Tenth International Conference on Computational Fluid Dynamics (ICCFD10), Barcelona,Spain, July 9-13, 2018

ICCFD10-2018-0305

Fractional step lattice Boltzmann methods for large eddy simulation on GPUs

A. Matyas*
Corresponding author: alparmat@gmail.com

* Continental AG, Division Chassis and Safety, Salzburgstr. 8, 550018 Sibiu, Romania.

15. June 2018

1 Introduction

Lattice Boltzmann methods (LBMs) play an important role for the modeling of fluid flow based on kinetic theory of gases. In contrast to computational fluid dynamics (CFD) governed by the Navier-Stokes equation (NSE), the LBM physical framework is based on the collision and streaming of particles in discrete lattice space [1]. The collision occurs between neighboring particles and the streaming step updates the new distribution functions (DFs) at each lattice point. The choice of the lattice defines the neighbor relations or velocity streaming direction, and usually lattices which have more streaming directions from a given cell offer a more accurate description of the flow problem. For example in the 3D case the D3Q27 lattice can be used for computing higher Reynolds number (Re) flows as compared to the D3Q15 lattice for the same lattice size. However, the former requires 27 DFs for each velocity direction, while the latter requires roughly half. This means that D3Q15 needs roughly half of the memory for computations. Such behavior is relevant on graphics processing units (GPUs), where the amount of memory can be limited. Since the evolution of the particle velocities is governed by the Boltzmann equation [2, 3], a specific numerical treatment is required for solving the flow problem which is different from solving NSEs. Initially a lattice is assigned to the continuous Boltzmann equation for defining its discrete form as well as a collision term is used [4]. Common lattices are the D2Q9 2D lattice with 9 velocity DF directions or the D3Q27 3D lattice with 27 velocity directions. The collision in LBMs can be performed by single or multiple relaxation terms, where the former can have simple interpretations and the latter can be more accurate [3]. Due to its simplicity the Bhatnagar-Gross-Krook single relaxation term [4] is commonly used as it provides a simple relation to the fluid viscosity and the calculation of pressure and velocity from the DFs is straightforward. Furthermore, an equivalence of the LBM to the NSE equation can be shown by Chapman-Enskog expansion [2, 3] up to second order in the Mach number. Such properties have proven that LBMs can be used as a general technique for modeling different scales of fluid flow. The numerical advantage of LBMs in contrast to NSE based methods lies in their straightforward implementation on Cartesian grids as well as the excellent parallel scalability of the algorithm on CPUs as well as GPUs. Several methods have been developed for improving compute, data storage and data structuring efficiency of LBM implementations [5, 6, 7]. For storage efficiency of LBMs the goal is mainly to have as less DFs stored as possible with the minimum amount of (single or half-float) precision. Efficient storage has been demonstrated with the shift algorithm [5] that systematically moves through 2D slices of the 3D lattice and stores the updated slice in the same memory array, however, several lattice points further in the update direction. Conseqently, it works with a single set of DFs for a given lattice and saves memory. A further aspect from LBM CFD is the choice of the hardware (CPU vs. GPU) which requires a data structuring scheme and can influence the compute efficiency. For CPUs the memory access time is very similar for all the data, however the floating point operations used in the collision step take most of the computing time. This aspect favors the so-called SoA (structure-of-arrays) data layout,

where each lattice point has arrays that contain all DFs and the evolution equations can be updated more efficiently due to vectorization. On the other hand for GPUs the AoS (array-of-structures) data layout is preferable [6], since the streaming step takes more time than the collision step and streaming to neighbor cells that are stored closer in memory is faster because the GPU bus can read several memory blocks during a clock cycle.

