Numerical Stabilisation of the Lattice Boltzmann method for higher Reynolds number fluid

N. Maquignon¹, J. Duchateau¹, G. Roussel¹, F. Rousselle¹, Christophe Renaud¹

¹LISIC - ULCO, 50 rue Ferdinand Buisson, 62228 Calais Cedex, France
INNOCOLD, 145 avenue Maurice Schumann - MREI 1, 59140 Dunkerque, France

Abstract—In this article, schemes of numerical stabilisation for a thermal multiphase Lattice Boltzmann Method are presented. As the model used is fully described in [6], it has been found unworthy to re-introduce it in this work. In this paper, we give techniques to widen its compatibility range. Application to real thermal flow using LBM is usually limited to low value for Reynolds and Péclet numbers if the simulation is to be completed with reasonable computational resources. Furthermore, the Prandtl number has to be kept close to unity. Similar problems occur when components have different viscosities. A new method that uses a per-component grid with two separate lattices for dynamics and thermal equations is introduced. Different grid resolutions should be used when components have important differences in viscosity, as well as when a component has higher Prandtl number. In order to keep reasonable need in computational resources, dynamic mesh refinement is used, and validated in the case of the increase of Reynolds number for isothermal case. The effect of the improved lattice on numerical stability and accuracy of the results are discussed. The decrease in need for computational resources is shown. A comparison between multiple relaxation scheme and single relaxation is shown, in terms of atteignable Reynolds number.

Keywords: Lattice Boltzmann method, mesh refinement, phase transition, multi-component, heat transfer.

1. Introduction

Understanding the behavior of flows containing multiple components in different fluid states has become essential for improvement of many industrial processes and environmental modeling. The lattice Boltzmann method (LBM) is a great candidate for the simulation of such phenomenons, because its mesoscopic nature has the ability to incorporate multiphysics. The usual LBM for multiphase flow are often made consistent with thermodynamics and mechanical properties as surface tension are adjusted sometimes very close to theoretical values [8] [4] [5]. But applications to real cases is complicated because the range of compatibility for physical properties is often too restrained. Furthermore, phase change is rarely evaluated for thermal cases, and heat transfer isn’t coupled to mass transfer. In this work, we introduce a new lattice Boltzmann method that is able to simulate a thermal flow with multiple components, phase transition and heat transfer. We introduce static and dynamic mesh refinement to allow the increase of Reynolds, Péclet, and Prandtl numbers as well as viscosity ratio. These criteria are important for medium or large scale real simulations and determine the computational resources needed. Coarse grids usually cannot simulate high Reynolds or Péclet numbers flows with reasonable spatial resolution without the use of LES or RANS filtering methods, as the Smagorinsky subgrid model [7]. Such filtering doesn’t exist yet for multiphase flows. The subgrid can be used for the BGK part of the Boltzmann equation but instabilities coming from the forcing term aren’t easily decreased. The present work shows that the use of multiphase model implies a cost in numerical stability that decreases compatibility. A solution would be to increase resolution but it indeed has a severe impact on computational time. In this work, an adaptive mesh allows to increase numerical stability while keeping reasonable computational needs.

As previously mentioned, compatibility range often restrains applications to a few cases, because of numerical instabilities. The different models should be applied on meshes of finer resolution. To the best of our knowledge, only the work of Fan & Yu [10] describes an adaptive mesh refinement for monocomponent isothermal case with two phases using traditional SC model. This model still needs an extension to real thermodynamics with temperature transport equation. Furthermore, it does not take into account the presence of multiple components.

2. Thermal multi-component lattice Boltzmann method with heat transfer and phase change

As described in [6], the new coupled model is of the following form :

$$f_{\sigma,i}(x + e_i \delta t, t + \delta t) - f_{\sigma,i}(x,t) = -\frac{f_{\sigma,i}(x,t) - f_{\sigma,i}^eq(x,t)}{\tau_{\sigma,f}} + \Delta f_{\sigma,i}$$

(1)
\[ g_{\sigma,i}(x + e_i \delta t, t + \delta t) - g_{\sigma,i}(x, t) = - \frac{g_{\sigma,i}(x, t) - g_{\sigma,i}^q(x, t)}{\tau_{\sigma,g}} + \omega_i \Phi_{\sigma} \delta t \]  

