Proceedings of the 7th European Congress on Computational Methods in Applied Sciences and Engineering,
ECCOMAS Congress 2016, pp. 1046-1060
DOI:10.7712/100016.1869.10678

ECCOMAS Congress 2016
VII European Congress on Computational Methods in Applied Sciences and Engineering
M. Papadrakakis, V. Papadopoulos, G. Stefanou, V. Plevris (eds.)
Crete Island, Greece, 5–10 June 2016

VALIDATION AND VERIFICATION OF A 2D LATTICE BOLTZMANN SOLVER FOR INCOMPRESSIBLE FLUID FLOW

Tamás István Józsa[1,3], Máté Szőke[2,3], Tom-Robin Teschner[3], László Könözsy[3*], Irene Moulitsas[3]

[1]University of Edinburgh
Institute for Energy Systems
Alrick Building, Max Born Crescent, Edinburgh, Scotland, United Kingdom, EH9 3BF
e-mail: t.jozsa@ed.ac.uk

[2]University of Bristol
Mechanical Engineering
Queen's Building, University Walk, Bristol, United Kingdom, BS8 1TR
e-mail: m.szoke@bristol.ac.uk

[3]Cranfield University
Centre for Fluid Mechanics and Computational Science
Building 52, Cranfield, Bedfordshire, United Kingdom, MK43 0AL
e-mail: {t.teschner, laszlo.konozsy, i.moulitsas}@cranfield.ac.uk

Keywords: Lattice Boltzmann Method, Validation, Verification, Computational Fluid Dynamics, LBM, CFD, CUDA, GPU, UPC, PGAS.

Abstract. *The lattice Boltzmann method (LBM) is becoming increasingly popular in the fluid mechanics society because it provides a relatively easy implementation for an incompressible fluid flow solver. Furthermore the particle based LBM can be applied in microscale flows where the continuum based Navier-Stokes solvers fail. Here we present the validation and verification of a two-dimensional in-house lattice Boltzmann solver with two different collision models, namely the BGKW and the MRT models [1]. Five different cases were studied, namely: (i) a channel flow was investigated, the results were compared to the analytical solution, and the convergence properties of the collision models were determined; (ii) the lid-driven cavity problem was examined [2] and the flow features and the velocity profiles were compared to existing simulation results at three different Reynolds number; (iii) the flow in a backward-facing step geometry was validated against experimental data [3]; (iv) the flow in a sudden expansion geometry was compared to experimental data at two different Reynolds numbers [4]; and finally (v) the flow around a cylinder was studied at higher Reynolds number in the turbulent regime. The first four test cases showed that both the BGKW and the MRT models were capable of giving qualitatively and quantitatively good results for these laminar flow cases. The simulations around a cylinder highlighted that the BGKW model becomes unstable for high Reynolds numbers but the MRT model still remains suitable to capture the turbulent von Kármán vortex street. The in-house LBM code has been developed in C and has also been parallelised for GPU architectures using CUDA [5] and for CPU architectures using the Partitioned Global Address Space model with UPC [6].*

1 Introduction

McNamara and Zanetti introduced the lattice Boltzmann Method (LBM) as a cure to remove the statistical noise encountered in its predecessor, the Lattice Gas Cellular Automata (LGCA) [7]. At the time, the impact of the LBM on research related to fluid dynamics could not have been foreseen. Almost three decades after its introduction, the LBM has enjoyed a wide range of applications including multiphase, porous media, magnetohydrodynamics, acoustics, medical and turbulent flows [8, 9] and of course the automotive industry [10, 11, 12]. The fast grown popularity stems from its ease of implementation and treatment of complex domains, including multi-physics problems which otherwise would require rather sophisticated numerical tools. At its core, it represents a discretised version of the Boltzmann equation and can be shown, via the Chapman–Enskog expansion, to adhere to the incompressible Navier–Stokes equations [13]. Therefore, the LBM exhibits favourable traits of a particle method, such as the ease of implementation for complex systems, while the modified Boltzmann equation represents a single, linear transport equation that only needs 10-20% of the computational time required by a conventional Navier–Stokes solver [14].

In this work we present the results we have obtained from a two-dimensional lattice Boltzmann solver with two different collision models, namely the BGKW and the MRT models [1]. We validate our solver against reference data for (i) a channel flow where the results are compared to the analytical solution, and the convergence properties of the collision models are determined; (ii) the lid-driven cavity problem [2] where the flow features and the velocity profiles are compared to existing simulation results at three different Reynolds number; (iii) the flow in a backward-facing step geometry is validated against experimental data [3]; (iv) the flow in a sudden expansion geometry is compared to experimental data at two different Reynolds numbers [4]; and finally (v) the flow around a cylinder is studied at higher Reynolds number in the turbulent regime.

With the current transistor technology plateauing, focus has shifted towards parallel scientific computing. A number of parallel software and hardware environments have emerged and combinations of these have been proposed in an attempt to tackle large scale problems accurately and efficiently [15]. Our LBM solver has further been parallelised using two different parallel computing environments and paradigms. The first approach relies of graphical processing units (GPUs) and utilises the Compute Unified Device Architecture (CUDA) framework [5] which has recently found interest in fluid dynamics, see for example Kuznik *et al.* [16]. The second approach relies on traditional central processing units and utilises the Unified Parallel C (UPC) framework [6]. UPC is based on the Partitioned Global Address Space (PGAS) programming model that aims to simplify the programming by abstracting the memory address space, hence allowing the programmer to concentrate on the solution of the underlying scientific problem. We further discuss the scalability of our solver for both parallelisation strategies followed.

2 Governing equations and computational methodology

To discretise the Boltzmann equation, particles are first represented by a Density Distribution Function (DDF) which is then restricted to propagate only in certain directions. These directions are determined by the chosen speed model. For the purposes of our study we have employed the D2Q9 speed model, which is a 2–dimensional model with 9 velocity directions. This model, presented in Figure 1, is a common choice for two-dimensional problems. The discretised Boltzmann transport equation is then formulated as

$$\frac{\partial f_i(\mathbf{x},t)}{\partial t} + (\mathbf{c}_i \cdot \nabla) f_i(\mathbf{x},t) = \Omega_i(f_i) \tag{1}$$

where i is the index of the restricted streaming direction and $f_i(\mathbf{x},t)$ is the DDF depending both on space and time. The discrete velocities, denoted by $\mathbf{c}_i$, take binary values and are defined by

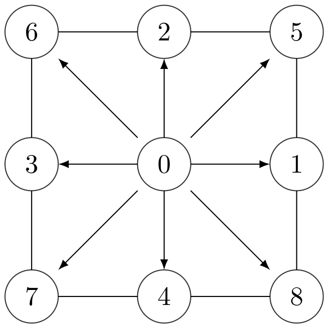


Figure 1: The D2Q9 speed model within a square lattice

$$\mathbf{c}_i = \begin{cases} 0, & i=0 \\ \left(\cos\left[(i-1)\,\pi/2\right], \sin\left[(i-1)\,\pi/2\right]\right), & i=1,2,3,4 \\ \sqrt{2}\left(\cos\left[(i-5)\,\pi/2+\pi/4\right], \sin\left[(i-5)\,\pi/2+\pi/4\right]\right). & i=5,6,7,8 \end{cases} \tag{2}$$

