mathrm{f}^{\mathrm{neq}}$ and $\bar{\mathrm{f}}^{(1)}$ needs to be perfect in order to have a good agreement between α_2 and α_3. This is only the case for D2Q9 at $t \approx t_3$. We also note that we are able to validate our expression for $\alpha_{@}$ Eq. (4.17) ash seen by comparing the red curve that represent the prediction made in Ref. [2], and the blue curve, representing the present result.

4.2.3 Step 3 - Using isotropy relations

We evaluate the sum on the numerator and on the denominator of α_2 Eq. (4.17) by using isotropy relations, while keeping in mind that $\mathbf{\Pi}^{(1)}$ is symmetric. Starting with the denominator, we can make use of the fourth order lattice isotropy relation, $\mathbf{\Delta}^{(4)} = \sum_\ell t_\ell \boldsymbol{c}_\ell \boldsymbol{c}_\ell \boldsymbol{c}_\ell \boldsymbol{c}_\ell = c_\mathrm{s}^4[\mathbf{11}]_3$, where the notation $([\mathbf{LM}]_3)_{ijkl} = L_{ij}M_{kl} + L_{ik}M_{jl} + L_{il}M_{kj}$, for $\mathbf{A}$ and $\mathbf{B}$, two symmetric second-order tensors denotes the fourth-order tensor created by taking the sum of the 3 fourth-order tensors with unique index combination of the two (symmetric) tensors. We have:

$$\begin{aligned}\sum_{\ell=0}^{q-1} t_\ell \left(\mathbf{Q}_\ell : \mathbf{\Pi}^{(1)}\right)^2 &= \mathbf{\Pi}^{(1)}\mathbf{\Pi}^{(1)} :: \sum_\ell t_\ell (\boldsymbol{c}_\ell \boldsymbol{c}_\ell - c_s^2 \mathbf{1})(\boldsymbol{c}_\ell \boldsymbol{c}_\ell - c_s^2 \mathbf{1}) \\ &= \mathbf{\Pi}^{(1)}\mathbf{\Pi}^{(1)} :: \sum_\ell t_\ell [\boldsymbol{c}_\ell \boldsymbol{c}_\ell \boldsymbol{c}_\ell \boldsymbol{c}_\ell - c_s^2(\boldsymbol{c}_\ell \boldsymbol{c}_\ell \mathbf{1} + \mathbf{1}\boldsymbol{c}_\ell \boldsymbol{c}_\ell) + c_s^4 \mathbf{11}] \\ = 2c_s^4 \,\mathrm{Tr}(\mathbf{\Pi}^{(1)}\mathbf{\Pi}^{(1)})\end{aligned} \tag{4.18}$$

In order to evaluate the numerator, we need to use a sixth order lattice isotropy relation $\mathbf{\Delta}^{(6)} = c_\mathrm{s}^6[\mathbf{111}]_{15}$. The notation $[\mathbf{LMN}]_{15}$, for $\mathbf{L}$, $\mathbf{M}$, and $\mathbf{N}$, ee symmetric second-order tensors, denotes the sixth-order tensor created by taking the sum of the 15 sixth-order tensors with unique index combination of the three (symmetric) tensors. This is not a requirement of C-E to recover the NSE and usual lattices such as D2Q9 do not possess this property, while D2Q17 does. Therefore we introduce the sixth-order anisotropic contributions $\boldsymbol{A}$ and write $\mathbf{\Delta}^{(6)} = c_\mathrm{s}^6[\mathbf{111}]_{15} + 6c_\mathrm{s}^6\mathbf{A}$. We obtain:

$$
\begin{aligned}
\sum_{\ell=0}^{q-1} t_\ell \left(\mathbf{Q}_\ell : \mathbf{\Pi}^{(1)}\right)^3 &= \mathbf{\Pi}^{(1)}\mathbf{\Pi}^{(1)}\mathbf{\Pi}^{(1)} ::: \sum_\ell t_\ell(\boldsymbol{c}_\ell\boldsymbol{c}_\ell - c_s^2\mathbf{1})(\boldsymbol{c}_\ell\boldsymbol{c}_\ell - c_s^2\mathbf{1})(\boldsymbol{c}_\ell\boldsymbol{c}_\ell - c_s^2\mathbf{1}) \\
&= \mathbf{\Pi}^{(1)}\mathbf{\Pi}^{(1)}\mathbf{\Pi}^{(1)} ::: \sum_\ell t_\ell[\boldsymbol{c}_\ell\boldsymbol{c}_\ell\boldsymbol{c}_\ell\boldsymbol{c}_\ell\boldsymbol{c}_\ell\boldsymbol{c}_\ell - c_s^2(\boldsymbol{c}_\ell\boldsymbol{c}_\ell\boldsymbol{c}_\ell\boldsymbol{c}_\ell\mathbf{1} + \boldsymbol{c}_\ell\boldsymbol{c}_\ell\mathbf{1}\boldsymbol{c}_\ell\boldsymbol{c}_\ell + \mathbf{1}\boldsymbol{c}_\ell\boldsymbol{c}_\ell\boldsymbol{c}_\ell\boldsymbol{c}_\ell) \\
&\quad + c_s^4(\boldsymbol{c}_\ell\boldsymbol{c}_\ell\mathbf{11} + \mathbf{1}\boldsymbol{c}_\ell\boldsymbol{c}_\ell\mathbf{1} + \mathbf{11}\boldsymbol{c}_\ell\boldsymbol{c}_\ell) - c_s^6\mathbf{111}] \\
&= c_s^6(8\,\mathrm{Tr}(\mathbf{\Pi}^{(1)}\mathbf{\Pi}^{(1)}\mathbf{\Pi}^{(1)}) + 6\,\mathbf{\Pi}^{(1)}\mathbf{\Pi}^{(1)}\mathbf{\Pi}^{(1)} ::: \mathbf{A})
\end{aligned}
\tag{4.19}
$$

As a result, ϕ_α yields

$$
\phi_\alpha = -\frac{1}{3\rho c_s^2}\frac{\mathrm{Tr}(\mathbf{\Pi}^{(1)}\mathbf{\Pi}^{(1)}\mathbf{\Pi}^{(1)})}{\mathrm{Tr}(\mathbf{\Pi}^{(1)}\mathbf{\Pi}^{(1)})} - \frac{1}{4\rho c_s^2}\mathbf{A} ::: \frac{\mathbf{\Pi}^{(1)}\mathbf{\Pi}^{(1)}\mathbf{\Pi}^{(1)}}{\mathrm{Tr}(\mathbf{\Pi}^{(1)}\mathbf{\Pi}^{(1)})} + \mathcal{O}(\nabla u^2) + \mathcal{O}(\Delta_t), \tag{4.20}
$$

and the outcome of step 3 reads:

$$
\alpha_3 = 2 - \frac{2\Delta_t}{3\rho c_s^2}\frac{\mathrm{Tr}(\mathbf{\Pi}^{(1)}\mathbf{\Pi}^{(1)}\mathbf{\Pi}^{(1)})}{\mathrm{Tr}(\mathbf{\Pi}^{(1)}\mathbf{\Pi}^{(1)})} - \frac{\Delta_t}{2\rho c_s^2}\mathbf{A} ::: \frac{\mathbf{\Pi}^{(1)}\mathbf{\Pi}^{(1)}\mathbf{\Pi}^{(1)}}{\mathrm{Tr}(\mathbf{\Pi}^{(1)}\mathbf{\Pi}^{(1)})} + \mathcal{O}(\Delta_t\nabla u^2) + \mathcal{O}(\Delta_t^2) \tag{4.21}
$$

We plot in Figs. 4.11 and 4.12 the summary of the numerical check corresponding to step 3 for the D2Q9 and D3Q17 simulations respectively. We first check whether the denominator relation Eq. (4.18) is checked. Because both lattices have 4th order isotropy properties, the matching is perfect. As for the numerator relation Eq. (4.19), we have again a perfect match for the D2Q17 simulation as it possesses 6th order isotropy properties which therefore leads to a perfect agreement between α_3 and α_4 on the last line of joint PDFs. The 6th order anisotropy arising for the D2Q9 is relatively small as the joint PDF between the LHS and RHS of Eq. (4.19) falls very close to the blue line highlighting a very good agreement. However, the impact of anisotropy is amplified in the formulation of α_4 and results in a second anisotropic branch on the joint PDF between α_4 and α_3.

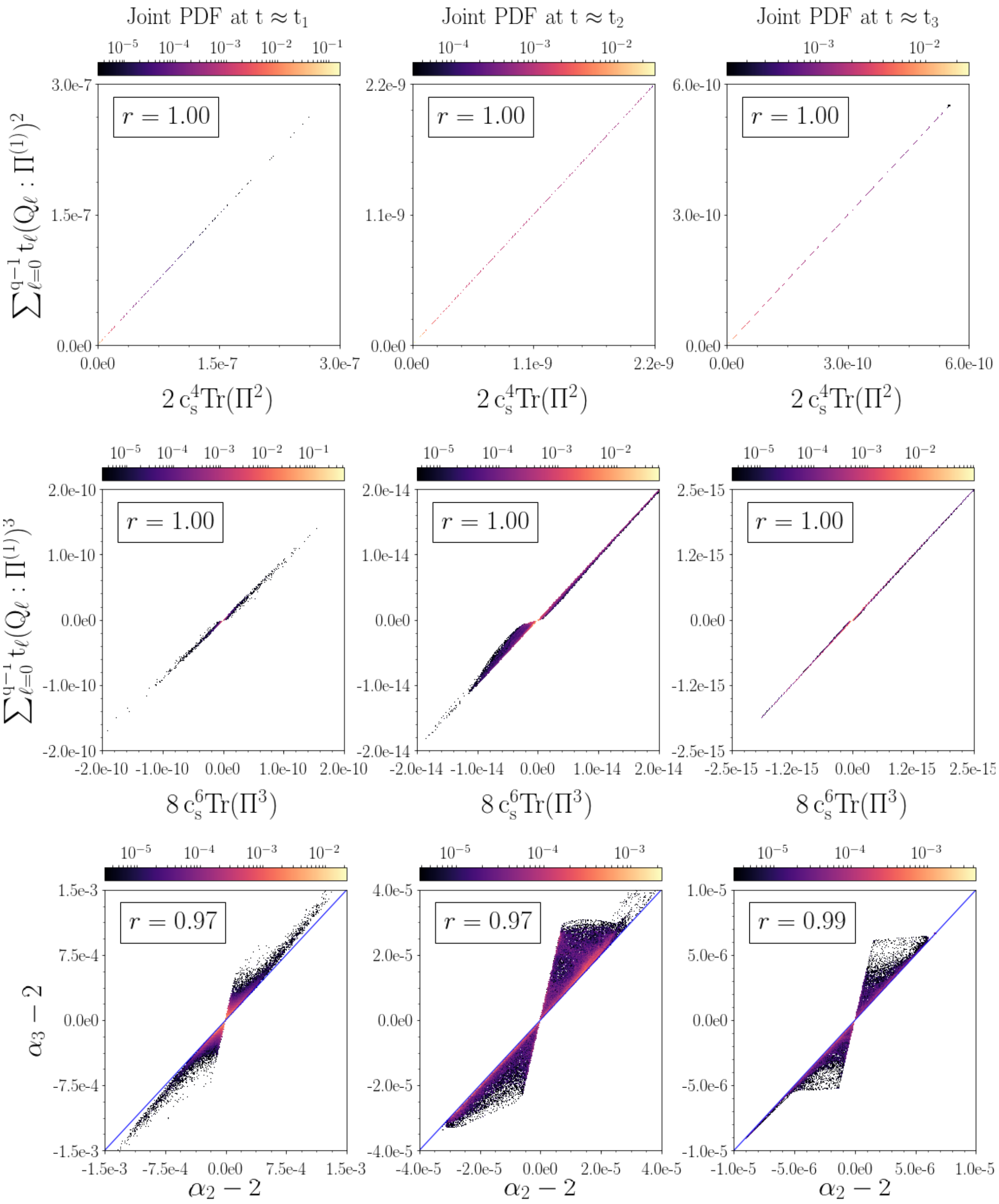


FIGURE 4.11. D2Q9: The first line shows the joint PDF between the LHS and the RHS of Eq. (4.18) based on 4th order isotropy relations. The first line shows the joint PDF between the LHS and the RHS of Eq. (4.19) based on 6th order isotropy relations. The last line shows the joint PDF between α_2 (Eq.(4.17)) and α_3 (Eq.(4.21)). Each column corresponds to one of the three studied times of the decaying t_1, t_2, and t_3. The blue curve shows a perfect matching while r is the Pearson's correlation coefficient.

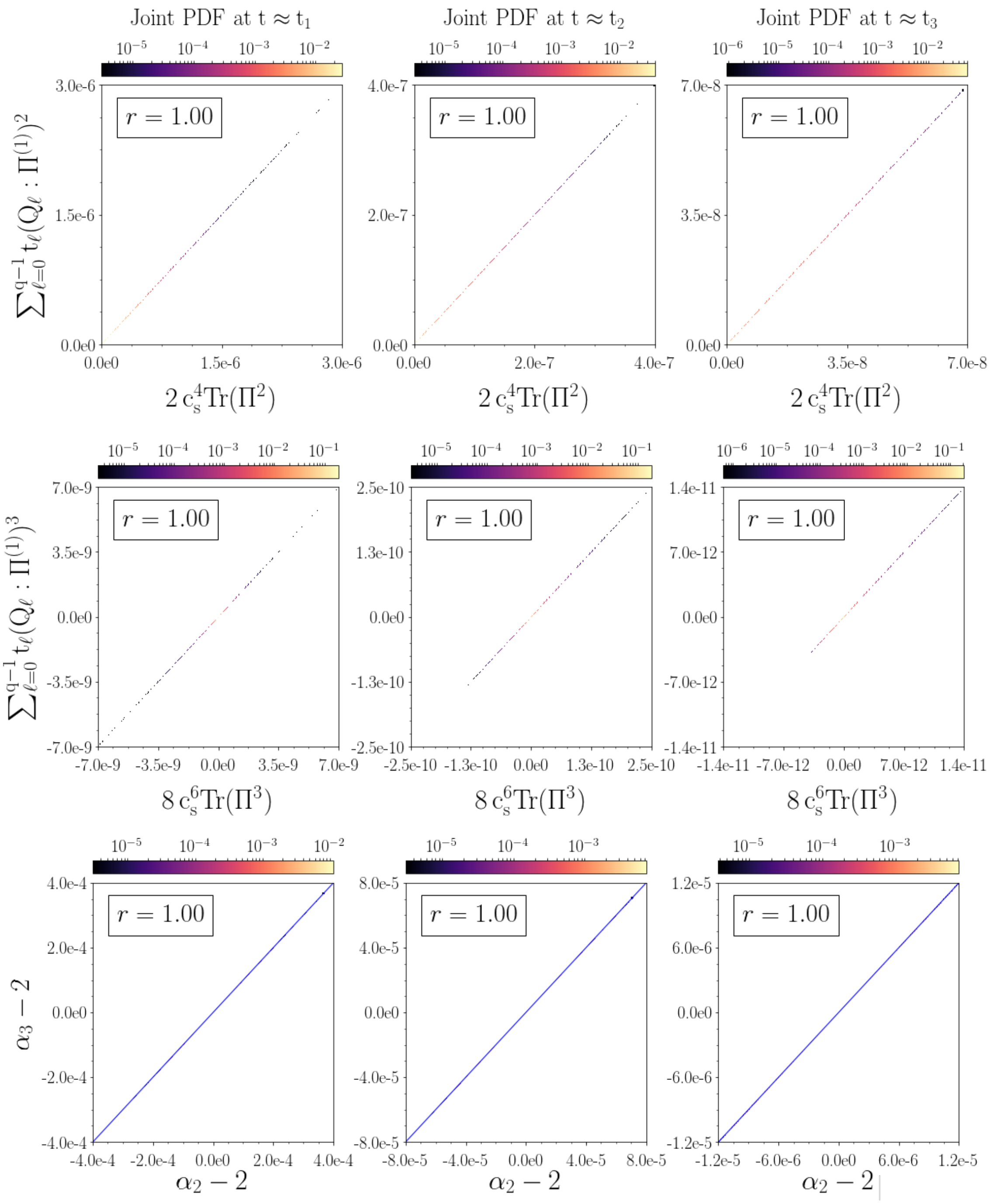


Figure 4.12. D2Q17: The first line shows the joint PDF between the LHS and the RHS of Eq. (4.18) based on 4th order isotropy relations. The first line shows the joint PDF between the LHS and the RHS of Eq. (4.19) based on 6th order isotropy relations. The last line shows the joint PDF between α_2 (Eq.(4.17)) and α_3 (Eq.(4.21)). Each column corresponds to one of the three studied times of the decaying t_1, t_2, and t_3. The blue curve shows a perfect matching while r is the Pearson's correlation coefficient.

4.2.4 Step 4 - Macroscopic closure

To close ϕ_α, we can directly use the relation between $\mathbf{\Pi}^{(1)}$ and $\mathbf{S}$ in Eq. (4.9) and we obtain

$$\phi_\alpha = \frac{1}{3\beta}\frac{\mathrm{Tr}(\mathbf{S}^3)}{\mathrm{Tr}(\mathbf{S}^2)} + \frac{1}{4\beta}\mathbf{A} ::: \frac{\mathbf{S}^3}{\mathrm{Tr}(\mathbf{S}^2)} + \mathcal{O}(\nabla u^2) + \mathcal{O}(\Delta_t), \tag{4.22}$$

and we obtain the closed approximation of α,

$$\alpha_4 = 2 + \frac{2\Delta_t}{3\beta}\frac{\mathrm{Tr}(\mathbf{S}^3)}{\mathrm{Tr}(\mathbf{S}^2)} + \frac{\Delta_t}{2\beta}\mathbf{A} ::: \frac{\mathbf{S}^3}{\mathrm{Tr}(\mathbf{S}^2)} + \mathcal{O}(\Delta_t\nabla u^2) + \mathcal{O}(\Delta_t^2) \tag{4.23}$$

We first look at the relation Eq. (4.9) component by component for the D2Q9 simulation on Fig. 4.13 and for the D2Q17 simulation on Fig. 4.14. Like at step 3, we observe an improvement of the matching between all components for the D2Q9 simulation as time goes to $t \approx t_3$. For the D2Q17 simulation, only the symmetric components develop in a good matching with time. This is most likely due to the lack of differentiability of the velocity field and will be discussed further in the next section.

Finally, we show on Fig. 4.7 and on Fig. 4.8, the summary of the numerical check for step 4 for the D2Q9 and the D2Q17 simulations respectively. For the D2Q9 simulation, the agreement between α_3 and α_4 goes from bad for $t \approx t_1$ and $t \approx t_2$, to perfect as $t \approx t_3$. As expected, given the numerical check between $\Pi^{(1)}$ and S for the D2Q17 simulation, we do not observe a good agreement between α_3 and α_4.

4.2.5 Overall agreement and discussion

The final correlation between α and α_4 is plotted for the D2Q9 and the D2Q17 respectively on Fig. 4.17 and Fig. 4.18. For D2Q9, while we are unable to reach a good agreement at $t \approx t_1$ and $t \approx t_2$, , we observe a very good matching at $t \approx t_3$ (first line, Fig. 4.17). We can further improve the accuracy by imposing a threshold under wish we filter out points to enforce $T2 \neq 0$ as done in step 1. On the second line of Fig. 4.17, we show the overall agreement imposing $T_2 > 10^{-10}$. On top of it, based on the results of step 3, we filter out anisotropic points observed at step 3 by imposing

$$\frac{\left|\sum_{\ell=0}^{q-1} t_\ell \left(\mathbf{Q}_\ell : \mathbf{\Pi}^{(1)}\right)^3 - 8c_s^6\Pi^{(1)}_{\kappa\theta}\Pi^{(1)}_{\theta\gamma}\Pi^{(1)}_{\gamma\kappa}\right|}{\left|8c_s^6\Pi^{(1)}_{\kappa\theta}\Pi^{(1)}_{\theta\gamma}\Pi^{(1)}_{\gamma\kappa}\right|} < 0.1,$$

thus selecting only points which validates the sixth order isotropy relations given in Eq. (4.19). We plot the resulting overall agreement between α and its hydrodynamic closure on the third line of Fig. 4.17 and observe further improvements in the agreement at $t \approx t_3$ between α calculated online using Newton-Raphson and its macroscopic closure α_5. As for the D2Q17 simulation, as expected from the results of step 2 and step 4, no

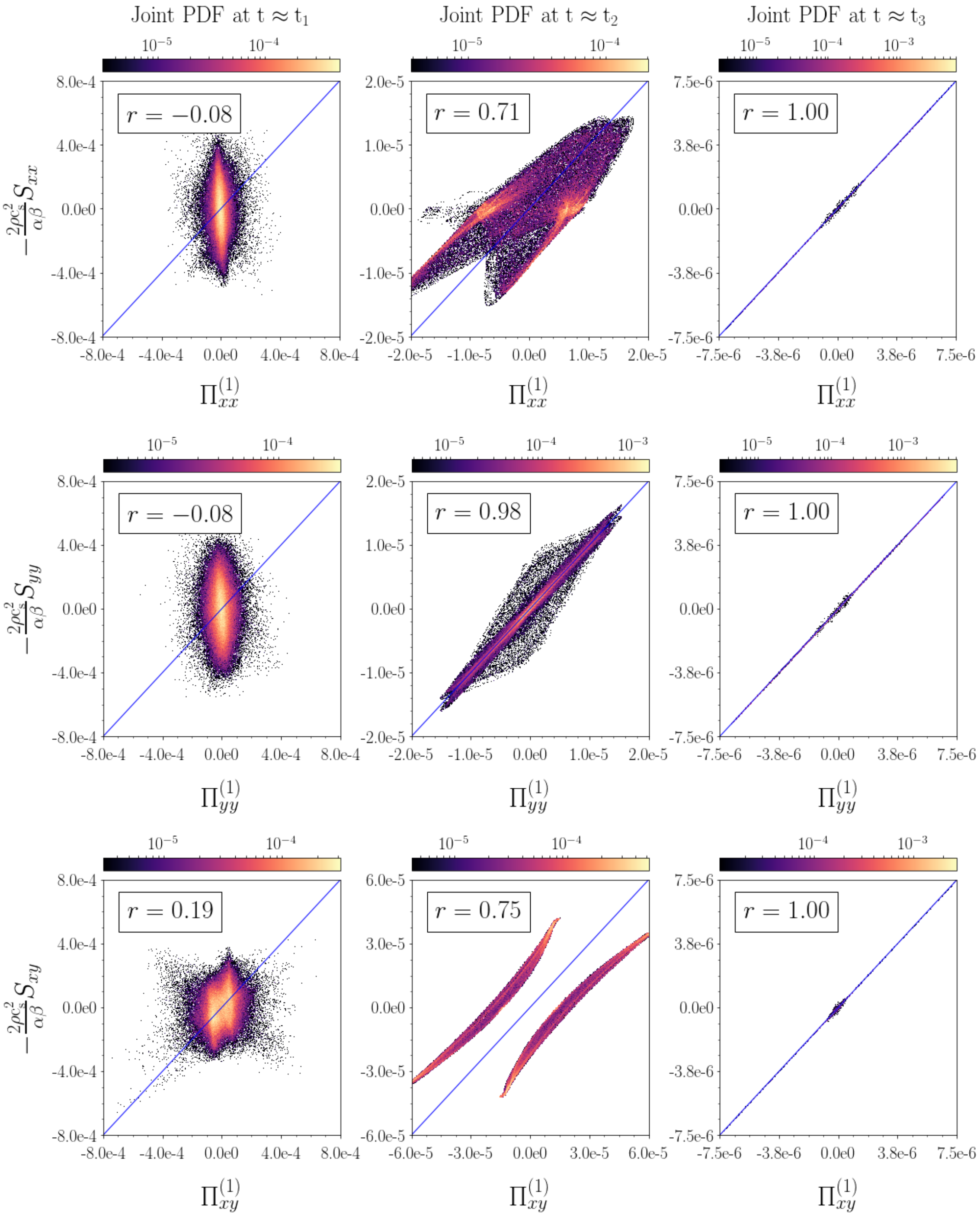


FIGURE 4.13. D2Q9: Joint PDF between $\Pi^{(1)}$ and $-\frac{2\rho c_s^2}{\alpha\beta}S$. Each line corresponds to the matching of a component of the tensors and each column corresponds to one of the three studied times of the decaying t_1, t_2, and t_3. The blue curve shows a perfect matching while r is the Pearson's correlation coefficient.

