Particle Dynamics Methods of Blood Flow Simulations,

Alen Tosenberger, Vsevolod Salnikov, Nikolai Bessonov, Evgeniya Babushkina, Vitaly Volpert

Math. Model. Nat. Phenom.
Vol. 6, No. 5 , 2011, pp. 320-332
DOI: 10.1051/mmnp/20116512

Particle Dynamics Methods of Blood Flow Simulations

A. Tosenberger[1] *, V. Salnikov[1], N. Bessonov[1,2], E. Babushkina[2], V. Volpert[1]

[1] Institut Camille Jordan, Université Lyon 1, UMR 5208 CNRS
69622 Villeurbanne, France
[2] Institute of Mechanical Engineering Problems, 199178 Saint Petersburg, Russia

Abstract. Various particle methods are widely used to model dynamics of complex media. In this work molecular dynamics and dissipative particles dynamics are applied to model blood flows composed of plasma and erythrocytes. The properties of the homogeneous particle fluid are studied. Capillary flows with erythrocytes are investigated.

Key words: molecular dynamics, dissipative particle dynamics, blood flows, erythrocyte dynamics
AMS subject classification: 76Z05, 92C05

1. Introduction

Dissipative particle dynamics is a method to simulate systems of particles. It originates from molecular dynamics where each particle moves according to Newton's second law. The difference between two methods is in the form of forces acting between the particles. In molecular dynamics, it is a pairwise potential force which depends on the distance between each two particles. In dissipative particle dynamics, there are two additional forces, dissipative and random. Their form depends on the medium which we want to describe. In the case of a homogeneous Newtonian fluid, dissipative and random forces are also pairwise and have a special form [3], [9], [11]. For properly chosen values of parameters, this approach is equivalent to Navier-Stokes equations. However, from the computational point of view, it is more expensive. Its application becomes justified to model complex media, in particular blood flows. Conventional approaches, where blood is considered as a homogeneous non-Newtonian fluid, is widely used and allows one to study numerous

*Corresponding author. E-mail: tosenberger@math.univ-lyon1.fr problems of hemodynamics (see, e.g., [8]). However it has its limitations when we need to describe in more detail cell-cell interaction in the flow or cell-vessel wall interaction. In this case we need to take into account individual cells with their geometry and structure (elastic cell membrane, intracellular medium). Several approaches are developed in order to study blood flows considered as a heterogeneous medium composed of plasma and blood cells: lattice Boltzmann method [12], smoothed particle dynamics [10], dissipative particle dynamics [5], immerse boundary method [7] and other methods to approximate the membrane [1]. We will briefly present some results for molecular dynamics (MD) and dissipative particles dynamics (DPD).

2. Molecular dynamics and dissipative particle dynamics

2.1. MD description

The first approach is based on Molecular Dynamics (MD) simulations. We introduce a conservative force in a form similar to Lennard-Jones potential but simpler from the computational point of view:

$$\mathbf{F}_{ij}^{C} = F^{C}(r_{ij})\hat{\mathbf{r}}_{ij}, \tag{2.1}$$

where $\mathbf{F}_{ij}$ is the force between the particles i and j, acting on the particle i, $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$, $r_{ij} = |\mathbf{r}_{ij}|$, $\hat{\mathbf{r}}_{ij} = \mathbf{r}_{ij}/r_{ij}$, and the function F is given by the equality

$$F^{C}(r_{ij}) = k\left(\frac{r_c}{r_{ij}} - 1\right)\left(\frac{nr_c}{r_{ij}} - 1\right). \tag{2.2}$$

Here r_c is the force cut-off radius, which determines the distance of force influence, and n is the parameter which determines the second root of the function F.

If the the roots of the conservative force function are equal to each other, then there is no attraction between the particles. In addition to the conservative force, on each particle a force of fixed amplitude, but random direction is applied. This random force, denoted $\mathbf{F}_i^R$, acts as an energy source. The motion of the particles is governed by Newton's second law of motion, in which the artificial viscosity term is introduced in order to adjust dynamic viscosity of a simulated fluid, but also to stabilize the system of particles, if needed:

$$d\mathbf{r}_i = \mathbf{v}_i dt, \qquad d\mathbf{v}_i = \frac{1}{m_i}\left(\mathbf{F}_i^R + \sum_{j\neq i}\mathbf{F}_{ij}^C - c\mathbf{v}_i\right)dt, \tag{2.3}$$

where $\mathbf{r}_i$ is the position vector and $\mathbf{v}_i$ is the velocity vector of particle i. The coefficient c is the strength of the artificial viscosity (dampening).