Hence the LBM can be separated into two steps: *collision* and *streaming*. The advection is represented by $(\mathbf{c}_i \cdot \nabla) f_i(\mathbf{x}, t)$ where the DDF is streaming to its neighbouring lattice sites. The collision is represented by $\Omega_i(f_i)$ and it is, unlike in the LGCA where a explicit collision treatment is possible, not straightforward. The collision can, at best, be approximated and different models have been proposed. We will limit ourselves to the Single Relaxation Time (SRT) scheme of Bhatnagar, Gross and Krook [17] (also referred to as the BGK or BGKW scheme due to the contribution of Welander [18]) and the Multi Relaxation Time (MRT) scheme of D'Humieres [19, 20]. Boundary conditions must be formulated for the DDF which are not readily available for incoming directions. We have followed the approach of Zou and He [21] which can be written in a compact form as

$$f_c = \mathrm{opp}(f_c) \pm \frac{2}{3}\varrho u_\perp, \tag{3}$$

$$f_+ = \mathrm{opp}(f_+) + \frac{1}{2}\left(f_{IP-} - f_{IP+}\right) \pm \frac{1}{6}\varrho u_\perp \pm \frac{1}{2}\varrho u_\parallel, \tag{4}$$

$$f_- = \mathrm{opp}(f_-) - \frac{1}{2}\left(f_{IP-} - f_{IP+}\right) \pm \frac{1}{6}\varrho u_\perp \pm \frac{1}{2}\varrho u_\parallel. \tag{5}$$

We have introduced f_c, f_+ and f_- as the unknown distribution functions on the boundary where f_c denotes the central one and $f_\pm$ its right and left neighbours, respectively. The function $\mathrm{opp}()$ returns the DDF pointing in the opposite direction while $f_{IP\pm}$ is the *inplane* DDF, i.e. parallel to the boundary. The boundary's normal and parallel velocity component are denoted by $u_\perp$ and $u_\parallel$ and the sign is positive if its component is pointing in the positive coordinate direction. The density on the boundary is obtained from

$$\varrho = \frac{1}{1 \pm u_\perp}\left[f_0 + \sum_i^{inplane} f_i + 2\left(\sum_j^{outside} f_j\right)\right], \tag{6}$$

where we sum over the DDFs on the boundary (inplane) and outside of the boundary. Dirichlet or Neumann type boundary conditions may be applied to open boundaries to obtain the velocity components required in Eq.(3) through Eq.(5). For solid boundaries, the simple bounce-back scheme suffices and we have incorporated the treatment of Bouzidi [22] for curved boundaries. The macroscopic quantities at the end of each time step are obtained from

$$\varrho = \sum_{i=0}^{8} f_i, \tag{7}$$

$$p = \frac{\varrho}{3}, \tag{8}$$

$$\mathbf{u} = \sum_{i=0}^{8} \frac{f_i \mathbf{c}_i}{\varrho}. \tag{9}$$

We have built on the work of [23] and implemented the above mentioned procedure using the C language for better binding with our parallel approaches. The implementation has been augmented in order to increase its computational performance and lower the memory requirements where possible. The parallel implementations of our solver can run either on GPUs using CUDA or CPUs using UPC. The corresponding validation results along with parallel scalability information are presented in the following section.

3 Results and discussion

In this section we present the verification and the validation of our LBM solver through a set of five different test cases. The geometries of the investigated domains and the applied boundary conditions are presented in Figure 2. In our discussion we will limit ourselves to only presenting results obtained with our serial LBM solver, upon which we rely when performing the relevant verification and validation studies. Our parallel implementations in CUDA and UPC have been verified by ensuring that they reproduce the same results as the serial solver. Finally we discuss the parallel strategies and present their scaling behaviour.

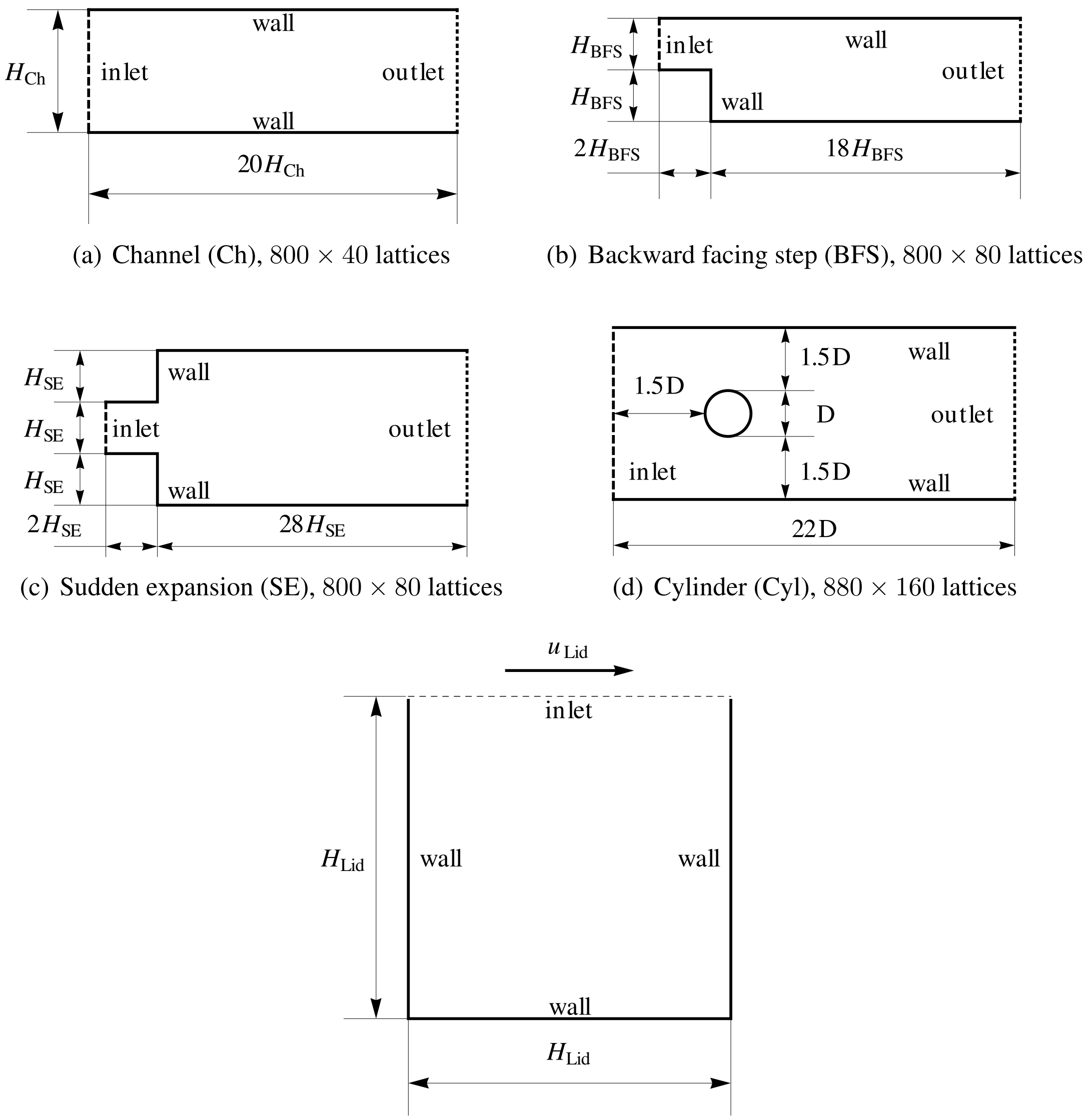
3.1 Laminar Flow in a Channel

Our first validation case is the widely used laminar channel flow where we can compare our results to an analytic velocity profile. It further lends itself to a detailed grid convergence study as we can make predictions about its exact profile.

