# A Non-uniform Staggered Cartesian Grid Approach for Lattice-Boltzmann Method *

Pedro Valero-Lara[1] and Johan Jansson[12]

[1] BCAM-Basque Center for Applied Mathematics, Bilbao, Spain.
[2] KTH Royal Institute of Technology, Stockholm, Sweden.

**Abstract**
We propose a numerical approach based on the Lattice-Boltzmann method (LBM) for dealing with mesh refinement of Non-uniform Staggered Cartesian Grid. We explain, in detail, the strategy for mapping LBM over such geometries. The main benefit of this approach, compared to others, consists of solving all fluid units only once per time-step, and also reducing considerably the complexity of the communication and memory management between different refined levels. Also, it exhibits a better matching for parallel processors. To validate our method, we analyze several standard test scenarios, reaching satisfactory results with respect to other state-of-the-art methods. The performance evaluation proves that our approach not only exhibits a simpler and efficient scheme for dealing with mesh refinement, but also fast resolution, even in those scenarios where our approach needs to use a higher number of fluid units.

*Keywords:* Lattice-Boltzmann Method, Mesh Refinement, Parallel Algorithms, Multicore.

## 1 Introduction

Advanced strategies for the efficient implementation of computationally intensive numerical methods have a strong interest in the industrial and academic community. The Lattice Boltzmann Method (LBM) is a clever discretization of the Boltzmann equation [15]. Multiple studies have compared the efficiency of LBM with other methods [1, 5], showing that it can achieve an equivalent numerical accuracy over a large number of applications. Due to the particular features of LBM, it has been adapted to numerous parallel computer architectures, such as multicore CPU [11], manycore accelerators [16] and cluster [9] being a very profitable method

*This research has been supported by EU-FET grant EUNISON 308874, the Basque Excellence Research Center (BERC 2014-2017) program by the Basque Government, the Spanish Ministry of Economy and Competitiveness MINECO: BCAM Severo Ochoa accreditation SEV-2013-0323 and the Project of the Spanish Ministry of Economy and Competitiveness with reference MTM2013-40824. We acknowledge PRACE for awarding us access to the supercomputer resources Hermit, Hornet and SuperMUC based in Germany at The High Performance Computing Center Stuttgart (HLRS) and Leibniz Supercomputing Center (LRZ), and from the Swedish National Infrastructure for Computing (SNIC) at PDC.

for parallel computing, showing a high MFLUPS (Millions of Fluid Lattice Updates per Second) ratio. Also, given the growing popularity of LBM, multiple tools [19, 7] have arisen which has consolidated this method into academia and industry. In particular, in this work we have considered the LBM-HPC framework [7] as our reference software tool.

LBM is an efficient and fast method, however the usage of Cartesian grids is expensive. Although scientific problems exist for which a homogeneous description of the domain is a reasonable choice, it is usually desirable to solve regions of high geometrical complexity with a finer grid to minimize the computational cost. In this sense, the main contribution of this work is a refinement approach based on a *Non-uniform Staggered Cartesian Grid* (NSCG) for LBM solvers. This approach has been previously considered over other methods [20].

Several refinement techniques have been implemented for LBM-based solvers, such as adaptive-mesh-refinement (ARM) [2], multi-grid [14], and multi-domain [6]. Each of these techniques exhibit its own advantages and disadvantages. For ARM and multi-grid the coarse grid is present all over the simulation domain. In the multi-domain refinement, the regions where refined patches are inserted are taken off the coarse grid. The multi-domain method exhibits better performances and higher memory savings and a more complex grids coupling with respect to the other two approaches. Nevertheless, those approaches based on ARM present the most complex scenario for data management, due mainly to its dynamic data structure, while the multi-grid operations are more profitable for programming and data management. All these approaches require the use of synchronization points as there exists a data dependence among the different refined levels, such that those refined regions must be computed at least several times per time-step. All this degrades the performance and disfavors the implementation on parallel processors. Our *NSCG* approach attempts to address these shortcommings by reducing considerably the complexity of the previous refinement approaches.

This paper is structured as follows: Section 2 introduces the main characteristics of LBM; Section 3 briefly describes the physical problem at hand and the general numerical framework that has been selected to cope with it (LBM coupled with *NSCG* technique). Section 4 and 5, numerical accuracy and performance studies are performed, respectively; finally, in Section 6 some conclusions are outlined.