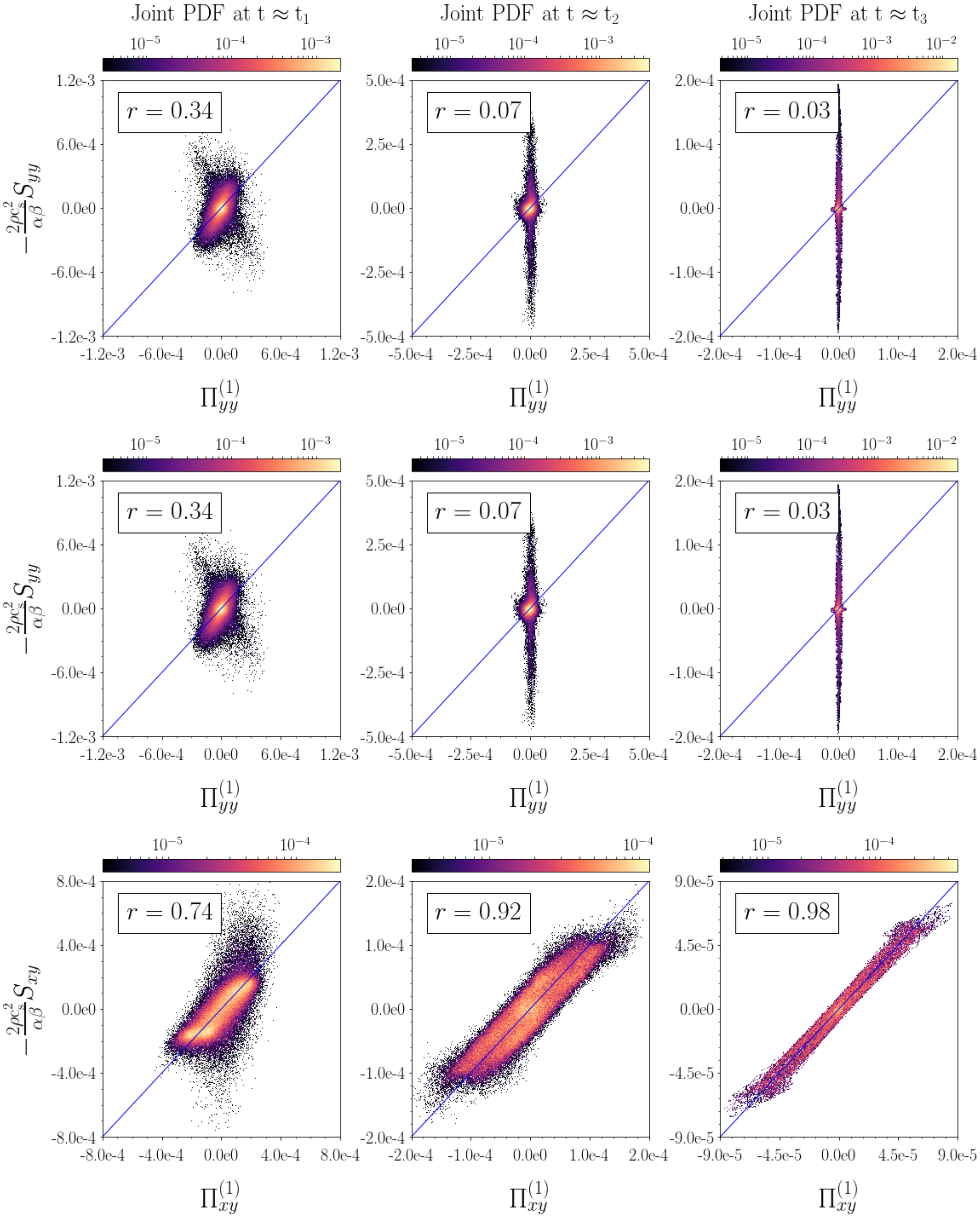


FIGURE 4.14. D2Q17: Joint PDF between $\Pi^{(1)}$ and $-\frac{2\rho c_s^2}{\alpha\beta}S$. Each line corresponds to the matching of a component of the tensors and each column corresponds to one of the three studied times of the decaying t_1, t_2, and t_3. The blue curve shows a perfect matching while r is the Pearson's correlation coefficient.

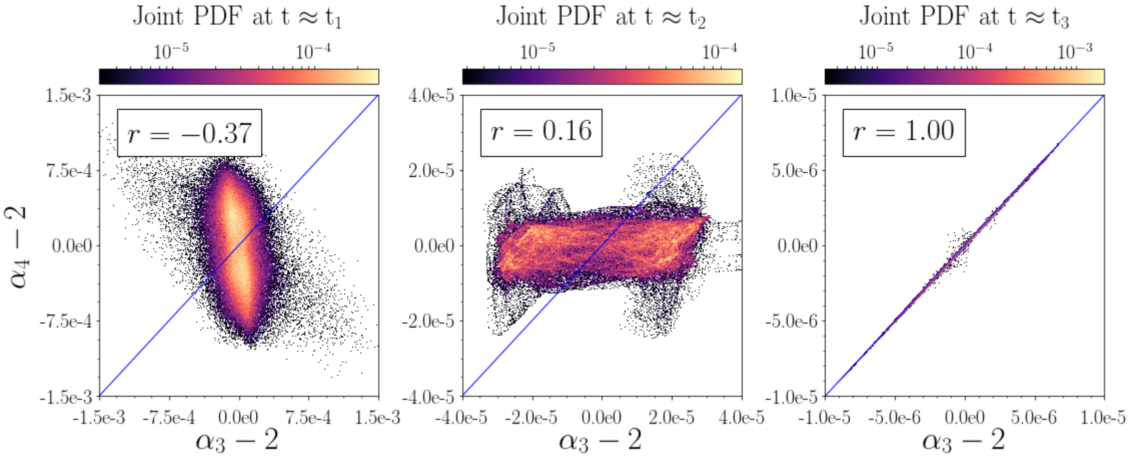


FIGURE 4.15. D2Q9: Joint PDF between α_3 (Eq. (4.21)) and α_4 (Eq. (4.23)). Each column corresponds to one of the three studied times of the decaying t_1, t_2, and t_3. The blue curve shows a perfect matching and the red curve shows the prediction made in Ref [2] while r is the Pearson's correlation coefficient.

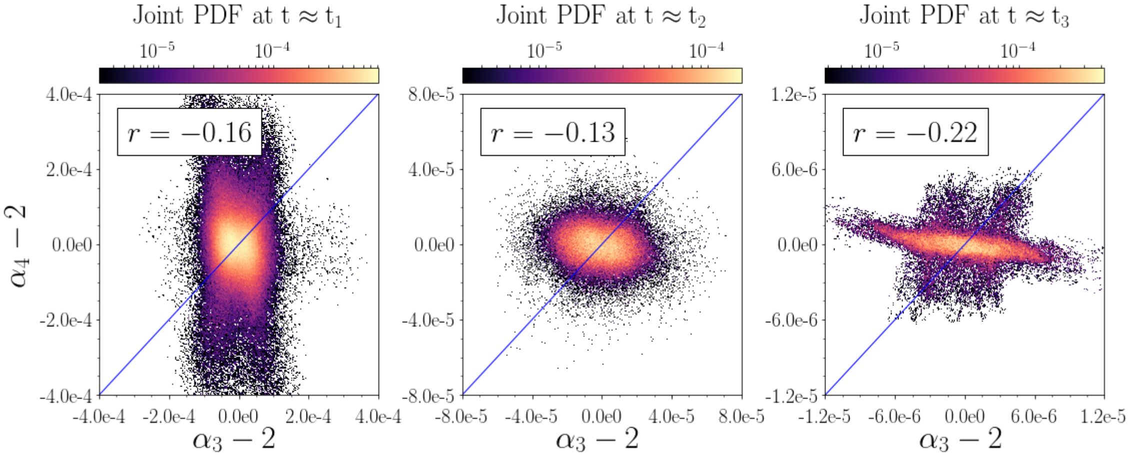


FIGURE 4.16. D2Q17: Joint PDF between α_3 (Eq. (4.21)) and α_4 (Eq. (4.23)). Each column corresponds to one of the three studied times of the decaying t_1, t_2, and t_3. The blue curve shows a perfect matching and the red curve shows the prediction made in Ref [2] while r is the Pearson's correlation coefficient.

agreement was obtained between α and α_5 (first line Fig. 4.18) and filtering out points to enforce $T_2 > 10^{-11}$ (second line line Fig. 4.18) is not sufficient to improve it.

Overall, our computations revealed a difference to Ref. [2] of a factor 2, which was numerically verified for the D2Q9 simulation as the simulation got increasingly resolved. Indeed, the outcome of the derivation of α in terms of macroscopic quantities α_4 (Eq. (4.23)) is valid at $\mathcal{O}(\Delta_t \nabla u^2)$. This validity criteria was hidden in Ref. [2] under the regularization assumption Eq. (4.15) (made at step 2 of this derivation). But if this C-E expansion for α is only valid for smooth velocity fields, it is by definition not applicable to fully developed turbulent flows (see section 1.1.4 on the dissipative anomaly). As for D2Q17, looking at the small scales on the spectra (Fig. 4.2 panel (a)), we can see that the velocity field is not obviously differentiable, and therefore we were not able to check relations involving velocity gradients.

4.3 Computation of the resulting macroscopic Sub-Grid Scale model

In the ELBM literature as well as in the original derivation of the hydrodynamic SGS in Ref. [2], the effective viscosity ν_{eff} is straightforwardly obtained from the effective relaxation time $\tau_{\text{eff}} = \frac{1}{\alpha\beta}$ as

$$\nu_{\text{eff}} = \Delta_t \left(\frac{1}{\alpha\beta} - \frac{1}{2} \right) c_s^2 + \mathcal{O}(Ma^3) + \mathcal{O}(\Delta_t^2). \tag{4.24}$$

and therefore,

$$\delta\nu_e^A \approx -c_s^2 \Delta t^2 \frac{1}{6\beta^2} \frac{S_{\lambda\mu} S_{\mu\gamma} S_{\gamma\lambda}}{S_{\gamma\delta} S_{\gamma\delta}}. \tag{4.25}$$

Notice that Eq. (4.25) differs from the output eddy viscosity of a factor 2 stemming from the approximation of α numerically verified in the last session, and a further factor Δ_t in the eddy viscosity, correcting the dimensionality of the result.

This expression is based on the fact that the momentum conservation equation obtained at the second order in Δ_t Eq. (4.26) for a fixed relaxation time $\tau = \tau_0$ is also valid when $\tau = \tau_{\text{eff}} = \frac{\alpha}{\beta}$. Therefore we have

$$\partial_t(\rho \boldsymbol{u}) + \boldsymbol{\nabla} \cdot (\rho \boldsymbol{u}\boldsymbol{u}) = -\boldsymbol{\nabla}(c_s^2 \rho) + \boldsymbol{\nabla} \cdot \left\{ \Delta_t \left(\frac{1}{\alpha\beta} - \frac{1}{2} \right) \left[c_s^2 \rho [\boldsymbol{\nabla} \boldsymbol{u} + (\boldsymbol{\nabla} \boldsymbol{u})^T] + \boldsymbol{A} :: \boldsymbol{\nabla}(\rho \boldsymbol{u}\boldsymbol{u}\boldsymbol{u}) \right] \right\} + \mathcal{O}(\Delta_t^2). \tag{4.26}$$

This expression can be considered correct up to $\mathcal{O}(\Delta_t^2)$, meaning that α has to be set equal to its leading order expression in Δ_t. As per Eq, (4.2), we have

$$\alpha = 2 + \mathcal{O}(\Delta_t), \tag{4.27}$$

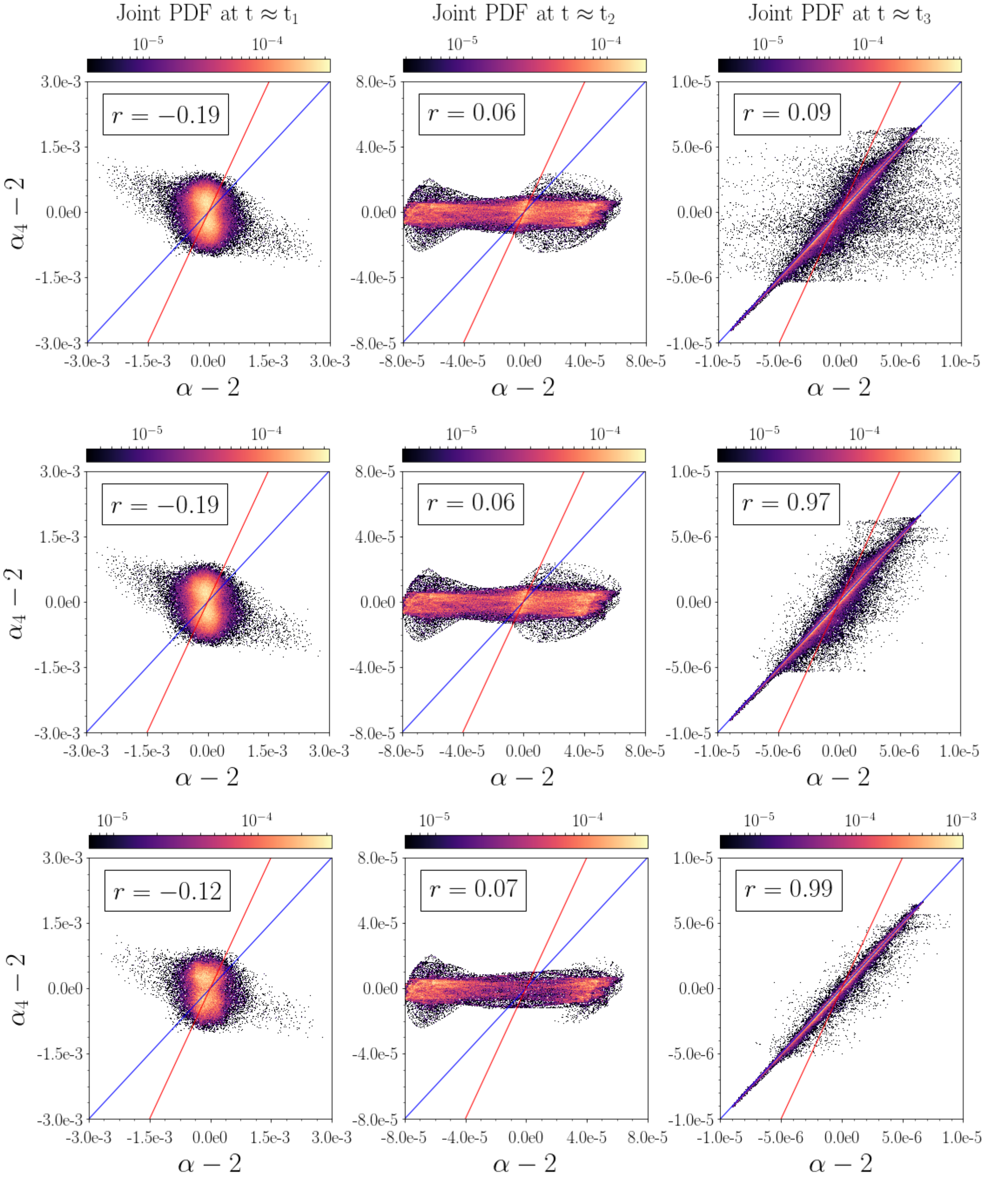


FIGURE 4.17. D2Q9: Joint PDF between α (calculated using Newton-Raphson) and α_4 (Eq.(4.23)). Each column corresponds to one of the three studied times of the decaying t_1, t_2, and t_3. The blue curve shows a perfect matching and the red curve shows the prediction made in Ref [2] while r is the Pearson's correlation coefficient.

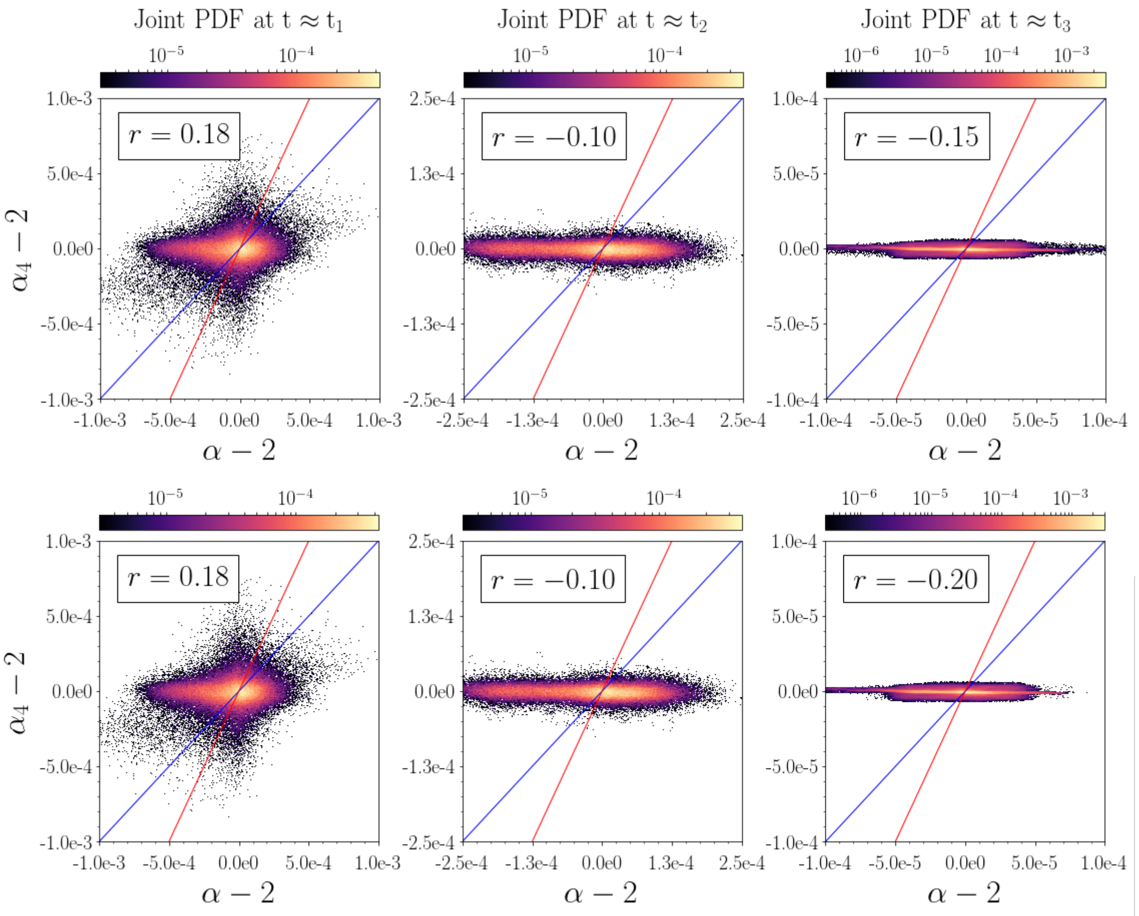


Figure 4.18. D2Q17: Joint PDF between α (calculated using Newton-Raphson) and α_4 (Eq.(4.23)). Each column corresponds to one of the three studied times of the decaying t_1, t_2, and t_3. The blue curve shows a perfect matching and the red curve shows the prediction made in Ref [2] while r is the Pearson's correlation coefficient.

and higher orders in the expansion of α are absorbed in $\mathcal{O}(\Delta_t^2)$ in Eq. (B.20). It is clear that the ELBM viscosity as identified in the momentum equation of second order in Δ_t, is actually given by

$$\nu = \Delta_t \frac{1-\beta}{2\beta} c_s^2 + \mathcal{O}(Ma^3) + \mathcal{O}(\Delta_t^2) = \nu_0 + \mathcal{O}(Ma^3) + \mathcal{O}(\Delta_t^2). \tag{4.28}$$

Therefore in order to observe the effect of the fluctuating entropic parameter α on the dynamics, it is necessary to go the third order C-E expansion. The full derivation is given in Appendix C and the resulting mass conservation equation is

$$\partial_t \rho + \boldsymbol{\nabla} \cdot (\rho \boldsymbol{u}) = \frac{\Delta_t^2}{12} \boldsymbol{\nabla}\boldsymbol{\nabla} : \left[c_s^2 \rho [\boldsymbol{\nabla} \boldsymbol{u}]_2 + \boldsymbol{A} :: \boldsymbol{\nabla}(\rho \boldsymbol{uuu}) \right] + \mathcal{O}(\Delta_t^3), \tag{4.29}$$

in which we use the notation $([\mathbf{A}]_2)_{ij} = A_{ij} + A_{ji}$, with $\mathbf{A}$ a second-order tensor. The momentum conservation equation is given here in the isotropic case ($\boldsymbol{A} = \mathbf{0}$) for simplicity:

$$\begin{aligned}
\partial_t(\rho \boldsymbol{u}) + \boldsymbol{\nabla} \cdot (\rho \boldsymbol{uu}) = &-\boldsymbol{\nabla}(c_s^2 \rho) \\
&+ \Delta_t \boldsymbol{\nabla} \cdot \left\{ \left(\tau_0 - \frac{1}{2} \right) c_s^2 \rho [\boldsymbol{\nabla} \boldsymbol{u}]_2 \right\} \\
&+ \Delta_t \boldsymbol{\nabla} \cdot \left\{ \frac{2-\alpha}{4\beta} c_s^2 \rho [\boldsymbol{\nabla} \boldsymbol{u}]_2 \right\} \\
&+ \Delta_t^2 \frac{(\beta-1)^2}{4\beta^2} \boldsymbol{\nabla} \cdot \left\{ c_s^2 \rho [\boldsymbol{\nabla} \boldsymbol{u} \cdot \boldsymbol{\nabla} \boldsymbol{u}]_2 + 2 c_s^4 \rho \boldsymbol{\nabla}\boldsymbol{\nabla} \ln \rho + \boldsymbol{\nabla}\boldsymbol{\nabla} : (\rho \boldsymbol{uuuu}) \right\} \\
&+ \frac{\Delta_t^2}{12} c_s^2 \boldsymbol{\nabla} \cdot \{ \boldsymbol{\nabla} \cdot (\rho [\boldsymbol{\nabla} \boldsymbol{u}]_2 \boldsymbol{u}) - 2\rho [\boldsymbol{\nabla} \boldsymbol{u} \cdot \boldsymbol{\nabla} \boldsymbol{u}]_2 \} \\
&- \frac{\Delta_t^2}{12} \boldsymbol{\nabla} \cdot \left\{ 4 c_s^4 \rho \boldsymbol{\nabla}\boldsymbol{\nabla} \ln \rho + \boldsymbol{\nabla}\boldsymbol{\nabla} : (\rho \boldsymbol{uuuu}) \right\} + \mathcal{O}(\Delta_t^3).
\end{aligned} \tag{4.30}$$

Replacing α by its macroscopic approximation Eq. (4.23), the momentum balance becomes

$$\begin{aligned}
\partial_t(\rho \boldsymbol{u}) + \boldsymbol{\nabla} \cdot (\rho \boldsymbol{uu}) = &-\boldsymbol{\nabla}(c_s^2 \rho) \\
&+ \Delta_t \boldsymbol{\nabla} \cdot \left\{ \left(\tau_0 - \frac{1}{2} \right) c_s^2 \rho [\boldsymbol{\nabla} \boldsymbol{u}]_2 \right\} \\
&- \frac{\Delta_t^2}{6\beta^2} \boldsymbol{\nabla} \cdot \left\{ \frac{\operatorname{Tr} \mathbf{S}^3}{\operatorname{Tr} \mathbf{S}^2} c_s^2 \rho [\boldsymbol{\nabla} \boldsymbol{u}]_2 \right\} \\
&+ \Delta_t^2 \frac{(\beta-1)^2}{4\beta^2} \boldsymbol{\nabla} \cdot \left\{ c_s^2 \rho [\boldsymbol{\nabla} \boldsymbol{u} \cdot \boldsymbol{\nabla} \boldsymbol{u}]_2 + 2 c_s^4 \rho \boldsymbol{\nabla}\boldsymbol{\nabla} \ln \rho \right\} \\
&+ \frac{\Delta_t^2}{12} c_s^2 \boldsymbol{\nabla} \cdot \{ \boldsymbol{\nabla} \cdot (\rho [\boldsymbol{\nabla} \boldsymbol{u}]_2 \boldsymbol{u}) - 2\rho [\boldsymbol{\nabla} \boldsymbol{u} \cdot \boldsymbol{\nabla} \boldsymbol{u}]_2 \} \\
&- \frac{\Delta_t^2}{12} \boldsymbol{\nabla} \cdot \left\{ 4 c_s^4 \rho \boldsymbol{\nabla}\boldsymbol{\nabla} \ln \rho \right\} + \mathcal{O}(\nabla^3 u^3 \Delta_t^2) + \mathcal{O}(\Delta_t^3).
\end{aligned} \tag{4.31}$$

Eq. (4.31) reveals that the implicit SGS model implied by ELBM is, even assuming the validity of sixth order isotropy relation, much more complex than the single eddy viscosity term usually put forward in the litterature [3, 69].