Table 1 shows the results obtained for the non-dimensional peak velocity at the centre of the channel for three different grids. We have given the normalised grid spacing in parenthesis, based on the finest grid, and further calculated the extrapolated value of the peak velocity for a grid independent domain. As discussed in detail in Roache [24], we have calculated the Grid Convergence Index (GCI) for the coarse-medium (1,2) and medium-fine (2,3) mesh and calculated the apparent order p$_a$ to judge the accuracy of the solution. We can see from Table 1 that the BGKW collision model initially performs better compared to the MRT scheme, however, the extrapolated value shows that the MRT scheme approaches the exact solution faster. This is also reflected by the step decrease of the GCI when compared to the convergence

Table 1: Grid convergence study of the non-dimensional center (peak) velocity. The normalized grid spacing has been given in parenthesis and a grid independent peak velocity has been extrapolated. The GCI and apparent order is given from the coarse to medium (1,2) and medium to fine (2,3) mesh.

	$u/u_{\max}[-]$				GCI$_{1,2}$	GCI$_{2,3}$	p$_a$
$n_x \times n_y$	400 × 20 (4)	800 × 40 (2)	1600 × 80 (1)	extrapolated			
BGKW	0.9459	0.9761	0.9856	0.9899	1.77	0.55	1.67
MRT	0.7302	0.9191	0.9732	0.9948	10.29	2.78	1.81



(a) Channel (Ch), 800 × 40 lattices

(b) Backward facing step (BFS), 800 × 80 lattices

(c) Sudden expansion (SE), 800 × 80 lattices

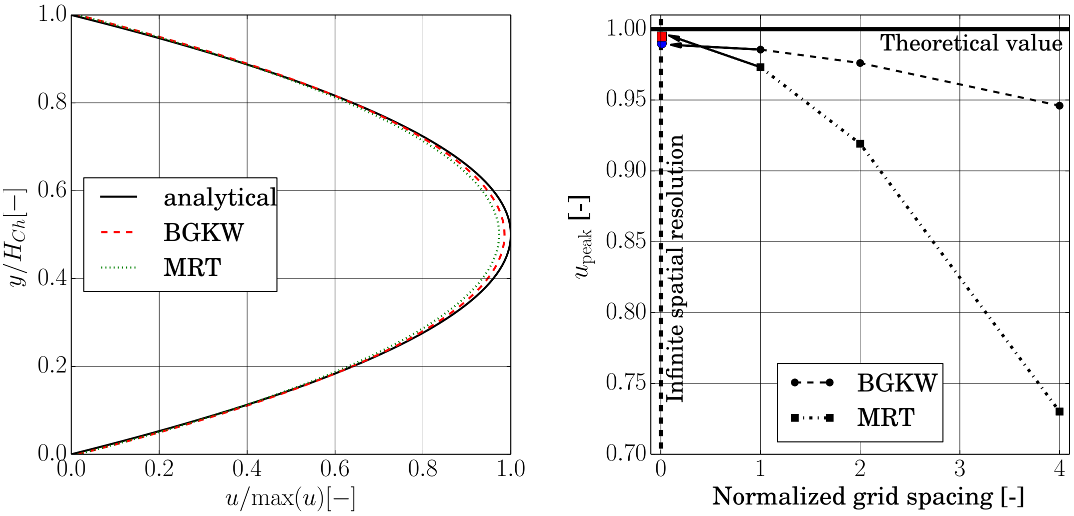
(d) Cylinder (Cyl), 880 × 160 lattices

(e) Lid-driven cavity (LC), 400 × 400 lattices

Figure 2: Computational domains with their dimensions and boundary conditions. Open or moving boundaries are given by dashed lines

behaviour of the BGKW scheme and the apparent order confirms this, which is a measure of how fast the error diminishes.

The obtained velocity profile at the channel's outlet is presented in Figure 3(a) for the fine mesh. Figure 3(b) shows the convergence of the peak velocity over the normalised grid spacing. Figure 3(a) shows that both collision models produce qualitatively the same velocity profile and capture the correct behaviour. In Figure 3(b) we have summarised our findings mentioned above and show the convergence properties of the BGKW and MRT scheme. In the limit of a grid independent result, the MRT scheme asymptotically approaches the theoretical value closer than the BGKW scheme which, however, produces more accurate results on the coarse grid. Although not evident from our presented results, it has to be noted that the MRT scheme is able to simulate lower lattice viscosities than the BGKW model which suffers from numerical instabilities. This is due to the nature of the model itself. While all DDFs are relaxed at the same rate in the BGKW model, the MRT scheme relaxes all of its DDFs in a transformed momentum space separately. Therefore it can treat each direction (Figure 1) individually and it is less prone to numerical instabilities. The lattice viscosity is used to calculate the Reynolds number and hence the BGKW scheme is not suited for high Reynolds number flows.



(a) Velocity profiles Re $= 100$ (fine mesh)

(b) Convergence of the collision models

Figure 3: Validation of the solver on the channel flow

3.2 Flow over a backward facing step

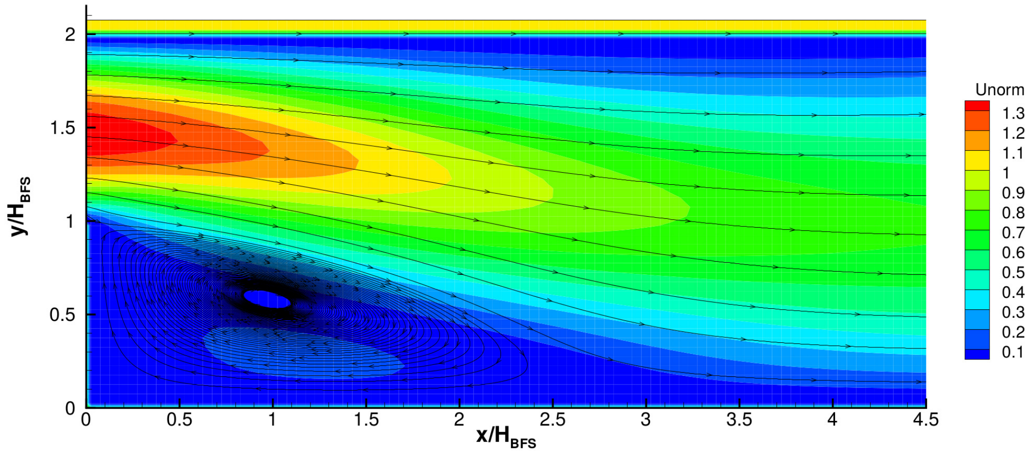
The Backward Facing Step (BFS) is another classical example for which an abundance of numerical and experimental data is available. We have examined the laminar flow at a Reynolds number of $Re = 100$ and we have chosen the experimental results of Armaly *et al.* [3] for our comparison.