## 2 Lattice-Boltzmann Method

Lattice-Boltzmann methods (LBM) combines those characteristics developed to solve the Boltzmann equation over a finite number of microscopic speeds. In this sense, LBM presents some lattice-symmetry features which allow the conservation of the macroscopic moments [3]. The standard lattice-Boltzmann method [13] is an explicit-time-step solver for incompressible flows. This numerical scheme divides each temporal iteration into two steps, one for propagation-advection (stream) and a collision step which represents inter-particle interactions, achieving a first order in time and second order in space method. This method describes the fluid behavior at mesoscopic level. At this level the fluid is modeled by a distribution function of the microscopic particles, $f$. Similarly to the Boltzmann equation, LBM solves the particle speed distribution by discretizing the speed space over a discrete finite number of possible speeds. The distribution function evolves according to the equation:

$$\frac{\partial f}{\partial t} + c\nabla f = \Omega \tag{1}$$

where $f$ is the particle distribution function, $c$ is the discrete space of speeds and $\Omega$ is the collision operator. By discretizing the distribution function $f$ in space, in time and in speed

$(\mathbf{c} = \mathbf{c}_i)$ we obtain $f_i(\mathbf{x}, t)$ which describes the probability of finding a particle at time $t$ located at $\mathbf{x}$ with speed $\mathbf{c}_i$.

The term $c\nabla f$ can be descritized as:

$$c\nabla f = c_i \nabla f_i = \frac{f_i(\mathbf{x} + c_i\Delta t, t + \Delta t) - f_i(\mathbf{x}, t + \Delta t)}{\Delta t} \tag{2}$$

In this way the particles can only move along the links of a regular Lattice (Figure 1) defined by the discrete speeds ($c_0 = c(0, 0)$; $c_i = c(\pm 1, 0), c(0, \pm 1), i = 1 \cdots 4$; $c_i = c(\pm 1, \pm 1), c(\pm 1, \pm 1), i = 5 \cdots 8$ with $c = \Delta x/\Delta t$), so that the synchronous particle displacements $\Delta \mathbf{x}_i = \mathbf{c}_i \Delta t$ never take the fluid particles away from the Lattice. In the present work, we follow the standard two-dimensional 9-speed lattice *D2Q9* [3].

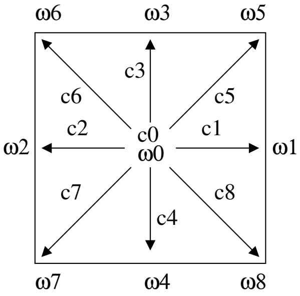

Figure 1: The standard two-dimensional 9-speed lattice (*D2Q9*) used.

The operator $\Omega$ describes the changes suffered by the collision of the microscopic particles which affect to distribution function $f$. To calculate the collision operator we consider the *BGK* (Bhatnagar-Gross-Krook) formulation [10] which relies upon a unique relaxation time $\tau$ toward the equilibrium distribution $f_i^{eq}$:

$$\Omega = -\frac{1}{\tau}\left(f_i\left(\mathbf{x}, t\right) - f_i^{eq}\left(\mathbf{x}, t\right)\right) \tag{3}$$

The equilibrium function $f^{eq}\left(\mathbf{x}, t\right)$ can be obtained by Taylor series expansion of the Maxwell-Boltzmann equilibrium distribution [13]:

$$f_i^{eq} = \rho\omega_i \left[1 + \frac{\mathbf{e_i} \cdot \mathbf{u}}{c_s^2} + \frac{\left(\mathbf{e_i} \cdot \mathbf{u}\right)^2}{2c_s^4} - \frac{\mathbf{u}^2}{2c_s^2}\right] \tag{4}$$

where $c_s$ is the speed of sound ($c_s = 1/\sqrt{3}$) and the weight coefficients $\omega_i$ are $\omega_0 = 4/9$, $\omega_i = 1/9,\ i = 1 \cdots 4$ and $\omega_5 = 1/36,\ i = 5 \cdots 8$ according to the current normalization. Through the use of this collision operator and substituting the term $\frac{\partial f_i}{\partial t}$ using a first order temporal discretization, the discrete Boltzmann equation can be written as:

$$\frac{f_i(\mathbf{x}, t + \Delta t) - f_i(\mathbf{x}, t)}{\Delta t} + \frac{f_i(\mathbf{x} + e_i\Delta t, t + \Delta t) - f_i(\mathbf{x}, t + \Delta t)}{\Delta t} = -\frac{1}{\tau}\left(f_i\left(\mathbf{x}, t\right) - f_i^{eq}\left(\mathbf{x}, t\right)\right) \tag{5}$$

which can be written compactly as:

$$f_i\left(\mathbf{x} + \mathbf{e}_i\Delta t, t + \Delta t\right) - f_i\left(\mathbf{x}, t\right) = -\frac{\Delta t}{\tau}\left(f_i\left(\mathbf{x}, t\right) - f_i^{eq}\left(\mathbf{x}, t\right)\right) \tag{6}$$