For GPUs, that are the hardware of choice in this paper, the limited device memory is the main issue that hinders accurate simulations with a lareg number of elementse. Starting from the implementation of LBMs on GPUs, the memory and memory bandwidth limitation has lead to the development of algorithms that focus on efficient storage of DFs. However, most of the present schemes store explicitly the velocity DFs during the flow evolution in time, and therefore require large amounts of memory especially in 3D. Even for the shift algorithm on a D3Q27 lattice, at least 27 arrays are required for storing the velocity direction DFs which is a lot more than having a NSE implemented that requires three velocity and one pressure array. An alternative to this GPU memory storage problem of the DFs is offered by the fractional-step implementation of lattice Boltzmann methods FSLBMs [8]. Here, the flow problem is separated into a LBM based predictive part performed on each GPU core and a viscosity-corrective part, which is a reverse diffusion equation for the velocity vector [8]. Such methods facilitate storage for the velocity vector and pressure in the GPU device memory and therefore are highly efficient, can calculate up to very large Reynolds numbers (Re) as well as keep the advantage of good parallel scalability. However, such FSLBMs are implemented only up to two dimensions to the author's knowledge, therefore their usage is limited. Furthermore, numerical tests for an equivalent 3D implementation of FSLBM show limitation to very low Re, and therefore 3D FSLBM cannot be practically used for common industrial flow problems. In this paper an attempt is made to implement 3D FSLBMs based on coarse corrective steps (CCS). It is found that CCS extend the Re range of 3D FSLBMs without compromising the flow-physics. Furthermore the memory requirement for storage is equivalent to NSE-based approach, i.e., the velocity vector and the pressure is stored after each time step and not the DFs. Additionally, the implementation of 3D FSLBM in this paper facilitates calculations with up to one billion lattice points on a single 11 GB memory GPU while offering the required accuracy as shown by benchmark problems for pipes.

2 Problem Statement

Fractional stepping is often a meaningful choice for solving CFD problems since it divides the equations into individual parts that can have separate numerical treatment. For the current approach the LBM has been implemented similarly as in Ref. [8], however using DFs for incompressible flow [9] which give the predictive step for the developed method. Here, the Boltzmann equation is approximated in the discrete evolution form

$$f_\alpha(\mathbf{x},t) = (1-\omega)\, f_\alpha(\mathbf{x}-\mathbf{e}_\alpha\Delta t, t-\Delta t) + \omega f_\alpha^{eq}(\mathbf{x}-\mathbf{e}_\alpha\Delta t, t-\Delta t), \tag{1}$$

with the DF f_α defined on the D3Q27 lattice with direction $\alpha \in \{0,1,..,26\}$ and coordinate vector $\mathbf{x}$. Furthermore, ω is the collision term, $\mathbf{e}_\alpha$ is the direction stencil defined on the small cube of Figure 1. (a), Δt is the time step and

$$f_\alpha^{eq}(\mathbf{x},t) = W_\alpha \left(p^k + 3\mathbf{e}_\alpha \mathbf{u}^k e^{-2} + 4.5 \left(\mathbf{e}_\alpha \mathbf{u}^k\right)^2 e^{-4} - 1.5 \mathbf{u}^k \mathbf{u}^k e^{-2} \right) \tag{2}$$

is the equilibrium DF. Here, W_α is the lattice weight for D3Q27 lattice parameters, $\mathbf{u}^k$ is the velocity vector for the previous time step ($k = t/\Delta t - 1$), p^k is the pressure, $e = \Delta x/\Delta t$ is the lattice speed and Δx the lattice spacing. Furthermore, $\omega = 1$ is chosen on the right hand side of Eq. (1) which computes to $f_\alpha^{eq}(\mathbf{x}-\mathbf{e}_\alpha\Delta t, t-\Delta t)$ that can be evaluated by using Eq. (2). The above equations provide the reconstruction of DFs from the pressure and velocity evolution in our method, furthermore the predicted (p^*,$\mathbf{u}^*$) quantities are calculated from the DFs with sum over the velocity directions as

$$p^* = \sum_\alpha f_\alpha(\mathbf{x},t),\ \mathbf{u}^* = \sum_\alpha \mathbf{e}_\alpha f_\alpha(\mathbf{x},t). \tag{3}$$