Figure 4(a) shows the normalised velocity magnitude, superimposed by the streamlines as a visual guide. We only show the contour plot obtained for the MRT model as the BGKW results were qualitatively indistinguishable. A fully developed channel flow profile is present at the exit location of the smaller channel at $x/H_{BFS} = 0$ which then transitions into the bigger channel. A circulation area is formed downstream which is fed by the upstream channel. We have plotted the velocity profiles at various locations downstream of the channel and compared the results to the experimental data. We can see the fully developed velocity profile at the exit of the smaller channel, as indicated before, and then the development of the downstream profiles. Both the BGKW and MRT model under-predict the velocity profiles in the vicinity of the recirculation area and only approach the experimental data further downstream. Specifically, the magnitude of the velocity of the reverse flow and the jet are both under-estimated. Both collision models performed with little to chose between the two, which is also shown in the reattachment length error with -6.13% for the BGKW model and -5.48% for the MRT model, respectively.

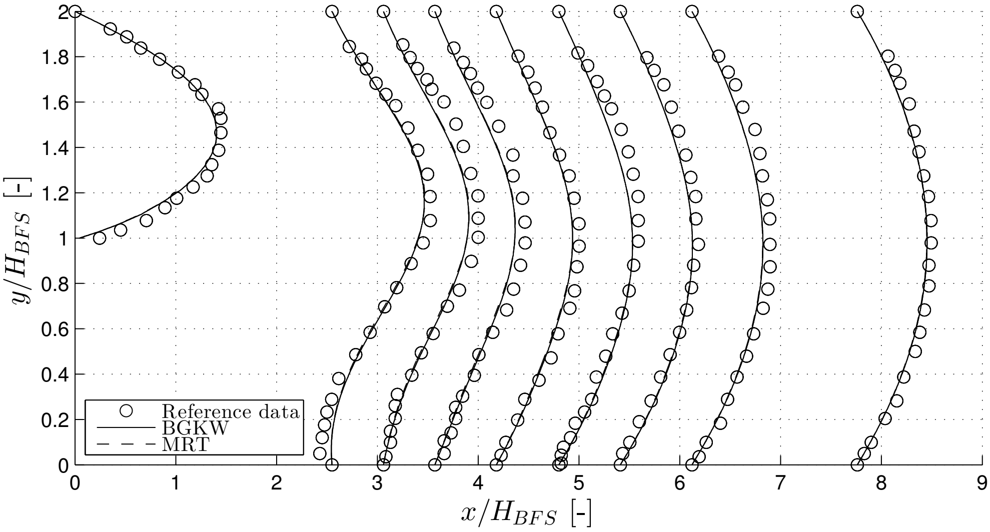
We conclude that our solver closely reproduces the velocity field and predicts the reattachment length within a few percent compared to the experimental data.

3.3 The bifurcation behaviour via a sudden expansion

The Sudden Expansion (SE) is a test case which provides further physical insight into the performance of our LBM solver. While its geometry is symmetrical, as seen in Figure 2(c), two different and distinct flow patterns are encountered for different Reynolds numbers. This is caused by the bifurcation of the fluid and is attributed to the non-linear behaviour of the Navier–Stokes equations. Hence, it is an interesting test case as the governing equation of the lattice Boltzmann method (Eq.(1)) is linear in nature. We



(a) Developed velocity field after the BFS (MRT)

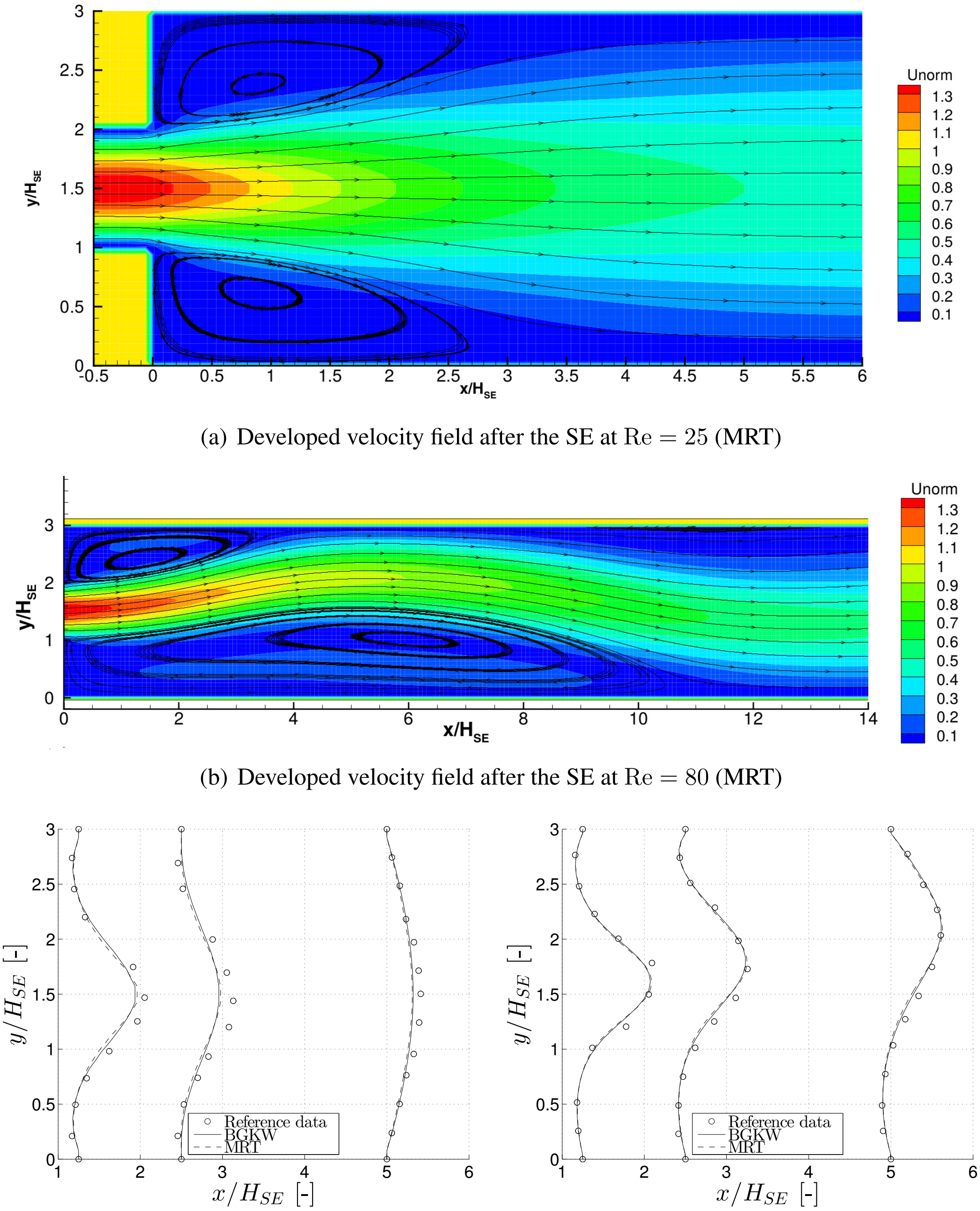


(b) Velocity profiles after the BFS

Figure 4: Validation of the solver on the BFS test case at $\mathrm{Re} = 100$

therefore performed simulations at Reynolds numbers of $Re = 25$, for a symmetrical flow field and $Re = 80$, for an anti-symmetrical, and compared our results against experimental data published by Fearn *et al.* [4].