Equation 6 is typically advanced in time in two stages, the collision step and the streaming step, i.e.:
Given $f_i(\mathbf{x}, t)$ compute:

$\rho = \sum f_i(\mathbf{x}, t)$ and
$\rho\mathbf{u} = \sum \mathbf{c}_i f_i(\mathbf{x}, t)$

Collision stage:

$$f_i^* (\mathbf{x}, t + \Delta t) = f_i (\mathbf{x}, t) - \frac{\Delta t}{\tau} \left(f_i (\mathbf{x}, t) - f_i^{eq} (\mathbf{x}, t)\right)$$

Streaming stage:

$$f_i (\mathbf{x} + \mathbf{c}_i \Delta t, t + \Delta t) = f_i^* (\mathbf{x}, t + \Delta t)$$

# 3 The Non-uniform Staggered Cartesian Grid (NSCG) Algorithm for Lattice-Boltzmann Method

This section describes the main idea behind our algorithm, which consists of the concept that the usage of different relaxation frequencies ($\delta = 1/\tau$) for the same lattice unit can be considered for dealing with refined grids. Figure 2 illustrates some classical simulation cases for uniform (top) and non-uniform (bottom) grid refinement.

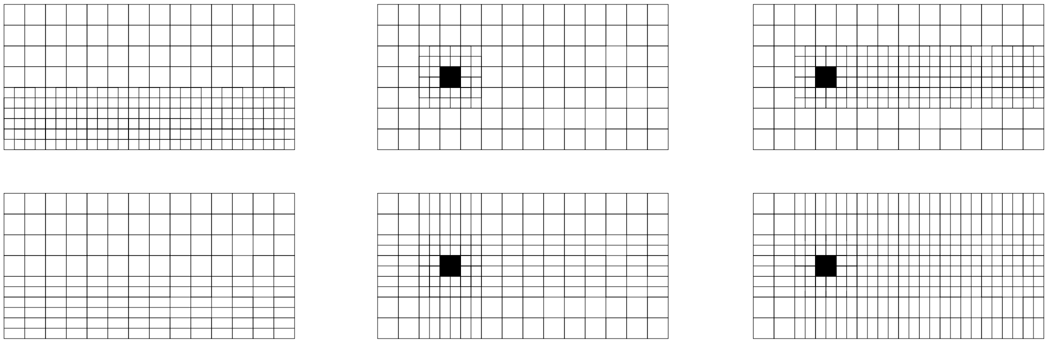

Figure 2: Grid geometry for the first test scenario, channel flow (left), and second one, flow past square cylinder (medium → A sub-scenario and right → B sub-scenario), for the *multi-domain* (top) and *NSCG* (bottom) approaches.

The rescaling of physical quantities is inspired by the work by [6]. In the following, we work in lattice units, the $c$ subindex is assigned to coarse grid units, while the $f$ subindex for fine grid units. To keep the discussion simple we chose to refine the grid by a factor of two only. Thus, $\Delta x_c$ and $\Delta x_f$ are the spatial discretization of the coarse and fine grid respectively, so it follows that: $\Delta x_f = \Delta x_c/2$.

Given the aforementioned relation, the rescaling of the relaxation time now needs to be discussed. Let the Reynolds number be: $Re_n = U_n L_n/v_n$, where $n$ refers to fine ($f$) or coarse ($c$) domain, $U$, $L$, $v$ are the characteristic velocity, the characteristic length-scale and the viscosity, respectively. $U_n$ and $L_n$ are given by: $U_n = U\Delta t_n/\Delta x_n$ and $L_n = L/\Delta x_n$. Forcing the Reynolds number to be independent of the grid, one obtains:

$$Re_c = Re_f \equiv \frac{UL\Delta t_c}{\Delta x_c^2 v_c} = \frac{UL\Delta t_f}{\Delta x_f^2 v_f} \tag{7}$$

Recalling that $\Delta x_f = \Delta x_c/2$, the rescaling of the viscosity's:

$$v_f = \frac{\Delta x_c}{\Delta x_f} v_c \tag{8}$$

By using the relation between the relaxation frequency and viscosity ($v = c_s^2(1/\delta - 1/2)$) [6], $\delta_f$ can be written as:

$$\delta_f = \frac{2\delta_c}{4 - \delta_c} \tag{9}$$

The rescaling process introduced above can be applied over a *Nonuniform Staggered Cartesian Grid* by following the scheme illustrated in Figure 3.

