Applicability of regularized particles-on-demand method to solve Riemann problem

E Zipunova$^1$, A Perepelkina$^1$ and A Zakirov$^2$

$^1$ Keldysh Institute of Applied Mathematics, Russia
$^2$ Kintech Lab, Russia
E-mail: e.zipunova@gmail.com

Abstract. The lattice Boltzmann method (LBM) is a promising rapidly evolving simulation tool designed to capture the physics of the mesoscopic flow. But it is well known that the standard LBM is severely restricted to problems with low flow speed and small temperature range. The novel Particle-on-demand method (PonD) allows us to numerically solve the discrete Boltzmann equation for high Mach numbers. But the propagation step becomes more computationally intensive. A large part of the computational cost comes from matrix inversions during the rescaling of discrete distribution functions from one system of reference to another. We propose another method of discrete distribution function rescaling by applying regularization and moment conversion to avoid matrix inversions. This improvement shows drastic acceleration in calculations in comparison with standard PonD. Regularized PonD was applied to solve Riemann problems for a range of parameters. Results obtained by this improved method were compared to results received by the standard PonD and the standard LBM. The conservation of mass, momentum and energy of the new numerical method was studied. The necessary conditions for the stability of the solution were obtained.

1. Introduction
The lattice Boltzmann method (LBM) [1, 2] is a modern approach to the simulation of complex flows and an efficient numerical solver that suits modern, highly parallel computing architectures. However, it is limited to low Mach numbers and near-isothermal conditions.

To overcome the limitations of LBM one can use different reference frames [3, 4]. To define reference frames we use a gauge, that is, a pair of values $\lambda = \{u, T\}$, so that $v = \frac{\xi - u}{\sqrt{T}}$ is the speed in the reference frame $\lambda$, and $\lambda_0 = \{0, 1\}$ is the original reference frame. Thus one can calculate moments $m^{(n)}$ using Hermite quadrature in the $\lambda_0$ reference frame:

$$m^{(n)}(x, t) = \int f(x, \xi, t)\xi^n d\xi = \sum_{i=1}^{Q} \left( w_i f(x, c_i, t) c_i^n \right) = \sum_{i=1}^{Q} f_{i0}^c c_i^n,$$

and in the $\lambda$ reference frame :

$$m^{(n)}(x, t) = \int f(x, \sqrt{T}v + u, t)(\sqrt{T}v + u)^n d(\sqrt{T}v + u) =$$

$$= \sum_{i=1}^{Q} \left( \sqrt{T}^{c_i} w_i f(x, \sqrt{T}c_i + u, t) \frac{\omega(c_i)}{\omega(c_i)} \right) (\sqrt{T}c_i + u)^n = \sum_{i=1}^{Q} f_{i\lambda}^c (\sqrt{T}c_i + u)^n.$$
These two equations can be written in a matrix form:

\[ M^{\lambda'} f^{\lambda'} = M^\lambda f^\lambda, \quad M^\lambda_{ij} = (\sqrt{T}c_i + u)^j. \]

\( f^{\lambda'} \) in a new gauge \( \lambda' \) can be calculated using the inverse matrix:

\[ f^{\lambda'} = M^{\lambda'-1} M^\lambda f^\lambda \]

With this approach, in the Particles-on-Demand method (PonD) [4], the expansions near Maxwellians with different \( u \) and \( T \) are used in the same simulation setup.

The streaming in PonD is performed from off-lattice points and is supplemented with interpolation. However, in the implementation reported in [4], the computational complexity is increased due to the gauge transformation and interpolation steps. Due to the interpolation the mass conservation is lost as well.

Thus, the PonD method is a promising step towards the future of computational fluid dynamics modelling, but at the present stage it seems not feasible for relevant 3D simulations. In [4, 5] the explicit form of the transformation matrix \( G(\lambda, \lambda') = M^{\lambda'-1}(u', T')M^\lambda(u, T) \) is found for D1Q3, D2Q9, D3Q27 stencils. Stencils are defined by the expression \( Dd\Omega_q \), where \( d \) is dimension of the problem, \( q \) is size of quadrature. The symbolic form of \( G(\lambda, \lambda') \) is complex for many-point 3D quadratures, and its numerical computation in each mesh node is expensive. A cheaper way for gauge transformation has been found in the Regularized PonD method [6].