Figures 5(a) and 5(b) show the normalised velocity plots for both the $Re = 25$ and $Re = 80$ cases and we have again shown the streamlines as a visual guide. For the symmetrical flow field we can identify two recirculation zones of the same magnitude, located in the lower and upper corner while the jet is expanding symmetrically downstream. The $Re = 80$ case is showing the expected symmetry breaking effect at higher Reynolds numbers and two distinct recirculation areas are formed. A third one, characteristic for even higher Reynolds numbers, is starting to form at the top before the jet is returning to a symmetric velocity profile. Figures 5(c) and 5(d) show the velocity profile compared to the experimental data at three different downstream locations. Qualitatively the results are matched although we see again



(a) Developed velocity field after the SE at Re = 25 (MRT)

(b) Developed velocity field after the SE at Re = 80 (MRT)

(c) Velocity profiles after the SE at Re = 25

(d) Velocity profiles after the SE at Re = 80

Figure 5: Validation of the solver on the SE test case at Re = 25 and at Re = 80

an under-prediction of the velocity magnitude in the transversal locations. The discrepancies are mainly found in the centre while the flow near the wall is resolved more closely. Interestingly, the $Re = 80$ case with the bifurcation exhibits closer agreement between the experimental and numerical results in terms of velocity profiles. Both the BGKW and MRT collisions models match well the experimental result.

In Table 2 we present the reattachment length L_{re} in non-dimensional units (normalised by H_{SE}, see Figure 2(c)) and its relative deviation to the experimental results. For both cases and for both recirculation zones in the $Re = 80$ case, the reattachment length has been under-predicted which is consistent with the observation of the backward facing step. However, we can see close agreement with the reference data. For the $Re = 25$ case we have less than 4% difference. The $Re = 80$ also shows acceptable

Table 2: Sudden expansion flow for two different Reynolds numbers. The $Re = 25$ is symmetrical and only one reattachment length needs to be calculated while the $Re = 80$ case produces a non symmetrical flow field which has two distinct (i.e. primary and secondary) vortex and hence two distinct reattachment lengths. The relative error corresponds to comparisons with Fearn *et al.* [4].

	$Re = 25$		$Re = 80$ (primary)		$Re = 80$ (secondary)	
	L_{re} [-]	$\varepsilon_{\text{relative}}$ [%]	L_{re} [-]	$\varepsilon_{\text{relative}}$ [%]	L_{re} [-]	$\varepsilon_{\text{relative}}$ [%]
BGKW	3.35	-2.33	3.45	-6.25	10.65	-9.44
MRT	3.30	-3.79	3.56	-3.26	10.60	-9.86

deviations of less than 7% for the primary and 10% for the secondary recirculation area, respectively. Hence the symmetric case was more accurately resolved which correlated well with the reattachment length prediction of the similar flow pattern over the backward facing step. The symmetry breaking at higher Reynolds numbers was captured, which is not surprising as the governing equation reproduces the Navier–Stokes equations via the Chapman–Enskog expansion, as mentioned in the introduction. However, it remains a linear equation and thus the higher difference compared to the symmetric case, in terms of reattachment length, is not surprising.

We can conclude that our solver performed well for the symmetric case and is capable of capturing the bifurcation behaviour of a fluid.

3.4 Flow structures in a lid-driven cavity flow

We now investigate the lid driven cavity flow which consists of simple boundary conditions. These can be accurately described and hence we can focus on the physical behaviour of the solver. We compare our results with the numerical data from Ghia *et al.* [2], where the vorticity-stream function approach was chosen to solve the 2D Navier–Stokes equations. It is of interest because it offers 2D data without any turbulence models, hence, we can validate our data against a Navier–Stokes solver which obtained results under the same conditions.

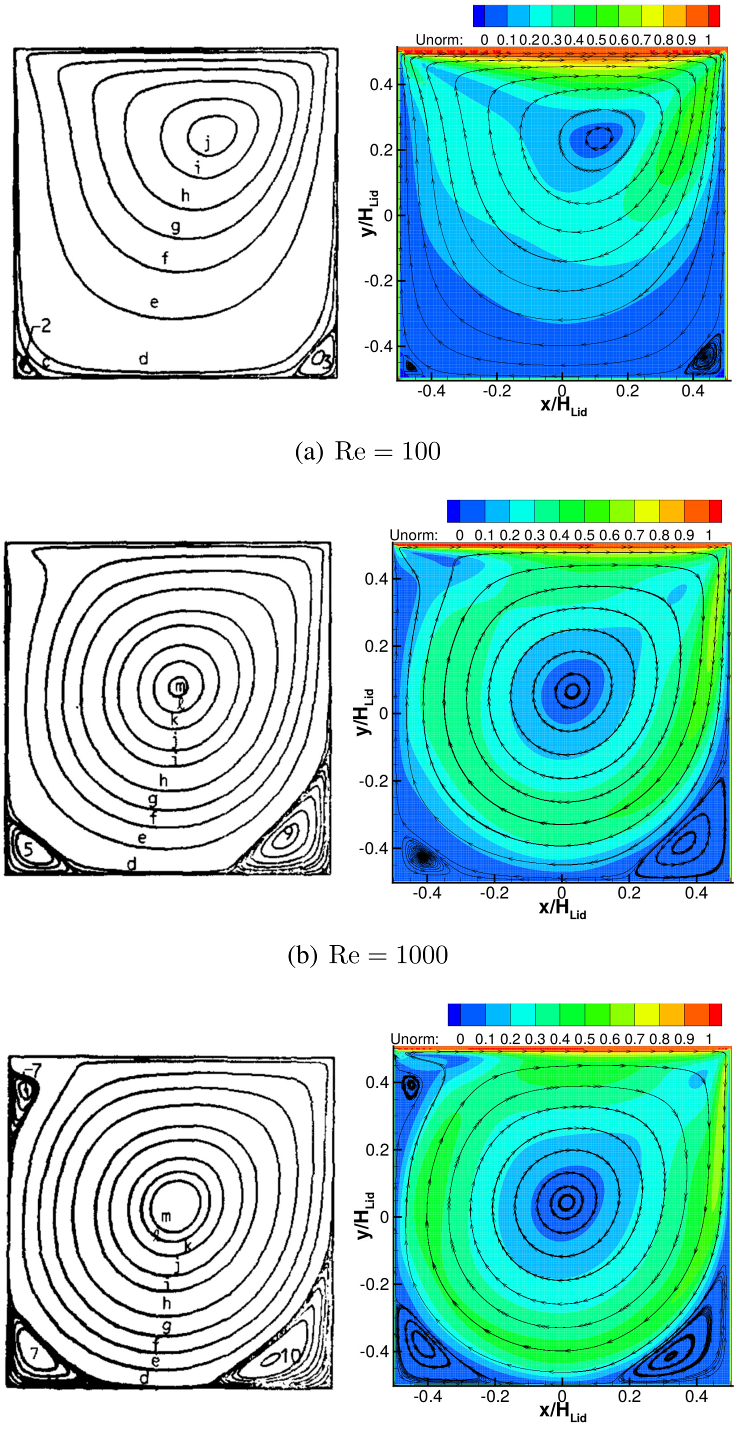
Figures 6(a)–6(c) show the streamlines obtained from the reference solutions (left) and our simulations (right) at $Re = 100$, $Re = 1000$ and $Re = 3200$, respectively. We see that the primary and secondary vortices in the lower corners are well-resolved for the $Re = 100$ and $Re = 1000$ case while the movement of the main vortex towards the centre is captured. For the high Reynolds number flow at $Re = 3200$, a tertiary vortex is formed near the lid which is consistent with the reference solution. Our simulations compare well with the reference data by pure visual means.