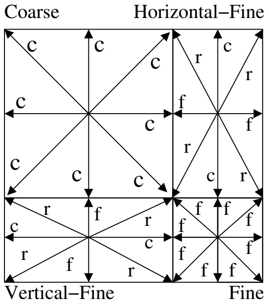

Figure 3: kind of refined lattices for the *NSCG* approach (right). $\delta_c$, $\delta_f$ and $\delta_r$ denoted in the graph as $c$, $f$ and $r$ respectively.

Depending on the direction to be refined, one of the three different $\delta$ factors, $\delta_c$, $\delta_f$, are considered $\delta_r$. Both, $\delta_c$ and $\delta_f$ have been described in the above subsection. $\delta_r$ is computed as arithmetic average of the other two $\delta$ factors ($\delta_r = (\delta_c + \delta_f)/2$). This scheme only affects the collision-LBM stage (Equation 9):

Next, we present the coupling algorithm: It consists of a simple communication from/to the different refined levels, which affects only the streaming-LBM stage. In essence, an adaptation to the local relaxation factor is performed before computing the streaming step over those lattice functions ($f_i$), which need to be communicated to boundary regions among the different levels. To visualize the differences among our NSCG algorithm with respect to standard LBM, the major steps of our method are now detailed:

Given $f_i(\mathbf{x}, t)$ compute:

$\rho = \sum f_i(\mathbf{x}, t)$ and
$\rho\mathbf{u} = \sum \mathbf{c}_i f_i(\mathbf{x}, t)$

Collision stage:

$f_i^* (\mathbf{x}, t + \Delta t) = f_i (\mathbf{x}, t) - \delta_* (f_i (\mathbf{x}, t) - f_i^{eq} (\mathbf{x}, t))$

Depending on the lattice direction ($i$) and the refinement level associated, $\delta_c$, $\delta_r$ or $\delta_f$ is considered.

Streaming stage:

If target ($f_i^t$) and source ($f_i^s$) lattice direction share the same refinement level do:
$f_i\left(\mathbf{x}+\mathbf{c}_i\Delta t, t+\Delta t\right) = f_i^*\left(\mathbf{x}, t+\Delta t\right)$
If the target lattice-direction refinement level is different do:
$f_i^t = f_i^s - \delta_*^t(f_i^s - f_i^{s-eq})$

To clarify, next, we show the transfers from/to the lattices, for the streaming major step, according to the simple geometry illustrated by Figure 3. The enumeration is in agreement with respect to Figure 1.

$$\begin{aligned} f_2^c &= f_2^{hor.-f} - \delta_c(f_2^{hor.-f} - f_2^{hor.-f-eq}) & \qquad f_1^{hor.-f} &= f_1^c - \delta_f(f_1^c - f_1^{c-eq}) \\ f_6^c &= f_6^f - \delta_c(f_6^f - f_6^{f-eq}) & \qquad f_4^{ver.-f} &= f_4^c - \delta_f(f_4^c - f_4^{c-eq}) \\ f_3^c &= f_3^{ver.-f} - \delta_c(f_3^{ver.-f} - f_3^{ver.-f-eq}) & \qquad f_8^f &= f_8^c - \delta_f(f_8^c - f_8^{c-eq}) \end{aligned}$$

The previous case can be easily generalized for the rest of streaming cases. Unlike other refinement approaches, [6, 2, 14], the LBM-steps are computed on every fluid node only once per time step. It is not necessary to have additional overlapping regions between the different refined levels and complex communication (interpolation) processes from/to each of the levels. Also, our new approach allows to refine only in one direction, which can be very appropriate and accurate-enough for some scenarios(Figure 2-left), demanding a lower computational cost. However, when dealing with refined domains over more than one direction, additional lattice units are necessary (Figure 2-medium and right).