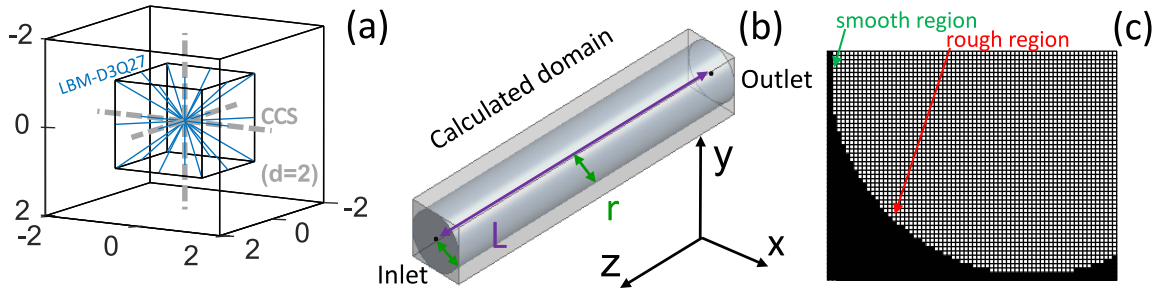

Figure 1: (a) D3Q27 stencils shown in the small box are used for the LBM-predictive step. The coarse corrective step (CCS) uses stencils with $d \geq 2$ shown by dashed lines. (b) Pipe domain used for the calculations, where the radius r and length L fully define the geometry. (c) Pipe cross-sections used in the calculations with some smooth and rough regions.

Eqs. (1), (2) and (3) are shown to be equivalent to solving the NSE[9, 8]

$$\frac{\mathbf{u}^* - \mathbf{u}^k}{\Delta t} + \left(\mathbf{u}^k \cdot \nabla\right) \mathbf{u}^k = -\nabla p^k + \nu^* \nabla^2 \mathbf{u}^k \tag{4}$$

with viscosity $\nu^* = \Delta t e^2/6$. Here, $p^* = p^{k+1}$ is the predicted pressure which equals the pressure for the current time step and $\mathbf{u}^*$ is the predicted velocity. For the corrective step the reverse $(\nu - \nu^* < 0)$ diffusion equation

$$\frac{\mathbf{u}^{k+1} - \mathbf{u}^*}{\Delta t} = (\nu - \nu^*) \Delta \mathbf{u} \tag{5}$$

is used. It can be observed that by adding Eq. (4) and (5) the NSE is obtained, therefore if the reverse diffusion equation is solved in the corrective step, one should obtain NSE solutions for fluid flow. For the reverse diffusion equation a lattice spacing d is introduced to find the discrete form

$$\frac{\mathbf{u}^{k+1} - \mathbf{u}^*}{\Delta t} = \frac{\nu - \nu^*}{d^2 \Delta x^2} \left[6\mathbf{u}^k(\mathbf{x}) - \sum_{\alpha=1}^{6} \mathbf{u}^k(\mathbf{x} + d\mathbf{e}_\alpha \Delta t) \right], \tag{6}$$

where ν is the kinematic viscosity of the fluid. The above CCS form of the reverse diffusion equation can use stencils pointing beyond the nearest lattice cells as shown in Figure 1. (a). Such choice enables calculations for several orders of magnitude higher Re than for a conventional stencil with d=1. The positive effect for $d > 1$ is attributed to reducing the time step sensitivity in the exponential solutions of reversed diffusion equations, and is discussed in the following subsection. Finally, for correctly modeling large Re flow with FSLBM, the wall adaptive local eddy viscosity (WALE) turbulence model [10]

$$\nu_t = (\Delta x C_w)^2 \frac{\left|S_{ij}^d\right|^3}{\left|\bar{S}_{ij}\right|^5 + \left|S_{ij}^d\right|^{5/2}}, \; \bar{S}_{ij} = \frac{\partial_{x_j} u_i + \partial_{x_i} u_j}{2}, \; S_{ij}^d = \frac{g_{ij}^2 + g_{ji}^2}{2} - \frac{\delta_{ij} g_{kk}}{3}, \; g_{ij}^2 = \partial_{x_k} u_i \partial_{x_j} u_k \tag{7}$$