This method is similar to regularized LBM. In a regularized LBM, the higher moments are found for D1Q3, D2Q9, D3Q27 stencils. Stencils are defined by the expression \( Dd\Omega_q \), where \( d \) is dimension of the problem, \( q \) is size of quadrature. The symbolic form of \( G(\lambda, \lambda') \) is complex for many-point 3D quadratures, and its numerical computation in each mesh node is expensive. A cheaper way for gauge transformation has been found in the Regularized PonD method [6].

This method can be modified to be either a variation of the regularized LBM or equivalent to the original PonD.

2. Method Description

In a moving reference frame \( \lambda \) a distribution function can be approximated by its projection onto the space spanned by the first \( N \) Hermite polynomials:

\[ f(x, \xi, t) \approx f^N(x, \sqrt{T}v + u, t) = \omega(v) \sum_{n=0}^{\infty} \frac{1}{n!} d^{(n)}(x, t) \mathcal{H}^{(n)}(v), \quad d^{(n)}(x, t) = \int f(x, \xi, t) \mathcal{H}^{(n)}(v) d\xi \]

where coefficients \( d^{(n)}(x, t) \) are the Hermite moments. Let us take a quadrature rule with degree of precision \( E \geq 2N \), nodes \( c_i \), \( i = 1..Q \), and weights \( w_i \). Given that \( f^N/\omega(v) \) is a polynomial of degree no more than \( N \), the moments up to the order \( N \) can be calculated as finite sums:

\[
\begin{aligned}
  d^{(n)}(x, t) &= \int f^N(x, \xi, t) \mathcal{H}^{(n)}(v) d\xi = \int \sqrt{T}^D f^N(x, \sqrt{T}v + u, t) \mathcal{H}^{(n)}(v) dv = \\
  &= \int \omega(v) \frac{\sqrt{T}^D f^N(x, \sqrt{T}v + u, t)}{\omega(v)} \mathcal{H}^{(n)}(v) d\xi = \\
  &= \sum_{i=1}^{Q} \left( \sqrt{T}^D w_i \frac{f^N(x, \sqrt{T}c_i + u, t)}{\omega(c_i)} \right) \mathcal{H}^{(n)}(c_i) = \sum_{i=1}^{Q} f_{i}^{\lambda} \mathcal{H}^{(n)}(c_i).
\end{aligned}
\]

where \( f_{i}^{\lambda} \) is:

\[
f_{i}^{\lambda}(x, t) = \sqrt{T}^D w_i \frac{f^N(x, \sqrt{T}c_i + u, t)}{\omega(c_i)} = \sqrt{T}^D w_i \sum_{n=0}^{N} \frac{1}{n!} d^{(n)}(x, t) \mathcal{H}^{(n)}(c_i). \tag{1}
\]
Thus, $d^{(n)}(x, t)$ are expressed through $f^\lambda(x, t)$ and $f^\lambda_t(x, t)$ are expressed through $d^{(n)}(x, t)$. In the regularization methods of LBM, they are sequentially computed through each other.

If the $d^{(n)}(x, t)$ values are known then the discrete distribution functions in the moving gauge $f^\lambda$ can be found with (1). So the method to calculate moments $d^{(n)}(x, t)$ in a moving reference frame $\lambda$ from moments $m^{(n)}(x, t)$ is developed, and $m^{(n)}(x, t)$ are easily calculated in all reference frames.

**2.1. Recursive method calculating $d$**

To calculate $d^{(n)}(x, t)$ let's write the known expression for Hermite polynomials:

$$H^{(n+1)}_{\alpha_0...\alpha_n}(v) = v_{\alpha_0} H^{(n)}_{\alpha_1...\alpha_n}(v) - \sum_{i=0}^{n} \delta_{\alpha_0\alpha_i} H^{(n-1)}_{\alpha_1...\alpha_{i-1}\alpha_{i+1}...\alpha_n}(v).$$

Then integrate both sides of the equation by $\xi$:

$$\int H^{(n+1)}_{\alpha_0...\alpha_n}(v)d\xi = \int \left( \frac{\xi_{\alpha_0} - u_{\alpha_0}}{\sqrt{T}} \right) H^{(n)}_{\alpha_1...\alpha_n}(v)d\xi - \sum_{i=0}^{n} \delta_{\alpha_0\alpha_i} \int H^{(n-1)}_{\alpha_1...\alpha_{i-1}\alpha_{i+1}...\alpha_n}(v)d\xi.$$