A set of equations for every component \( \sigma \) is necessary. The density of component \( \sigma \) is:

\[ \sum_i f_{\sigma,i}(x, t) = \rho_{\sigma}(x, t) \quad (3) \]

Speed of component \( \sigma \) is:

\[ \sum_i f_{\sigma,i}(x, t)e_i = \rho_{\sigma}(x, t)u_{\sigma}(x, t) \quad (4) \]

The temperature of component \( \sigma \) is:

\[ \sum_i g_{\sigma,i}(x, t) = T_{\sigma}(x, t) \quad (5) \]

A composite temperature is necessary to model the whole fluid temperature:

\[ T(x, t) = \frac{\sum_{\sigma} \frac{\rho_{\sigma}C_{P\sigma}T_{\sigma}}{\tau_{\sigma,g}}}{\sum_{\sigma} \frac{\rho_{\sigma}L_{\sigma}P_{\sigma}}{\tau_{\sigma,g}}} \quad (6) \]

The associated algorithm and details are fully described in [6].

2.1 3D visualisation

We present here a result of a simulation that allows the separation of a fluid consisting of an assembly of two components. The simulation domain consists of a lightweight component and a heavy component. The domain initialization is done as in the algorithm presented above. The purpose of the simulation is to highlight the efficiency of the algorithm for processing multi-component simulations. In our case, we present a simulation with two components but the algorithm remains valid for a larger number of components.

We can see on figure 1 the initialization corresponding to the mixture with two components. The heavy component (blue) and the light component (yellow) will gradually separate throughout the simulation. Figure 2 allows us to observe this phenomenon of separation. We can see the simulation images after 1000, 2000, 3000 and 3500 iterations. It is clear that the separation is more obvious with a significant number of iterations.

The visualization of the simulation was performed using the VTK (Visualization Toolkit) library. VTK is an open-source, freely available software system for 3D computer graphics, image processing and visualization. VTK supports a wide variety of visualization algorithms including: scalar, vector, tensor, texture, and volumetric methods. In our case, we opted for a display of the scalar field density using a texture mapping method to visualize the global 3D domain.

3. Mesh refinement

3.1 Static mesh refinement

3.1.1 Increase of atteignable Reynolds or Péclet numbers using static mesh refinement

The model described in the previous sections needs improvement of numerical stability to be applied to real cases. Real flows can have high Reynolds or Péclet numbers, which determine the need of mesh refinement. Equation (8) and (7) show that for a given value of characteristic speed \( U \) and length \( L \), higher Reynolds or Péclet numbers require finer mesh size or smaller relaxation parameter. Assuming \( \delta x = \delta t \):

\[ \frac{UL}{Re} = \nu = \frac{\delta x^2}{3\delta t} (\tau_f - 1/2) = \frac{\delta x}{3} (\tau_f - 1/2) \quad (7) \]
\[ \frac{UL}{Pe} = \alpha = \frac{\delta x^2}{3\delta t}(\tau_g - 1/2) = \frac{\delta x}{3}(\tau_g - 1/2) \] (8)

The relaxation parameters determine the stability of the numerical scheme. The model diverges more easily with small values of \( \tau_g \) and \( \tau_f \). They cannot be chosen smaller than 1/2 and solution becomes unstable when they get close to the limit value. A possibility to reach smaller \( \nu \) or \( \alpha \) with stable numerical scheme is to decrease spatial and time steps \( \delta x \) and \( \delta t \). It indeed increases the need for computational resources. Refinement of factor 2 needs four times more memory and is height times slower. A solution for memory and time saving is to refine mesh size locally where turbulence is expected. This is what next subsection is about.

3.1.2 Partial mesh refinement

In order to validate mesh refinement for density and increase of atteignable Reynolds number, a domain with two different grid sizes, as shown in figure 3, is defined and Poiseuille flow profile is recovered on each of the areas, as shown in figure 4, 5 and 6. A simulation of Karman instability is performed in figure 7, refined by a factor 2 around and after the cylinder where instabilities are supposed to appear. This configuration allows to increase Reynolds number by 35 %, in comparison with coarse grid simulation. This value has been established experimentally through the simulation of figure 7. In refined area, computational time is increased by a factor 8.