Moreover, this finding also strongly puts in question the control one has when using an eddy viscosity SGS within LBM. As discussed in section 1.3.5, this is traditionally done by modifying the local relaxation time to obtain a local effective viscosity. For example, in the Smagorinsky model [4], the eddy viscosity is $\delta\nu_e^S = (C_S\Delta)^2\sqrt{2S_{ij}S_{ij}}$ which scales as $\mathbf{S}$. Calculating the effective relaxation time to enforce this model is equivalent to coming up with a parameter $\alpha_{\mathrm{S}} = 2(1+\Delta_t\phi_\alpha^S)$, [71], where the fluctuation corresponding to the Smagorinsky eddy viscosity are contained in $\phi_\alpha^S = \mathcal{O}(1)$. Thus in that case as well, the eddy viscous dissipation term is of order $\mathcal{O}(\Delta_t^2)$ as the terms contained in the last four lines of Eq. (4.31) and therefore this LBM-LES cannot be considered solely as a LES with a Smagorinsky eddy viscosity model.

4.4 Concluding remarks

In this chapter, we have studied in detail the formulation of the implicit SGS implied by ELBM both at macroscale and at mesoscale. In the context of decaying 2D HIT from a fully developed turbulent flow configuration, we have reformulated the C-E expansion of α carried out in Ref. [2]. Step-by-step, we numerically checked for simulations using the D2Q9 (isotropic to the fourth order) and the D2Q17 (isotropic to the sixth order), all the approximation to obtain a formulation of the entropic parameter in terms of macroscopic quantities.

Our computations revealed a difference to Ref. [2] of a factor 2 for the approximation of α, which was numerically verified, and a further factor Δ_t in the eddy viscosity, correcting the dimensionality of the result. We have determined that the macroscopic approximation for α is valid for a smooth velocity field with low gradients. Indeed, we observed that for the D2Q9 simulation by filtering out anisotropic points, we could only recover a good matching with the measured α at the end of the decaying. As a result, the macroscopic approximation of the entropic parameter was shown to be valid only for well-resolved flows, out of the range of interest of a LES.

As for the D2Q17 simulation, while the steps based on isotropy relations highlighted a perfect agreement, it was not possible to recover an agreement between α and its macroscopic approximation. The simulated flow was unphysical from the fully developed turbulent initial configuration on, as its energy spectrum was showing a set of small scales bumps. Those bumps, due to the multi-speed characteristic of the D2Q17 lattice, leads to velocity field that was not differentiable.

Furthermore, we have shown that the implicit ELBM model does not limit itself to a sole eddy viscous dissipation depending on the input relation time and on the entropic parameter. Indeed, this term appears in a macroscopic equation of motion that requires a C-E expansion of third order, while the N-S equations are recovered at the second order. A number of extra third-order terms are therefore part of the implicit ELBM SGS model.

However, a few points need to be highlighted here. The first one is that the macroscopic eddy viscosity term of the ELBM implicit model still has a very interesting formulation. It is similar to the Smagorinsky model in the sense that scales with the strain-rate tensor, but, at the same time, it is not positive-definite and therefore allows backscatter of energy to the resolved scales. Secondly, even if there is more to the ELBM implicit SGS model than the eddy viscous dissipation, it does not mean that ELBM does not model turbulence properly. Both of those points should be further explored to conclude about the ELBM's implicit modeling of turbulence.

CHAPTER 5

INERTIAL RANGE STATISTICS OF THE ENTROPIC LATTICE BOLTZMANN AND LARGE-EDDY SIMULATIONS IN 3D TURBULENCE

In this chapter, we present preliminary results on the comparison of inertial range statistics of turbulent velocity fields obtained by ELBM, with those coming from a high-resolution DNS of the NSE conducted with a PS code. Additionally, we also study the approximated eddy viscosity against the Smagorinsky [4] closures by implementing them in a PS LES code. This chapter is organized as follows. After presenting the set of forced 3D HIT ELBM and PS LES simulations in section 5.1, we study low-order inertial range statistics in section 5.2; in section 5.3; in order to understand how the compared closures capture non-linear intermittent behavior [72], we analyze high-order statistics of those simulations; some concluding remarks follows in section 5.4.

5.1 Compared ELBM and LES closure in 3D homogeneous isotropic turbulence

In chapter 4, we have shown that ELBM cannot be considered to be recovering the Navier-Stokes equations with a sole eddy viscosity term of the form

$$\delta\nu_e^M = c_s^2 \frac{2-\alpha}{2\alpha\beta}\Delta_t. \tag{5.1}$$

However, the fact that the actual ELBM closure is more complex than a simple eddy viscosity model does not mean, in principle, that ELBM is unable to model turbulence. At the same time, even if invalid for the range of Reynolds numbers of interest, the macroscopic approximation of the ELBM eddy viscosity has a very interesting formulation

$$\delta\nu_e^A = (C_A\Delta)^2 \frac{S_{\lambda\mu}S_{\mu\gamma}S_{\gamma\lambda}}{S_{\gamma\delta}S_{\gamma\delta}} \propto o(\mathbf{S}), \tag{5.2}$$

where Δ is the LES cut-off length and C_A a dimensionless coefficient. In Chapter 4, we have shown that the C-E expansion of the ELBM entropic parameter yields $C_A = \frac{c_s \Delta_t}{\beta\sqrt{6}}$ and $\Delta = 1$ as the filtering is implicit. Indeed, it is similar to a Smagorinsky model [4],

$$\delta\nu_e^S = (C_S\Delta)^2\sqrt{2S_{\lambda\mu}S_{\lambda\mu}} \propto o(\mathbf{S}), \tag{5.3}$$

where C_S is a dimensionless coefficient typically taken equal to 0.16 and Δ is the cut-off length, as they both scale as the strain-rate tensor $\mathbf{S}$. Moreover, the approximated eddy viscosity $\delta\nu_e^A$ Eq. (5.2) is not positive-definite and therefore allows backscatter of energy, *i.e.* energy transfer from the unresolved to the resolved scales. Indeed, while energy should in average cascade towards the small scales to model properly the small-scale dissipation, a realistic SGS should also intermittently transfer energy in the other direction. The approximate eddy viscosity $\delta\nu_e^A$ possesses this property and therefore is of interest.

We conduct a set of simulations of forced 3D homogeneous isotropic turbulence (HIT) on a periodic three-dimensional 512^3 computational grid. The homogeneous isotropic turbulence forcing acts on a shell of (dimensionless) wavenumbers $\boldsymbol{k}$ of magnitude from 1 to 2 with a constant phase ϕ and we take its rotational to ensure that it does not input any incompressibility in the system.

The ELBM simulation uses a lattice with 27 discrete velocities (see Fig. 5.1), the D3Q27 [33, 34, 35]. We enforce the spectral forcing using the exact-difference method forcing scheme [61] for a relaxation time $\tau_0 = 0.5001$ corresponding to $\beta \approx 0.9998$.

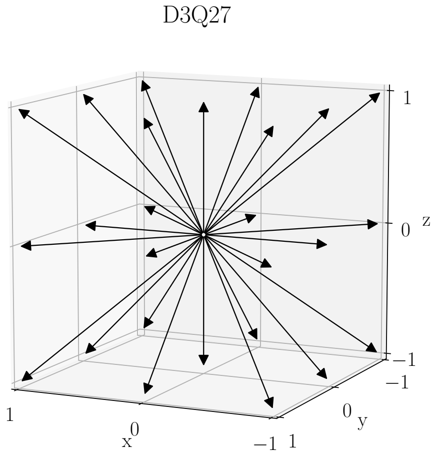


Figure 5.1. D3Q27 lattice stencil used for the ELBM simulation.

In order to benchmark the approximated eddy viscosity model against the Smagorinsky model, we have implemented them as LES closures within a PS code. We refer hereafter to the LES simulation with an approximated eddy viscosity model as *LES-A* and to the one with a Smagorinsky model as *LES-S*. In the expression of Smagorinsky eddy viscosity Eq (5.3), we use the standard value of $C_S = 0.16$, while for the approximated eddy viscosity Eq. (5.2), we use $C_A = 0.45$, and for both we have $\Delta = \frac{3\pi}{512} \approx 0.0184$. Additionally, as a reference, we run a Direct Numerical Simulation (DNS) conducted on a 1024^3 grid, which will be denoted as *DNSx2*.

The snapshots Fig. 5.2 of the approximated eddy viscosity $\delta\nu_e^A$ for the LES-A simulation and the approximated $\delta\nu_e^A$ and measured eddy viscosity $\delta\nu_e^M$ for ELBM provide an insight on their spatial behavior. All eddy viscosities appear to be small scales quantities, which do not seem to be organized in structures. As expected for such a high Reynolds number, the measured and approximated eddy viscosities do not seem to be correlated.

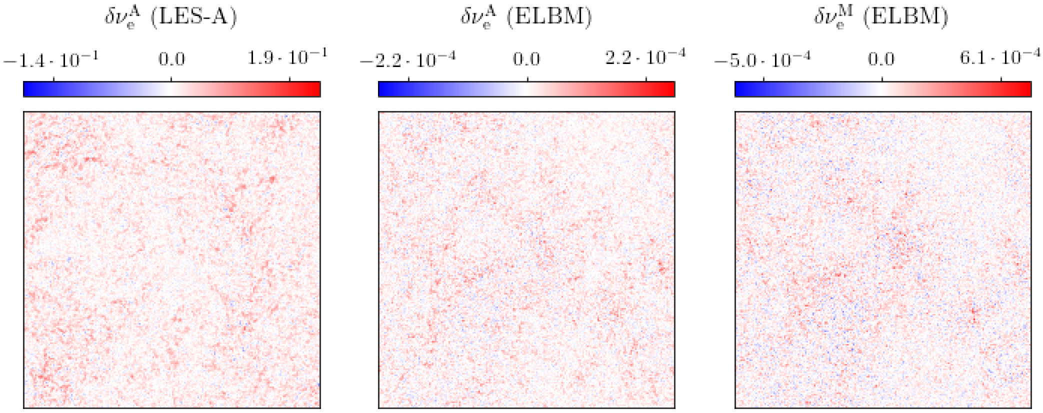


FIGURE 5.2. 2D-sliced snapshot of the approximated eddy viscosity $\delta\nu_e^A$ for the LES simulation (LES-A) and of the approximated eddy viscosity $\delta\nu_e^A$ and measured eddy viscosity $\delta\nu_e^M$ for the ELBM simulation.

Moreover, we show on Fig. 5.3 the PDF of the ELBM measured eddy viscosity $\delta\nu_e^M$, the ELBM approximated eddy viscosity $\delta\nu_e^A$ and the approximated eddy viscosity of the LES simulation $\delta\nu_e^A$. Only the ELBM measured eddy viscosity exhibits a strong positive skewness, highlighting that information on the presence of a forward cascade is contained in the tails of its PDF. However, all eddy viscosities are positive on average and therefore their corresponding SGS is always overall dissipative. Moreover, they show deviations of one order of magnitude of their standard deviations, a sign of a strong non-Gaussian behavior.

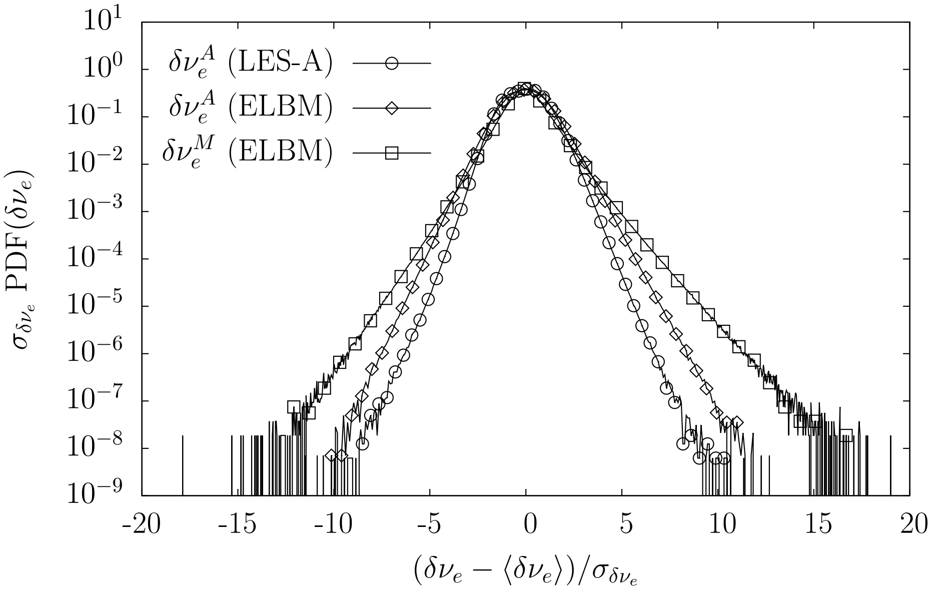


FIGURE 5.3. Standardized PDF of the eddy viscosity corresponding to LES with the approximated model (LES-A), ELBM with the approximated model $\delta\nu_e^A$ model, and ELBM with the measured model $\delta\nu_e^M$ model.

5.2 Analysis of low-order inertial range statistics

We show in Fig. 3.10 the superposed time-averaged spectrum for all the conducted simulations. First, it is worth mentioning that the ELBM and LES-A simulation remain stable even though their eddy viscosities can take negative values. Overall, each simulation highlights a nicely extended inertial range with an energy cascade slope matching the Kolmogorov prediction of $-5/3$ [28, 14], except for the ELBM simulation that shows a slight small-scale accumulation of energy. To ensure a fair comparison, an ELBM simulation exhibiting a $-5/3$rd energy spectra slope need to be produced. We will pursue our analysis nonetheless and present preliminary results.

According to the phenomenological theory of Kolmogorov (K41) [12] and as described in Chapter 1.1.4, the scaling behavior of the p-th order longitudinal structure function is

$$S_p(r) = \langle(\delta_r \boldsymbol{u}_{\parallel})^p\rangle \sim C_p(\varepsilon_{\nu_0} r)^{\xi_p}, \text{ with } \xi_p = p/3. \tag{5.4}$$

Nevertheless, both experimental and numerical studies have highlighted that the scaling exponent of p-th order structure function deviates from the K41 predicted value as a result of intermittency. She-Lêveque showed in Ref. [16] that the real exponent can be

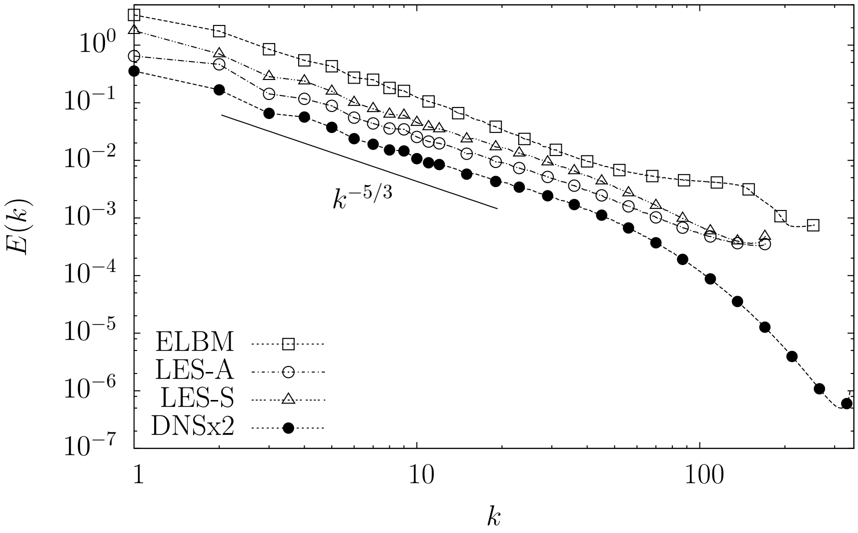


FIGURE 5.4. Superposed time-averaged spectrum for the conducted simulations at 512^3 grid points, using ELBM, LES with the approximated model (LES-A), LES with Smagorinsky model (LES-S) and DNS at 1024^3 grid points (DNSx2). The Kolmogorov predicted slope of the forward energy cascade of $-5/3$ [28, 14] is given as a reference.

modelled as

$$\xi_p = \frac{p}{9} + 2\left(1 - \left(\frac{2}{3}\right)^{p/3}\right). \tag{5.5}$$

The energy spectrum can be directly linked to the second-order structure functions. Therefore, we start by taking look at Fig. 5.5, where the longitudinal second-order structure functions S_2 (left panel) and their scaling exponent (right panel) are plotted. The exponent is obtained by taking the derivative of the logarithm of the S_2. The range of scales exhibiting a power-law defines the inertial range of scales. Therefore the scaling exponent is of particular interest and it is interesting to observe that ELBM inertial range predictions seem particularly expanded. Looking closer on panel (b), we can see that the ELBM simulation exponent starts to deviate from the constant She-Lêveque value at the same scale than other simulations and comes back to it at a smaller scale. In this preliminary stage, it is unclear whether this is only a consequence of the small bump observed on the energy spectra (Fig. 5.4). The LES-A and LES-S closures show similar inertial range behavior that the DNSx2 simulation. However, from second-order statistics, it is not possible to estimate whether the non-linear inertial range physics is preserved by the closures and we need to perform higher-order statistical analysis.

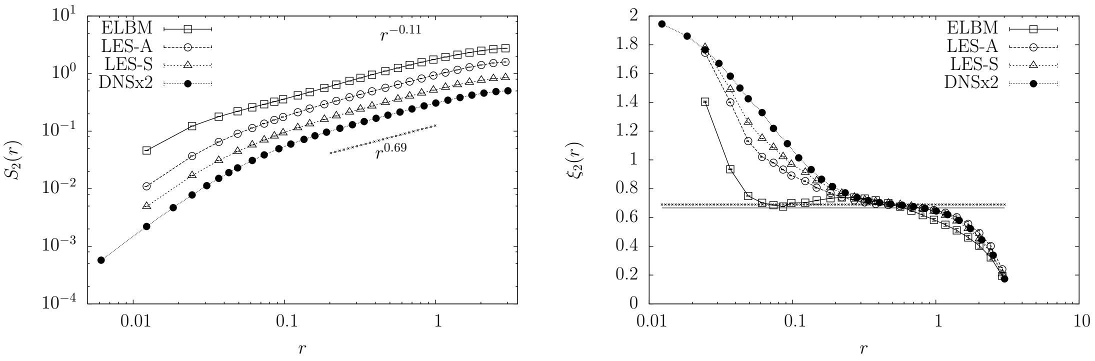


FIGURE 5.5. Second-order longitudinal structure functions (left) and corresponding local slopes (right) for the conducted simulations at 512^3 grid points, using ELBM, LES with the approximated model (LES-A), LES with Smagorinsky model (LES-S) and DNS at 1024^3 grid points (DNSx2). The dashed line corresponds to the K41 prediction in the inertial range (Eq. (5.4)), while the straight line corresponds to the intermittent corrected prediction from the She-Lêveque model (Eq. (5.5)).

5.3 Analysis of high-order inertial range statistics

We extend the analysis of the second-order structure function by conducting a similar one with the fourth-order structure functions S_4 on Fig. 5.6. The analyzed scaling behavior of S_4 is very similar than the one of S_2. To get an insight on whether the intermittent behavior is captured by the closures, we look on Fig. 5.7 at the Extended Self-Similarity [73],

$$ESS(r) = \frac{\xi_4}{\xi_2}. \tag{5.6}$$

A linear K41 behavior would recover an inertial range value of the ESS of 2. The She-Lêveque predictions, accounting for intermittency, highlight the presence of a plateau for an ESS value of 1.86. We observe, that while at large scale the ESS is dominated by the forcing, both ELBM and DNSx2 exhibit a plateau on an inertial range of scales of similar length. The LES-based closures, LES-A and LES-S, both clearly fail to capture the right intermittent physics in the inertial range. At small scales, all closures deviate away from the right non-linear physics to a linear behavior.

Looking further at the Kurtosis of the velocity increment,

$$K(r) \equiv \frac{\left\langle((\delta_r u)^4\right\rangle}{\left\langle((\delta_r u)^2\right\rangle^2} = \frac{S^{(4)}(r)}{\left(S^{(2)}(r)\right)^2}, \tag{5.7}$$

on Fig. 5.8, we see that for all simulations at large scale, where the forcing is active, the velocity increments are Gaussian with a K value of 3. At decreasing scales, the statistics are changing following predictions of the She-Lêveque model. In the inertial range of

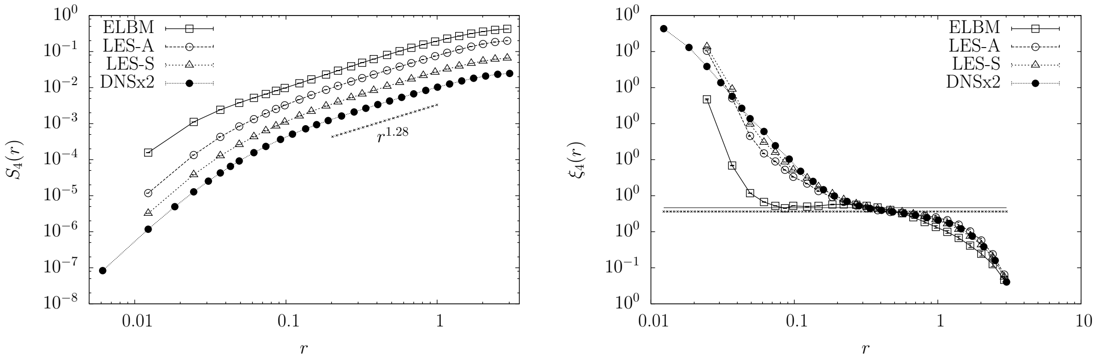


FIGURE 5.6. Fourth-order longitudinal structure functions (left) and corresponding local slopes (right) for the conducted simulations at 512^3 grid points, using ELBM, LES with the approximated model (LES-A), LES with Smagorinsky model (LES-S) and DNS at 1024^3 grid points (DNSx2). The dashed line corresponds to the K41 prediction in the inertial range (Eq. (5.4)), while the straight line corresponds to the intermittent corrected prediction from the She-Lêveque model (Eq. (5.5)).

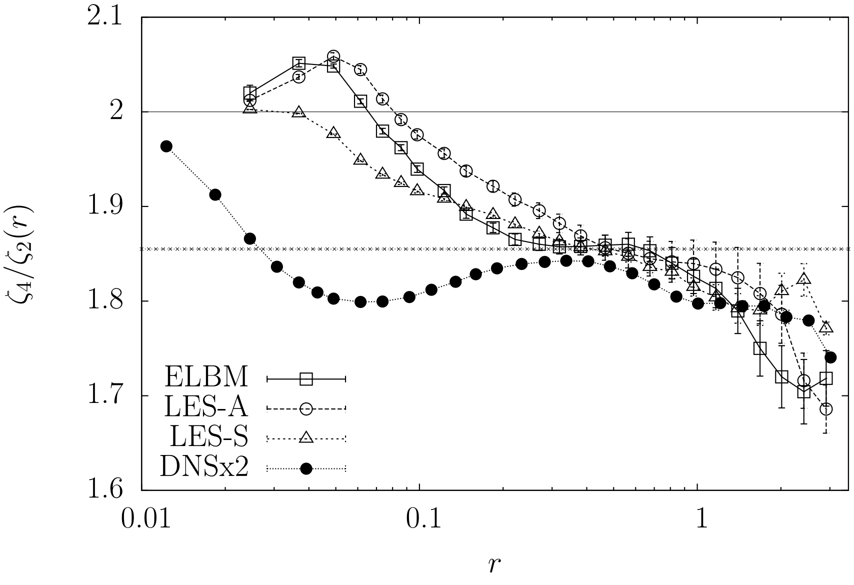


FIGURE 5.7. Longitudinal Extended Self-Similarity for the conducted simulations at 512^3 grid points, using ELBM, LES with the approximated model (LES-A), LES with Smagorinsky model (LES-S) and DNS at 1024^3 grid points (DNSx2). The dashed line corresponds to the K41 prediction in the inertial range (Eq. (5.4)), while the straight line corresponds to the intermittent corrected prediction from the She-Lêveque model (Eq. (5.5)).

scale, the DNSx2 physics appeared to be still captured by all closures. At $r \approx 0.2$, the statistics of the DNSx2 simulation depearts from inertial range predictions as the dissipation effects are appearing. The closure-based simulations, ELBM, LES-A and LES-S do not have dissipative scales, but deviate simultaneously as DNSx2 simulations from the correct inertial range physics.