Let:

$$H^{(n)}_{\alpha_1...\alpha_n}(v) = P(\xi_{\beta_1...\beta_n}, \xi_{\gamma_1...\gamma_n-1}, ..., \xi_{1}, \xi_{0}, u, T),$$

where $P$ affine function $\xi^{(n)}$, then:

$$\int H^{(n+1)}_{\alpha_0...\alpha_n}(v)d\xi = \int \frac{1}{\sqrt{T}} \left( P(\xi_{\alpha_0\beta_1...\beta_n}, \xi_{\alpha_0\gamma_1...\gamma_n-1}, ..., \xi_{0\alpha_\tau}, \xi_{0\alpha_0}, u, T) - u_{\alpha_0} P(\xi_{\beta_1...\beta_n}, \xi_{\gamma_1...\gamma_n-1}, ..., \xi_{1}, \xi_{0}, u, T) \right) d\xi - \sum_{i=0}^{n} \delta_{\alpha_0\alpha_i} \int H^{(n-1)}_{\alpha_1...\alpha_{i-1}\alpha_{i+1}...\alpha_n}(v)d\xi.$$

From equation (2):

$$d^{(n+1)}_{\alpha_1...\alpha_n} = P(m^{(n)}_{\beta_1...\beta_n}, m^{(n-1)}_{\gamma_1...\gamma_n-1}, ..., m^{(1)}_{1}, m^{(0)}_{0}, u, T).$$

So we obtain a recursive expression for moments $d^{(n+1)}$ through $d^{(n)}$ and $d^{(n-1)}$:

$$d^{(n+1)}_{\alpha_0...\alpha_n} = T^{-1/2} \left( P(m^{(n+1)}_{\alpha_0\beta_1...\beta_n}, m^{(n)}_{\alpha_0\gamma_1...\gamma_n-1}, ..., m^{(2)}_{\alpha_0\tau}, m^{(1)}_{\alpha_00}, u, T) - u_{\alpha_0} d^{(n)}_{\alpha_1...\alpha_n} \right) - \sum_{i=0}^{n} \delta_{\alpha_0\alpha_i} d^{(n)}_{\alpha_1...\alpha_{i-1}\alpha_{i+1}...\alpha_n}.$$

The first two moments are: $d^{(0)} = m^{(0)}$, $d^{(1)}_{\alpha} = T^{-1/2} \left( m^{(1)}_{\alpha} - u_{\alpha} m^{(0)} \right)$, and:

$$d^{(2)}_{\alpha\beta} = T^{-1/2} \left[ T^{-1/2} \left( m^{(2)}_{\alpha\beta} - u_{\alpha} m^{(1)}_{\beta} \right) - T^{-1/2} u_{\beta} \left( m^{(1)}_{\alpha} - u_{\alpha} m^{(0)} \right) \right] - m^{(0)} \delta_{\alpha\beta} =$$

$$= T^{-1} \left[ m^{(2)}_{\alpha\beta} - u_{\alpha} m^{(1)}_{\beta} - u_{\beta} m^{(1)}_{\alpha} + (u_{\alpha} u_{\beta} - T \delta_{\alpha\beta}) m^{(0)} \right].$$

(3)
2.2. Algorithm

So we have the following algorithm of calculations. All data is stored in moments \( m^{(n)} \) in the standard gauge \( \lambda_0 \). For each mesh point \( x \) at each time step \( t \) (Fig.1)

- streaming starts with a prediction of a new gauge as the gauge from the previous time step 
  \[ \lambda'(x, t) = \lambda(x, t - \Delta t) = \{u, T\} \]. In this gauge,
    - for each \( c_i \),
      * calculate the point \( x' = x - (\sqrt{T}c_i + u) \) from which we stream the discrete distribution function;
      * in \( x' \), get moments \( m^{(n)} \) with Lagrange interpolation;
      * convert \( m^{(n)} \) to the moments in the moving frame \( d^{(n)} \) with (3);  
      * calculate the discrete distribution function \( f_i^{\lambda'} \) from \( d^{(n)} \) using (1) and stream it to the point \( x \).
    - calculate the new values of moments \( m^{(n)} \) as \( \sum f_i^{\lambda'} H^{(n)}(\sqrt{T}c_i + u) \), and get the values of speed and temperature \( \{u, T\} \)
    - correct the predicted gauge \( \lambda' \).

