Multi-Particle collision dynamics

A mesoscale model for hydrodynamics

Francesco Musso

University of Trento

The Multi-Particle collision dynamics is an efficient fluid dynamic model, which can integrate Navier-Stokes equations. The model is briefly described and results for a simple system at different Reynolds number are reported. Excellent agreement was found with previous works and with some reference calculations.

1 Introduction

Simulation of matter in complex fluids is a challenging task due to the very different time and length scales involved: chemical reaction in a non-stationary solvent, solvated polymers and proteins, self-propelled objects and many others phenomena are examples of this scenario.

All atom simulations of such systems are impractical due to the long integration time required to observe interesting phenomena arising from thermal fluctuations and hydrodynamics interaction. Lattice models and coarse-grained MD models were developed for the solvent dynamics, but while the former lack Galilean invariance and are impractical for complicated geometries, the latter are still computationally not efficient enough.

In 2000 a mesoscopic algorithm called Multi-Particle collision dynamics (MPC), or Stochastic rotation dynamics (SRD), for simulation of fluids was introduced by Malevanets et al. [1, 2]: on a discrete time step particles are propagated via a stream step and a collision step. Space it's discretized to define the interaction range of the multi-particle collision, but the coordinates are continuous variables, which results in numerical stability and an H-theorem [1].

It has been widely used and studied, achieving great results in reproducing long-time hydrodynamics behaviour and transport coefficients. Its widespread use comes from its efficiency and versatility: it can be tuned to reproduce different fluids by just changing parameters of the algorithm without increasing the computational cost and hydrodynamics interaction can be easily turned off to study their relative importance [3]. Its simplicity allows also for analytical prediction of transport coefficients for very different free-path scales, which is something very uncommon in mesoscale algorithms.

One of the limitation of the algorithm, which is shared with the others considering only local interaction, is that it can simulate only compressible fluid since the local interaction takes time to propagate to the entire fluid: in the limit of the velocity of the flow going to zero, MPC becomes very inefficient.

2 Model

The original algorithm consists of two parts: the streaming step and the collision step. The streaming step propagates the position of particles via

$$\mathbf{r}_i(t + \Delta t) = \mathbf{r}_i(t) + \mathbf{v}_i(t)\Delta t \tag{1}$$

while the collision step update the velocity by dividing space in lattice cells where particle interacts with each others via

$$\mathbf{v}_i(t + \Delta t) = \mathbf{u}(t) + \Omega(\mathbf{v}_i(t) - \mathbf{u}(t)) \tag{2}$$

where $\mathbf{u}(t)$ is the mean velocity of the particles of the cell where the i-th particle is in, and Ω is a stochastic matrix representing a rotation of angle α or $-\alpha$ with equal probability.

The properties of the simulated fluid can be controlled by the parameters of the algorithm (Δt and α) and by minor modifications to the algorithm setup e.g. the boundary conditions.

2.1 Simulating out of equilibrium systems at different temperatures

The conservation of energy allow to simulate the Microcanical ensemble almost straightforwardly, while more precautions have to be taken to ensure conservation of momentum in an out of equilibrium system, as a moving fluid, or for the Canonical ensemble.

In the particular case of a fluid flow, early works about the model showed how the conservation of momentum depends heavily on the mean free path of particles in the system [4]. In fact it was proven that in the case of small mean free path, compared to the cell lattice size a, particle stays in the same interaction cell for long times. This behaviour introduces correlation in the velocities of particle in the same cell, breaking the molecular chaos assumption.

In order to overcome this issue a cell shifting procedure was proposed: before the collision step all lattice cells are shifted by the same random vector with coordinates in $[-a/2, a/2]$: this procedure breaks the correlation and restore Galilean invariance also for small mean free path systems.

2.2 Thermostatting

Various thermostatting techniques have been devised for MPC. One of the easiest could be to rescale the velocities locally (in a cell) by multiplying them for the correct factor.

In the case of a fluid flow another method could be to reassign the velocities of the particles in the incoming region with the Maxwell distribution at the right temperature.

In order to simulate the system in the Canonical ensemble the collision step has to be changed: instead of rotation of the relative velocities, new v_i^{ran} are drawn

from a Gaussian distribution with variance $k_B T/m$ and the collision step will be

$$\mathbf{v}_i(t+\Delta t) = \mathbf{u}(t) + v_i^{ran} - \sum_{j \in cell} v_i^{ran}/N_c \tag{3}$$

where N_c is the number of the particles in the cell. This kind of thermostatting procedure changes completely the algorithm making it more difficult to treat analytically and, above all, less computationally efficient [5].