![Fig. 3: Initialisation of a domain with two different resolutions.](image)

![Fig. 4: Magnitude of velocity profile for Poiseuille flow simulation at Re = 200.](image)

A possibility for computation time and memory saving would be to dynamically refine the grid, only at times and positions at which instability appears. The refinement would occur when one or more criteria would go above a threshold value. This will be introduced in the section about dynamic mesh refinement using a refined patch.

3.1.3 Increase of atteignable Prandtl numbers using static mesh refinement

For a single component, the Prandtl number is the ratio of viscosity over thermal diffusivity. It also can been seen as the ratio of Péclet over Reynolds numbers, \( Pr = \nu/\alpha = Pe/Re \). Usually, the use of same mesh size for density and temperature distribution functions restrains applications to species that have a Prandtl number close to unity. In order to allow simulations of \( Pr << 1 \) or \( Pr >> 1 \), different mesh sizes should be used for mass and heat transfer processes. Relaxation parameters have to be determined so that reference values for \( \alpha_{ref} \) and \( \nu_{ref} \) are conserved. The next formulas give relationship between lattice discrete properties :

\[ \nu_{ref} = c_s^2(\tau_{fn} - 1/2)\delta t_f \] (9)
\[ \alpha_{ref} = c_s^2 (\tau_{gn} - 1/2) \delta_t g \] (10)

\[ \delta t = \delta x \] is better for stability, and the lattice speed of sound is kept as \( c_s = 1/\sqrt{3} \). A refinement factor can be defined as \( n_{fg} = \delta t_f / \delta t_g = \delta x_f / \delta x_g \). If \( Pr >> 1 \), \( n_{fg} >> 1 \) and reciprocally \( Pr << 1 \) implies \( n_{fg} << 1 \). This new lattice layout means new rules have to be adopted for communication between density and temperature. If refinement factor is not unity, interpolations of some moments are necessary. Temperature has to be either averaged or interpolated when pressure is needed. Interpolation is performed with Taylor expansion. Speed is to be distributed equally or interpolated among cells of different resolution.

### 3.1.4 Increase of atteignable viscosity ratio using static mesh refinement

When two components have different viscosities, they cannot easily be simulated on the same grid resolution. The problem is the same with thermal diffusivity. Taking previous section into account, a per component mesh resolution for temperature and density should be determined. At communication steps between two components, the used macroscopic values have to be averaged or interpolated to a different resolution. Two components communicate at heat exchange step and by the external interaction force term. Those are the two macroscopic values that have to be converted to different resolutions.

### 3.2 Dynamic Mesh refinement (DMR)

#### 3.2.1 DMR using a refinement patch

DMR is used to refine mesh locally when instability starts to develop. The refinement avoids instability to spread and is activated only on areas that require stabilisation. This step is performed with the use of a patch that translates distribution functions from coarse grid to a patch of finer resolution. The copy can be done with interpolation or with values equally distributed. After collision and advection are computed within the refined patch, values can be averaged back to coarse grid or kept on finer mesh. The rules for relaxation parameters are the same as for static mesh refinement.

In figure 8, Karman instability is simulated at \( Re = 195 \). Patch allows to increase maximum atteignable Reynolds number up to 31%. As at every time step values are copied from the patch to the coarse mesh, the only cost in memory is the size of the patch, which can be up to 24 times less than the cost of partial static refinement. The value of factor 24 has been established through the simulation of figure 8. The patch shouldn’t be too small to keep stability. The computational time is also decreased by a factor depending on the number of patches used. The more the patches are, the slower the simulation is.