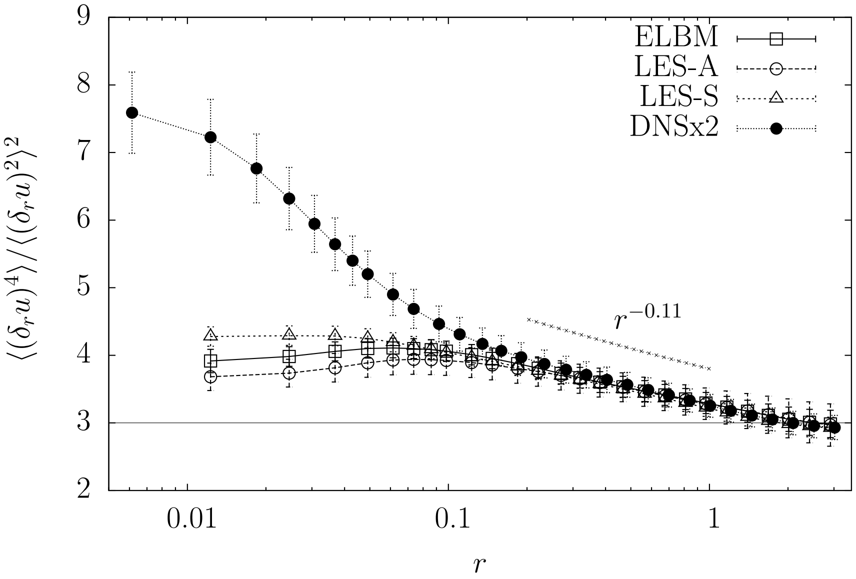


Figure 5.8. Kurtosis of the velocity increment for the conducted simulations at 512^3 grid points, using ELBM, LES with the approximated model (LES-A), LES with Smagorinsky model (LES-S) and DNS at 1024^3 grid points (DNSx2). The dashed horizontal line corresponds to the value of a Gaussian.

5.4 Concluding remarks

In this chapter, in order to assess the capabilities of the ELBM closure, we compared the inertial range statistics of turbulent velocity fields obtained by ELBM, with those coming from a higher-resolution DNS conducted with a PS code. Preliminary results showed that ELBM is able to increase the inertial scaling range and partially captures the correct intermitent behaviors. However, there is a need to perform a new ELBM simulation with an increased input viscosity to have fair comparisons with simulations exhibiting similar energy spectra small-scale slopes.

Besides, we have implemented the approximated ELBM eddy viscosity closure as the SGS in a PS LES code, we observed that it has similar inertial range statistics than a Smagorinsky model implemented with the same code. Even in this preliminary study ELBM exhibits a more extended inertial range and captures the non-linear intermittent physics slightly better than those closures. It means observe that the added physics of ELBM compared to a sole eddy viscosity leads to, in this preliminary study at least, improved inertial range statistics.

CHAPTER 6

ACCELERATING LATTICE BOLTZMANN FLOWS SIMULATION USING NVSHMEM MODEL FOR GPU-INITIATED COMMUNICATIONS

The top 500 list of the most powerful supercomputers has seen the rise of accelerated systems [74]. In, particular, NVIDIA V100 Graphics Processing Units (GPUs), which are capable of more than 5 tera floating-point operations (TFLOPS), provide a significant part of the computational power of the current two most powerful computer systems, Summit [75] and Sierra [76], which together make use of more than 40,000 of them. The success of GPUs in High-Performance Computing (HPC) is due to its inherent capacity to perform massively parallel calculations. A number of algorithms in scientific computing have been able to harness these computational capabilities. In particular, codes making use of a stencil are well ported on GPUs as they mostly require a set of local operation which needs to be performed similarly across a usually large number of nodes and requires interactions only between nearest neighbor nodes.

The Lattice Boltzmann Method (LBM) implementation on parallel architectures is of interest to allow the simulation of turbulent flows. As seen in Chapter 1, such flows require a very high-resolution computational grid because of their multi-scale nature. A number of works have been carried out to optimize LBM codes on single and multiple GPUs [77, 78, 79]. State-of-the-art implementations commonly offload computations to the GPUs and rely on the CPU to initiate communications between processes using Message-Passing Interface (MPI) calls [80].

Recently, NVSHMEM, a programming interface that implements the OpenSHMEM programming model across a cluster of peer-to-peer (P2P) connected NVIDIA GPUs was introduced. It provides an in-kernel interface that allows CUDA threads to access any location in symmetrically-distributed memory, thus making possible to initiate communication from the GPU [81]. The motivation behind NVSHMEM lies in the fact that depending on the CPU for communication limits strong scalability, *i.e.* how the time-to-solution of a fixed problem varies as the number of processors increases. This is an important metric for scientific codes as they run on systems relying on an increasing number of GPUs.

In this chapter, we present a work-in-progress of the performance scaling of a LBM code on multi-GPUs using MPI with the aim of using them as a reference to evaluate NVSHMEM implementations based on in-kernel communications. We have developed metaLBM [5], an open-source high-performance implementation of LBM that support multiple 2D and 3D lattice stencils and have been used to conduct high-resolution flow simulations using different turbulence models and spectral forces. metaLBM makes possible the use of any DdQq stencil, but in this work, we focus on the simulation of 3D flows using the D3Q19 lattice, for which a number of benchmarks have already been carried out [82, 83]. NVSHMEM is not yet available to the public, and for that reason, we will not be able to show comparative performance benchmarks. However, we will detail our NVSHMEM implementations, showing how its API can be used to implement GPU-centric communications. We will first detail the LBM algorithm in section 6.1. Then, the details of the GPU architecture and the DGX-1 systems used for benchmarks are presented in section 6.2. In section 6.3, we highlight the optimization featured by metaLBM on single GPU, while in section 6.4, we introduce all the studied CPU- and GPU-centric communication multi-GPUs implementations; we show preliminary results on the benchmark of the CPU-initiated MPI one in section 6.5; we conclude on the presented results and highlight the next steps in section 6.7.

6.1 The metaLBM software: algorithmic aspects

As described in chapter 1, the LBM equation describe the streaming and collision of particle distribution functions f_ℓ:

$$f_\ell(\boldsymbol{x} + \boldsymbol{c}_\ell \Delta_t, t + \Delta_t) = f_\ell(\boldsymbol{x}, t) - \frac{1}{\tau_0} \left[f_\ell(\boldsymbol{x}, t) - f_\ell^{eq}(\boldsymbol{x}, t) \right], \tag{6.1}$$

with τ_0 the dimensionless relaxation time and in this chapter, we take f^{eq} as the equilibrium distribution projected on the first two Hermite polynomials:

$$f_\ell^{\mathrm{eq}} = \rho t_\ell \left(1 + \frac{\boldsymbol{u} \cdot \boldsymbol{c}_\ell}{c_s^2} + \frac{\boldsymbol{u}\boldsymbol{u} : \boldsymbol{c}_\ell \boldsymbol{c}_\ell - c_s^2 |\boldsymbol{u}|^2}{2 c_s^4} \right), \tag{6.2}$$

where c_s the speed of sound in the lattice.

We study 3D flows using a lattice with 19 discrete velocities, the D3Q19. However, to depict algorithmic details in a clear manner, we will use in what follows diagrams representing behaviors for the 2D projection of the D3Q19, the D2Q9. Those two stencils are shown in Fig. 6.1.

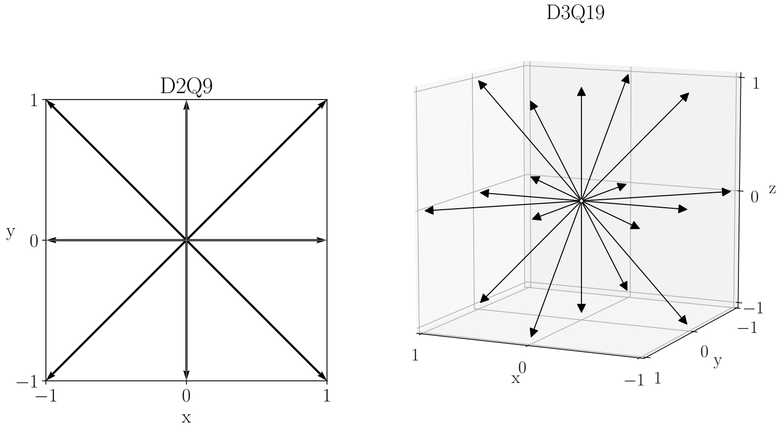


FIGURE 6.1. D2Q9 and D3Q19 lattice stencils. The D2Q9 stencil is given to illustrate all algorithmic concepts, while the D3Q19 is used in benchmarks.

After, initializing the distribution functions, Eq. (6.1) is iterated on for lattice node $\boldsymbol{x}$ and each discrete velocity direction ℓ until the required maximum iteration is reached. At each time step, boundary conditions are applied by copying appropriates distribution functions at the boundaries of the computational domain. Each iteration of Eq. (6.1) consists in two phases. The first one is the collision step, in which the post-collision distributions are calculated as

$$f_\ell^{\text{post}}(\boldsymbol{x},t) = f_\ell(\boldsymbol{x},t) - \frac{1}{\tau_0})-f_\ell^{eq}(\boldsymbol{x},t)\big] , \qquad (6.3)$$

and the second one is the streaming step

$$f_\ell(\boldsymbol{x}+\boldsymbol{c}_\ell\Delta_t, t+\Delta_t) = f_\ell^{\text{post}}(\boldsymbol{x},t). \qquad (6.4)$$

This straightforward algorithm is called the *push-algorithm* and is characterized by local read but non-local write operations. On GPUs, it is less computationally expensive to have local write but non-local read operations. Therefore, in the following we will adopt the *pull-algorithm* [84] for which the streaming step comes first and becomes

$$f_\ell^{\text{pre}}(\boldsymbol{x}, t+\Delta_t) = f_\ell(\boldsymbol{x}-\boldsymbol{c}_\ell\Delta_t, t-\Delta_t). \qquad (6.5)$$

In practice to minimize the read and write operations, the metaLBM algorithm iteratively calls a single step called streamAndCollide, which fuses the streaming and collision step together [85], as well as periodic boundary conditions kernels. A high-level summary of the implemented streamAndCollide functions called at each iteration is presented in Algorithm 3.

Algorithm 3 Fused streamAndCollide pull algorithm as implemented in metaLBM.

```
Set $\rho = 0$ and $\boldsymbol{u} = \mathbf{0}$
for each lattice node $\boldsymbol{x}$ do
    for each discrete velocity number $\ell$ do
        Store $f_\ell^{\text{pre}}(\boldsymbol{x}, t) = f_\ell(\boldsymbol{x} - \boldsymbol{c}_\ell \Delta_t, t - \Delta_t)$
        $\rho \mathrel{+}= f_\ell^{\text{pre}}(\boldsymbol{x}, t)$
        $\boldsymbol{u} \mathrel{+}= f_\ell^{\text{pre}}(\boldsymbol{x}, t)\, \boldsymbol{c}_\ell$
    end for
    $\boldsymbol{u} \mathrel{/}= \rho$
    Calculate and store $\mathrm{f}^{\text{neq}} = \mathrm{f}^{\text{pre}} - \mathrm{f}^{\text{eq}}$
    for each discrete velocity $\ell$ do
        Stream and collide $f_\ell(\boldsymbol{x}, t) = f_\ell^{\text{pre}}(\boldsymbol{x}, t) - \frac{1}{\tau_0} f_\ell^{\text{neq}}(\boldsymbol{x}, t)$
    end for
end for
```

6.2 GPU arthitecture

6.2.1 NVIDIA GPU characteristics

In this chapter, we will work with two generations of NVIDIA GPUs, *Pascal* and *Volta*. The main characteristics of the Pascal P100 and the Volta V100 GPUs are given in Table 6.1. Such GPUs have a massively parallel architecture and can perform a number of floating-point operations per second (FLOPS) of the order of 10^{13}.

GPUs are designed to process Single Instruction on Multiple Threads (SIMT). This is different from CPUs' Single Instruction, Multiple Data (SIMD) as the branch of instructions executed on each thread on the GPU can diverge at the expense of performance. NVIDIA GPUs processing units are called Streaming Multiprocessors (SMX), each of them combining compute units called CUDA cores. A CUDA thread is scheduled on a CUDA core and parallelism is commonly achieved by mapping each CUDA thread to one node of the computational grid. The scheduling of CUDA threads on CUDA cores is done by grouping 32 CUDA threads into warps. It allows latency hiding by putting on hold the warps waiting for data and executing the ones ready to run. This is a very important feature for our work. Indeed, one of the ideas behind NVSHMEM is to use these intrinsic latency hiding capabilities to hide remote direct memory access (RDMA) access from/to other memory physically located on another GPU.

The P100 GPU delivers a single-precision (SP) peak performance of 10.6 TFLOPS and 5.3 Tflops in double-precision (DP); The V100 GPU greatly improves on those numbers with a peak performance of 15.7 TFLOPS in SP and 7.8 in DP, corresponding to a $\approx 50\%$ increase of peak performance compared to the Pascal generation system.

Table 6.1. Features of the most NVIDIA GPUs used in this work: the Pascal P100 and the Volta V100. DP and and SP stands for double- and single-precision.

	Pascal P100	Volta V100
# Streaming Multiprocessors	56	80
# CUDA cores	3584	5120
Clock (GHz)	1.328	1.530
DP peak performance (GFLOPS)	5300	7800
SP peak performance (GFLOPS)	10600	15700
L2 cache (MB)	4.96	6.14
Global Memory (GB)	16	16
Memory bandwidth (GB/s)	732	900

The on-chip memory hierarchy of NVIDIA GPUs gets increasingly complex at each new generation. Overall, the large throughput of GPUs is backed up by significant memory bandwidth. The available global memory of the P100 and V100 GPUs is 16 GB and has a bandwidth of 732 and 900 GB for the P100 and the V100 GPU respectively. Modern GPUs are characterized by a number of memory levels from L2 cache to registers, which access times from the lowest to the highest level differs by up to two orders of magnitude.

From the GPU, access to the CPU memory (RAM) is very expensive. While the RAM is often much larger than the GPU memory, it is connected to it through a PCI express connection, with a bandwidth of only 16 GB/s. This implies that often CPU-GPU memory transfers can be the bottleneck of applications' performance. Therefore, one needs to ensure that the data required for computation remain as high as possible in the GPU memory hierarchy for as long as possible and that accesses to the RAM are avoided.

6.2.2 The DGX-1 cluster

In order to benchmark implementations based on NVSHMEM, we need a system of P2P-connected NVIDIA GPUs. As shown on Fig 6.2, the DGX-1 machine is an ideal machine for our single node experiments. It comes in two flavors: the DGX-1P composed of 8 P100 GPUs and the DGX-1P, which makes use of 8 V100 GPUs. In both cases, the GPUs are densely P2P-connected using NVIDIA NVLink connections. As GPUS are connected directly together, when data need to be transferred, it is no longer required to suffer from

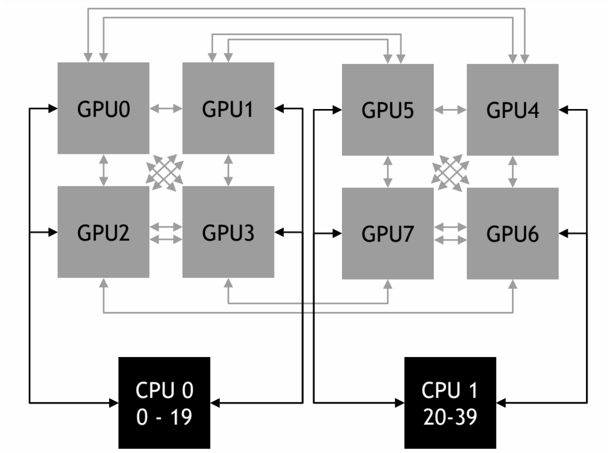


Figure 6.2. Network topology in a DGX-1 system. The PCIe connections are shown as black arrows and the NVLink connections are shown as green arrows. If the GPUs are P100s, the system is referred to as DGX-1P, while if they are V100s, it is referred to as DGX-1P. Taken from [86].

the low PCIe connection bandwidth going through the host CPUs. As a result, NVLink connections improves the bandwidth implied by GPU-GPU communications by a factor 10 to 160 GB/s compared to PCIe. Each GPU having four NVlink connections, the DGX-1 system links two groups of four fully-connected GPUs together in the shape of a hybrid-cube mesh. On the host side, the DGX-1 machine equips a dual-socket of Xeon E5-2698 v4, each of them having 20 cores. It should be noted that each CPU-GPU PCIe bandwidth is shared among two GPUs using PCIe switches, thus reducing further the memory bandwidth between the CPU and the GPU. In our application, this proved to strongly impact input/output (I/O) performances.

6.2.3 Programming NVIDIA GPUs

metaLBM code is written in an object-oriented fashion in `C++` and utilize a few `C++11` features as well as `C++` templates for genericity. In order to port it on GPUs, we make use of `CUDA C/C++`, an NVIDIA API for GPU programming, which, since version 7.5, is compatible with a number of `C++11` features. We offload all computationally heavy parts of the code to the devices (GPUs), while functions with small parallelism run on the host (CPUs). Parallel function running on the GPUs are denoted as kernels and identified using the *__global__* keyword. They can call device functions (keyword *__device__*), while host functions (keyword *__host__*) only runs on the CPUs only. To run a kernel, one needs to indicate how CUDA threads should be grouped in blocks and how blocks should be grouped in a grid. The grid is a one-, two-, or three-dimensional arrangement of blocks, which are themselves one-, two-, or three-dimensional arrangements of threads. The kernel terminates once, every thread of every block in the grid has executed all of its instructions.

It is important to notice that CUDA kernels run in parallel with CPU functions, so it is possible to overlap executions of functions on the host and of kernels on the device. This feature is usually exploited to overlap computation and communication in a multi-GPU setting (see section 6.4)

6.3 Single GPU implementation

6.3.1 Data dependency and access pattern

In metaLBM, we use the most common pattern of accessing the distribution functions: the AB-pattern. According to the AB access pattern, there are two copies of the distribution functions in global memory (A and B). At each iteration, the pointers to distribution functions A and distribution functions B is swapped and therefore, we alternatively read from A and write to B, and then read from B and write to A. This choice ensures that there is no data dependency involved in an iteration of the LBM algorithm and therefore make its parallelization straightforward. However, the global memory requirements implied by this access pattern are significant. Other LBM memory access patterns have been studied [85] and some of them allow to use only a single copy of the distributions in memory, while still allowing distributed-memory parallelism. The AA-pattern and the swap pattern are two of such memory access patterns, yet they come at the price of high complexity, especially in the generic case of a DdQq stencil, and therefore we did not explore them further.

As we saw, in the previous section, access to the CPU memory is very expensive and should be minimized to preserve performances. Nevertheless, research-oriented LBM codes such as metaLBM require to store and output a number of field arrays. In addition to the usual macroscopic arrays (velocity and density), we commonly need to store the distribution functions to create a checkpoint of the simulation and be able to restart the code. To minimize device memory requirements, we use host pinned memory. Apart from having the advantage to make the host memory directly reachable within a CUDA kernel, it also makes it possible to cut some of the cost of CPU-GPU memory copy [87].

6.3.2 Memory layout

The memory layout of the distribution functions array is typically based on either the array of structures (AoS) or the structure of arrays (SoA) schemes. On the one hand, the

AoS scheme, while reminding of object-oriented programming, is not suitable on GPUs as it does not provide the coalesced memory that is required for warps to perform grouped read/write operations. On the other hand, the SoA scheme is a popular choice [88, 89, 90] as for a given an index, populations of all lattice nodes can be written on a contiguous slab of memory. Those memory layouts are usually implemented as index functions and we give in Listing the index functions corresponding to AoS and SoA memory layouts.

```
size_t AoS_index_population(size_t index_node,
size_t dir) {
  return index_node * num_dir + dir;
}

size_t SoA_index_population(size_t index_node,
size_t dir) {
  return dir * volume + index_node;
}
```

FIGURE 6.3. Implementation of the AoS (top) and SoA (bottom) memory layouts. index_node is the node index in column-major order and dir is the discrete velocity number while num_dir is the total number of discrete velocities (for D3Q19, it is 19) and volume is the total number of nodes.

More complex memory layouts have also been suggested. For instance, in the Cluster SoA (CSoA) [91] memory layout for which a number, multiple of the number of threads in a warp (32), of consecutive elements of each population are grouped together to enable aligned read operations. However, while benchmarks showed significant improvements for Intel Accelerators, on GPUs, the improvements were strongly limited. As a result, the SoA memory layout was adopted in metaLBM.

6.3.3 Periodic boundary conditions

In order to allow an efficient implementation of the periodic boundary conditions, we surround the lattice by a halo as shown in Fig. 6.4. Therefore, each of the two distribution functions array allocated on the GPU have a size of $(L_x+2H_x)\times(L_y+2H_y)\times(L_z+2H_z)$. Since for the D3Q19 lattice, populations can move a maximum of a single point at each iteration, we take a halo thickness $H_x = H_y = H_z = 1$. In practice at the beginning of each iteration, we pull each distribution pointing outside of the computational domain to the corresponding halo node to allow computations at the boundary to happen seamlessly. For D3Q19 this means that for each site only five populations per boundary node need to be transferred, while three populations per boundary node need to be transferred for D2Q9.

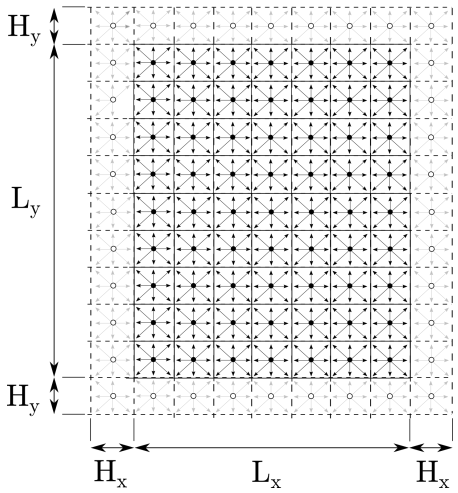


FIGURE 6.4. Two-dimensional D2Q9 visualization of the computational domain surrounded by a halo region as allocated on the GPU.

6.4 Multi-GPUs implementation

In practice, turbulent flows simulations require very large domains and are therefore limited by the amount of memory available on the GPU. To deal with this issue, we can run the simulations on multiple GPUs, distributing the global computational grid into smaller local domains. To do that, we create a one to one map between processes, called MPI processes for MPI or processing elements for NVSHMEM, and GPUs. We present two different approaches to GPU-GPU communications, one which relies on the host CPU to initiate communication, and the other solely based on the NVSHMEM API, which allows the GPU itself to initiate communications to P2P connected GPUs.

6.4.1 Domain partitioning

For three-dimensional flow simulations, the computational domain can be partitioned either across one, two or three dimensions. Each of these partitions forms a sub-domain allocated to one GPU. Going to higher partitioning dimensions have the advantage of being able to scale the communication time with the number of processes. However, this *a priori* does not impact the scaling at a low number of processes [92] and as we work on a single-node multi-GPUs (< 8) system, we adopt a 1D partitioning along the X-direction.

In order to enforce a 1D ring of N_p NVLink-connected GPUs, we can use *nvidia-smi topo -m* to get the DGX-1 system topology and force ordering of the GPU using the environmental variable $CUDA\_VISIBLE\_DEVICES$. As seen in Fig. 6.5, each local grid is also surrounded by a halo and allocated a local volume of $(L_x + 2H_x)/N_p \times$

$(L_y + 2H_y) \times (L_z + 2H_z)$ nodes. At the beginning of an iteration, GPUs exchange data for nodes close to the right and left boundaries of their sub-domains with the GPUs that are previous and next on the 1D ring. This implies that along the direction of the partitioning, periodic boundary conditions are automatically enforced.