2.3 Dependence of transport coefficients on the parameters

The dependence on Δt of the algorithm can be exploited to control the Schmidt number [6], which is defined as

$$S_c = \frac{\nu}{D} \tag{4}$$

where ν is the kinematic viscosity and D is the mass diffusivity.

Having a fine control over S_c allows to simulate very different fluids, since it regulates the ratio between the hydrodynamic contributions and thermal fluctuations in the system. This is one of the main advantages of this mesoscopic algorithm with respect to the others, since the interval of S_c which can be simulated is very big for very different geometries.

Another advantages of MPC is the great accordance between analytical solution for transport coefficient and the numerical simulation results, which allows to have a much finer control over the simulation setup. In Figure 1- 2 are reported the simulation results with their analytical counterpart and the theoretical prediction S_c as a function of Δt from Ripoll et al. [6].

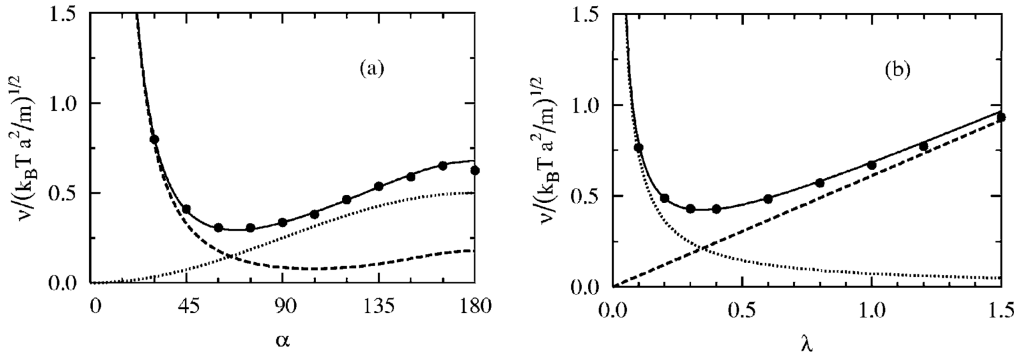

Figure 1: The points represent the simulation results, while the full line the theoretical prediction. The dashed lines are the theoretical contributions coming from the streaming and collisional part. (λ is the dimensionless equivalent of Δt)

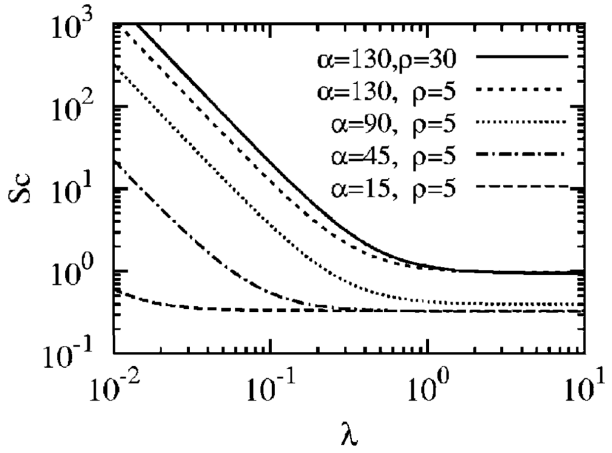

Figure 2: Theoretical dependence of S_c on the time parameter (λ is the dimensionless equivalent of Δt)

3 Implementation and results

The following section is dedicated to reproduce the results from Lamura et al. [7], about the flow around a square cylinder in a 2-dimensional channel, using MPC.

3.1 System under study

One of the most studied and well-known system in hydrodynamics is a cylinder immersed in a 2D channel with a solvent in motion which create a flow. In this case the "cylinder" is a square of edge D in a channel of height $H = 8D$ and length $L = 50D$, as in Figure 3 from Lamura et al. [7]. Every length is given in term of a, which for convenience is set to 1.

The flow is driven in the positive x direction, by, after every step, replacing the x component of the velocity of particles in the inflow region I, which is the one for $x < 10$, with Maxwell-Boltzmann distributed velocities with average value

$$u_x(y) = \frac{4v_{\max}}{H^2}(H - y)y \tag{5}$$

in the x-direction and zero average in the y-direction. This gives rise to the Pouisille flow which is well described theoretically and numerically.

The average density is set to $\rho = 10$ per lattice cell and the masses to $m = 1$. The velocities distribution width is then given by the square root of $k_B T = 0.4\ ma^2/\Delta t^2$, where again Δt is set to 1 for computational convenience.