#### 3.2.2 Criteria for activation of patch

The chosen criterium for activation of patch in figure 8 is the magnitude of non equilibrium viscous stress tensor, which is used in the Smagorinsky model to increase relaxation parameter. As it can be found in reference [7], using the Smagorinsky subgrid model consists in computing a correction step on relaxation parameter:

\[ \tau^* = \tau + \tau_t \] (11)

\[ \tau_t = \frac{1}{2} \left( -\tau + \sqrt{\tau^2 + \frac{2C_{Smagno} \delta^2}{\rho_0 c_s^2 \delta^2} \left\| \Pi_{\alpha\beta}^{neq} \right\|} \right) \] (12)

\[ \left\| \Pi_{\alpha\beta}^{neq} \right\| = \sum_{i=0}^{8} \varepsilon_{i\alpha} \varepsilon_{i\beta} (f_i - f_i^{eq}) \] (13)

Parameter \( \tau \) is corrected to \( \tau^* \) by adding a vortex relaxation parameter \( \tau_f \). \( \left\| \Pi_{\alpha\beta}^{neq} \right\| \) is the magnitude of non equilibrium viscous stress tensor. \( \alpha \) and \( \beta \) are 2D space directions. \( \rho_0 \) is a density reference value. \( C_{Smagno} \) is the Smagorinsky parameter, which has to be empirically determined. The Smagorinsky subgrid model has the advantage that it allows to simulate higher Reynolds number flows without increasing spatial resolution. The maximum attainable Reynolds number depends on the magnitude of \( C_{Smagno} \). Increasing this parameter means increasing maximum attainable Reynolds number. But the physical signification of \( C_{Smagno} \) can remain quite unclear and some information loss can be suspected. Dynamic mesh refinement aims to increase Reynolds number without drastically increasing need in computational ressources, and without calculating correction.
steps from equation (11) and (12), using a quite unclear $C_{Smago}$ parameter.

DMR rather uses the non equilibrium viscous stress tensor $\|\Pi_{\alpha\beta}\|$ as a criteria to activate refinement patch locally, as shown in figure 8. The magnitude is averaged on a sub-domain. If the obtained value is above a threshold value, a patch is activated on the sub-domain. For a thermal flow without source term the instability is of the same kind, because the numerical scheme is basically the same as LBGK model. Thus, a tensor of the same form of $\|\Pi_{\alpha\beta}\|$ can be calculated and used to determine patched zones. Let us call this tensor $V_{\alpha\beta}^{neq}$:

$$\|V_{\alpha\beta}^{neq}\| = \sum_{i=0}^{8} e_{i\alpha} e_{i\beta} (g_i - g_i^{eq})$$ (14)

For the case of a phase transition flow, the only criterium of viscous tensor norm is insufficient. The numerical instability does not come from the same phenomenon. The density ratio is usually the criterium responsible for divergence [9]. This means it has to be taken into account for refinement of multiphase flow. Thus, the criterium to be chosen is density gradient, which is significant at the interfaces between gas and liquid domains.

The calculation of this gradient can prevent high density ratios within the entire domain and thus easily refine the unstable places in order to gain stability. We present here a result of a simulation for evaluating the density gradient as a criterion for refinement. On images of figure 9, we take a benchmark of condensation for monocomponent fluid, on which we have added our criterion. An uncondensed domain is initiated and separation between liquid and gas phases occurs in isothermal condition. We can view the red locations where it is necessary to refine:

Fig. 9: Patched areas (red) where density gradient is significant.

We observe that only the areas where the density ratio is important are concerned. This criterion seems to be a good compromise to obtain a gain in stability for simulations of phase transition with large density ratios. It is important to remark that the purpose here is to show that the criterion of refinement with the density gradient is a relevant criterion, the threshold is currently set manually. It will be interesting to observe whether it is possible to automatically manage the threshold calculation according to the needs of the simulation.

3.2.3 Algorithm for DMR with patch

The whole coarse domain can be subdivided into subdomains which can be of the size of the patch or larger. In figure 10, the sub-domains drawn in yellow are limited to regions of interest, after the cylinder, where instabilities are expected. Knowledge of the flow allows to decrease the number of regions of interest.

![Fig. 10: Magnitude of velocity profile at different time steps for Karmann instability simulation with Re = 195, using DMR. The number of patched zone (red) increases with time.](image)

The viscous tensor norm is calculated and averaged on a first interest zone from the twelve subdomains delimited in yellow. If the value is above a threshold, a patch is activated. At activation, a copy with or without interpolation from coarse grid to fine patch is computed. Then collison and advection steps are performed two times, if temporal resolution is refined with a factor 2. Notice that a factor two refined patch requires 4 times more elements in comparison to its corresponding coarse area. After those calculations, values are averaged and copied back to coarse grid. Then, the same work is performed on each of the remaining subdomains of interest.