is implemented with the turbulent viscosity ν_t and with $(\nu - \nu^*)$ replaced in Eq. (6) by $(\nu + \nu_t - \nu^*)$. The WALE model defined by Eq. (6) accounts for the strain and rotation rate for the smallest vortices in the flow as well as can model the transition to turbulence with $C_w = 0.5$ fixed [10]. For the WALE implementation in FSLBM central differences have been used for the velocity derivatives defining the $\bar{S}_{ij}$, S_{ij}^d and g_{ij} tensors and the model expressions are identical with the cited reference.

2.1 CCS and expansion solutions

FSLMB corrective steps give the solution to general 3D reverse diffusion equations. The intention of this section is to justify mathematically the use of CCS and show its behavior for the solution process. The expressions within this section are not used in our method they are just meant to provide a hint on the

numerical behavior. For the 3D reverse diffusion equation our assumption is that its general solution can be found in the form

$$u(x,y,z,t) = X(x)Y(y)Z(z)T(t). \tag{8}$$

Furthermore, we treat only one of the velocity vector components from the reverse diffusion equation since the other have the same properties and are updated by CCS independently. Using Eq. (8) with the reverse diffusion equation (5) leads to the expansion solution

$$u(x,y,z,t) = \sum_{k,l,m} a_k \sin\left(k\frac{\pi x}{L} + \varphi_0\right) b_l \sin\left(l\frac{\pi y}{L} + \xi_0\right) c_m \sin\left(m\frac{\pi y}{L} + \eta_0\right) e^{-\frac{\pi^2 t}{d^2\Delta x^2}\left(k^2+l^2+m^2\right)\left(\nu - \frac{\Delta t e^2}{6}\right)} \tag{9}$$

which is a linear combination of orthogonal basis functions. Here, $L = 2d\Delta x$ is the cell length where we apply CCS and φ_0, ξ_0, η_0 are the phases that satisfy the boundary conditions of the discrete cubes of length $d\Delta x$. Generally the phases can be complex numbers, therefore the solution is built from exponentially growing or decaying waves and also periodic waves. Within the above solution one can observe that the larger integer d is, the slower the temporal growth. Furthermore, hyperbolic basis solutions (e.g. when $\sin\left(k\frac{\pi x}{L} + \varphi_0\right) \in \mathbb{C}$) have smaller variation due to L being increased in the denominator. Since the coefficients a_k, b_l and c_m correspond to Fourier modes of the general solution u, they are finite as long as the solution is finite. The above properties favor increased ($d > 1$) discrete steps (CCS) for correcting the LBM viscosity.

2.2 Conditions for pipe flow

Several pipe geometries have been defined on Cartesian grids for the benchmark of our solver, where a schematic representation is shown in Figure 1 (b). For all of the computations the inlet conditions are set by $\mathbf{u}_{inlet} \rightarrow \mathbf{u}_{outlet^-}$ and $p_{inlet} \rightarrow p_{outlet^-} + p_1$. Here the notation $outlet^-$ stands for the streamwise position of 100 elements before the outlet. Furthermore, for the actual outlet the zero gradient condition $\partial_z \mathbf{u}_{outlet} = \mathbf{0}$ and zero pressure $p_{outlet} = 0$, is used. The above choice mimics periodic boundary conditions for long pipe geometries as our results show in the next sections. A further step applied in our solver is the subtraction of the residual pressure from all the cells since the NSE and FSLBM methods generally give solutions to pressure gradients and the solutions are always found in terms of a pressure difference. Numerically, however, the larger the background pressure between two points, the larger the chances for round-off errors. Therefore, after a predefined number of time-iterations (e.g. 100) we subtract the average outlet pressure from every lattice point, including the inlet. Finally, since velocity in pipes has several scales and our solver can handle Mach numbers up to ≈0.4, we truncate the outlet Mach number to 0.15 for avoiding numerical overflow. This value is above the calculated average Ma in our methods and does not influence the average pressure and velocity. Quantities like friction factor, normalized velocities or turbulence spectra are calculated in the range of $l_1 = 12r$ to $l_2 = L - 4r$ in our domains to avoid the effects of the truncation. Furthermore the truncation influences the velocity fluctuations near the inlet due to the periodic conditions, however, if the pipe is long enough and the analyzed behavior is far from the inlet the relevant quantities are correctly computed.