Periodic boundary condition are imposed in the x direction, while no-slip boundary condition in y direction. No-slip boundary condition imply that the average velocity at the boundaries is zero and it is enforced by applying the bounce-back rule: when a particle hit the wall is shot right back in the direction it came from.

Since the mean-free-path of the system is $\sim 0.63\ a$, which is comparable with the lattice constant, the shift-grid procedure described before is not required. In any

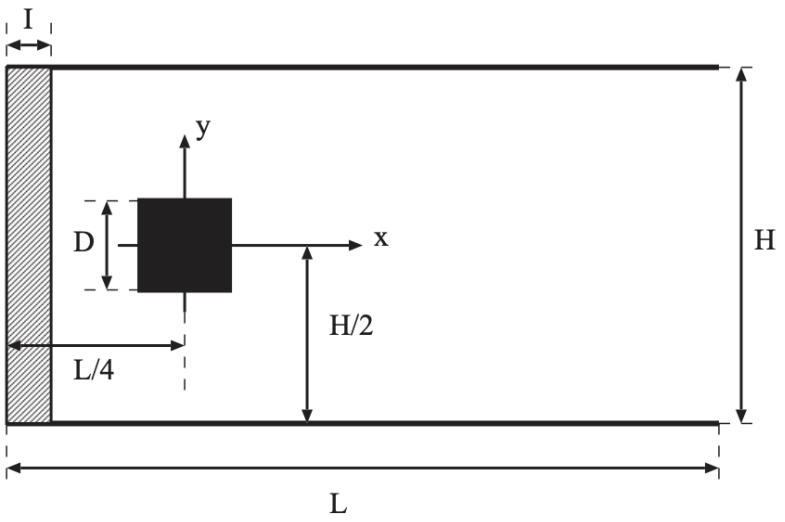

Figure 3: Geometry of the system

case the mean-free-path is not big enough to guarantee Galilean invariance, so a simpler trick is used: the walls are shifted by a small amount so that the cell lattice fills the channel asymmetrically [8].

3.2 Viscosity calculation

The flow patterns in hydrodynamics can be characterized by a dimensionless number called Reynolds number (R_e), which measure the ratio between the inertial forces and viscous ones. In the case of high R_e the inertial forces dominate and the flow is turbulent, while when the viscous forces prevail, R_e will be low and the flow is laminar.

In the case of the planar Puoisille flow under study here, the Reynolds number is given by

$$R_e = \frac{v_{\max} D}{\nu}$$

By fixing $v_{\max}$, in this case $v_{\max} = 0.2$, and changing D, one can study the flow around the cylinder at different R_e in order to find the phase transition between the laminar and turbulent flow phase.

In order to characterize the R_e, the kinematic viscosity has to be computed. In the case of the planar Poiseuille flow, the velocity distribution in the direction parallel to the flow is given by [1]

$$v_x(y) = \frac{G}{2\mu} y(H - y)$$

[1] check out this brief recap of planar Poiseuille flow laws.

where G is the constant pressure gradient $G = -\frac{dp}{dx}$ across the channel and μ is the dynamic viscosity, which is equal to ν multiplied by the density. It can also be shown that the shear stress at the walls is given by

$$\tau(y=0) = \frac{H}{2}\left(-\frac{dp}{dx}\right) \qquad \tau(y=H) = -\frac{H}{2}\left(-\frac{dp}{dx}\right)$$

So by computing the shear stress at the walls one can obtain G and with the velocity profile one can obtain ν. The velocity profile can be assumed the same as the one imposed in the inflow region (Eq. 5) or can be computed during the simulation. All the calculation for these quantities has to be executed without the cylinder in the channel.

The results for the velocity profile are reported in Figure 4, where the channel is simulated with $D = 6$.