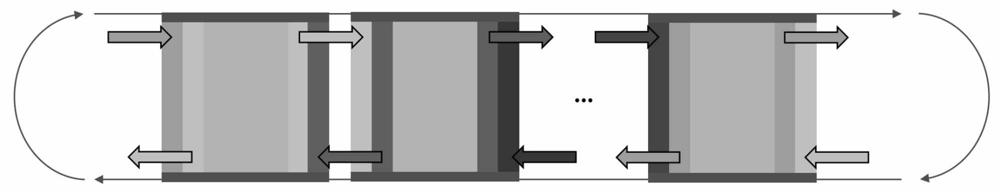


FIGURE 6.5. Two-dimensional D2Q9 visualization of the one-dimensional set of sub-domains on N_p GPUs virtually ordered along a ring. Adapted from Ref. [86].

6.4.2 CPU-initiated communications

CUDA+MPI model

The most common approach to port a single GPU code to multiple GPUs is to use the MPI communication library. Typically, to send distribution functions from one GPU to the next GPU on the 1D ring, one has to allocate a buffer, pack contiguously into this buffer the distribution functions to be sent, transfer the buffer to the CPU and send it with MPI communication routines. Once the buffer is received by the neighboring process, it is transferred to the receiving GPU and unpacked into the distribution functions array. As mentioned in section 6.2, communication between CPU and GPU are a common bottleneck of multi-GPUs implementation scaling. However, the situation can be partially dealt with on NVIDIA architectures using CUDA-Aware MPI [93]. It allows to directly use memory addresses located on the GPU as arguments of MPI communication routines. While it simplifies the code by allowing the ommitment of explicit CPU-GPU communications, it also improves performances by pipelining all the operations required in a data transfer and allowing RDMA communications.

Moreover, to further simplify MPI communications and improve efficiency, we can cut the packing/unpacking steps by noticing that, when the SoA memory layout is used, for each direction pointing outside of the local sub-domain at the boundary, the distribution functions for all nodes at the boundary are already coalesced slabs of memory. Therefore, at the cost of 3 (for D2Q9) or 5 (for D3Q19) MPI communication calls instead of one, the packing/unpacking procedure can be skipped as shown in Fig. 6.6. Moreover, we can further improve on this by making use of non-blocking send and receives communications *MPI_Isend / _Irecv*.

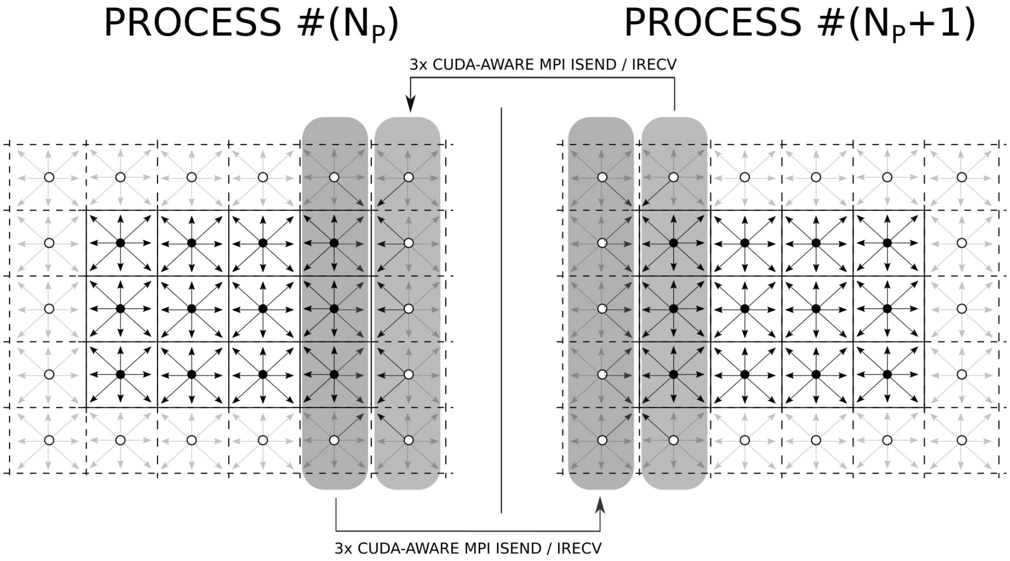

PROCESS #(N_P)
PROCESS #(N_P+1)
3x CUDA-AWARE MPI ISEND / IRECV
3x CUDA-AWARE MPI ISEND / IRECV

FIGURE 6.6. Two-dimensional D2Q9 visualization of the communication procedure at a single time step. For each direction pointing outside of the local domain. a slab of memory corresponding to distribution functions at that direction at all boundary nodes is sent and received to/from the next GPU on the virtual onde-dimensional ring. Adapted from Ref. [80] with permissions.

CUDA+NVSHMEM model

NVSHMEM API also supports CPU-side communication in a very similar manner than CUDA-Aware MPI and therefore we add this implementation to our comparative performance analysis. In order to use NVSHMEM for CPU-initiated communication we simply replace the *MPI_Isend / _Irecv* calls by the corresponding *shmem_putmem* NVSHMEM API calls. Alternatively, there is a similar API, *shmem_putmem_on_stream*, that offloads communications on a CUDA stream. For this implementation as well as others that use NVSHMEM, all GPU memory allocation is pinned GPU device memory allocated symmetrically on each GPU memory using the *shmem_alloc* API.

Communication and computation overlapping

As a kernel running on GPU devices and host-side function can run in parallel, it is possible to hide communications by overlapping them with computations through the use of CUDA streams. Indeed, as described in Fig. 6.7, after running the periodic boundary conditions on the Y- and Z-axis, we can launch on stream 1 the kernel to process the bulk of the sub-domain, *i.e.*, the sub-domain without the X-axis boundary nodes, and proceed with communications. Once communications are done, we launch a kernel on X-axis boundary nodes to process them as well.

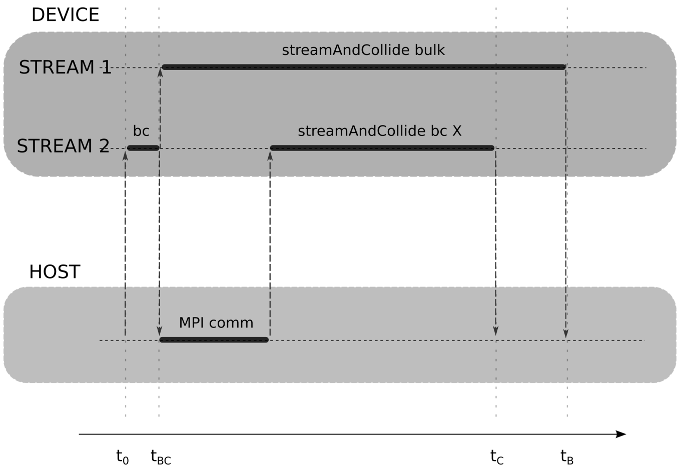


FIGURE 6.7. Timeline of the overlapping logic implemented in metaLBM. Adapted from Ref. [91] with permissions.

6.4.3 GPU-initiated communications

CUDA+NVSHMEM model

Once the memory is allocated symmetrically on all GPUs using *shmem_alloc*, it is possible to get, from any GPU, a pointer to the memory residing on its P2P-connected GPUs using *shmem_ptr*. At this point, from a practical perspective, communicating at the boundary along the X-axis is the same than applying periodic boundary condition in that direction. Therefore, the implementation of an NVSHMEM kernel is very simple. On top of that, having kernels that include both communication and computation enable overlapping between computation and communication using the GPU intrinsic capabilities based on warp scheduling. Moreover, if communications are handled by the device, there is no input needed from the host, but it stills launch the required kernels at each iteration. Launching kernel includes an overhead which can be skipped by writing a persistent kernel that takes care of iterating over time. Therefore this implementation, while having the advantage of being easy to code, could also result in better performances than a CPU-centric CUDA+MPI implementation with communication-computation overlapping.

6.5 Comparative performance benchmarks

We present in this section benchmarks on a single GPU and the weak and strong scaling of the reference CUDA+MPI implementation. In both cases, we use two systems: the DGX-1P, based on P100 GPUs and the DGX-1V, based on V100 GPUs.

6.6 Performance metrics

Performances of LBM codes are commonly evaluated based on an indicator called million lattice updates per second (MLUPS) defined as

$$\frac{V_0 \cdot n}{t \cdot 10^6}, \tag{6.6}$$

where t is the time required to perform n iterations of the LBM algorithm on V_0 nodes of the lattice.

In the following, we will also be interested in measuring times. First, the computation time, *i.e.* the cumulative time taken by the *streamAndCollide* kernel to proceed with all nodes in the case of no communication-computation overlapping or all nodes in the bulk, in the case of overlapping. Then in the case without overlapping, we also measure the communication time, which is the cumulative time taken to proceed with periodic boundary conditions and MPI communications and the processing of the streamAnd-Collide kernel on the boundary along the X-axis.

6.6.1 Single GPU

We first present a benchmark of the performance of metaLBM on a single GPU at increasing size of the domain along the X-axis while setting the size along the Y- and Z-axis to $L_y = L_z = 256$ for both a Pascal P100 GPU and a Volta V100 GPU. On Fig. 6.8, we observe that on domain such as $L_x > 64$, performances are constant reaching ≈ 1000 MLUPS on P100 and ≈ 1500 on V100. The theoretical increase of 50% in performances on the specs between the GPU of the Pascal generation to the one of the Volta generation is, therefore, also observed in practice. Below $L_x = 64$, we see that performances decrease quickly, most likely because kernels are faster to execute and we start seeing the overhead of starting kernels on the GPUs.

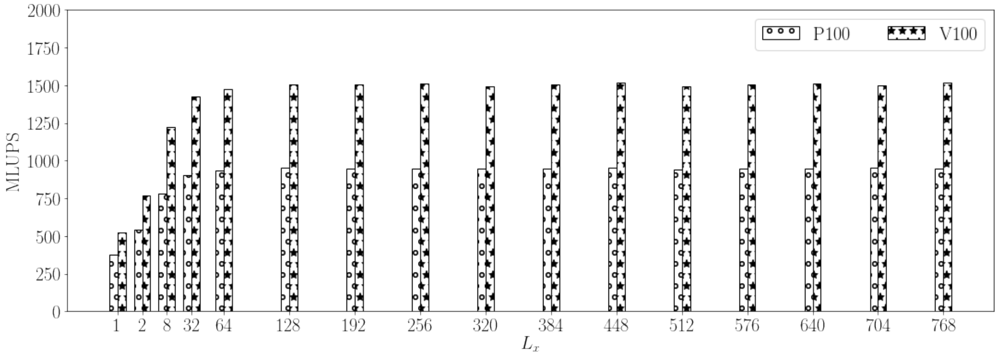


FIGURE 6.8. One-dimensional ring of a lattice on N_p GPUs virtually ordered along a ring.

6.6.2 Weak scaling

We measure the code performance at increasing the number of processes while maintaining the local domain size per process constant. This so-called *week scaling* experiment was conducted imposing a domain size of $V_0 = (N_p \cdot 480) \times 256 \times 256$, where N_p is the number of processes. We show on Figs. 6.9 and 6.10 the results of the weak scaling benchmark on the DGX-1P and the DGX-1V respectively. The benchmark on both systems looks very similar if we account for the performance differences observed on a single GPU. Overall, we observe a perfect scaling in the MPI with overlapping and a close to perfect scaling when overlapping is not implemented. From the measurement of the communication and computation times, it can be surprising than the weak scaling appears so good. However, here the communication times includes the time taken to process periodic boundary conditions, which, as the sub-domain treated on each GPU is the same for all, does not impact the results.

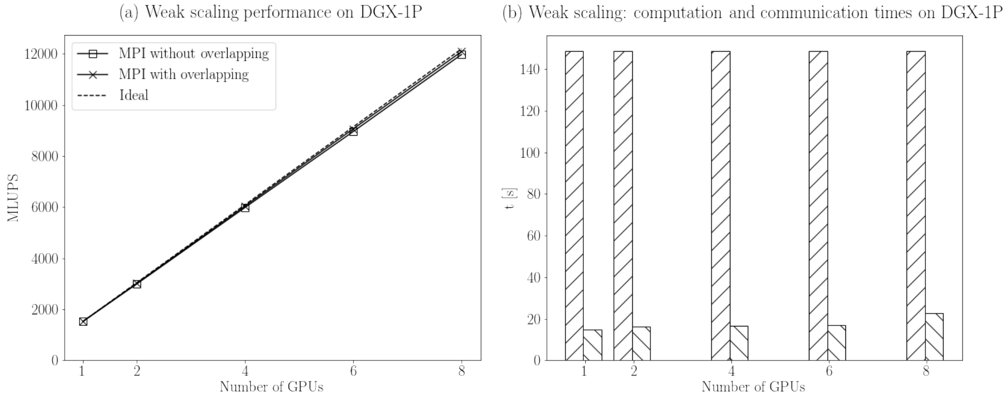


Figure 6.9. Weak scaling of the metaLBM code on a DGX-1P (8 P100 GPUs) for a domain of $(N_P \cdot 480) \times 256 \times 256$, where N_P is the number of processes (= number of GPUs). Panel (a) shows the performance scaling for the MPI implementations with and without overlapping, while Panel (b) shows the communicaiton and computation times for the implementation without overlapping.

6.6.3 Strong scaling

To assess how an application scale on multiple GPUs, one can also look at the code performance on a constant global domain size, while gradually increasing the number of GPUs. This measurement is denoted as *strong scaling* and performed on a computational domain of size $V_0 = 480 \times 256 \times 256$. On Figs. 6.11 and 6.12, we show the result of the strong scaling benchmark for the DGX-1P and the DGX-1V respectively. Again, except for the higher performances of V100 GPUs compared to P100 GPUs, the

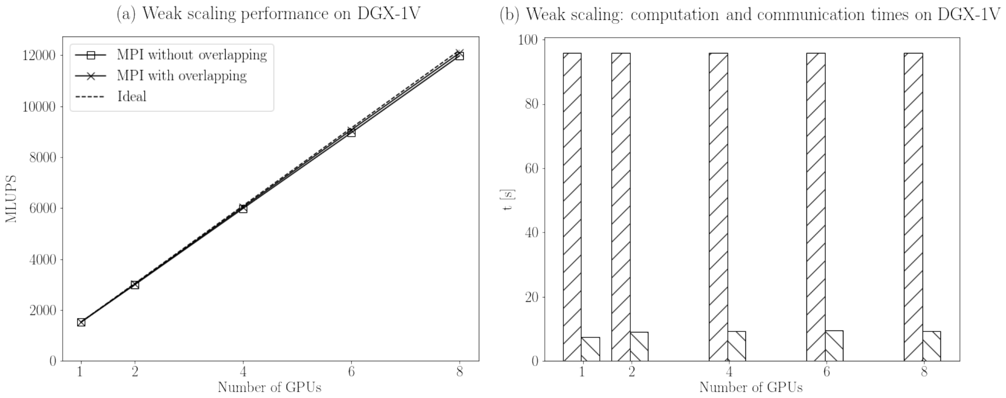


FIGURE 6.10. Weak scaling of the metaLBM code on a DGX-1V (8 V100 GPUs) for a domain of $(N_P \cdot 480) \times 256 \times 256$, where N_P is the number of processes (= number of GPUs). Panel (a) shows the performance scaling for the MPI implementations with and without overlapping, while Panel (b) shows the communicaiton and computation times for the implementation without overlapping.

results on both systems look similar. Here, we observe more clearly the impact of the fact that periodic boundary conditions are not overlapped. Indeed, while the communication times are much smaller than the computational times, we do not see a perfect scaling in the implementation involving communication-computation overlapping. This implementation, nevertheless, is still exhibiting better scaling than the non-overlapped implementation as MPI communications are properly hidden.

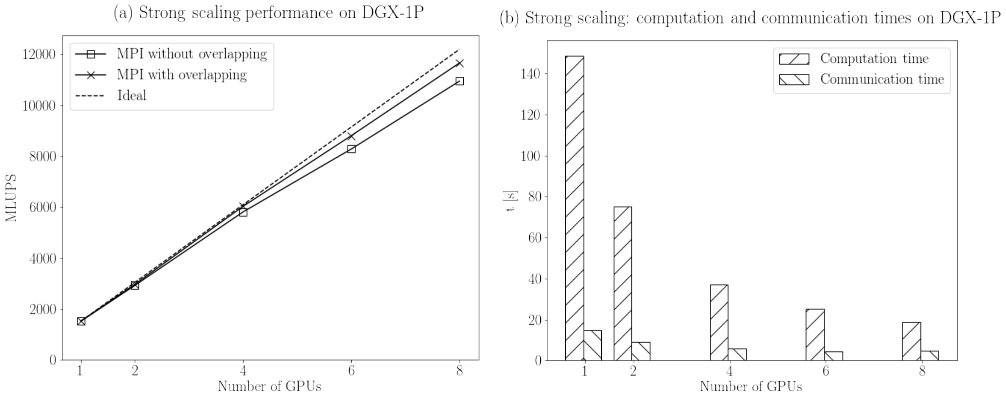


FIGURE 6.11. Strong scaling of the metaLBM code for a domain of $(N_P \cdot 480) \times 256 \times 256$, where N_P is the number of processes (= number of GPUs).

From the strong scaling results, we can infer the number of GPUs up to which our application should scale. Indeed, looking at the communication times for 8 GPUs, we see that they are still at least four times higher than the computation time, meaning

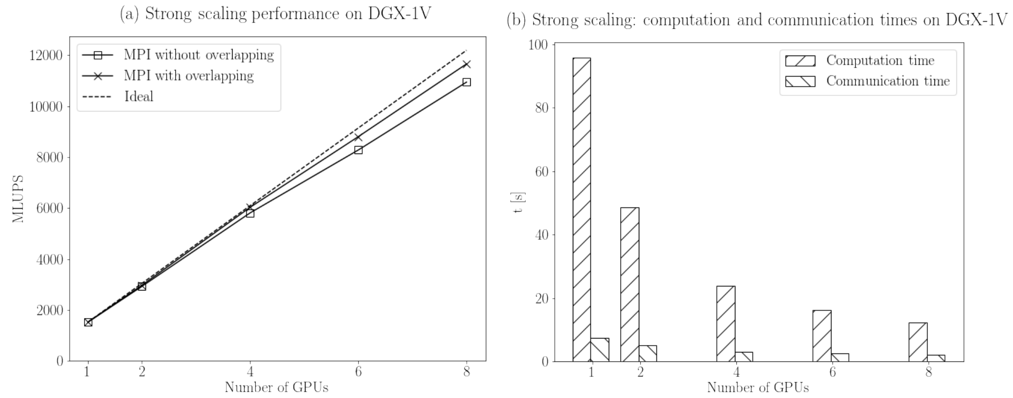


Figure 6.12. Strong scaling of the metaLBM code for a domain of $(N_P \cdot 480) \times 256 \times 256$, where N_P is the number of processes (= number of GPUs).

that the application could scale up to 32 GPUs. To further improve the scaling on more GPUs, it is then required to adopt a higher-dimensional domain partitioning. indeed, in 2D and 3D domain partitioning, communications times scales with the number of processes. However, for the present work, we deal with single node systems of 8 GPUs and therefore the 1D domain partitioning adopted here is perfectly suitable.

6.7 Concluding remarks & future work

In this chapter, we have presented a work-in-progress involving the benchmark of NVSHMEM, a new communication library that allows GPU to initiate data exchange across a set of P2P-connected GPUs. We have presented all implementations based on NVSHMEM for our code, metaLBM, and presented a detailed benchmark of a single GPU and MPI-based multi-GPUs implementations on DGX-1 systems composed of either Pascal or Volta generation GPUs. While results could be improved by also be overlapping periodic boundary conditions with computations, we have observed a near-to-perfect weak scaling and a very good strong scaling for implementation with and without communication-computation overlapping.

Based on those reference CUDA+MPI implementations, we can benchmark the performances of NVSHMEM implementations, for both standard and persistent kernels that allows saving performances from the overhead of repetitive kernel launches. NVSHMEM is yet only available for single node runs. When extended to multi-nodes, it could be interesting to explore further the scalings for a larger number of GPUs. This would also give us the opportunity to explore its performances in 2D or even 3D domain partitioning, for which the packing/unpacking required by MPI communications can no longer be skipped.

CHAPTER 7

CONCLUSION

7.1 Summary of the results

In this thesis, we shed some light on the implicit sub-grid scale modeling implied by the Entropic Lattice Boltzmann Method (ELBM) in the context of Homogeneous Isotropic Turbulence (HIT). The apparent unconditional stability of this sub-class of Lattice Boltzmann Method (LBM) is due to the enforcement of a H-theorem through the local calculation of an entropic parameter $\alpha(\boldsymbol{x}, t)$. ELBM has been put forward as an implicit Large-Eddy Simulation (LES) with an eddy viscosity Sub-Grid Scale (SGS) model, resulting from the assumption that the equation bridging the mesoscopic relaxation time with the macroscopic viscosity still holds when the relaxation time is fluctuating. Our first efforts focussed on numerically validating this assumption. To do that we developed a tool to assess the accuracy with which Navier-Stokes hydrodynamics is recovered by LBM and validated it against Pseudo-Spectral (PS) simulations of the Navier-Stokes Equations (NSE). Extend this tool with additional terms stemming from the eddy viscous dissipation implied by the fluctuating relaxation time, we applied it to a set of 2D and 3D HIT flows simulations of increasing Reynolds numbers. On the one hand, we could not clearly confirm, the assumption stating that ELBM can be macroscopically described as a LES with an eddy viscosity type SGS model. On the other hand, we have shown that ELBM extends the inertial range up to 20 times the Reynolds number of the last stable LBM simulation, demonstrating the existence of an implicit turbulence model while providing an upper limit of its range of validity.

In parallel for those two sets of simulations, we numerically assessed the hydrodynamic approximation of the assumed ELBM eddy viscosity from Ref. [2]. We find that it captures the dynamics of the ELBM eddy viscosity only at low Reynolds numbers, while it fails in fully developed turbulent regimes. To identify analytically the weak points of this approximation, we numerically check the assumptions made at every step of a C-E expansion of the entropic parameter $\alpha(\boldsymbol{x}, t)$. We explain why the macroscopic eddy viscosity formulation fails to be recovered at high Reynolds number by highlighting the presence of extra terms whose magnitude grows with the velocity gradients.

We performed a C-E expansion of ELBM at the second order showing that we recover the Navier-Stokes Equations (NSE) with only the constant zeroth order contribution of the entropic parameter $\alpha(\boldsymbol{x}, t) = 2 + \mathcal{O}(\Delta_t)$ appearing. Thus, we show that a third order C-E expansion is required to analytically observe the implicit ELBM model. Performing this higher-order expansion, we have shown that it is not only composed of an eddy viscous dissipation that depends on the effective relaxation time, but also of extra terms of the same order of magnitude. The latter result is of particular importance as it raises questions on the validity of many other eddy viscosity SGS models implemented in LBM.

The fact that the ELBM closure is more complex than a simple eddy viscosity model did not mean, that it was unable to model turbulence. Indeed, we have assessed the capability of the ELBM closure by comparing inertial range statistics of turbulent velocity fields obtained from an ELBM simulation, with those coming from a high-resolution Direct Numerical Simulation (DNS) of the NSE conducted with a PS code. First results showed that ELBM is able to increase the inertial scaling range and partially captures the correct intermittent behaviors. Intrigued by the interesting functional shape of the approximated eddy viscosity closure, we have shown, conducting the same inertial range statistical analysis, that it does not capture intermittency any better than a Smagorinsky model.

A part of this thesis was also dedicated to high-performance computing and this thesis work has required significant computing power. We developed an in-house open-source high-performance implementation of LBM [5] that support multiple 2D and 3D lattice stencils and can be used to conduct high-resolution flow simulations using different turbulence models and spectral HIT forcings. The code was optimized on multi-GPUs architectures by making use of NVSHMEM, a novel GPU-centric communication library. As a part of on-going work, this implementation is benchmarked against the standard CPU-centric communication implementation based on the Message-Passing Interface (MPI) library. We introduced the motivation behind this work and presented the results of the reference MPI implementation in terms of both strong and weak scaling properties, highlighting excellent scaling behaviors.

7.2 Future work

As a further step, it could be interesting to look at multi-relaxation times ELBM, the KBC family of methods [60]. Different flavors of those methods exist, but while they have been benchmarked in terms of mean flow properties, it is still unclear whether they are able to capture intermittent behaviors.