# 4 Numerical Validation

To validate our NSCG approach for the LBM we carry out two different test scenarios: the first being a classical channel flow, while the second one is a simulation of the flow past a square cylinder. In both scenarios, three different LBM implementations are compared: *No refinement*, where no refined fluid domain is considered; *multi-domain* approach (MDA) based on the work of D. Lagrava et al. [6]; and our *NSCG* approach.

In the first scenario, the refined region is located in the bottom-half of the fluid domain. The *multi-domain* consists of a one-step coarse and a two-step fine algorithm over the horizontal and vertical directions [6]. In contrast, the *NSCG* considers only a one-step fine refinement over the horizontal direction. Figure 2-left graphically illustrates the different grid geometries with respect to *MDA* and *NSCG* approaches, for the first test scenario. To clarify Table 1 contains the different settings for every approach in terms of the number of lattice units. Several Reynolds numbers (50, 100, 150 and 200) have been tested for the same configuration. In LBM the Reynolds number equation ($Re = \frac{UL}{\upsilon}$) is related to the relaxation time $\tau$ (Equation 6) as: $\tau = [(3 * UL)/Re] + 0.5$ [15].

To visualize the numerical accuracy of every approach, we illustrate, in Figure 4, the horizontal velocity, for each Reynolds number, on the plane $x = 100$ over the fine-grid for time step 500.

On the other hand, in the second scenario are considered two sub-scenarios, $A$ and $B$ (Figure 2-medium and right). The finer domain is located around and behind the square cylinder. For the *NSCG*, two additional refined regions are necessary. One located in the top-bottom around the main refined region, and one located in the left ($B$ sub-scenario) and left-right ($A$ sub-scenario) of the cylinder. For the cylinder diameter D, the grid flow is set as $21D \times 14D$. When the Reynolds number is lower than 100, there is no vortex structure formed during the

| | | | |
|---|---|---|---|
| Channel Flow | | | |
| **Approach** | No Refine. | MDA | NSCG |
| **Coarse Grid** | 100 × 200 | 50 × 200 | 50 × 200 |
| **Fine Grid** | | 100 × 400 | 100 × 200 |
| **Fluid Units** | 20000 | 50000 | 30000 |
| Flow Past Square Cylinder (A Sub-scenario) | | | |
| **Approach** | No Refine. | MDA | NSCG |
| **Coarse Grid** | 210 × 140 | 210 × 140 | 190 × 120 |
| | | −20 × 20 (fine grid) | |
| **Fine Grid** | | 40 × 40 | 40 × 140 (top-bottom) |
| | | | 210 × 40 (left-right) |
| **Fluid Units** | 29400 | 30600 | 36800 |
| Flow Past Square Cylinder (B Sub-scenario) | | | |
| **Approach** | No Refine. | MDA | NSCG |
| **Coarse Grid** | 210 × 140 | 210 × 140 | 20 × 120 |
| | | −190 × 20 (fine grid) | |
| **Fine Grid** | | 380 × 40 | 380 × 120 (top-bottom) |
| | | | 20 × 40 (left) |
| | | | 240 × 40 (around-behind) |
| **Fluid Units** | 29400 | 40800 | 61600 |

Table 1: Test scenarios setting.

| **Reference** | $C_D$ | **Reference** | $C_D$ | **Reference** | $C_D$ |
|---|---|---|---|---|---|
| Numerical Data: | | Experimental Data: | | This work: | |
| Verstappen and Veldman [17] | 2.09 | Lyn et al. [8] | 2.1 | No Refinement | 2.48 |
| Porquie et al. [12] | 2.2 (UK1) | | | MDA | 2.02 |
| | 2.3 (UK2) | | | NSCG | 2.01 |
| | 2.23 (UK3) | | | | |
| Wang and Vanka [18] | 2.03 | | | | |
| Kawashima and Kawamura [4] | 2.72 (ST2) | | | | |
| | 2.73 (ST5) | | | | |

Table 2: Comparison between the numerical results yield by this work and other previous studies.

evolution, i.e., the flow field is laminar and steady. In contrast, for a Reynolds number of 100, the symmetric rectangular zones disappear and an asymmetric pattern is formed. The vorticity is shed behind the circular cylinder, and vortex structures are formed downstream. This phenomenon is well-capture by our *NSCG* approach. It is graphically illustrated in Figure 5.

One important dimensionless number is studied, the drag ($C_D = \frac{F_D}{0.5\rho U^2 D}$) coefficient. $F_D$ corresponds to the resistance force of the square cylinder to the fluid in the streamwise direction, $\rho$ is the density of the fluid, and $U$ is the velocity of inflow. In order to verify the numerical results, the coefficients were calculated and compared with the results of previous studies (Table 2).