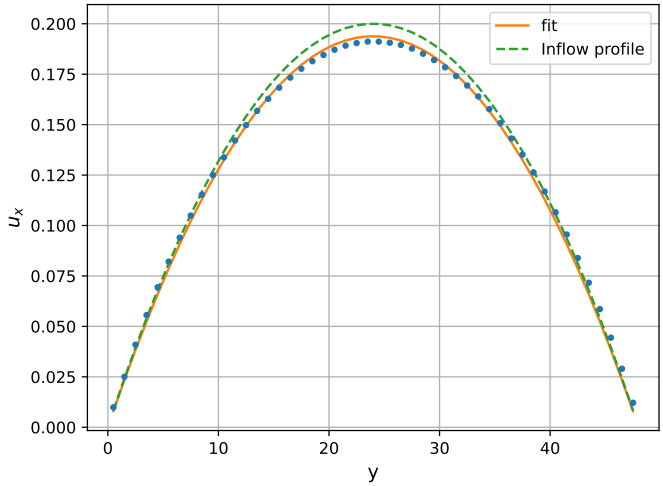

Figure 4: Velocity profile for a channel with $D = 6$

The pressure gradient is equal to $G = (0.00072 \pm 0.00001)\ a/\Delta t^2$ and the two estimates for the viscosity are $\nu = (0.104 \pm 0.002)\ a^2/\Delta t$, assuming Eq. 5, and $\nu = (0.108 \pm 0.002)\ a^2/\Delta t$ using the computed velocity profile. Both estimates are compatible with $\nu = (0.110 \pm 0.004)\ a^2/\Delta t$ reported by Lamura et al. [7]. From now on, to compute the Reynolds number, the second estimate will be used.

3.3 Recirculation length and drag coefficient computations

Knowing the viscosity ν, the system can be studied at different R_e by changing D. Due to computational limitation, the calculations here stop at $R_e \simeq 40$, which implies $D = 24$ and a number of particle of about 16 millions.

One of the most interesting quantities to describe the wake of an object in a fluid flow is the so called *recirculation length*: behind the cylinder a zone of low pressure arises and the mean velocity on the central axis become negative up to a certain length where it become positive again. The point where the velocity become positive again is defined as the recirculation length L_r and it gives a measure of the dimension of the wake. In Figure 5 is reported an example of the velocity distribution behind the central axis of the square.

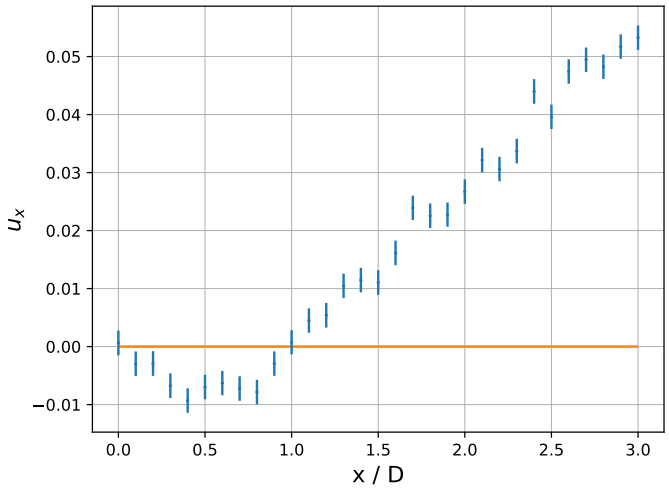

Figure 5: Example of velocity distribution behind the cylinder, with $D = 12$

The plot for L_r as a function of R_e is reported in Figure 6, with results from the original paper and from a paper with high precision calculation using different algorithms. The value obtained match with the values coming from the original paper, which are less then the one from Breuer et al. [9], which in this case can be considered as the theoretical reference. The MPC algorithms seems to struggle more to replicate the desired behaviour at bigger R_e. That is probably due to the more turbulent flow, since R_e is closer to the critical value of $R_e = 49$, which makes the recirculation length a lesser precise quantity to evaluate. In any case the implementation given here seems to better simulate the wanted behaviour, coming closer to the "reference" by Breuer et al. [9].

One of most crucial quantity describing an object in a fluid flow is the *drag coefficient*, which measure the resistance of the fluid to the movement of the object in it. The drag coefficient C_d is a dimensionless number that can be computed with the formula

$$C_d = \frac{2F_x}{\rho v_{\max}^2 D}$$

where F_x is the force of the flow on the cylinder parallel to the direction flow.

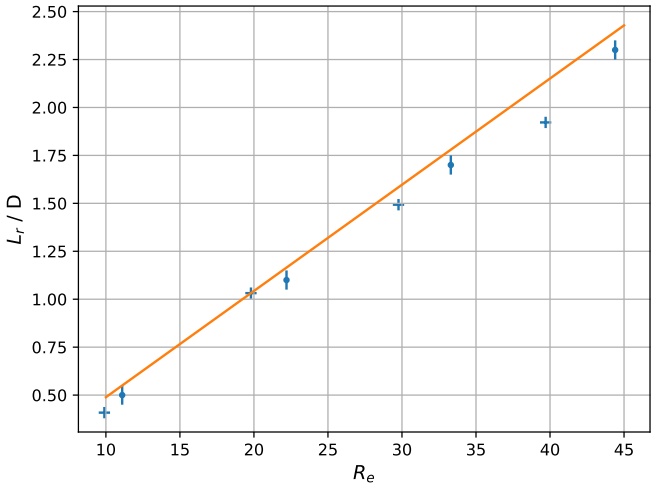

Figure 6: Recirculation length as a function of the Reynolds number: this study (·), Lamura et al. [7] (+), Breuer et al. [9] (−).