- Iterations proceed until we find the right values of speed and temperature \( \{u, T\} \). Once iterations stop, the collision is performed and the new values of moments \( a^{(n)} \) in the standard frame are computed.

![Figure 1. RegPond algorithm.](image1)

![Figure 2. Results obtained by RegPond with unfixed size L template for moving contact discontinuities with speed u = 10, t = 10.](image2)

We call this method the Regularized Particles-on-Demand method (RegPonD). In contrast with [4], \( a^{(n)} \) values are stored instead of \( f^{\lambda_i} \), and the highlighted steps in the algorithm above differ.

3. Results

To compare PonD, RegPonD and LBM we studied contact discontinuities moving with different speeds (Fig. 3, 4).

\[
\tau = 0.7, \quad P = T \rho, \quad P|_{x<0} = P|_{x\geq0} = 1, \quad u|_{x<0} = u|_{x\geq0} = u, \quad \rho|_{x<0} = 1.1, \quad \rho|_{x\geq0} = 1,
\]

For this case, interpolation in PonD and RegPonD is with unfixed template with size \( L=5 \).
As can be seen, results obtained by PonD, RegPond and LBM match for $u = 0.1$ (Fig. 3). For $u = 0.2$ (Fig. 4), LBM is unstable when results obtained by PonD and RegPond still look adequate.

For example, in Fig. 2, results obtained by RegPond for moving contact discontinuities with speed $u = 10$ for various size $L$ are presented. Interpolation was made with an unfixed template, because interpolation with a fixed template is not stable for $P = 1$ even for small $u$.

To research the applicability of RegPond with different interpolation templates we studied a shock wave:

$$\tau = 0.51, \quad \rho|_{x<0} = \rho|_{x>0} = 1, \quad u|_{x<0} = u|_{x>0} = 0, \quad P|_{x<0} = 0.1, \quad t = 400,$$

with various values $P|_{x>0}$. As can be seen in Fig. 5 and 6, for smaller $P|_{x>0}$ values unfixed interpolation templates are more accurate and for higher the fixed template shows better results.
This is due to the fact that fixed interpolation templates conserve mass and momentum [5], and this mostly leads to a more accurate result for shock-wave solutions. But for higher values of speed and temperature, the point from which we stream the $f_i$ values falls outside of the template. Thus, interpolation loses stability.

4. Conclusion

We developed the RegPond method as an alternative method for gauge transformation in PonD. Pond and RegPond are the same scheme for $M = Q$. We made a code for testing all PonD schemes. RegPond is validated to be rigorous and efficient. Both Pond and RegPond are applicable for higher Mach numbers than LBM. With the fixed interpolation template, mass conservation is satisfied. For the Riemann problem, an unfixed interpolation template is applicable for a wider range of parameters.

Acknowledgements

The work is supported by Russian Science Foundation, grant # 18-71-10004.

References

[1] Succi S 2001 *The lattice Boltzmann equation: for fluid dynamics and beyond* (Oxford university press)
[2] Shan X, Yuan X and Chen H 2006 Kinetic theory representation of hydrodynamics: a way beyond the Navier–Stokes equation *J. Fluid Mech.* **550** 413–41
[3] Sun C and Hsu A T 2003 Three-dimensional lattice Boltzmann model for compressible flows *Phys. Rev. E* American Physical Society **68** 016303.
[4] Dorschner B, Bösch F and Karlin IV 2019 Particles-on-Demand for Kinetic Theory *Phys. Rev. Lett.* **12** 13 130602
[5] Zakirov A, Korneev B, Levchenko V and Perepelkina A 2019 On the conservativity of the Particles-on-Demand method for the solution of the Discrete Boltzmann Equation, *Preprint /Keldysh Institute Preprints* **35** 19
[6] Zipunova E, Perepelkina A and Zakirov A 2020 Regularization and the Particles-on-Demand method for the solution of the discrete Boltzmann equation, *DSFD* **29**
[7] Latt J and Chopard B 2006 Lattice Boltzmann method with regularized pre-collision distribution functions *Mathematics and Computers in Simulation* **72**(2) 165-8