3 Sample results

For the validation of the developed model the FSLBM friction factor is calculated as $f = 4r(p_1 - p_2)/3\rho u_z^2(l_1 - l_2)$ where p_1 is defined at l_1 and p_2 is defined at l_2, additionally, all of the quantities in the above expression are given in terms of lattice units and $\rho = 1$. Furthermore the pressures and velocity are time-averaged as $p_{1,2} = \sum_{k=1}^{N} p_{1,2}^k/N$ and $u_z = \sum_{k=1}^{N} u_z^k/N$. A second quantity of interest is the Reynolds number $Re = 2er\Delta x u_z/\nu_{LBM}$ given in terms of lattice units. Here, u_z is the same velocity used for the calculation of the friction factor, $e = 1$ is the lattice speed and $\Delta x = 10^{-3}$ for all of the calculations. Consequently, Re can be changed geometrically by adjusting r or through the lattice viscosity by adjusting ν_{LBM}. Further parameters like the friction velocity, u^+, y^+, and turbulence parameters are discussed in the subsections.

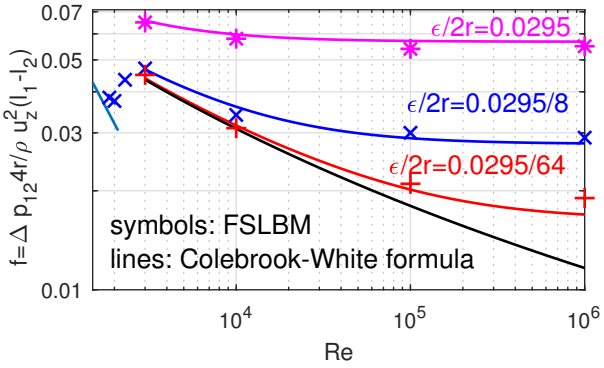

Figure 2: Comparison for the friction factors of pipe domains with various roughness as calculated by FSLBM and by the Colebrook-White equation. The pipes have decreasing roughness from top to bottom which relates to the dimensions of ($L = 1200$, $r = 25$); ($L = 2400$, $r = 50$) and ($L = 4800$, $r = 100$). The lowest solid line is for the case of a smooth pipe.

3.1 Comparison for the friction factors

Turbulent pipe flow strongly depends on the surface roughness, a quantity which is usually determined from friction factor measurements fit to results from the Colebrook-White equation [11]. Here, we perform the same approach for a single calculation point and we start with a rough pipe that has dimensions of $r = 25$, $L = 1200$ and Re=10^6, therefore the flow is expected to be in the fully turbulent regime. From this calculation the ratio of surface roughness to pipe diameter is determined to be 0.0295. For the case of a finer lattice, e.g., $r = 50$ and $L = 2400$ the roughness is reduced by a factor of 8, since each of the (x,y,z) directions are doubled, therefore the roughness value of $0.0295/8$ is used to validate the new calculations. Figure 2 shows such a validation by comparison of FSLBM friction factors to results from the Colebrook-White equation given by the solid lines. A good agreement is found between the FSLBM symbols and the reference curves for the range of Re and friction factors given by the Colebrook-White equation which shows the robustness of FSLBM calculations. For the $\epsilon/2r = 0.0295/8$ case the transition between laminar and turbulent flow is reproduced at Re$\approx$ 2000 with minor differences to the $64/Re$ laminar scaling. This can be regarded to slight overestimation of the intermittency in FSLBM as well as the $L = 2400$ pipe length which is already small for these lower Re calculations with the used periodic boundary conditions.