The value obtained for C_d as function of R_e are reported in Figure 7, alongside the values from the other works.

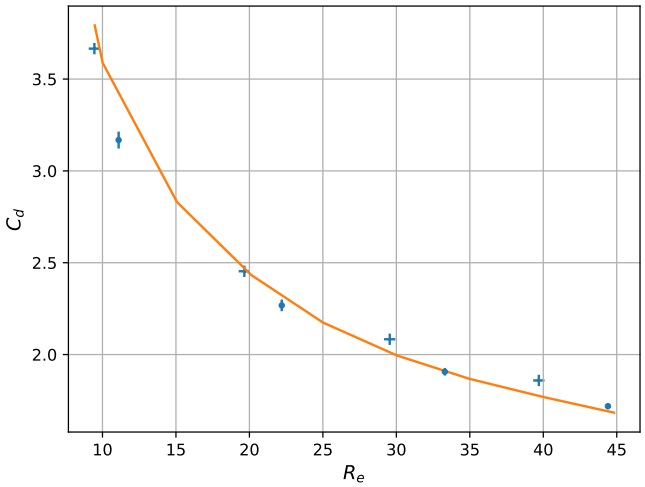

Figure 7: Drag coefficient as a function of the Reynolds number: this study (·), Lamura et al. [7] (+), Breuer et al. [9] (−).

The discrepancies at small R_e could be the result of some wall effect since the system is smaller. Wall effects are less prominent in Lamura et al. [7] because of the different technique to assure Galilean invariance was used. The methods show

their difference also for bigger R_e, where the fixed-shifted cell of this study seems to reproduce better the desired behaviour.

3.4 Computational cost

All the calculation were run on the university cluster because of the large number of particle involved (16 millions for $R_e \simeq 40$), which implies a high memory requirement and high computational power. Nonetheless the algorithm is very efficient since, due to the relatively simple operation required in the steps (additions and matrix vector multiplications in 2/3 dimensions) and the independence of operations between different cells, it can be easily and heavily parallelized to be run extremely efficiently on GPUs, also for much bigger systems.

4 Conclusion

The MPC algorithm has proven over the year to be an efficient and reliable algorithm for hydrodynamics simulations. Here some results for a benchmark system were computed, and the adaptability of the algorithm was highlighted: by changing the cell-shifting procedure it was shown that for some regimes the algorithm can perform better in replicating the wanted behaviour. The versatility of the algorithm is also proven by the countless modification proposed over the years, in order to simulate different hydrodynamics phenomena [5].

All the code used in this work can be found here.

References

[1]Malevanets et al., "Mesoscopic model for solvent dynamics", The Journal of Chemical Physics **110**, 8605–8613 (1999).

[2]Malevanets et al., "Solute molecular dynamics in a mesoscale solvent", The Journal of Chemical Physics **112**, 7260–7269 (2000).

[3]Kikuchi et al., "Polymer collapse in the presence of hydrodynamic interactions", The European Physical Journal E **9**, 63–66 (2002).

[4]Ihle et al., "Stochastic rotation dynamics: a galilean-invariant mesoscopic model for fluid flow", Phys. Rev. E **63**, 020201 (2001).

[5]Gompper et al., "Multi-particle collision dynamics: a particle-based mesoscale simulation approach to the hydrodynamics of complex fluids", in *Advanced computer simulation approaches for soft matter sciences iii* (Springer Berlin Heidelberg), pages 1–87.

[6]Ripoll et al., "Dynamic regimes of fluids simulated by multiparticle-collision dynamics", Physical Review E **72** (2005).

[7]Lamura et al., "Numerical study of the flow around a cylinder using multi-particle collision dynamics", The European Physical Journal E **9**, 477–485 (2002).

[8]Lamura et al., "Multi-particle collision dynamics: flow around a circular and a square cylinder", Europhysics Letters (EPL) **56**, 319–325 (2001).

[9]Breuer et al., "Accurate computations of the laminar flow past a square cylinder based on two different methods: lattice-boltzmann and finite-volume", International Journal of Heat and Fluid Flow **21**, 186–196 (2000).