Figure 10 shows increase of the number of patched zones, that gradually have viscous tensor norm above a defined
threshold value, at different time steps.

**4. Multiple relaxation parameters parametrisation.**

Another way to increase the attainable Reynolds number is to use a multiple relaxation times (MRT) instead of a single relaxation time (SRT). This MRT scheme supposes a change from distribution functions space to the momentum space. The calculation is done with a transformation matrix and a collision matrix.

As previously reminded, the usual SRT LBM with force term has the following expression:

$$f_{\sigma,i}(x+e_i\delta t,t+\delta t) - f_{\sigma,i}(x,t) = -\frac{f_{\sigma,i}(x,t) - f_{eq,\sigma,i}(x,t)}{\tau_{\sigma,f}} + \Delta f_{\sigma,i}$$  \hspace{1cm} (15)

A transformation matrix $M$ allows the determination of corresponding per momentum equations:

$$Mf = m$$  \hspace{1cm} (16)

Where $f = (f_0, ..., f_8)$ is the distribution function vector and $m = (m_0, ..., m_8)$ is the corresponding momentum vector. The transformation matrix $M$ has the following expression:

$$M = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
-4 & 1 & -1 & -1 & -1 & 2 & 2 & 2 & 2 \\
4 & -2 & -2 & -2 & -2 & 1 & 1 & 1 & 1 \\
0 & 1 & 0 & -1 & 0 & 1 & -1 & -1 & 1 \\
0 & -2 & 0 & 2 & 0 & 1 & -1 & -1 & 1 \\
0 & 0 & 1 & 0 & -1 & 1 & 1 & -1 & -1 \\
0 & 0 & -2 & 0 & 2 & 1 & 1 & -1 & -1 \\
0 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \\
\end{bmatrix}$$  \hspace{1cm} (17)

The new equation written in momentum space and with multiple relaxation parameters is:

$$m_{\sigma,i}(x+e_i\delta t,t+\delta t) = m_{\sigma,i}(x,t) - s_i(m_{\sigma,i}(x,t) - m_{eq,\sigma,i}(x,t)) + \Delta m_{\sigma,i}$$  \hspace{1cm} (18)

This new formula takes $s_i$ as the relaxation parameter. It comes from the relaxation matrix which is given by:

$$S = \begin{bmatrix}
s_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & s_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & s_2 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & s_3 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & s_4 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & s_5 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & s_6 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & s_7 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & s_8 \\
\end{bmatrix}$$  \hspace{1cm} (19)

This formulation allows to increase the attainable Reynolds number, as shown in figures 11 and 12.

The usual limit for a non refined grid is $Re = 150$, but we came up to $Re = 1000$ with the MRT model. This shows the superiority of the MRT scheme over SRT.

![Fig. 11: SRT Karman instability at $Re = 150$.](image1)

![Fig. 12: MRT Karman instability at $Re = 1000$.](image2)

**5. Conclusions**

Static and dynamic mesh refinement have allowed to increase Reynolds number from up to 35 % and 31% for classic LBGK model. Criterion of viscous tensor norm shows to be efficient to activate patch and insure stability. Though stability is validated when the model does not diverge, and though we have computed Poiseuille flow, we believe further validation is necessary. Furthermore, the patch validity and tests should be extended to multiphase multicomponent flow. External and internal interaction force terms have to be established for multi resolution problems. Some work like reference [10] gives formula for multigrid resolution, but it still needs to be extended to multicomponent case and validated with the use of a patch. The criterium of density ratio for phase transition flow appears to be correct to determine the regions that should be patched.

The MRT [1] is another method of stabilisation which showed to be performant. The method allows to reach higher Reynolds numbers than with classic BGK collision term. GPU implementation that uses CUDA or OpenCL language allows parallel implementation of LBM [2] [3], and acceleration of computation time. Combination of the present work with MRT and GPU implementation can be a great improvement for the simulation of multicomponent flows in real cases.

**References**


**Acknowledgment**

This work has been made possible thanks to a collaboration between academic and industrial groups, gathered by the INNOCOLD association.