# 5 Performance Study

Before visualizing the performance results (Figure 6), it is important to note that *multi domain* approach requires a higher number of points compared to *NSCG* approach for the first scenario (Table 1). However, in the second scenario the *NSCG* approach requires of a higher number of

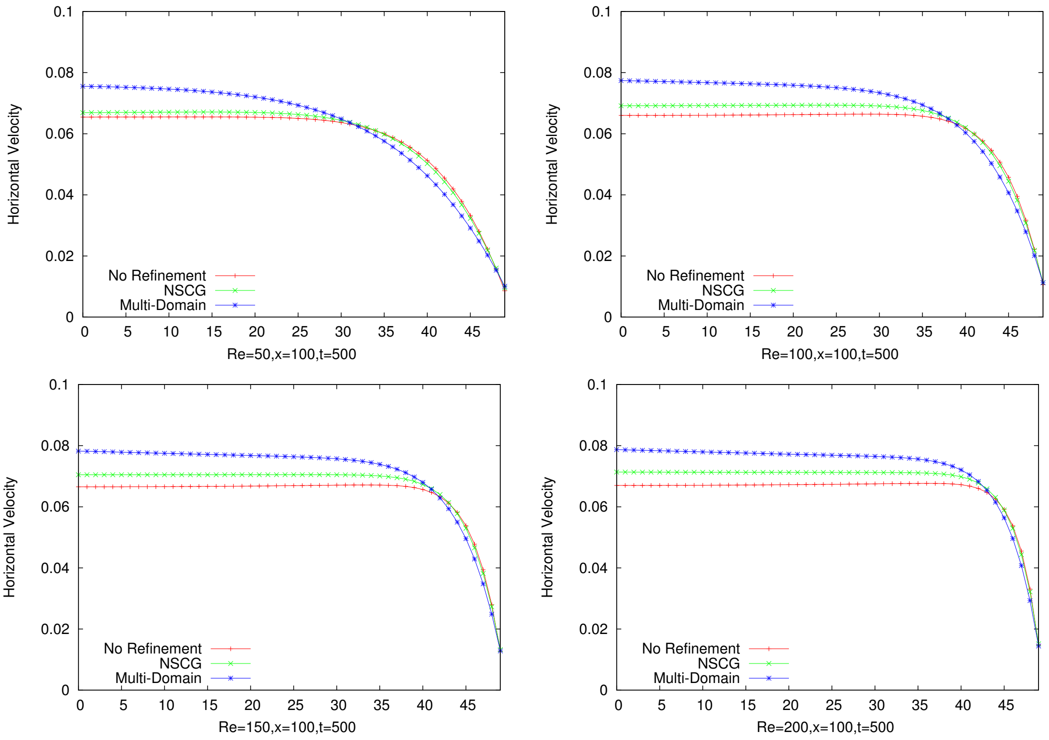

Figure 4: Numerical results for the first test scenario for characteristic velocity of 0.1.

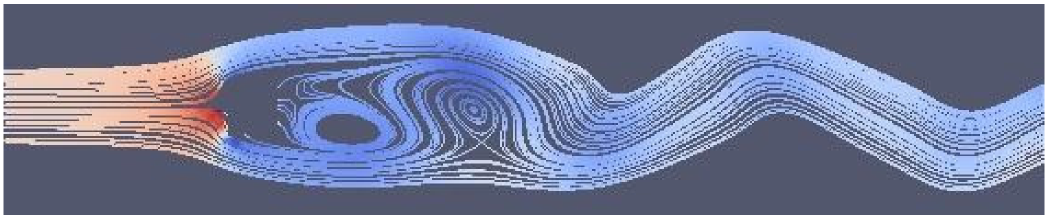

Figure 5: Streamlines for flow past a square cylinder at Re = 100.

fluid points.

To evaluate the efficiency of our approach, we perform a set of tests which consists of the same scenarios studied for the numerical validation. The computational platform consists of a single computational node of the supercomputing Lindgren Cray XE6 system based on AMD Opteron 12-core Magny-Cours (2.1 GHz) processors at KTH. In particular, we have carried out all our performance studies over a node composed of 24 cores divided between 2 sockets, with 12 cores each ( L1 cache: 64KB data cache, L2 cache: 512KB per core, L3 cache: 12MB per processor) and 32 GB DDR3 per node using the *cc* compiler and *-fopenmp -O3* as optimization flags.