We further investigate the velocity profiles at the centerlines of the box, i.e. horizontally and vertically, and present our results in Figure 7(a)–7(c). For all investigated cases we see a good agreement between the velocity profiles of the reference data and both the BGKW and MRT scheme. Minor differences exist which can be either due to the numerical scheme used in the reference solution or the accuracy in the boundary condition imposition. The order of accuracy at the wall depends on the placement of the lattice, i.e. either coinciding with the DDFs or in-between them. See Succi [1] for a discussion.

We conclude that the two-dimensional flow features are accurately resolved and good agreement could be obtained with respect to the reference data.

3.5 Turbulent Flow around a cylinder

The solver presented in [23] has been extensively tested for the laminar flow around the channel. The von Kármán vortex street has been properly resolved and thus the sinusoidal lift and drag behaviour recovered. Here we present initial results for our LBM solver applied to the same cylinder flow at $Re = 10000$ to demonstrate the capabilities of the solver to resolve turbulent structures. The cylinder



(a) $\mathrm{Re} = 100$

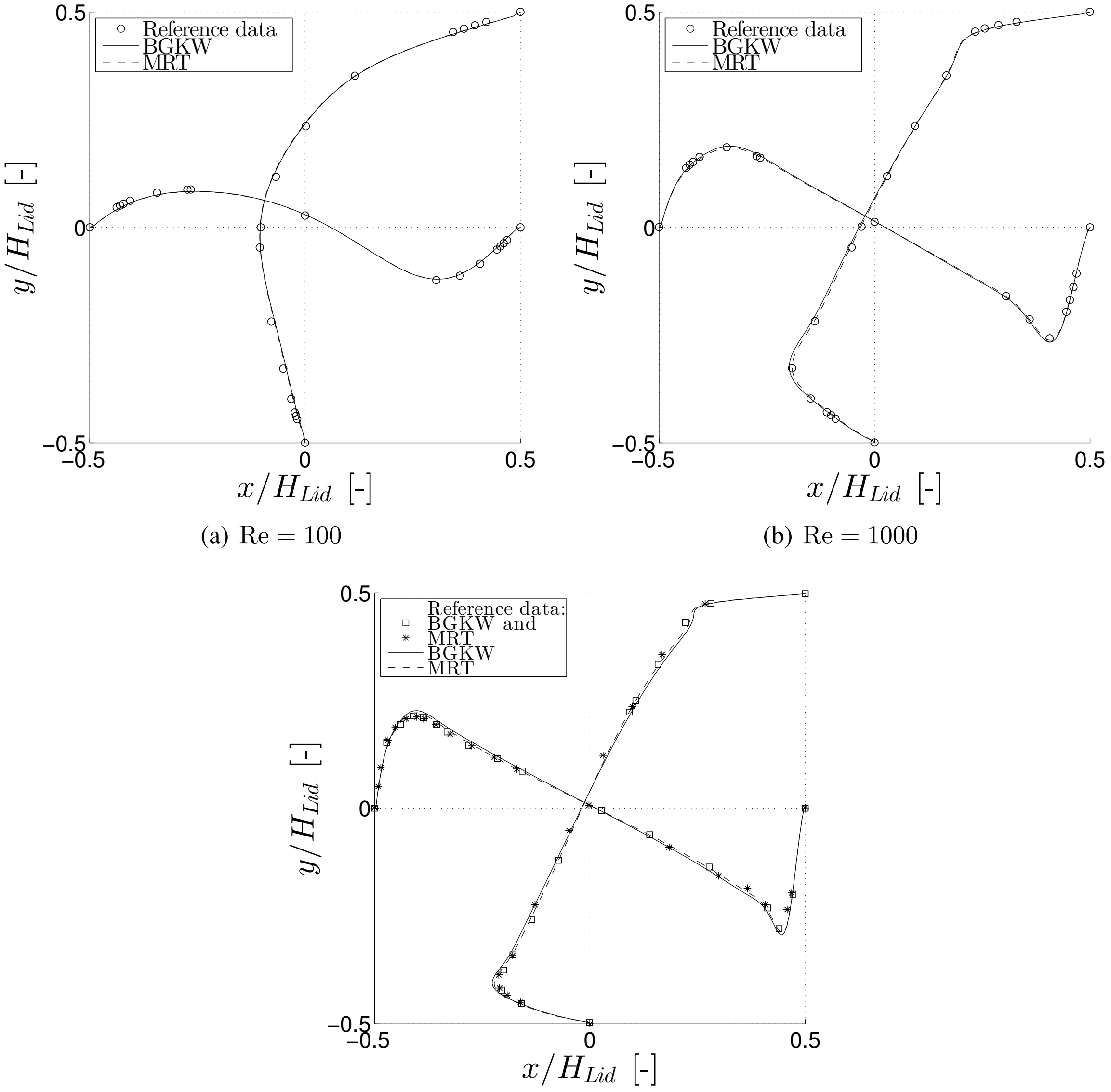
(b) $\mathrm{Re} = 1000$

(c) $\mathrm{Re} = 3200$

Figure 6: Velocity field comparison between Ghia *et al.* [2] (left) and the results obtained with the MRT collision model (right)

itself is an interesting test case as it consists of curved boundaries. Despite the stair-stepping profile around the cylinder, the treatment of Bouzidi *et al.* [22] accounts for the curvature of the geometry which is seen by the smooth vorticity distribution around the cylinder.

The presented results were obtained by the MRT collision model as the BGKW, as mentioned in section 3.1, is not suitable for high Reynolds number flows due to its inherent numerical instabilities. Fig-



(a) Re = 100

(b) Re = 1000

(c) Re = 3200

Figure 7: Velocity profiles at the horizontal and vertical centerline of the lid driven cavity at $Re = 100$, $Re = 1000$ and $Re = 3200$, respectively

ure 8 shows the contours of instantaneous, normalised vorticity magnitude in the downstream cylinder location. We can see that vortices are forming just downstream of the cylinder which then detach and are convected into the wake region. We show positive rotation in red and negative in blue so as to show the asymmetric vortex formation at the cylinder and its pairwise existence close to it, before turbulent mixing diffuses the overall vorticity.

In this proof of concept study we wanted to show the applicability of our solver to curved boundaries and high Reynolds number flows. Before we can properly validate our solver for turbulent flows, the solver needs to be extended to three dimensions.