3.2 General properties

Turbulent flows have some mostly geometry independent scaling laws that are general for validating FSLBM calculations. Such validations are based on physics that have been observed, furthermore they have well established and widely accepted models.

First of such is the law of the wall, since the investigated pipes have some smooth surfaces surrounded by rough surfaces (see Figure 1 (c)). We consider the geometry cases of ($L = 2400$, $r = 50$) and ($L = 4800$, $r = 100$) and calculate the average velocity tangential to the smooth region for turbulent flow. The comparison is performed by calculating the friction velocity $u^* = \sqrt{2u_{z_w}e^2}$, where $u_{z_w} = r\sum_{k=1}^{N} u_{z_w}^k/N$ is the time-averaged streamwise velocity at the closest point to the smooth surface in terms of lattice units. With the above definition the normalized closest point to the wall becomes $u^*/2e$. After having the friction velocity defined the calculated results are shown in Figure 3 (a) where the averaged tangential velocity normalized to the friction velocity shows excellent agreement to the laws of the wall for both the linear and logarithmic regions and all the considered cases. Here, we see a nearly perfect agreement for $Re = 10^6$ and ($r = 100$, $L = 4800$) shown by the green solid line, which is only slightly lower in the logarithmic region. Furthermore, we see a reduction of u^+ in the logarithmic region for $Re = 10^6$ and ($r = 50$, $L = 2400$) shown by the gray dashed line. This is a result of surface roughness that is in the neighboring of the smooth surface and it is visible only for calculations of increased Re for the coarse pipe case. Such behavior is regarded

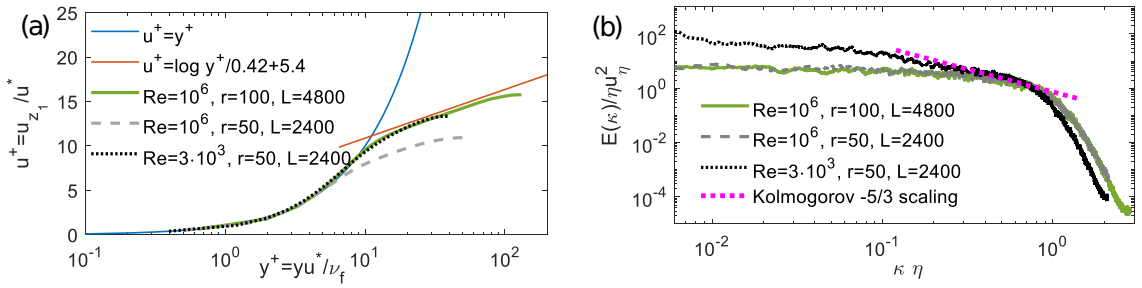

Figure 3: (a) Law of the wall compared for moderately rough ($r = 50$, $L = 2400$) and smooth pipes ($r = 100$,$L = 4800$) in their smooth region (see Figure 2 (c)) and for large and medium Re. The small reduction shown by the dashed line in the log-region is attributed to the influence of the pipe rough region on the pipe smooth region.

to efficient mixing of the turbulent structures and to the overall influence of the rough region parts on the log-region when the roughness is not covered by the boundary layer. If the pipe is finer or Re is reduced as the other calculations show the law of thee wall is completely smooth. A second aspect is the extension of the log-region for higher y^+ for the finer (smoother) pipe or for increased Re when comparing the cases. Such behavior is also expected since in the given direction the results calculated for the smoother pipe have twice as many elements as the coarse pipe therefore the fully turbulent region gets more resolved. Furthermore for increased Re the turbulent region intrinsically expands.