In this performance analysis (Figure 6), we obtain execution time (ms) and MFLUPS (Millions of Fluid Units Per Second) ratio ($MFLUPS = (\#Fluid\ Units/time(s))/10^6$), considering 1, 12 and 24 OpenMP-threads. Table 3 illustrates the benefit (speedup) of our ap-

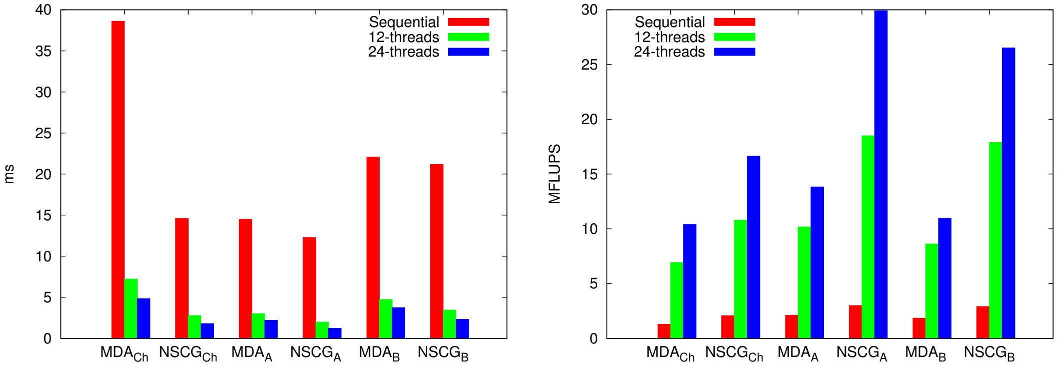

Figure 6: Time (left) and MFLUPS (right) for both refinement approaches, MDA and NSCG, over the three test scenarios, Channel flow ($MDA_{Ch}$, $NSCG_{Ch}$), flow past cylinder, A ($MDA_A$, $NSCG_A$) and B ($MDA_B$, $NSCG_B$) sub-scenarios.

proach compared to *MDA* in terms of execution time ($time_{MDA}/time_{NSCG}$) and MFLUPS ($MFLUPS_{NSCG}/MFLUPS_{MDA}$).

| | **Time** | | | **MFLUPS** | | |
|---|---|---|---|---|---|---|
| **Test Scenario** | Sequential | 12 Threads | 24 Threads | Sequential | 12 Threads | 24 Threads |
| Channel flow | 2.65 | 2.60 | 2.67 | 1.59 | 1.56 | 1.60 |
| A | 1.18 | 1.51 | 1.8 | 1.42 | 1.82 | 2.16 |
| B | 1.04 | 1.37 | 1.59 | 1.58 | 2.07 | 2.41 |

Table 3: Speedup reached by the NSCG approach against MDA approach.

# 6 Conclusions

In this work we have proposed, developed and analyzed a new approach for dealing with mesh refinement over LBM based on *Non-uniform Staggered Cartesian Grid* (NSCG). This approach exhibits a similar numerical accuracy compared to LBM refinement method based on multi-domain, Also it requires a lower computational and programming effort. The *NSCG* approach has been validated and analyzed for three different test scenarios. Although the gain of using our approach is reflected in every test, the benefit is greater when dealing with a higher number of threads, thus, it is a profitable scheme for parallel processors.

As a future research topic we plan to investigate more complex physical scenarios which require a higher amount of memory, making the use of memory distributed platforms mandatory. In addition, we plan to analyze other parallel platforms such as the nVidia GPU and the Intel Xeon Phi, as well as to implement more elaborate strategies for memory management and data distribution.

# References

[1] Lilit Axner, Alfons G. Hoekstra, Adam Jeays, Pat Lawford, Rod Hose, and Peter MA Sloot. Simulations of time harmonic blood flow in the mesenteric artery: comparing finite element and

lattice boltzmann methods. *BioMedical Engineering OnLine*, 2000.

[2] Abbas Fakhari and Taehun Lee. Finite-difference lattice boltzmann method with a block-structured adaptive-mesh-refinement technique. *Phys. Rev. E*, 89:033310, Mar 2014.

[3] Xiaoyi He and Li-Shi Luo. A priori derivation of the lattice boltzmann equation. *Phys. Rev. E*, 55:R6333–R6336, Jun 1997.