On the implementation side, ELBM was shown to be a computationally expensive algorithm, that is also intrinsically unadapted to GPUs. Indeed, the fact that at any lattice node the number of Newton-Raphson steps required to solve the entropic step equation can be different damages performances on massively parallel architectures. Indeed, it means that each CUDA thread can diverge causing the other CUDA threads within the same warp to wait. Also, each thread perform writes operations at each iteration of the Newton-Raphson algorithm and therefore, it might be worthwhile to use the Clustered Structure of Array [91] that allows aligned read operations or to make used of a shared memory-based tilling technique.

APPENDIX A

DERIVATION OF THE BALANCE EQUATIONS FROM THE WEAKLY COMPRESSIBLE NAVIER-STOKES

A.1 Kinetic energy balance equations

A.1.1 Derivation of the equation

To get the kinetic energy balance, we start by multiplying Eq. (1.45) by u_i and it reads (reapeted indices are meant summed upon)

$$\rho u_i \partial_t u_i + \rho u_j u_i \partial_j u_i = -c_s^2 u_i \partial_i \rho + \nu u_i \partial_j \left(\rho \left(\partial_j u_i + \partial_i u_j\right)\right) + u_i F_i. \tag{A.1}$$

Using the continuity Eq. (1.45), we can rewrite the L.H.S. of Eq. (A.1) as

$$\begin{aligned}
\rho u_i \partial_t u_i + \rho u_j u_i \partial_j u_i &= \rho \partial_t \frac{\boldsymbol{u}^2}{2} + \rho u_j \partial_j \frac{\boldsymbol{u}^2}{2} \\
&= \partial_t \frac{\rho \boldsymbol{u}^2}{2} - \frac{\boldsymbol{u}^2}{2} \partial_t \rho + \partial_j \frac{\rho \boldsymbol{u}^2}{2} u_j - \frac{\boldsymbol{u}^2}{2} \partial_j \left(\rho u_j\right) \\
&= \partial_t \frac{\rho \boldsymbol{u}^2}{2} + \partial_j \frac{\rho \boldsymbol{u}^2}{2} u_j - \frac{\boldsymbol{u}^2}{2} \left(\partial_t \rho + \partial_j \left(\rho u_j\right)\right) \\
&= \partial_t \frac{\rho \boldsymbol{u}^2}{2} + \partial_j \frac{\rho \boldsymbol{u}^2}{2} u_j.
\end{aligned}$$

We can also rewritte the second term of the R.H.S. of Eq. (A.1) as

$$\nu u_i \partial_j \left(\rho \left(\partial_j u_i + \partial_i u_j\right)\right) = \partial_j \left(\nu \rho u_i \left(\partial_j u_i + \partial_i u_j\right)\right) - \nu \rho \left(\partial_j u_i + \partial_i u_j\right) \partial_j u_i.$$

We obtain the balance equation for kinetic energy:

$$\begin{aligned}
\partial_t \frac{\rho \boldsymbol{u}^2}{2} =& \partial_j \left[-\frac{\rho \boldsymbol{u}^2}{2} u_j + \nu \rho u_i \left(\partial_j u_i + \partial_i u_j\right)\right] \\
& - c_s^2 u_i \partial_i \rho - \nu \rho \left(\partial_j u_i + \partial_i u_j\right) \partial_j u_i + u_i F_i.
\end{aligned} \tag{A.2}$$

A.1.2 Averaging over a subvolume

To check the balance equations, we average them over a volume V of surface ∂V. this volume can be, theoretically speaking any subvolume of the computational domain. We refer with $\langle \dots \rangle$ to the integration over V: $\iiint_V \dots \mathrm{d}V$. This gives us for the kinetic energy balance Eq. (A.2)

$$\begin{aligned}\langle \partial_t \frac{\rho \boldsymbol{u}^2}{2} \rangle = &\iiint_V \partial_j \left[-\frac{\rho \boldsymbol{u}^2}{2} u_j + \nu \rho u_i \left(\partial_j u_i + \partial_i u_j \right) \right] \mathrm{d}V \\ &- c_s^2 \langle u_i \partial_i \rho \rangle - \nu \langle \rho \left(\partial_j u_i + \partial_i u_j \right) \partial_j u_i \rangle + \langle u_i F_i \rangle.\end{aligned} \tag{A.3}$$

Using the divergence theorem, we get

$$\begin{aligned}\partial_t \langle \frac{\rho \boldsymbol{u}^2}{2} \rangle = &\oiint_{\partial V} \left[-\frac{\rho \boldsymbol{u}^2}{2} u_j + \nu \rho u_i \left(\partial_j u_i + \partial_i u_j \right) \right] n_j \, \mathrm{d}S \\ &- c_s^2 \langle u_i \partial_i \rho \rangle - \nu \langle \rho \left(\partial_j u_i + \partial_i u_j \right) \partial_j u_i \rangle + \langle u_i F_i \rangle.\end{aligned} \tag{A.4}$$

A.1.3 Case of a 2D system

When we integrate over the whole grid, $V = [0, L-1] \times [0, L-1]$, with periodic boundaries, the fluxes become 0 and the averaged kinetic energy balance Eq. (A.4) becomes

$$\partial_t \langle \frac{\rho \boldsymbol{u}^2}{2} \rangle = -c_s^2 \langle u_i \partial_i \rho \rangle - \nu \langle \rho \left(\partial_j u_i + \partial_i u_j \right) \partial_j u_i \rangle + \langle u_i F_i \rangle. \tag{A.5}$$

A.2 Enstrophy balance equations

A.2.1 Derivation of the equation

To get the enstrophy balance, we start by deriving the equation for the vorticity $\boldsymbol{\omega} = \boldsymbol{\nabla} \times \boldsymbol{u}$. In order to do that, we drop the index notation for the vector notation in Eq. (1.45) and divide by ρ to obtain

$$\frac{D\boldsymbol{u}}{Dt} = -c_s^2 \frac{1}{\rho} \boldsymbol{\nabla} + \frac{1}{\rho} \boldsymbol{\nabla} \cdot \left[\nu \rho \left(\boldsymbol{\nabla} \boldsymbol{u} + \left(\boldsymbol{\nabla} \boldsymbol{u} \right)^T \right) \right] + \frac{1}{\rho} \boldsymbol{F}. \tag{A.6}$$

Taking the curl of the L.H.S. of Eq. (A.6) holds

$$\begin{aligned}\boldsymbol{\nabla} \times \frac{D\boldsymbol{u}}{Dt} &= \boldsymbol{\nabla} \times \left(\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{\nabla} \frac{\boldsymbol{u}^2}{2} - \boldsymbol{u} \times \left(\boldsymbol{\nabla} \times \boldsymbol{u} \right) \right) \\ &= \frac{\partial}{\partial t} \left(\boldsymbol{\nabla} \times \boldsymbol{u} \right) + \boldsymbol{\nabla} \times \boldsymbol{\nabla} \left(\frac{\boldsymbol{u}^2}{2} \right) - \boldsymbol{\nabla} \times \left(\boldsymbol{u} \times \boldsymbol{\omega} \right) \\ &= \frac{\partial \boldsymbol{\omega}}{\partial t} + \boldsymbol{\omega} \left(\boldsymbol{\nabla} \cdot \boldsymbol{u} \right) - \left(\boldsymbol{\omega} \cdot \boldsymbol{\nabla} \right) \boldsymbol{u} + \left(\boldsymbol{u} \times \boldsymbol{\nabla} \right) \boldsymbol{\omega}.\end{aligned}$$

and taking the curl of the pressure gradient term in Eq. (A.6) we have

$$\begin{aligned}\boldsymbol{\nabla} \times c_s^2 \frac{1}{\rho} \boldsymbol{\nabla} \rho &= c_s^2 \frac{1}{\rho} \left(\boldsymbol{\nabla} \times \boldsymbol{\nabla} \rho \right) + c_s^2 \left(\boldsymbol{\nabla} \frac{1}{\rho} \right) \times \boldsymbol{\nabla} \rho \\ &= -c_s^2 \frac{1}{\rho^2} \boldsymbol{\nabla} \rho \times \boldsymbol{\nabla} \rho = 0.\end{aligned}$$

Thus, we obtained the vorticity equation

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + \boldsymbol{\omega} \left(\boldsymbol{\nabla} \cdot \boldsymbol{u} \right) - \left(\boldsymbol{\omega} \cdot \boldsymbol{\nabla} \right) \boldsymbol{u} + \left(\boldsymbol{u} \times \boldsymbol{\nabla} \right) \boldsymbol{\omega} = \boldsymbol{\nabla} \times \frac{1}{\rho} \boldsymbol{\nabla} \cdot \left[\nu \rho \left(\boldsymbol{\nabla} \boldsymbol{u} + \left(\boldsymbol{\nabla} \boldsymbol{u} \right)^T \right) \right] + \boldsymbol{\nabla} \times \frac{1}{\rho} \boldsymbol{F}. \tag{A.7}$$

To get the enstrophy balance equation, we take the scalar product of $\boldsymbol{\omega}$ with Eq. (A.7). Starting from the L.H.S., we go back to index notations and it reads

$$\begin{aligned}&\boldsymbol{\omega} \cdot \left\{ \frac{\partial \boldsymbol{\omega}}{\partial t} + \boldsymbol{\omega} \left(\boldsymbol{\nabla} \cdot \boldsymbol{u} \right) - \left(\boldsymbol{\omega} \cdot \boldsymbol{\nabla} \right) \boldsymbol{u} + \left(\boldsymbol{u} \times \boldsymbol{\nabla} \right) \boldsymbol{\omega} \right\} \\ &= \boldsymbol{\omega} \cdot \frac{\partial \boldsymbol{\omega}}{\partial t} + \boldsymbol{\omega} \cdot \boldsymbol{\omega} \left(\boldsymbol{\nabla} \cdot \boldsymbol{u} \right) - \boldsymbol{\omega} \cdot \left(\boldsymbol{\omega} \cdot \boldsymbol{\nabla} \right) \boldsymbol{u} + \boldsymbol{\omega} \cdot \left(\boldsymbol{u} \times \boldsymbol{\nabla} \right) \boldsymbol{\omega} \\ &= \partial_t \frac{\boldsymbol{\omega}}{2} + \omega_i u_j \partial_j \omega_i - \omega_i \omega_j \partial_j u_i + \omega_i \omega_i \partial_j u_j.\end{aligned}$$

However, as

$$\begin{aligned}\omega_i u_j \partial_j \omega_i &= \frac{1}{2} u_j \partial_j \left(\omega_i \omega_i \right) \\ &= \frac{1}{2} \partial_j \left(u_j \omega_i \omega_i \right) - \frac{1}{2} \omega_i \omega_i \partial_j u_j,\end{aligned}$$

we have

$$\boldsymbol{\omega} \cdot \left\{ \frac{\partial \boldsymbol{\omega}}{\partial t} + \boldsymbol{\omega} \left(\boldsymbol{\nabla} \cdot \boldsymbol{u} \right) - \left(\boldsymbol{\omega} \cdot \boldsymbol{\nabla} \right) \boldsymbol{u} + \left(\boldsymbol{u} \times \boldsymbol{\nabla} \right) \boldsymbol{\omega} \right\} = \partial_t \frac{\boldsymbol{\omega}^2}{2} + \partial_j \left(\frac{\boldsymbol{\omega}^2}{2} u_j \right) + \frac{\boldsymbol{\omega}^2}{2} \partial_j u_j - \omega_i \omega_j \partial_j u_i.$$

Setting $\boldsymbol{H} = \frac{1}{\rho} \boldsymbol{\nabla} \cdot \left[\rho \left(\boldsymbol{\nabla} \boldsymbol{u} + \left(\boldsymbol{\nabla} \boldsymbol{u} \right)^T \right) \right]$, we take the scalar product of $\boldsymbol{\omega}$ and the dissipation term of the R.H.S. of Eq. (A.7):

$$\begin{aligned}\boldsymbol{\omega} \cdot \left(\nu \boldsymbol{\nabla} \times \boldsymbol{H} \right) &= \nu \omega_i \epsilon_{ijk} \partial_j H_k \\ &= \nu \epsilon_{ijk} \left[\partial_j \left(\omega_i H_k \right) - H_k \partial_j \omega_i \right] \\ &= \partial_j \left(\nu \epsilon_{ijk} \omega_i H_k \right) + \nu \epsilon_{kji} H_k \partial_j \omega_i \\ &= \partial_j \left(\nu \epsilon_{ijk} \omega_i H_k \right) + \nu \boldsymbol{H} \cdot \left(\boldsymbol{\nabla} \times \boldsymbol{\omega} \right),\end{aligned}$$

where ϵ is the Levi-Civita symbol.

The enstrophy balance equation is:

$$\begin{aligned}\partial_t \frac{\boldsymbol{\omega}^2}{2} =& \partial_j \left[-\frac{\boldsymbol{\omega}^2}{2} u_j + \nu \epsilon_{ijk} \omega_i H_k \right] \\ & - \frac{\boldsymbol{\omega}^2}{2} \partial_j u_j + \omega_i \omega_j \partial_j u_i + \nu \boldsymbol{H} \cdot (\boldsymbol{\nabla} \times \boldsymbol{\omega}) + \boldsymbol{\omega} \cdot \left(\boldsymbol{\nabla} \times \frac{1}{\rho} \boldsymbol{F} \right)\end{aligned} \tag{A.8}$$

where $\boldsymbol{H} = \frac{1}{\rho} \boldsymbol{\nabla} \cdot \left[\rho \left(\boldsymbol{\nabla u} + (\boldsymbol{\nabla u})^T \right) \right]$

A.2.2 Averaging over a subvolume

Again, to check the balance equations, we average them over a volume V of surface ∂V. this volume can be, theoretically speaking any subvolume of the computational domain. Similarly, integrating the entropy balance Eq. (A.8) over V, we get

$$\begin{aligned}\left\langle \partial_t \frac{\boldsymbol{\omega}^2}{2} \right\rangle =& \iiint_V \partial_j \left[-\frac{\boldsymbol{\omega}^2}{2} u_j + \nu \epsilon_{ijk} \omega_i H_k \right] \mathrm{d}V \\ & - \left\langle \frac{\boldsymbol{\omega}^2}{2} \partial_j u_j \right\rangle + \left\langle \omega_i \omega_j \partial_j u_i \right\rangle + \nu \left\langle \boldsymbol{H} \cdot (\boldsymbol{\nabla} \times \boldsymbol{\omega}) \right\rangle + \left\langle \boldsymbol{\omega} \cdot \left(\boldsymbol{\nabla} \times \frac{1}{\rho} \boldsymbol{F} \right) \right\rangle.\end{aligned}$$

Using the divergence theorem, we get

$$\begin{aligned}\partial_t \left\langle \frac{\boldsymbol{\omega}^2}{2} \right\rangle =& \oiint_{\partial V} \left[-\frac{\boldsymbol{\omega}^2}{2} u_j + \nu \epsilon_{ijk} \omega_i H_k \right] n_j \, \mathrm{d}V \\ & - \left\langle \frac{\boldsymbol{\omega}^2}{2} \partial_j u_j \right\rangle + \left\langle \omega_i \omega_j \partial_j u_i \right\rangle + \nu \left\langle \boldsymbol{H} \cdot (\boldsymbol{\nabla} \times \boldsymbol{\omega}) \right\rangle + \left\langle \boldsymbol{\omega} \cdot \left(\boldsymbol{\nabla} \times \frac{1}{\rho} \boldsymbol{F} \right) \right\rangle,\end{aligned} \tag{A.9}$$

where $\boldsymbol{H} = \frac{1}{\rho} \boldsymbol{\nabla} \cdot \left[\rho \left(\boldsymbol{\nabla u} + (\boldsymbol{\nabla u})^T \right) \right]$

A.2.3 Case of a 2D system

In 2D, the vortex streching term is zero and the Levi-Civita symbol gets simpler, therefore the averaged enstrophy balance Eq. (A.9) becomes

$$\begin{aligned}\partial_t \left\langle \frac{\omega^2}{2} \right\rangle =& \oiint_{\partial V} \left[-\frac{\omega^2}{2} u_j + \nu \epsilon_{zjk} \omega H_k \right] n_j \, \mathrm{d}V \\ & - \left\langle \frac{\omega^2}{2} \partial_j u_j \right\rangle + \nu \left\langle H_x \partial_y \omega - H_y \partial_x \omega \right\rangle + \left\langle \omega \left(\partial_x \frac{F_y}{\rho} - \partial_y \frac{F_x}{\rho} \right) \right\rangle,\end{aligned} \tag{A.10}$$

where ω is the component of the vorticity vector along the z-axis, $\boldsymbol{\omega} = \omega \boldsymbol{e}_z$.

Moreover, when integrating over the whole grid, $V = [0, L-1] \times [0, L-1]$, with periodic boundaries, the fluxes become 0 and the averaged enstrophy balance Eq. (A.10) becomes

$$\partial_t \left\langle \frac{\omega^2}{2} \right\rangle = - \left\langle \frac{\omega^2}{2} \partial_j u_j \right\rangle + \nu \left\langle H_x \partial_y \omega - H_y \partial_x \omega \right\rangle + \left\langle \omega \left(\partial_x \frac{F_y}{\rho} - \partial_y \frac{F_x}{\rho} \right) \right\rangle. \tag{A.11}$$

APPENDIX B

CHAPMAN-ENSKOG EXPANSION FOR LBGK

To unroll the C-E expansion, we use the formalism introduced in section 1.3.4. The following notation will also be used for the n-th order isotropy relation

$$\boldsymbol{\Delta}^{(n)} = \sum_{\ell} t_\ell \overbrace{\boldsymbol{c}_\ell \dots \boldsymbol{c}_\ell}^{n}, \tag{B.1}$$

Moreover, for **L**, **M**, and **N**, three symmetric second-order tensors, we use the following notations to shorten the proof:

$$\begin{aligned}
([\mathbf{L}]_2)_{ij} &= L_{ij} + L_{ji}, \text{ meaning that } [\mathbf{A}]_2 \\
([\boldsymbol{u}\mathbf{L}]_2)_{ij} &= u_i L_{jk} + u_j L_{ik} + u_k L_{ij}, \text{ meanin} \\
([\mathbf{LM}]_3)_{ijkl} &= A_{ij} B_{kl} + A_{ik} B_{jl} + A_{il} B_{kj}, \text{ me} \\
\text{and } [\mathbf{LMN}]_{15} &\text{ is the sixth-order tensor created}
\end{aligned}$$

tensors with unique index combination of the three (symmetric) tensors.

(B.2)

We know that, thanks usual lattice possess second and third order isotropy, therefore we will assume that $\boldsymbol{\Delta}^{(2)} = c_s^2 \mathbf{1}$ and $\boldsymbol{\Delta}^{(4)} = c_\mathrm{s}^4 [\mathbf{11}]_3$, while the sixth order isotropy relation is written as

$$\boldsymbol{\Delta}^{(6)} = c_\mathrm{s}^6 [\mathbf{111}]_{15} + 6 c_\mathrm{s}^6 \boldsymbol{A}, \tag{B.3}$$

with $[\mathbf{111}]_{15} = [\mathbf{1}[\mathbf{11}]_3]_5$, where the extra contribution proportional to $\boldsymbol{A}$ are due to (eventual) lattice anisotropies.

B.1 First order: Euler equations

For $N = 1$, Eq. (1.41) gives

$$D_\ell f_\ell^{\mathrm{eq}} = -\alpha\beta f_\ell^{\mathrm{eq}} \phi_\ell + \mathcal{O}(\Delta_t). \tag{B.4}$$

Taking the zeroth order moment of Eq (B.4) leads straightforwardly to mass conservation

$$\partial_t \rho + \boldsymbol{\nabla} \cdot (\rho \boldsymbol{u}) = \mathcal{O}(\Delta_t), \tag{B.5}$$

and its first order moment gives the momentum balance

$$\partial_t (\rho \boldsymbol{u}) + \boldsymbol{\nabla} \cdot \sum_\ell \boldsymbol{c}_\ell \boldsymbol{c}_\ell f_\ell^{\mathrm{eq}} = \mathcal{O}(\Delta_t). \tag{B.6}$$

Since

$$\sum_\ell \boldsymbol{c}_\ell \boldsymbol{c}_\ell f_\ell^{\mathrm{eq}} = c_s^2 \rho \mathbf{1} + \rho \boldsymbol{u}\boldsymbol{u}, \tag{B.7}$$

momentum balance becomes

$$\partial_t (\rho \boldsymbol{u}) + \boldsymbol{\nabla} \cdot (\rho \boldsymbol{u}\boldsymbol{u}) = -\boldsymbol{\nabla}(c_s^2 \rho) + \mathcal{O}(\Delta_t). \tag{B.8}$$

By using the mass conservation equation Eq. (B.5), momentum balance can also be written as

$$\rho \partial_t \boldsymbol{u} + \rho \boldsymbol{u} \cdot \boldsymbol{\nabla} \boldsymbol{u} = -\boldsymbol{\nabla}(c_s^2 \rho) + \mathcal{O}(\Delta_t). \tag{B.9}$$

B.2 Second order: Athermal weakly compressible Navier-Stokes equations

Going to $N = 2$, Eq. (1.41) writes

$$D_\ell f_\ell^{\mathrm{eq}} + \frac{\Delta_t}{2} D_\ell^2 f_\ell^{\mathrm{eq}} = -\alpha\beta f_\ell^{\mathrm{eq}} \phi_\ell - \Delta_t D_\ell (f_\ell^{\mathrm{eq}} \phi_\ell) + \mathcal{O}(\Delta_t^2). \tag{B.10}$$

By applying the $D_\ell = \partial_t + \boldsymbol{c}_\ell \cdot \nabla$ operator, we get

$$D_\ell^2 f_\ell^{\mathrm{eq}} = -\frac{1}{\tau_0} D_\ell (f_\ell^{\mathrm{eq}} \phi_\ell) + \mathcal{O}(\Delta_t). \tag{B.11}$$

By inserting the latter in the former we obtain

$$D_\ell f_\ell^{\mathrm{eq}} = -\frac{1}{\tau_0} f_\ell^{\mathrm{eq}} \phi_\ell + \Delta_t D_\ell \left[\left(\frac{\alpha\beta}{2} - 1 \right) f_\ell^{\mathrm{eq}} \phi_\ell \right] + \mathcal{O}(\Delta_t^2), \tag{B.12}$$

Taking the zeroth order of Eq. (B.12), we get the mass conservation equation,

$$\partial_t \rho + \boldsymbol{\nabla} \cdot (\rho \boldsymbol{u}) = \mathcal{O}(\Delta_t^2), \tag{B.13}$$

and taking its first order moment, we obtain the momentum balance,

$$\partial_t (\rho \boldsymbol{u}) + \boldsymbol{\nabla} \cdot (\rho \boldsymbol{u}\boldsymbol{u}) = -\boldsymbol{\nabla}(c_s^2 \rho) + \Delta_t \boldsymbol{\nabla} \cdot \left[\left(\frac{1}{2\tau_0} - 1 \right) \sum_\ell \boldsymbol{c}_\ell \boldsymbol{c}_\ell f_\ell^{\mathrm{eq}} \phi_\ell \right] + \mathcal{O}(\Delta_t^2), \tag{B.14}$$

where (B.7) has been used. Starting from (B.4), we obtain

$$\begin{aligned}\sum_\ell \boldsymbol{c}_\ell \boldsymbol{c}_\ell f_\ell^{\mathrm{eq}} \phi_\ell &= -\tau_0 \sum_\ell \boldsymbol{c}_\ell \boldsymbol{c}_\ell D_\ell f_\ell^{\mathrm{eq}} + \mathcal{O}(\Delta_t) \\ &= -\tau_0 \left[\partial_t \left(c_s^2 \rho \mathbf{1} + \rho \boldsymbol{u}\boldsymbol{u} \right) + \boldsymbol{\nabla} \cdot \sum_\ell \boldsymbol{c}_\ell \boldsymbol{c}_\ell \boldsymbol{c}_\ell f_\ell^{\mathrm{eq}} \right] + \mathcal{O}(\Delta_t),\end{aligned} \tag{B.15}$$

where, again, we used (B.7). By using mass conservation and momentum balance at the order $N = 1$ (Eq. (B.5) and Eq. (B.8)), we have

$$\begin{aligned}\partial_t (c_s^2 \rho \mathbf{1} + \rho \boldsymbol{u}\boldsymbol{u}) &= c_s^2 \partial_t \rho \mathbf{1} + \partial_t (\rho \boldsymbol{u}) \boldsymbol{u} + \rho \boldsymbol{u} \partial_t \boldsymbol{u} \\ &= -c_s^2 \boldsymbol{\nabla} \cdot (\rho [\boldsymbol{u}\mathbf{1}]_3) + c_s^2 \rho [\boldsymbol{\nabla} \boldsymbol{u}]_2 - \boldsymbol{\nabla} \cdot (\rho \boldsymbol{u}\boldsymbol{u}\boldsymbol{u}) + \mathcal{O}(\Delta_t).\end{aligned} \tag{B.16}$$

Furthermore,

$$\begin{aligned}\sum_\ell \boldsymbol{c}_\ell \boldsymbol{c}_\ell \boldsymbol{c}_\ell f_\ell^{\mathrm{eq}} &= \rho \left[\frac{\boldsymbol{u} \cdot \boldsymbol{\Delta}^{(4)}}{c_s^2} + \frac{\boldsymbol{u}\boldsymbol{u}\boldsymbol{u} :\cdot \boldsymbol{\Delta}^{(6)} - 3 c_s^2 |\boldsymbol{u}|^2 \boldsymbol{u} \cdot \boldsymbol{\Delta}^{(4)}}{6 c_s^6} \right] \\ &= c_s^2 \rho [\boldsymbol{u}\mathbf{1}]_3 + \rho \boldsymbol{u}\boldsymbol{u}\boldsymbol{u} + \boldsymbol{A} :\cdot \rho \boldsymbol{u}\boldsymbol{u}\boldsymbol{u}.\end{aligned} \tag{B.17}$$

Therefore, we obtain

$$\sum_\ell \boldsymbol{c}_\ell \boldsymbol{c}_\ell f_\ell^{\mathrm{eq}} \phi_\ell = -\frac{1}{\alpha\beta} c_s^2 \rho [\boldsymbol{\nabla} \boldsymbol{u}]_2 - \frac{1}{\alpha\beta} \boldsymbol{A} :: \boldsymbol{\nabla} (\rho \boldsymbol{u}\boldsymbol{u}\boldsymbol{u}) + \mathcal{O}(\Delta_t). \tag{B.18}$$

Momentum balance Eq. (B.16) then becomes

$$\partial_t (\rho \boldsymbol{u}) + \boldsymbol{\nabla} \cdot (\rho \boldsymbol{u}\boldsymbol{u}) = -\boldsymbol{\nabla} (c_s^2 \rho) + \boldsymbol{\nabla} \cdot \left\{ \Delta_t \left(\frac{1}{\alpha\beta} - \frac{1}{2} \right) \left[c_s^2 \rho [\boldsymbol{\nabla} \boldsymbol{u}]_2 + \boldsymbol{A} :: \boldsymbol{\nabla} (\rho \boldsymbol{u}\boldsymbol{u}\boldsymbol{u}) \right] \right\} + \mathcal{O}(\Delta_t^2). \tag{B.19}$$

This allows us to write, in the isotropic case ($\boldsymbol{A} = \mathbf{0}$) or in the limit of low Mach numbers, the following expression kinematic viscosity ν_0:

$$\nu_0 = \Delta_t \left(\tau_0 - \frac{1}{2} \right) c_s^2 + \mathcal{O}(Ma^3) + \mathcal{O}(\Delta_t^2). \tag{B.20}$$

APPENDIX C

CHAPMAN-ENSKOG EXPANSION FOR ELBM

To perform this C-E expansion, we use the formalism introduced in section 1.3.4 as well as the tensorial notations described in Eq. (B.2)

The entropic parameter α is computed at each lattice node as the solution of

$$\sum_\ell f_\ell \ln\left(\frac{f_\ell}{t_\ell}\right) = \sum_\ell [\alpha f_\ell^{\mathrm{eq}} + (1-\alpha) f_\ell] \ln\left(\frac{\alpha f_\ell^{\mathrm{eq}} + (1-\alpha) f_\ell}{t_\ell}\right). \tag{C.1}$$

Notice that α is a parametric function of Δ_t through its dependence on f_ℓ.