An additional comparison is now made for the turbulence kinetic energy spectra and their scaling. The calculations are performed using the time-dependent streamwise velocity sampled at the pipe mid-point and at a length of $l_1 = 12r$. Furthermore, $E(\kappa) = u_z S(f)/4\pi$ is assumed, where $u_z = u_z(r, r, l_1)$ is the average streamwise velocity at the sampled point and $S(f)$ is the frequency spectrum calculated as $S(f) = e^2 \left|F\left\{u_z^k - u_z\right\}\right|^2 / \Delta t_s$, with $\Delta t_s = 10\Delta t$ defined as the sampling time. From $S(f)$ the energy dissipation is calculated as $\varepsilon = 4\pi^2(6\nu^*)\,(u_z)^{-2} \int f^2 S(f) \mathrm{d}f$. Since the viscosity ν^* and energy dissipation is known, the Kolmogorov scales can be calculated straightforwardly and are used for normalization of the spectra. Such results are shown in Figure 3 (b) for the same geometries and Re that have been considered for validating the law of the wall. Furthermore, the $\kappa^{-5/3}$ scaling is also shown by the dashed line for reference. A first observation is the small inertial subrange which can be typical for high Re pipe flow [12]. For the case of low Re an extended inertial subrange scaling is observed in contrast with the results for higher Re. The narrow inertial subrange is attributed to increased homogeneity of the flow structures when the turbulence is fully developed and its scales are more uniform. All in all the calculated spectra show the typical -5/3 scaling in the inertial subrange, furthermore the energy containing range as well as the dissipation range are identified which are a further validation for the developed FSLBMs.

3.3 Development of turbulent flow resolved with 10^9 lattice points on a single GPU

A final benchmark for the FSLBM method is now provided to demonstrate the strength of its implementation algorithm. Namely for calculating very large problems efficiently on GPUs the combination of the shift algorithm, half-CUDA precision and pressure-velocity storage instead of DF storage are required. While the former can be implemented for every type of LBM the latter is only a benefit of the presented FSLBM to the authors knowledge. By including all the above features in FSLBM the calculation can handle $\approx 92.7 \cdot 10^6$ elements per Gb of GPU memory which is considered an efficiency record for LBM calculations on GPUs. For the turbulent flow in this last section a domain of ($r = 350$,$L = 8400$) is considered which has roughly 1.02 billion elements. Furthermore, $Re = 1.3 \cdot 10^6$ is chosen with the initial Mach number of 0.1 distributed equally in the domain. Figure 4 (a) shows the development of a turbulent slug-like region after 14000 time steps, where the choice for the visualization of the structures is the $\lambda_2 < 0$ criterion [13] colored by the Mach number. Additionally, the inset shows a zoom to the flow structures between the laminar and turbulent region of the developing flow which are resolved to very small scales by the presented high resolution FSLBM. After