[4] Norifusa Kawashima and Hiroshi Kawamura. Numerical analysis of les of flow past a long square cylinder. In Jean-Pierre Chollet, PeterR. Voke, and Leonhard Kleiser, editors, *Direct and Large-Eddy Simulation II*, volume 5 of *ERCOFTAC Series*, pages 413–422. Springer Netherlands, 1997.

[5] S. Kollmannsberger, S. Geller, A. Dster, J. Tlke, C. Sorger, M. Krafczyk, and E. Rank. Fixed-grid fluidstructure interaction in two dimensions based on a partitioned lattice boltzmann and p-fem approach. *International Journal for Numerical Methods in Engineering*, 79(7):817–845, 2009.

[6] D. Lagrava, O. Malaspinas, J. Latt, and B. Chopard. Advances in multi-domain lattice boltzmann grid refinement. *Journal of Computational Physics*, 231(14):4808 – 4822, 2012.

[7] LBM-HPC. http://www.bcamath.org/en/research/lines/cfdct/software.

[8] D. A. Lyn and W. Rodi. The flapping shear layer formed by flow separation from the forward corner of a square cylinder. *Journal of Fluid Mechanics*, 267:353–376, 5 1994.

[9] Christian Obrecht, Frdric Kuznik, Bernard Tourancheau, and Jean-Jacques Roux. Scalable lattice boltzmann solvers for cuda gpu clusters. *Parallel Computing*, 39(67):259 – 270, 2013.

[10] E. Gross P. Bhatnagar and M. Krook. A model for collision processes in gases. i: small amplitude processes in charged and neutral one-component system. *Phys. Rev. E*, 94:511–525, 1954.

[11] T. Pohl, M. Kowarchik, J. Wilke, and U. Rüde K. Iglberger. Optimization and profiling of the cache performance of parallel lattice boltzmann codes. *Parallel Processing Letters*, 13(4):549560, 2003.

[12] M. Pourquie, M. Breuer, and W. Rodi. Computed test case: Square cylinder. In Jean-Pierre Chollet, PeterR. Voke, and Leonhard Kleiser, editors, *Direct and Large-Eddy Simulation II*, volume 5 of *ERCOFTAC Series*, pages 375–379. Springer Netherlands, 1997.

[13] Y. H. Qian, D. D'Humires, and P. Lallemand. Lattice bgk models for navier-stokes equation. *EPL (Europhysics Letters)*, 17(6):479, 1992.

[14] M. Schnherr, K. Kucher, M. Geier, M. Stiebler, S. Freudiger, and M. Krafczyk. Multi-thread implementations of the lattice boltzmann method on non-uniform grids for {CPUs} and {GPUs}. *Computers & Mathematics with Applications*, 61(12):3730 – 3743, 2011. Mesoscopic Methods for Engineering and Science - Proceedings of ICMMES-09 Mesoscopic Methods for Engineering and Science.

[15] Sauro Succi. *The Lattice Boltzmann Equation for Fluid Dynamics and Beyond (Numerical Mathematics and Scientific Computation)*. Numerical mathematics and scientific computation. Oxford University Press, USA, August 2001.

[16] Pedro Valero-Lara, Alfredo Pinelli, and Manuel Prieto-Matias. Accelerating solid-fluid interaction using lattice-boltzmann and immersed boundary coupled simulations on heterogeneous platforms. *Procedia Computer Science*, 29(0):50 – 61, 2014. 2014 International Conference on Computational Science.

[17] R.W.C.P. Verstappen and A.E.P. Veldman. Fourth-order dns of flow past a square cylinder: First results. In Jean-Pierre Chollet, PeterR. Voke, and Leonhard Kleiser, editors, *Direct and Large-Eddy Simulation II*, volume 5 of *ERCOFTAC Series*, pages 381–384. Springer Netherlands, 1997.

[18] G. Wang and S.P. Vanka. Les of flow over a square cylinder. In Jean-Pierre Chollet, PeterR. Voke, and Leonhard Kleiser, editors, *Direct and Large-Eddy Simulation II*, volume 5 of *ERCOFTAC Series*, pages 397–400. Springer Netherlands, 1997.

[19] Next Generation of CFD XFlow. http://www.xflowcfd.com/.

[20] Dong-Hyeog Yoon, Kyung-Soo Yang, and Choon-Bum Choi. Flow past a square cylinder with an angle of incidence. *Physics of Fluids (1994-present)*, 22(4):–, 2010.