C.1 Second order: Athermal weakly compressible Navier-Stokes equations

The momentum balance obtained at the second order Eq. (4.26) for a fixed relaxation time $\tau = \tau_0$ is also valid when $\tau = \tau_{\mathrm{eff}} = \frac{\alpha}{\beta}$. Therefore we have

$$\partial_t(\rho \boldsymbol{u}) + \boldsymbol{\nabla}\cdot(\rho \boldsymbol{u}\boldsymbol{u}) = -\boldsymbol{\nabla}(c_s^2 \rho) + \boldsymbol{\nabla}\cdot\left\{\Delta_t\left(\frac{1}{\alpha\beta} - \frac{1}{2}\right)\left[c_s^2 \rho [\boldsymbol{\nabla}\boldsymbol{u}]_2 + \boldsymbol{A} :: \boldsymbol{\nabla}(\rho \boldsymbol{u}\boldsymbol{u}\boldsymbol{u})\right]\right\} + \mathcal{O}(\Delta_t^2). \tag{C.2}$$

Therefore, the kinematic viscosity ν is given by:

$$\nu = \Delta_t\left(\frac{1}{\alpha\beta} - \frac{1}{2}\right) c_s^2 + \mathcal{O}(Ma^3) + \mathcal{O}(\Delta_t^2). \tag{C.3}$$

Notice that this expression can be considered correct up to $\mathcal{O}(\Delta_t)$. This actually means that α, which itself admits an expansion in Δ_t, has to be set equal to its leading order expression. In order to estimate the leading order of α, we rewrite Eq. (C.1), by using Eq. (1.42), as

$$\sum_\ell f_\ell^{\mathrm{eq}}(1+\Delta_t \phi_\ell) \ln\left(f_\ell^{\mathrm{eq}} \frac{1+\Delta_t \phi_\ell}{t_\ell}\right) - \sum_\ell f_\ell^{\mathrm{eq}}[1+(1-\alpha)\Delta_t \phi_\ell] \ln\left(f_\ell^{\mathrm{eq}} \frac{1+(1-\alpha)\Delta_t \phi_\ell}{t_\ell}\right) = 0, \tag{C.4}$$

and expand in Δ_t, to get

$$\alpha \sum_\ell f_\ell^{\mathrm{eq}} \phi_\ell \left(1 + \ln\left(\frac{f_\ell^{\mathrm{eq}}}{t_\ell}\right)\right) + \Delta_t \alpha \left(1 - \frac{\alpha}{2}\right) \sum_\ell f_\ell^{\mathrm{eq}} \phi_\ell^2 = \mathcal{O}(\Delta_t^2). \tag{C.5}$$

By excluding the trivial solution $\alpha = 0$ and simplifying (the first term vanishes), we obtain

$$\alpha = 2 + \mathcal{O}(\Delta_t), \tag{C.6}$$

and higher orders in the expansion of α are absorbed in $\mathcal{O}(\Delta_t^2)$ in Eq. (B.20), that is

$$\nu = \Delta_t \frac{1-\beta}{2\beta} c_s^2 + \mathcal{O}(Ma^3) + \mathcal{O}(\Delta_t^2) = \Delta_t \left(\tau_0 - \frac{1}{2}\right) c_s^2 + \mathcal{O}(Ma^3) + \mathcal{O}(\Delta_t^2). \tag{C.7}$$

C.2 Third order: Added contributions from the fluctuating entropic parameter

Therefore in order to observe the effect of the fluctuating entropic parameter α on the dynamics, it is necessary to go the third order C-E expansion. For $N = 3$ we obtain

$$D_\ell f_\ell^{\mathrm{eq}} + \frac{\Delta_t}{2} D_\ell^2 f_\ell^{\mathrm{eq}} + \frac{\Delta_t^2}{6} D_\ell^3 f_\ell^{\mathrm{eq}} = -\alpha\beta f_\ell^{\mathrm{eq}} \phi_\ell - \Delta_t D_\ell (f_\ell^{\mathrm{eq}} \phi_\ell) + \frac{\Delta_t^2}{2} D_\ell^2 (f_\ell^{\mathrm{eq}} \phi_\ell) + \mathcal{O}(\Delta_t^3). \tag{C.8}$$

The following relations will be systematically used:

$$\sum_\ell \overbrace{\boldsymbol{c}_\ell \ldots \boldsymbol{c}_\ell}^{n} D_\ell (\alpha f_\ell) = \partial_t \left(\alpha \sum_\ell \overbrace{\boldsymbol{c}_\ell \ldots \boldsymbol{c}_\ell}^{n} f_\ell\right) + \boldsymbol{\nabla} \cdot \left(\alpha \sum_\ell \overbrace{\boldsymbol{c}_\ell \ldots \boldsymbol{c}_\ell}^{n+1} f_\ell\right), \tag{C.9}$$

and

$$\begin{aligned} \sum_\ell \overbrace{\boldsymbol{c}_\ell \ldots \boldsymbol{c}_\ell}^{n} D_\ell^2 (\alpha f_\ell) = \partial_t^2 \left(\alpha \sum_\ell \overbrace{\boldsymbol{c}_\ell \ldots \boldsymbol{c}_\ell}^{n} f_\ell\right) + 2\partial_t \boldsymbol{\nabla} \cdot \left(\alpha \sum_\ell \overbrace{\boldsymbol{c}_\ell \ldots \boldsymbol{c}_\ell}^{n+1} f_\ell\right) \\ + \boldsymbol{\nabla}\boldsymbol{\nabla} : \left(\alpha \sum_\ell \overbrace{\boldsymbol{c}_\ell \ldots \boldsymbol{c}_\ell}^{n+2} f_\ell\right), \end{aligned} \tag{C.10}$$

By applying the D_ℓ operator we get

$$D_\ell^2 f_\ell^{\mathrm{eq}} + \frac{\Delta_t}{2} D_\ell^3 f_\ell^{\mathrm{eq}} = -\beta D_\ell (\alpha f_\ell^{\mathrm{eq}} \phi_\ell) - \Delta_t D_\ell^2 (f_\ell^{\mathrm{eq}} \phi_\ell) + \mathcal{O}(\Delta_t^2). \tag{C.11}$$

while a second application gives

$$D_\ell^3 f_\ell^{\mathrm{eq}} = -\beta D_\ell^2 (\alpha f_\ell^{\mathrm{eq}} \phi_\ell) + \mathcal{O}(\Delta_t). \tag{C.12}$$

By inserting the last two in the first one we obtain

$$D_\ell f_\ell^{\mathrm{eq}} = -\alpha\beta f_\ell^{\mathrm{eq}} \phi_\ell + \Delta_t D_\ell \left[\left(\frac{\alpha\beta}{2} - 1\right) f_\ell^{\mathrm{eq}} \phi_\ell\right] - \Delta_t^2 \frac{\beta}{12} D_\ell^2 (\alpha f_\ell^{\mathrm{eq}} \phi_\ell) + \mathcal{O}(\Delta_t^3). \tag{C.13}$$

Taking the zeroth order moment of Eq. (C.13) leads to mass conservation

$$\partial_t \rho + \boldsymbol{\nabla} \cdot (\rho \boldsymbol{u}) = -\Delta_t^2 \frac{\beta}{6} \boldsymbol{\nabla}\boldsymbol{\nabla} : \sum_\ell \boldsymbol{c}_\ell \boldsymbol{c}_\ell f_\ell^{\text{eq}} \phi_\ell + \mathcal{O}(\Delta_t^3), \tag{C.14}$$

and its first order moment gives the momentum balance

$$\begin{aligned} \partial_t(\rho\boldsymbol{u}) + \boldsymbol{\nabla} \cdot (\rho\boldsymbol{u}\boldsymbol{u}) = -\boldsymbol{\nabla}(c_s^2\rho) &+ \Delta_t \boldsymbol{\nabla} \cdot \left[\left(\frac{\alpha\beta}{2} - 1 \right) \sum_\ell \boldsymbol{c}_\ell \boldsymbol{c}_\ell f_\ell^{\text{eq}} \phi_\ell \right] \\ &- \Delta_t^2 \frac{\beta}{6} \boldsymbol{\nabla} \cdot \left[2\partial_t \sum_\ell \boldsymbol{c}_\ell \boldsymbol{c}_\ell f_\ell^{\text{eq}} \phi_\ell + \boldsymbol{\nabla} \cdot \sum_\ell \boldsymbol{c}_\ell \boldsymbol{c}_\ell \boldsymbol{c}_\ell f_\ell^{\text{eq}} \phi_\ell \right] + \mathcal{O}(\Delta_t^3), \end{aligned} \tag{C.15}$$

where Eqs. (B.7) and (C.6) were used. Further using Eq. (B.18), mass conservation Eq.(C.14) becomes

$$\partial_t \rho + \boldsymbol{\nabla} \cdot (\rho \boldsymbol{u}) = \frac{\Delta_t^2}{12} \boldsymbol{\nabla}\boldsymbol{\nabla} : \left[c_s^2 \rho [\boldsymbol{\nabla}\boldsymbol{u}]_2 + \boldsymbol{A} :: \boldsymbol{\nabla}(\rho\boldsymbol{u}\boldsymbol{u}\boldsymbol{u}) \right] + \mathcal{O}(\Delta_t^3). \tag{C.16}$$

Starting from Eq. (C.13), we obtain

$$\begin{aligned} \sum_\ell \boldsymbol{c}_\ell \boldsymbol{c}_\ell f_\ell^{\text{eq}} \phi_\ell &= -\frac{1}{\alpha\beta} \sum_\ell \boldsymbol{c}_\ell \boldsymbol{c}_\ell D_\ell f_\ell^{\text{eq}} + \Delta_t \frac{\beta - 1}{2\beta} \sum_\ell \boldsymbol{c}_\ell \boldsymbol{c}_\ell D_\ell (f_\ell^{\text{eq}} \phi_\ell) + \mathcal{O}(\Delta_t^2) \\ &= -\frac{1}{\alpha\beta} \left[\partial_t \left(c_s^2 \rho \boldsymbol{1} + \rho\boldsymbol{u}\boldsymbol{u} \right) + \boldsymbol{\nabla} \cdot \left(c_s^2 \rho [\boldsymbol{u}\boldsymbol{1}]_3 + \rho\boldsymbol{u}\boldsymbol{u}\boldsymbol{u} + \boldsymbol{A} :\cdot \rho\boldsymbol{u}\boldsymbol{u}\boldsymbol{u} \right) \right] \\ &\quad + \Delta_t \frac{\beta - 1}{2\beta} \left[\partial_t \sum_\ell \boldsymbol{c}_\ell \boldsymbol{c}_\ell f_\ell^{\text{eq}} \phi_\ell + \boldsymbol{\nabla} \cdot \sum_\ell \boldsymbol{c}_\ell \boldsymbol{c}_\ell \boldsymbol{c}_\ell f_\ell^{\text{eq}} \phi_\ell \right] + \mathcal{O}(\Delta_t^2), \end{aligned} \tag{C.17}$$

where Eqs. (B.7), (B.17) and (C.6) have been used. Consequently, momentum balance becomes

$$\begin{aligned} \partial_t(\rho\boldsymbol{u}) + \boldsymbol{\nabla} \cdot (\rho\boldsymbol{u}\boldsymbol{u}) &= -\boldsymbol{\nabla}(c_s^2\rho) \\ &\quad + \Delta_t \boldsymbol{\nabla} \cdot \left[\left(\frac{1}{\alpha\beta} - \frac{1}{2} \right) \left[\partial_t \left(c_s^2 \rho \boldsymbol{1} + \rho\boldsymbol{u}\boldsymbol{u} \right) + \boldsymbol{\nabla} \cdot \left(c_s^2 \rho [\boldsymbol{u}\boldsymbol{1}]_3 + \rho\boldsymbol{u}\boldsymbol{u}\boldsymbol{u} + \boldsymbol{A} :\cdot \rho\boldsymbol{u}\boldsymbol{u}\boldsymbol{u} \right) \right] \right] \\ &\quad + \Delta_t^2 \frac{(\beta - 1)^2}{2\beta} \boldsymbol{\nabla} \cdot \left[\partial_t \sum_\ell \boldsymbol{c}_\ell \boldsymbol{c}_\ell f_\ell^{\text{eq}} \phi_\ell + \boldsymbol{\nabla} \cdot \sum_\ell \boldsymbol{c}_\ell \boldsymbol{c}_\ell \boldsymbol{c}_\ell f_\ell^{\text{eq}} \phi_\ell \right] \\ &\quad - \Delta_t^2 \frac{\beta}{6} \boldsymbol{\nabla} \cdot \left[2\partial_t \sum_\ell \boldsymbol{c}_\ell \boldsymbol{c}_\ell f_\ell^{\text{eq}} \phi_\ell + \boldsymbol{\nabla} \cdot \sum_\ell \boldsymbol{c}_\ell \boldsymbol{c}_\ell \boldsymbol{c}_\ell f_\ell^{\text{eq}} \phi_\ell \right] + \mathcal{O}(\Delta_t^3), \end{aligned} \tag{C.18}$$

where Eq. (C.6) has been used. By using mass and momentum conservation equations at the order $N = 2$ (Eqs. (B.13) and (4.26)), we have

$$
\begin{aligned}
\partial_t(c_s^2\rho\mathbf{1} + \rho\boldsymbol{u}\boldsymbol{u}) &= c_s^2\partial_t\rho\mathbf{1} + \partial_t(\rho\boldsymbol{u})\boldsymbol{u} + \rho\boldsymbol{u}\partial_t\boldsymbol{u} \\
&= -c_s^2\boldsymbol{\nabla}\cdot(\rho[\boldsymbol{u}\mathbf{1}]_3) + c_s^2\rho[\boldsymbol{\nabla}\boldsymbol{u}]_2 - \boldsymbol{\nabla}\cdot(\rho\boldsymbol{u}\boldsymbol{u}\boldsymbol{u}) \\
&\quad - \Delta_t\frac{\beta-1}{2\beta}\left\{c_s^2[\boldsymbol{\nabla}\cdot(\rho[\boldsymbol{\nabla}\boldsymbol{u}]_2)\,\boldsymbol{u}]_2 + [\boldsymbol{A}\,::\!\cdot\,\boldsymbol{\nabla}\boldsymbol{\nabla}(\rho\boldsymbol{u}\boldsymbol{u}\boldsymbol{u})\,\boldsymbol{u}]_2\right\} + \mathcal{O}(\Delta_t^2),
\end{aligned}
\tag{C.19}
$$

where Eq. (C.6) has been used. Momentum balance then becomes

$$
\begin{aligned}
\partial_t(\rho\boldsymbol{u}) + \boldsymbol{\nabla}\cdot(\rho\boldsymbol{u}\boldsymbol{u}) = -\boldsymbol{\nabla}(c_s^2\rho)& \\
&+ \Delta_t\boldsymbol{\nabla}\cdot\left\{\left(\frac{1}{\alpha\beta} - \frac{1}{2}\right)\left[c_s^2\rho[\boldsymbol{\nabla}\boldsymbol{u}]_2 + \boldsymbol{A} :: \boldsymbol{\nabla}(\rho\boldsymbol{u}\boldsymbol{u}\boldsymbol{u})\right]\right\} \\
&+ \Delta_t^2\frac{(\beta-1)^2}{4\beta^2}c_s^2\boldsymbol{\nabla}\cdot[\boldsymbol{\nabla}\cdot(\rho[\boldsymbol{u}\boldsymbol{\nabla}\boldsymbol{u}]_6) - \boldsymbol{\nabla}\cdot(\rho\boldsymbol{u}[\boldsymbol{\nabla}\boldsymbol{u}]_2)] \\
&+ \Delta_t^2\frac{(\beta-1)^2}{4\beta^2}\boldsymbol{\nabla}\cdot[\boldsymbol{A}\,::\!\cdot\,\boldsymbol{\nabla}\boldsymbol{\nabla}(\rho\boldsymbol{u}\boldsymbol{u}\boldsymbol{u})\,\boldsymbol{u}]_2 \\
&+ \Delta_t^2\frac{(\beta-1)^2}{2\beta}\boldsymbol{\nabla}\cdot\left[\partial_t\sum_\ell\boldsymbol{c}_\ell\boldsymbol{c}_\ell f_\ell^{\mathrm{eq}}\phi_\ell + \boldsymbol{\nabla}\cdot\sum_\ell\boldsymbol{c}_\ell\boldsymbol{c}_\ell\boldsymbol{c}_\ell f_\ell^{\mathrm{eq}}\phi_\ell\right] \\
&- \Delta_t^2\frac{\beta}{6}\boldsymbol{\nabla}\cdot\left[2\partial_t\sum_\ell\boldsymbol{c}_\ell\boldsymbol{c}_\ell f_\ell^{\mathrm{eq}}\phi_\ell + \boldsymbol{\nabla}\cdot\sum_\ell\boldsymbol{c}_\ell\boldsymbol{c}_\ell\boldsymbol{c}_\ell f_\ell^{\mathrm{eq}}\phi_\ell\right] + \mathcal{O}(\Delta_t^3),
\end{aligned}
\tag{C.20}
$$

where we used again Eq. (C.6). A derivation of Eq. (B.18) with respect to time gives

$$
\begin{aligned}
\partial_t\sum_\ell\boldsymbol{c}_\ell\boldsymbol{c}_\ell f_\ell^{\mathrm{eq}}\phi_\ell &= -\frac{1}{2\beta}c_s^2\left(\partial_t\rho[\boldsymbol{\nabla}\boldsymbol{u}]_2 + \rho[\boldsymbol{\nabla}\partial_t\boldsymbol{u}]_2\right) - \frac{1}{2\beta}\boldsymbol{A} :: \boldsymbol{\nabla}\left[\partial_t(\rho\boldsymbol{u})\boldsymbol{u}\boldsymbol{u} + 2\rho\boldsymbol{u}\boldsymbol{u}\partial_t\boldsymbol{u}\right] + \mathcal{O}(\Delta_t) \\
&= \frac{1}{2\beta}c_s^2\left[\boldsymbol{\nabla}\cdot(\rho\boldsymbol{u}[\boldsymbol{\nabla}\boldsymbol{u}]_2) + \rho[\boldsymbol{\nabla}\boldsymbol{u}\cdot\boldsymbol{\nabla}\boldsymbol{u}]_2 + 2c_s^2\rho\boldsymbol{\nabla}\boldsymbol{\nabla}\ln\rho\right] \\
&\quad + \frac{1}{2\beta}\boldsymbol{A} :: \left[\boldsymbol{\nabla}\boldsymbol{\nabla}\cdot(\rho\boldsymbol{u}\boldsymbol{u}\boldsymbol{u}\boldsymbol{u}) + 3c_s^2\boldsymbol{\nabla}\boldsymbol{\nabla}(\rho\boldsymbol{u}\boldsymbol{u}) - 6c_s^2\boldsymbol{\nabla}(\rho\boldsymbol{u}\boldsymbol{\nabla}\boldsymbol{u})\right] + \mathcal{O}(\Delta_t).
\end{aligned}
\tag{C.21}
$$

where Eqs. (C.6), together with mass conservation Eq. (B.5) and momentum balance Eq. (B.8) at the order $N = 1$ were used. Starting from Eq. (C.13), we obtain

$$
\begin{aligned}
\sum_\ell\boldsymbol{c}_\ell\boldsymbol{c}_\ell\boldsymbol{c}_\ell f_\ell^{\mathrm{eq}}\phi_\ell &= -\frac{1}{2\beta}\sum_\ell\boldsymbol{c}_\ell\boldsymbol{c}_\ell\boldsymbol{c}_\ell D_\ell f_\ell^{\mathrm{eq}} + \mathcal{O}(\Delta_t) \\
&= -\frac{1}{2\beta}\left[\partial_t(c_s^2\rho[\boldsymbol{u}\mathbf{1}]_3 + \rho\boldsymbol{u}\boldsymbol{u}\boldsymbol{u} + \boldsymbol{A} :\!\cdot\,\rho\boldsymbol{u}\boldsymbol{u}\boldsymbol{u}) + \boldsymbol{\nabla}\cdot\sum_\ell\boldsymbol{c}_\ell\boldsymbol{c}_\ell\boldsymbol{c}_\ell\boldsymbol{c}_\ell f_\ell^{\mathrm{eq}}\right] + \mathcal{O}(\Delta_t),
\end{aligned}
\tag{C.22}
$$

where, again, Eqs. (C.6), mass conservation Eq. (B.5) and momentum balance Eq. (B.8) at the order $N = 1$ were used, we have

$$\begin{aligned}\partial_t(c_s^2\rho[\boldsymbol{u}\mathbf{1}]_3 + \rho\boldsymbol{uuu} + \boldsymbol{A} :\cdot \rho\boldsymbol{uuu}) &= c_s^2[\partial_t(\rho\boldsymbol{u})\mathbf{1}]_3 + \partial_t(\rho\boldsymbol{u})\boldsymbol{uu} + \rho\boldsymbol{u}[\boldsymbol{u}\partial_t\boldsymbol{u}]_2 \\ &\quad + \boldsymbol{A} :\cdot [\partial_t(\rho\boldsymbol{u})\boldsymbol{uu} + 2\rho\boldsymbol{uu}\partial_t\boldsymbol{u}] \\ &= -c_s^2[\boldsymbol{\nabla}\cdot(\rho\boldsymbol{uu})\,\mathbf{1}]_3 - c_s^2[\boldsymbol{\nabla}(\rho\boldsymbol{uu})]_3 + c_s^2\rho[\boldsymbol{u}\boldsymbol{\nabla}\boldsymbol{u}]_6 \\ &\quad - \boldsymbol{\nabla}\cdot(c_\mathrm{s}^4\rho[\mathbf{11}]_3) - \boldsymbol{\nabla}\cdot(\rho\boldsymbol{uuuu}) + \mathcal{O}(\Delta_t) \\ &\quad - \boldsymbol{A} :\cdot [\boldsymbol{\nabla}\cdot(\rho\boldsymbol{uuuu}) + 3c_s^2\boldsymbol{\nabla}(\rho\boldsymbol{uu}) - 6c_s^2\rho\boldsymbol{u}\boldsymbol{\nabla}\boldsymbol{u}],\end{aligned} \tag{C.23}$$