3.6 Scaling behaviour of the solver

We conclude our discussion on validation and verification with the scaling behaviour of our parallel LBM solvers. We start by introducing the notions of parallel speed-up P_{SU} and parallel efficiency P_E which

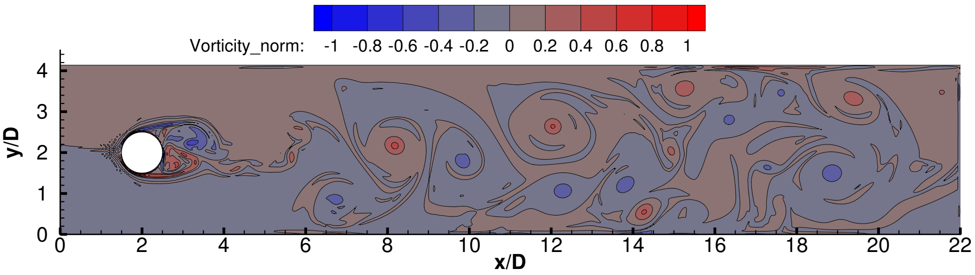


Figure 8: Flow past a circular cylinder at $Re = 10000$ showing the normalised vorticity magnitude

are defined respectively as

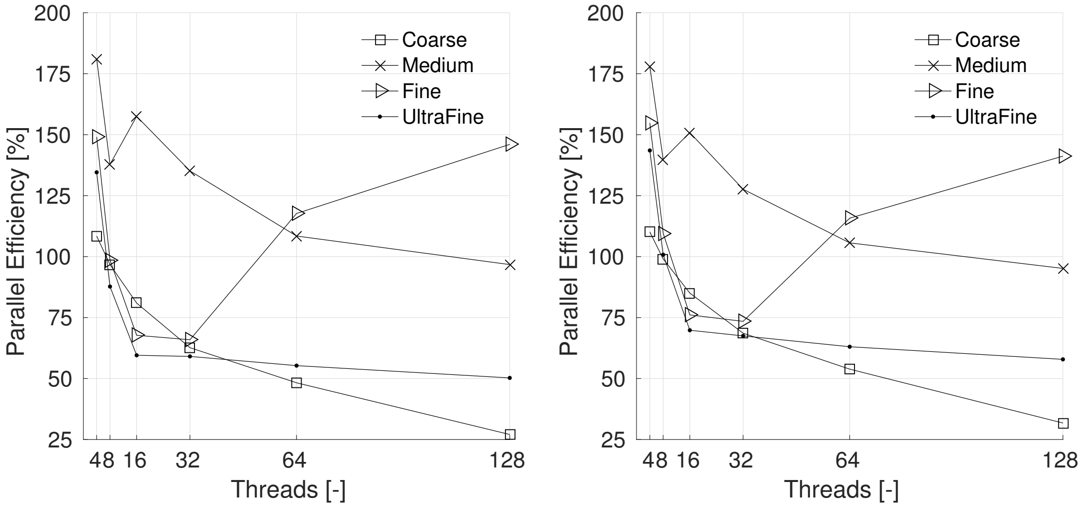
$$P_{SU} = t_{\text{serial}}/t_{\text{parallel}} \tag{10}$$

and

$$P_E = (P_{SU}/n_{\text{cores}})100\% = t_{\text{serial}}/(t_{\text{parallel}} \cdot n_{\text{cores}})100\% \tag{11}$$

By t_{serial} we denote the time required for a simulation to complete serially, by t_{parallel} we denote the time required for a parallel simulation to complete and by n_{cores} we denote the number of cores employed during a parallel simulation. We have tested our solver on the lid driven cavity flow for which we have created four different meshes with increasing number of points: the spatial resolutions 128^2, 256^2, 512^2, and 1024^2 correspond to the coarse, medium, fine and ultrafine grids, respectively. The results for the UPC and CUDA strategy are summarised in Figure 9 and Figure 10 respectively.

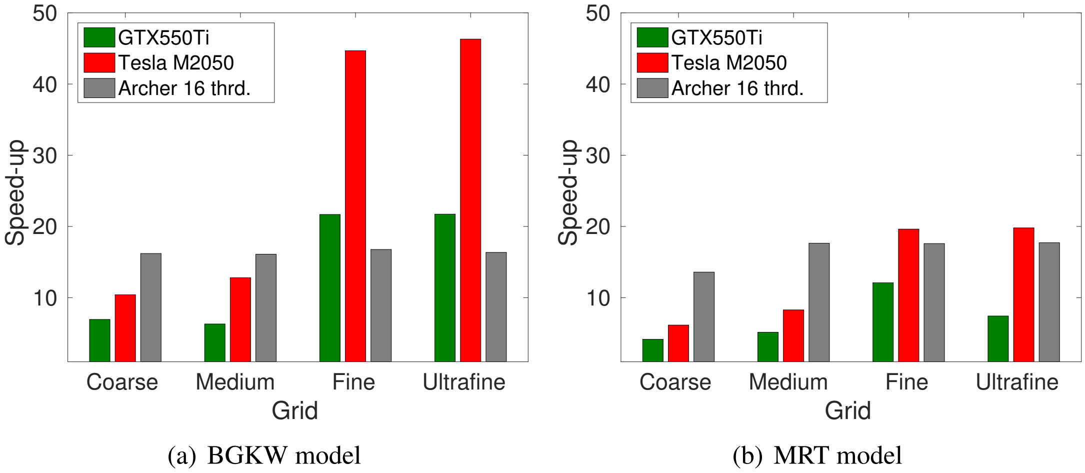
The UPC implementation shows that the parallel efficiency can be greater than 100%. Opposed to the serial computation, tasks can be overlapped in the UPC environment which enables the solver to achieve such high rates of efficiency. However, we also see that it drops for a larger amount of threads and all, except the fine mesh, experience degeneration of the efficiency due to increasing overheads. The fact that the fine mesh does not follow this trend may have several roots. Although PGAS gives a coherent view on memory, physically it is still separated. The way the partitioning and communications are done in the background may explain the trend of the fine grid in Figure 9 but also highlights the need for code profiling and compiler optimisation. Results obtained with the BGKW and the MRT models show similar



(a) BGKW model

(b) MRT model

Figure 9: Parallel performance of the parallel UPC code using double precision arithmetic



(a) BGKW model

(b) MRT model

Figure 10: Speed-up evaluation of the parallel codes using single precision arithmetic on two different GPUs

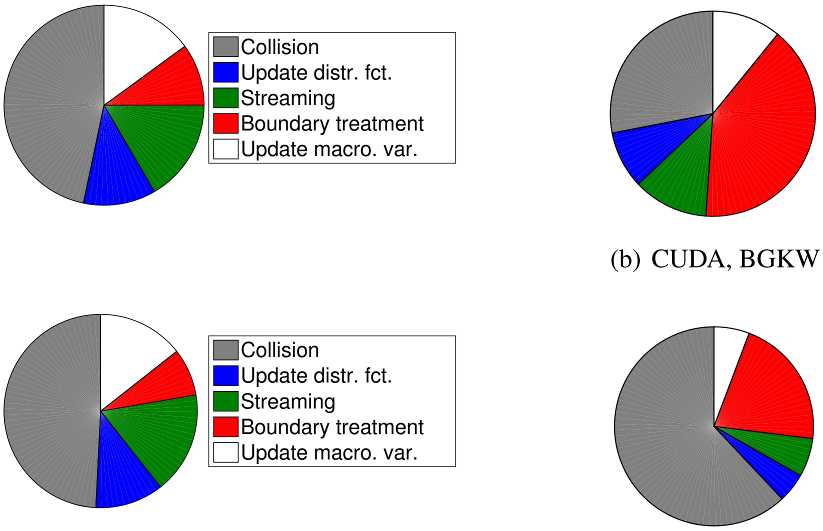
parallel efficiency trends as seen in Figure 9(a) and 9(b). It is worth to mention that the performance, measured with single and double precision, showed only marginal differences.