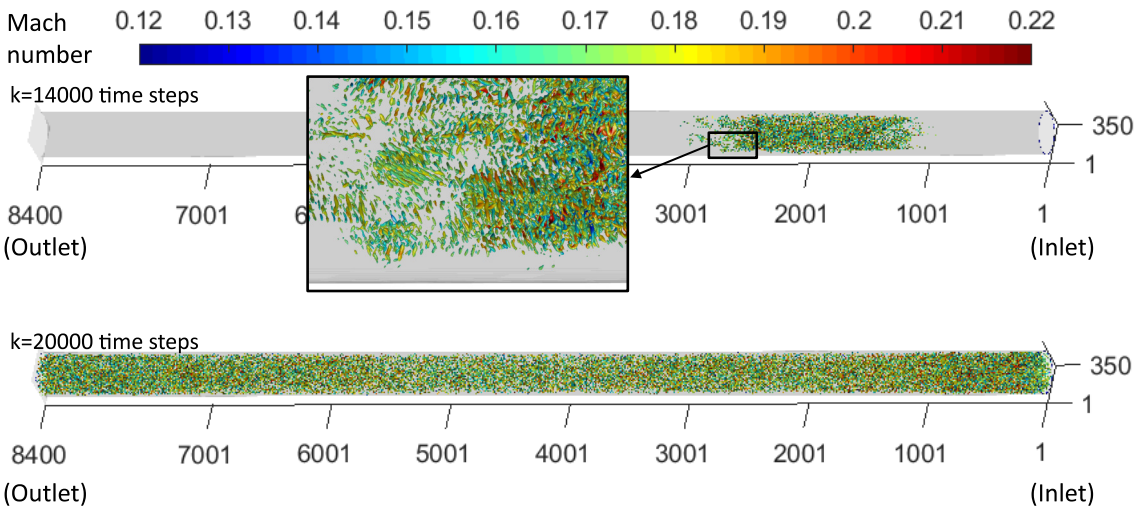

Figure 4: Velocity colored $\lambda_2 < 0$ criterion showing the development of turbulence from the inlet of the pipe at (a) 14000 time steps and (b) to the complete pipe region at 20000 time steps. The inset shows the fine resolution of the $\lambda_2 < 0$ structures near the onset region.

a further 6000 time steps the full turbulence is developed as shown in Figure 4 (b), which is typical for a slug-type flow.

4 Conclusion and Future Work

Calculations with 3D fractional step lattice Boltzmann methods using coarse corrective steps for the reverse diffusion equation have been presented in this paper. The advantage of the method lies in the treatment of the correction step with large stencils that skip the nearest neighbors of the given lattice cell. This enables increased Re range for the computations that have been performed for validating the developed model on pipe domains. Here, good agreement is found for calculated friction factors compared to results from the Colebrook-White equation as well as the law of the wall and Kolmogorov scaling are well reproduced by the presented scheme. FSLBMs enable calculations with $\approx 92.7 \cdot 10^6$ elements per Gb of GPU memory which is considered an efficiency record for current 3D LBM implementations, furthermore FSLBMs can resolve turbulent structures for very small scales. Developments including models for compressible flow and further speedup of the FSLBM calculation code are planned for the future.

References

[1] G. R. McNamara and G. Zanetti. Use of the Boltzmann equation to simulate lattice-gas automata. *Phys. Rev. Lett.*, 61:2332–2335, 1988.

[2] S. Chen and G. D. Doolen. Lattice boltzmann method for fluid flows. *Annu. Rev. Fluid Mech.*, 30(1):329–364, 1998.

[3] Cyrus K. Aidun and Jonathan R. Clausen. Lattice-boltzmann method for complex flows. *Annu. Rev. Fluid Mech.*, 42(1):439–472, 2010.

[4] 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.

[5] K. Mattila, J. Hyväluoma, J. Timonen, and T. Rossi. Comparison of implementations of the lattice-boltzmann method. *Comput. Math. Appl.*, 55(7):1514 – 1524, 2008.

[6] C. Obrecht, F. Kuznik, B. Tourancheau, and J. J. Roux. A new approach to the lattice boltzmann method for graphics processing units. *Comput. Math. Appl.*, 61(12):3628 – 3638, 2011.
[7] M. Astorino, J. Becerra Sagredo, and A. Quarteroni. A modular lattice boltzmann solver for gpu computing processors. *SeMA Journal*, 59(1):53–78, 2012.
[8] C. Shu, X.D. Niu, Y.T. Chew, and Q.D. Cai. A fractional step lattice boltzmann method for simulating high reynolds number flows. *Math. Comput. Simulat.*, 72(2):201 – 205, 2006.
[9] X. He and L.-S. Luo. Lattice boltzmann model for the incompressible navier–stokes equation. *J. Stat. Phys.*, 88(3):927–944, Aug 1997.
[10] F. Nicoud and F. Ducros. Subgrid-scale stress modelling based on the square of the velocity gradient tensor. *Flow Turbul. Combust.*, 62(3):183–200, 1999.
[11] C. F. Colebrook. Turbulent flow in pipes, with particular reference to the transition region between the smooth and rough pipe laws. *Journal of the Institution of Civil Engineers*, 11(4):133–156, 1939.
[12] R. Örlü, T. Fiorini, A. Segalini, G. Bellani, A. Talamelli, and P. H. Alfredsson. Reynolds stress scaling in pipe flow turbulence—first results from ciclope. *Philos. T. Roy. Soc. A*, 375(2089), 2017.
[13] J. Jeong and F. Hussain. On the identification of a vortex. *J. Fluid Mech.*, 285:69–94, 1995.