Furthermore,

$$\begin{aligned}\sum_\ell \boldsymbol{c}_\ell\boldsymbol{c}_\ell\boldsymbol{c}_\ell\boldsymbol{c}_\ell f_\ell^{\mathrm{eq}} &= \rho\left(1 - \frac{|\boldsymbol{u}|^2}{2c_s^2}\right)\boldsymbol{\Delta}^{(4)} + \frac{\rho\boldsymbol{uu}}{2c_s^4} : \boldsymbol{\Delta}^{(6)} \\ &= c_\mathrm{s}^4\rho[\mathbf{11}]_3 + c_s^2\rho[\boldsymbol{uu}\mathbf{1}]_6 + 3c_s^2\boldsymbol{A} : \rho\boldsymbol{uu}.\end{aligned} \tag{C.24}$$

Therefore, since $\boldsymbol{\nabla}\cdot(\rho[\boldsymbol{uu}\mathbf{1}]_6) = [\boldsymbol{\nabla}\cdot(\rho\boldsymbol{uu})\mathbf{1}]_3 + [\boldsymbol{\nabla}(\rho\boldsymbol{uu})]_3$, we obtain

$$\begin{aligned}\sum_\ell \boldsymbol{c}_\ell\boldsymbol{c}_\ell\boldsymbol{c}_\ell f_\ell^{\mathrm{eq}}\phi_\ell &= -\frac{1}{2\beta}\sum_\ell \boldsymbol{c}_\ell\boldsymbol{c}_\ell\boldsymbol{c}_\ell D_\ell f_\ell^{\mathrm{eq}} + \mathcal{O}(\Delta_t) \\ &= -\frac{1}{2\beta}\left\{c_s^2\rho[\boldsymbol{u}\boldsymbol{\nabla}\boldsymbol{u}]_6 - \boldsymbol{\nabla}\cdot(\rho\boldsymbol{uuuu})\right\} \\ &\quad - \frac{1}{2\beta}\boldsymbol{A} :\cdot [6c_s^2\rho\boldsymbol{u}\boldsymbol{\nabla}\boldsymbol{u} - \boldsymbol{\nabla}\cdot(\rho\boldsymbol{uuuu})] + \mathcal{O}(\Delta_t).\end{aligned} \tag{C.25}$$

For the sake of simplicitly, we give the resulting momentum balance in the isotropic case ($\boldsymbol{A} = \mathbf{0}$):

$$\begin{aligned}\partial_t(\rho\boldsymbol{u}) + \boldsymbol{\nabla}\cdot(\rho\boldsymbol{uu}) &= -\boldsymbol{\nabla}(c_s^2\rho) \\ &\quad + \Delta_t\boldsymbol{\nabla}\cdot\left\{\left(\tau_0 - \frac{1}{2}\right)c_s^2\rho[\boldsymbol{\nabla}\boldsymbol{u}]_2\right\} \\ &\quad + \Delta_t\boldsymbol{\nabla}\cdot\left\{\frac{2-\alpha}{2\alpha\beta}c_s^2\rho[\boldsymbol{\nabla}\boldsymbol{u}]_2\right\} \\ &\quad + \Delta_t^2\frac{(\beta-1)^2}{4\beta^2}\boldsymbol{\nabla}\cdot\left\{c_s^2\rho[\boldsymbol{\nabla}\boldsymbol{u}\cdot\boldsymbol{\nabla}\boldsymbol{u}]_2 + 2c_s^4\rho\boldsymbol{\nabla\nabla}\ln\rho + \boldsymbol{\nabla\nabla} : (\rho\boldsymbol{uuuu})\right\} \\ &\quad + \frac{\Delta_t^2}{12}c_s^2\boldsymbol{\nabla}\cdot\{\boldsymbol{\nabla}\cdot(\rho[\boldsymbol{\nabla}\boldsymbol{u}]_2\boldsymbol{u}) - 2\rho[\boldsymbol{\nabla}\boldsymbol{u}\cdot\boldsymbol{\nabla}\boldsymbol{u}]_2\} \\ &\quad - \frac{\Delta_t^2}{12}\boldsymbol{\nabla}\cdot\left\{4c_s^4\rho\boldsymbol{\nabla\nabla}\ln\rho + \boldsymbol{\nabla\nabla} : (\rho\boldsymbol{uuuu})\right\} + \mathcal{O}(\Delta_t^3),\end{aligned} \tag{C.26}$$

where we used that $\beta = \frac{1}{\tau_0}$ and $\boldsymbol{\nabla\nabla} : (\rho[\boldsymbol{u}\boldsymbol{\nabla}\boldsymbol{u}]_6) = \boldsymbol{\nabla\nabla} : (2\rho\boldsymbol{u}[\boldsymbol{\nabla}\boldsymbol{u}]_2 + \rho[\boldsymbol{\nabla}\boldsymbol{u}]_2\boldsymbol{u})$.

BIBLIOGRAPHY

[1] P. L. Bhatnagar, E. P. Gross, and M. Krook. A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems. *Phys. Rev.*, 94:511–525, 1954.

[2] O. Malaspinas, M. Deville, and B. Chopard. Towards a physical interpretation of the entropic Lattice Boltzmann method. *Physical Review E*, 78(6):066705, 2008.

[3] I. V. Karlin, F. Bösch, S. S. Chikatamarla, and S. Succi. Entropy-assisted computing of low-dissipative systems. *Entropy*, 17(12):8099–8110, 2015.

[4] J. Smagorinsky. General circulation experiments with the primitive equations. 91(3):99–194, 1963.

[5] metaLBM: An open-source high-performance implementation of LBM for simulation of turbulent flows. `https://gitlab.com/gtauzin/metaLBM_private`.

[6] G. Tauzin, L. Biferale, M. Sbragaglia, A. Gupta, F. Toschi, A. Bartel, and M. Ehrhardt. A numerical tool for the study of the hydrodynamic recovery of the Lattice Boltzmann method. *Computers & Fluids*, 172:241 – 250, 2018.

[7] G. Tauzin, D. Belardinelli, L. Biferale, and M. Sbragaglia. Study of the implicit sub-grid scale modeling within the entropic Lattice Boltzmann. In preparation.

[8] G. Tauzin, M. Buzzicotti, and L. Biferale. Inertial range statistics of the entropic Lattice Boltzmann and large-eddy simulations in 3d turbulence. In preparation.

[9] G. Tauzin, T. Nikolov, M. Wagner, and J. Kraus. Accelerating Lattice Boltzmann flows simulation using gpu-initiated communications with nvshmem. In preparation.

[10] B. Galperin and S. A. Orszag. *Large eddy simulation of complex engineering and geophysical flows.* Cambridge University Press, 2010.

[11] L. F. Richardson. Weather prediction by numerical process. *Quarterly Journal of the Royal Meteorological Society*, 48(203):282–284, 1922.

[12] A. N. Kolmogorov. The local structure of turbulence in incompressible viscous fluid for very large Reynolds numbers. In *Dokl. Akad. Nauk SSSR*, volume 30, pages 301–305. JSTOR, 1941.

[13] A. M. Obukhov. On the distribution of energy in the spectrum of turbulent flow. In *Dokl. Akad. Nauk SSSR*, volume 32, pages 22–24, 1941.

[14] U. Frisch. *Turbulence : the legacy of A.N. Kolmogorov*. Cambridge University Press, 1995.

[15] A. Alexakis and L. Biferale. Cascades and transitions in turbulent flows. *Physics Reports*, 767-769:1 – 101, 2018. Cascades and transitions in turbulent flows.

[16] Z.-S. She and E. Leveque. Universal scaling laws in fully developed turbulence. *Phys. Rev. Lett.*, 72:336–339, 1994.

[17] K. R. Sreenivasan. An update on the energy dissipation rate in isotropic turbulence. *Physics of Fluids (1994-present)*, 10(2):528–529, 1998.

[18] T. Ishihara, T. Gotoh, and Y. Kaneda. Study of high-Reynolds number isotropic turbulence by direct numerical simulation. *Annual Review of Fluid Mechanics*, 41:165–180, 2009.

[19] Y. Kaneda, T. Ishihara, M. Yokokawa, K. Itakura, and A. Uno. Energy dissipation rate and energy spectrum in high resolution direct numerical simulations of turbulence in a periodic box. *Physics of Fluids*, 15(2):L21, 2003.

[20] Fjortoft R. On the changes in the spectral distribution of kinetic energy for twodimensional, nondivergent flow. *Tellus*, 5(3):225–230, 1953.

[21] R. H. Kraichnan. Inertial ranges in two-dimensional turbulence. *Physics of Fluids*, 10:1417–1423, 1967.

[22] C. E. Leith. Diffusion approximation for two-dimensional turbulence. *The Physics of Fluids*, 11(3):671–672, 1968.

[23] G. K. Batchelor. Computation of the energy spectrum in homogeneous two-dimensional turbulence. *The Physics of Fluids*, 12(12):II–233–II–239, 1969.

[24] H. Pitsch. Large-eddy simulation of turbulent combustion. *Annual Review of Fluid Mechanics*, 38(1):453–482, 2006.

[25] C. A. Wagner, T. Hüttl, and P. Sagaut. *Large-eddy simulation for acoustics.* Cambridge University Press, 2007.

[26] P. P. Sullivan, J. C. McWilliams, and C.-H. Moeng. A subgrid-scale model for large-eddy simulation of planetary boundary-layer flows. *Boundary-Layer Meteorology*, 71(3):247–276, 1994.

[27] M. Lesieur, O. Métais, and P. Comte. *Large-eddy simulations of turbulence.* Cambridge University Press, 2005.

[28] S. B. Pope. *Turbulent Flows.* Cambridge University press, 2000.

[29] C. Meneveau and J. Katz. Scale-invariance and turbulence models for large-eddy simulation. *Annual Review of Fluid Mechanics*, 32(1):1–32, 2000.

[30] X. He and L.-S. Luo. Theory of the Lattice Boltzmann method: From the Boltzmann equation to the Lattice Boltzmann equation. *Phys. Rev. E*, 56:6811–6817, 1997.

[31] X. Shan, X.-F. Yuan, and H. Chen. Kinetic theory representation of hydrodynamics: a way beyond the navier–stokes equation. *Journal of Fluid Mechanics*, 550:413–441, 2006.

[32] P. C. Philippi, L. A. Hegele, L. O. E. dos Santos, and R. Surmas. From the continuous to the Lattice Boltzmann equation: The discretization problem and thermal models. *Phys. Rev. E*, 73:056702, 2006.

[33] S. Succi. *The Lattice Boltzmann Equation for Fluid Dynamics and Beyond.* Oxford University Press, 2001.

[34] D. A. Wolf-Gladrow. *Lattice-gas cellular automata and Lattice Boltzmann models : an introduction.* Springer, 2000.

[35] T. Krüger, H. Kusumaatmaja, A. Kuzmin, O. Shardt, G. Silva, and E. M. Viggen. *The Lattice Boltzmann Method.* Graduate Texts in Physics. Springer International Publishing, Cham, 2017.

[36] E. M. Viggen. *The Lattice Boltzmann Method with Applications in Acoustics.* PhD thesis, NTNU, 2009.

[37] G. Jin, S. Wang, Y. Wang, and G. He. Lattice Boltzmann simulations of high-order statistics in isotropic turbulent flows. *Applied Mathematics and Mechanics*, 39(1):21–30, 2018.

[38] K. N. Premnath, M. J. Pattison, and S. Banerjee. Dynamic subgrid scale modeling of turbulent flows using Lattice-Boltzmann method. *Physica A: Statistical Mechanics and its Applications*, 388(13):2640–2658, 2009.

[39] O. Filippova, S. Succi, F. Mazzocco, C. Arrighetti, G. Bella, and D. Hänel. Multi-scale Lattice Boltzmann schemes with turbulence modeling. *Journal of Computational Physics*, 170(2):812–829, 2001.

[40] Y.-H. Dong and P. Sagaut. A study of time correlations in Lattice Boltzmann-based large-eddy simulation of isotropic turbulence. *Physics of Fluids*, 20(3):035105, 2008.

[41] Y.-H. Dong, P. Sagaut, and S. Marie. Inertial consistent subgrid model for large-eddy simulation based on the Lattice Boltzmann method. *Physics of Fluids*, 20(3):035104, 2008.

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

[43] O. Malaspinas and P. Sagaut. Consistent subgrid scale modelling for Lattice Boltzmann methods. *Journal of Fluid Mechanics*, 700:514–542, 2012.

[44] B. M. Boghosian, J. Yepez, P. V. Coveney, and A. Wager. Entropic Lattice Boltzmann methods. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 457(2007):717–766, 2001.

[45] A. J. Wagner. An H-theorem for the Lattice Boltzmann approach to hydrodynamics. *Europhysics Letters (EPL)*, 44(2):144–149, 1998.

[46] I. V. Karlin, A. Ferrante, and H. C. Öttinger. Perfect entropy functions of the Lattice Boltzmann method. *Europhysics Letters (EPL)*, 47(2):182–188, 1999.

[47] S. Ansumali and I. V. Karlin. Stabilization of the Lattice Boltzmann method by the H-theorem: A numerical test. *Physical Review E*, 62(6):7999–8003, 2000.

[48] I. V. Karlin, S. Ansumali, E. DE Angelis, H. C. Öttinger, and S. Succi. Entropic Lattice Boltzmann method for large scale turbulence simulation. 2003.

[49] S. S. Chikatamarla and I. V. Karlin. Entropic Lattice Boltzmann method for turbulent flow simulations: Boundary conditions. *Physica A*, 392:1925–1930, 2013.

[50] N. Frapolli, S. S. Chikatamarla, and I. V. Karlin. Multispeed entropic Lattice Boltzmann model for thermal flows. *Physical Review E*, 90(4):043306, 2014.

[51] I. V. Karlin, D. Sichau, and S. S. Chikatamarla. Consistent two-population Lattice Boltzmann model for thermal flows. *Physical Review E*, 88(6):063310, 2013.

[52] G. Pareschi, N. Frapolli, S. S. Chikatamarla, and I. V. Karlin. Conjugate heat transfer with the entropic Lattice Boltzmann method. *Physical Review E*, 94(1):013305, 2016.

[53] A. M. Mazloomi, S. S. Chikatamarla, and I. V. Karlin. Entropic Lattice Boltzmann method for multiphase flows: Fluid-solid interfaces. *Physical Review E*, 92(2):023308, 2015.

[54] S. Ansumali, I. V. Karlin, C. E. Frouzakis, and K. B. Boulouchos. Entropic Lattice Boltzmann method for microflows. *Physica A*, 359:289–305, 2006.

[55] F. Tosi and S. Succi. An Introduction to Entropic Lattice Boltzmann Scheme. *SIMAI e-Lecture Notes*, 1(July):1–42, 2008.

[56] S. Succi, I. V. Karlin, and H. Chen. Role of the H-theorem in Lattice Boltzmann hydrodynamic simulations. *Review of Modern Physics*, 74(4):1203–1220, 2002.

[57] B. Keating, G. Vahala, J. Yepez, M. Soe, and L. Vahala. Entropic Lattice Boltzmann representations required to recover Navier-Stokes flows. *Physical Review E*, 75(3):036712, 2007.

[58] S. Ansumali, I. V. Karlin, and H. C. Öttinger. Minimal entropic kinetic models for hydrodynamics. *Europhysics Letters (EPL)*, 63(6):798–804, 2003.

[59] M. Atif, P. K. Kolluru, C. Thantanapally, and S. Ansumali. Essentially entropic Lattice Boltzmann model. *Phys. Rev. Lett.*, 119:240602, 2017.

[60] Fabian Bösch, Shyam S. Chikatamarla, and Ilya V. Karlin. Entropic multirelaxation lattice boltzmann models for turbulent flows. *Phys. Rev. E*, 92:043309, Oct 2015.

[61] A. L. Kuperstokh. New method of incorporating a body force term into the Lattice Boltzmann equation. In *Proceedings of the 5th International EDH Workshop*, pages 241–246, Poitiers, France, 2004.

[62] Z. Guo, C. Zheng, and B. Shi. Discrete Lattice effects on the forcing term in the Lattice Boltzmann method. *Phys. Rev. E*, 65:046308, 2002.

[63] B. Dorschner, F. Bösch, S. S. Chikatamarla, K. Boulouchos, and I. V. Karlin. Entropic multi-relaxation time Lattice Boltzmann model for complex flows. *Journal of Fluid Mechanics*, 801:623–651, 2016.

[64] N. Frapolli, S.S. Chikatamarla, and I.V. Karlin. Entropic Lattice Boltzmann simulation of thermal convective turbulence. *Computers & Fluids*, 175:2 – 19, 2018.

[65] P. A. Davidson. *Turbulence: An Introduction for Scientists and Engineers.* Oxford University Press, 2015.

[66] G. Boffetta and R. E. Ecke. Two-dimensional turbulence. *Annual Review of Fluid Mechanics*, 44(1):427–451, 2012.

[67] A. Kurganov and C.-T. Lin. On the reduction of numerical dissipation in central-upwind schemes. *Communications in Computational Physics*, 2:141–163, 2007.

[68] Z. Guo, C. Zheng, and B. Shi. Force imbalance in Lattice Boltzmann equation for two-phase flows. *Phys. Rev. E*, 83:036707, 2011.

[69] S. S. Chikatamarla, S. Ansumali, and I. V. Karlin. Entropic Lattice Boltzmann models for hydrodynamics in three dimensions. *Physical Review Letters*, 97(1):010201, 2006.

[70] J. Latt. *Hydrodynamic limit of Lattice Boltzmann equations.* PhD thesis, Univ. Geneve, 2007.

[71] S. Hou, J. Sterling, S. Chen, and G. D. Doolen. A Lattice Boltzmann Subgrid Model for High Reynolds Number Flows. *arXiv e-prints*, pages comp–gas/9401004, 1994.

[72] A. Arnèodo, R. Benzi, J. Berg, L. Biferale, E. Bodenschatz, A. Busse, E. Calzavarini, B. Castaing, M. Cencini, L. Chevillard, R. T. Fisher, R. Grauer, H. Homann, D. Lamb, A. S. Lanotte, E. Lévèque, B. Lüthi, J. Mann, N. Mordant, W.-C. Müller, S. Ott, N. T. Ouellette, J.-F. Pinton, S. B. Pope, S. G. Roux, F. Toschi, H. Xu, and P. K. Yeung. Universal intermittent properties of particle trajectories in highly turbulent flows. *Phys. Rev. Lett.*, 100:254504, 2008.

[73] R. Benzi, S. Ciliberto, R. Tripiccione, C. Baudet, F. Massaioli, and S. Succi. Extended self-similarity in turbulent flows. *Phys. Rev. E*, 48:R29–R32, 1993.

[74] Top 500 list - June 2018. `https://www.top500.org/list/2018/06/`.

[75] Summit supercomputer. `https://www.olcf.ornl.gov/summit/`.

[76] Sierra supercomputer. `https://computation.llnl.gov/computers/sierra`.

[77] P. Bailey, J. Myre, S. D. C. Walsh, D. J. Lilja, and M. O. Saar. Accelerating Lattice Boltzmann fluid flow simulations using graphics processors. In *2009 International Conference on Parallel Processing*, pages 550–557, 2009.

[78] L. Biferale, F. Mantovani, M. Pivanti, F. Pozzati, M. Sbragaglia, A. Scagliarini, S. F. Schifano, F. Toschi, and R. Tripiccione. An optimized d2q37 Lattice Boltzmann code on gp-gpus. *Computers & Fluids*, 80:55 – 62, 2013. Selected contributions of the 23rd International Conference on Parallel Fluid Dynamics ParCFD2011.

[79] E. Calore, A. Gabbana, J. Kraus, E. Pellegrini, S.F. Schifano, and R. Tripiccione. Massively parallel Lattice–Boltzmann codes on large gpu clusters. *Parallel Computing*, 58:1 – 24, 2016.

[80] A. Gabbana. Accelerating the d3q19 Lattice Boltzmann model with openacc and mpi, 2015.

[81] S. Potluri, D. Rossetti, D. Becker, D. Poole, M. Gorentla Venkata, O. Hernandez, P. Shamis, M. G. Lopez, M. Baker, and W. Poole. Exploring openshmem model to program gpu-based extreme-scale systems. In *Revised Selected Papers of the Second Workshop on OpenSHMEM and Related Technologies. Experiences, Implementations, and Technologies - Volume 9397*, OpenSHMEM 2015, pages 18–35, New York, NY, USA, 2015. Springer-Verlag New York, Inc.

[82] A. Kaufman, Z. Fan, and K. Petkov. Implementing the Lattice Boltzmann model on commodity graphics hardware. *Journal of Statistical Mechanics: Theory and Experiment*, 2009(06):P06016, 2009.

[83] E. Calore, Al. Gabbana, J. Kraus, S. F. Schifano, and R. Tripiccione. Performance and portability of accelerated Lattice Boltzmann applications with openacc. *Concurrency and Computation: Practice and Experience*, 28(12):3485–3502, 2016.

[84] G. Wellein, T. Zeiser, G. Hager, and S. Donath. On the single processor performance of simple Lattice Boltzmann kernels. *Computers & Fluids*, 35(8):910 – 919, 2006. Proceedings of the First International Conference for Mesoscopic Methods in Engineering and Science.

[85] M. Wittmann, T. Zeiser, G. Hager, and G. Wellein. Comparison of different propagation steps for Lattice Boltzmann methods. *Computers & Mathematics with Applications*, 65(6):924 – 935, 2013. Mesoscopic Methods in Engineering and Science.

[86] Multi GPU Programming models, GTC On-Demand. `http://on-demand-gtc.gputechconf.com/gtc-quicklink/4qjeW`.

[87] How to optimize data transfers in CUDA C/C++, NVIDIA Developer Blog. `https://devblogs.nvidia.com/how-optimize-data-transfers-cuda-cc/`.

[88] C. Feichtinger, J. Habich, H. Köstler, G. Hager, U. Rüde, and G. Wellein. A flexible patch-based Lattice Boltzmann parallelization approach for heterogeneous gpu–cpu clusters. *Parallel Computing*, 37(9):536 – 549, 2011. Emerging Programming Paradigms for Large-Scale Scientific Computing.

[89] Mark J. Mawson and Alistair J. Revell. Memory transfer optimization for a Lattice Boltzmann solver on kepler architecture nvidia gpus. *Computer Physics Communications*, 185(10):2566 – 2574, 2014.

[90] C. Obrecht, F. Kuznik, B. Tourancheau, and J.-J. Roux. A new approach to the Lattice Boltzmann method for graphics processing units. *Computers & Mathematics with Applications*, 61(12):3628 – 3638, 2011. Mesoscopic Methods for Engineering and Science — Proceedings of ICMMES-09.

[91] E. Calore, A. Gabbana, S. F. Schifano, and R. Tripiccione. Optimization of Lattice Boltzmann simulations on heterogeneous computers. *CoRR*, abs/1703.04594, 2017.

[92] W. Xian and A. Takayuki. Multi-gpu performance of incompressible flow computation by Lattice Boltzmann method on gpu cluster. *Parallel Computing*, 37(9):521 – 535, 2011. Emerging Programming Paradigms for Large-Scale Scientific Computing.

[93] An introduction to CUDA-Aware MPI, NVIDIA Developer Blog. `https://devblogs.nvidia.com/introduction-cuda-aware-mpi/`.