Figure 10 presents the speed-up of the CUDA implementation. We have tested the code on two different GPUs; the GTX550Ti which is targeted as a consumer graphics card and the Tesla M2050 which is purely designed for scientific computations. We have given speed-ups obtained with UPC for 16 threads on all four grid levels as a reference. The Tesla M2050 consistently yielded higher speed-ups compared to the GTX550Ti which were about twice as much. While the UPC approach yielded a speed-up approximately equal to its number of cores, the CUDA implementation, due to its different architecture, was able to achieve speed-ups of up to 45 compared to the serial code, see Figure 10(a). In the case of the CUDA implementation we experienced a significant drop for double precision computations which only yielded about one third of the single precision speed-ups. We measured a relatively high speed-up and strong scalability on the GPUs with the BGKW model, however, a considerable performance drop was experienced with the MRT collision model when we compare Figures 10(a) and 10(b).

The profiling results, presented in Figure 11, help to understand the speed-up behaviour of the solver. In Figure 11(c) we can see that the collision is the most computationally expensive operation in the case of the MRT collision model. As a matter of fact in the case of the CUDA implementation it takes approximately 60% of every time step, seen in Figure 11(d). While the performance difference between the BGKW and the MRT model seems to be negligible based on the serial profiling results of Figure 11(a) and Figure 11(c), the MRT collision step is the bottleneck of the time marching on the GPUs as can bee seen in Figure 11(b) and Figure 11(d). The distribution functions are stored in physically different locations for the MRT collision model. Hence, during the summation of all nine discrete velocity directions, data needs to be transferred via the global memory which has a relatively low bandwidth. Therefore, the MRT scheme exhibits a slower computational time, compared to the BGKW model, which could be circumvented by using shared memory on the GPU side.

4 Conclusions

In this study, we have validated our 2-dimensional lattice Boltzmann solver against the classical benchmark problems of a channel, backward facing step, sudden expansion, lid driven cavity and turbulent cylinder flow. We have shown that our solver is capable of treating complex geometries with curved boundaries at high Reynolds number flows and have demonstrated the accuracy of our solver to be within a small percentage of reference data. We extended our solver with high performance capabilities and included CUDA as a GPU-based, and UPC, as a CPU-based parallelsiation strategy, respectively. High efficiencies were obtained for the UPC approach for moderate numbers of CPUs while overheads



(b) CUDA, BGKW

(c) Serial code, MRT

(d) CUDA, MRT

Figure 11: Profiling results on the medium mesh using single precision arithmetic

were introduced at higher numbers. CUDA was able to give speed-ups as high as 45 compared to the serial solver which decreased drastically for double precision computations. The extension of solver to three dimensions is left to the future which will allow for turbulent flow applications.

5 Acknowledgement

This work used the ARCHER UK National Supercomputing Service (`http://www.archer.ac.uk`), and the Eddie2 local cluster at the University of Edinburgh for the parallel performance evaluation; we are grateful for the support of their staff. Special thanks to Gennaro Abbruzzese for the original C++ version of the code and the mesh generator which was used for our simulations.

REFERENCES

[1] S. Succi, *The lattice Boltzmann equation: for fluid dynamics and beyond*. Oxford University Press, Oxford, 2001.

[2] U. Ghia, K. N. Ghia, and C. T. Shin, High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. *Journal of Computational Physics*, **48**, 387–411, 1982.

[3] B. F. Armaly, F. Durst, J. C. F. Pereira, and B. Schnung, Experimental and theoretical investigation of backward-facing step flow. *Journal of Fluid Mechanics*, **127**, 473–496, 1983.

[4] R. M. Fearn, T. Mullin, and K. A. Cliffe, Nonlinear flow phenomena in a symmetric sudden expansion. *Journal of Fluid Mechanics*, **211**, 595–608, 1990.

[5] J. Sanders and E. Kandrot, *CUDA by example: an introduction to general-purpose GPU programming*. Addison-Wesley Professional, 2010.

[6] T. El-Ghazawi, W. Carlson, T. Sterling, K. Yelick, *UPC: Distributed Shared Memory Programming*. Wiley, 2005.

[7] G. R. McNamara and G. Zanetti, Use of the Boltzmann equation to simulate lattice-gas automata. *Physical Review Letters*, **61**, 2332, 1988.

[8] S. Chen, G. D. Doolen, Lattice-Boltzmann method for fluid flows. *Annual Review of Fluid Mechanics*, **30**, 329–364, 1998.

[9] C. K. Aidun, J. R. Clausen, Lattice-Boltzmann method for complex flows. *Annual Review of Fluid Mechanics*, **42**, 439–472, 2010.

[10] D.M. Holman, R.M Brionnaud, Z. Abiza, Solution to industry benchmark problems with the lattice-Boltzmann code XFlow. *6th European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2012)*, Vienna, Austria, September 10–14, 2012.

[11] C.L.A. Lafferty, A.C. Alajbegovic, K.B. Horrigan, Under-hood Thermal Simulation of a Class 8 Truck. *SAE Technical Paper*, 2007.

[12] G. Balasubramanian, L.A. Raghu Mutnuri, Z. Sugiyama, S. Senthooran, D. Freed, A computational process for early stage assessment of automotive buffeting and wind noise. *SAE International Journal of Passenger Cars - Mechanical Systems*, **6**, 2, 1231–1238, 2013.

[13] X. He, L.-S. Luo, Lattice Boltzmann Model for the Incompressible Navier–Stokes Equation. *Journal of Statistical Physics*, **88**, 3, 927–944, 1997.

[14] M.R. Salimi, M. Taeibi-Rahni, F. Jam, Heat transfer analysis of a porously covered heated square cylinder, using a hybrid Navier–Stokes-lattice Boltzmann numerical method. *International Journal of Thermal Sciences*, **91**, 59–75, 2015.

[15] J.A. Diaz, C.B. Muñoz-Caro, A.B. Niño, A survey of parallel programming models and tools in the multi and many-core era. *IEEE Transactions on Parallel and Distributed Systems*, **23**, 8, 1369–1386, 2012.

[16] F. Kuznik, C. Obrecht, G. Rusaouen, J.-J. Roux, LBM based flow simulation using GPU computing processor. *Computers and Mathematics with Applications*, **59**, 7, 2380–2392, 2010.

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

[18] P. Welander, On the temperature jump in a rarefied gas. *Arkiv Fysik*, **7**, 1954

[19] D. d'Humières, Generalized lattice-Boltzmann equations. In *Rarefied gas dynamics: Theory and simulations* (ed. B. D. Shizgal & D. P. Weaver), 1992.

[20] D. d'Humières, Multiplerelaxationtime lattice Boltzmann models in three dimensions. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, **360**, 437–451, 2002.

[21] Q. Zou and X. He, On pressure and velocity boundary conditions for the lattice Boltzmann BGK model. *Physics of Fluids*, **9**, 1591–1598, 1997.

[22] M. A. Bouzidi, M. B. Firdaouss, P. A. Lallemand, Momentum transfer of a Boltzmann-lattice fluid with boundaries. *Physics of Fluids*, **13**, 11, 3452–3459, 2001.

[23] G. Abbruzzese, Development of a 2D lattice Boltzmann code. *Master's thesis, Cranfield University*, 2013.

[24] P. J. Roache, *Verification and Validation in Computational Science and Engineering*. Hermosa Publishers, Albuquerque, New Mexico, 1998.