2.2. DPD description

The second approach considered in this work is Dissipative Particle Dynamics (DPD). We use it in the form described in literature [9], [11], [6]. The DPD is governed by three equations describing conservative, dissipative and random force acting between each two particles:

$$\mathbf{F}_{ij}^{C} = F_{ij}^{C}(r_{ij})\hat{\mathbf{r}}_{ij}, \tag{2.4}$$

$$\mathbf{F}_{ij}^{D} = -\gamma\omega^{D}(r_{ij})(\mathbf{v}_{ij}\cdot\hat{\mathbf{r}}_{ij})\hat{\mathbf{r}}_{ij}, \tag{2.5}$$

$$\mathbf{F}_{ij}^{R} = \sigma\omega^{R}(r_{ij})\frac{\xi_{ij}}{\sqrt{dt}}\hat{\mathbf{r}}_{ij}, \tag{2.6}$$

where $\mathbf{r}_i$ is the vector of position of the particle i, $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$, $r_{ij} = |\mathbf{r}_{ij}|$, $\hat{\mathbf{r}}_{ij} = \mathbf{r}_{ij}/r_{ij}$, and $\mathbf{v}_{ij} = \mathbf{v}_i - \mathbf{v}_j$ is the difference between velocities of two particles. γ and σ are coefficients which determine strength of the dissipative and the random force respectively, while ω^D and ω^R are weight functions; ξ_{ij} is a normally distributed random variable with zero mean, unit variance, and $\xi_{ij} = \xi_{ji}$. The conservative force is given by the equality

$$F_{ij}^{C}(r_{ij}) = \begin{cases} a_{ij}\left(1 - r_{ij}/r_c\right) & \text{for } r_{ij} \le r_c, \\ 0 & \text{for } r_{ij} > r_c, \end{cases} \tag{2.7}$$

where a_{ij} is the conservative force coefficient between particles i and j, and r_c is the cutoff radius.

The random and dissipative forces form a thermostat. If the following two relations are satisfied, system will preserve its energy and maintain the equilibrium temperature:

$$\omega^{D}(r_{ij}) = \left[\omega^{R}(r_{ij})\right]^{2}, \qquad \sigma^{2} = 2\gamma k_B T, \tag{2.8}$$

where k_B is the Boltzmann constant and T is the temperature. The weight functions are determined by:

$$\omega^{R}(r_{ij}) = \begin{cases} \left(1 - r_{ij}/r_c\right)^{k} & \text{for } r_{ij} \le r_c, \\ 0 & \text{for } r_{ij} > r_c, \end{cases} \tag{2.9}$$

where $k = 1$ for the original DPD method, but it can be also scaled in order to change the dynamic viscosity of the simulated fluid. Similar to the previous method, the motion of particles is determined by Newton's second law of motion:

$$d\mathbf{r}_i = \mathbf{v}_i dt, \qquad d\mathbf{v}_i = \frac{1}{m_i}\sum_{j\neq i}\left(\mathbf{F}_{ij}^{C}dt + \mathbf{F}_{ij}^{D}dt + \mathbf{F}_{ij}^{R}\sqrt{dt}\right). \tag{2.10}$$

2.3. Boundary conditions

A practical way to verify the correctness of MD and DPD is to compare the velocity and density profiles in the Poiseuille flow with the corresponding solution of the Navier-Stokes equations. Velocity and density profiles are averaged through a period of time which length depends on parameters, and it begins once the system is near its stable solution. The stability of the solution can be evaluated by the stability of the system energy.

One of the problems that raises when working with particle simulation methods, like MD and DPD, is the implementation of boundary conditions. Choosing a too simple implementation of a no-slip boundary condition can result in distortion of overall velocity and density profiles (Figure 1). Various implementations that try to arrange these problems can be found in literature [4]-[6].

For computational purposes, in our simulations we have tried to use as simple implementations as possible, while keeping the velocity and density profiles sufficiently precise. Therefore, we have started from a simple and straightforward implementation of no-slip boundary conditions where particles rebound from the boundary, but in the moment when a center of a particle is in contact with the wall the tangential to the boundary velocity is set zero. With no-slip boundary conditions, implemented like that, both measured profiles, velocity and density, have a deviation from the analytical solution in the region close to the boundary (Figure 1). These deviations can be decreased directly by increasing the number density while reducing the cutoff radius, but this makes the method much less robust. To correct them, the problem of velocity and density profiles can be studied separately. Density oscillations appear near the no-slip boundary because there is no particles on the other side of the wall, so the forces acting on the particles at the distance less than r_c from the boundary are not in balance.

To solve this problem we consider two approaches. The first approach, which we call half-periodic boundary condition, consists of allowing the particles which are next to one wall to feel the particles which are close to the opposite wall, just as it is case with periodic boundary conditions, but particles are still rebounded from the boundary. Interaction of particles on the opposite parts of the domain is done by taking into account only the part of forces between them which is orthogonal to the wall. The result is a correct density profile (Figure 1). For simple rectangular domains half-periodic boundary conditions are effective and elegant solution for the density problem. However, it can not be easily generalized on the more complex domains. Therefore, for complex domains it is better to measure the average force density and to apply missing part on each particle which is near the boundary [6].

To solve the velocity profile distortions near the boundary, tangential part of velocity needs to be corrected. In the simplest approach, a parabolic velocity profile can be imposed in a narrow boundary layer. More precise and sophisticated methods are also described in literature (see, e.g., [4]).

3. Fluid and flow properties: density, viscosity, velocity

Once we have sufficiently precise velocity and density profiles it is very important to know how the properties of the fluid depend on the choice of method and on parameters. One of the most

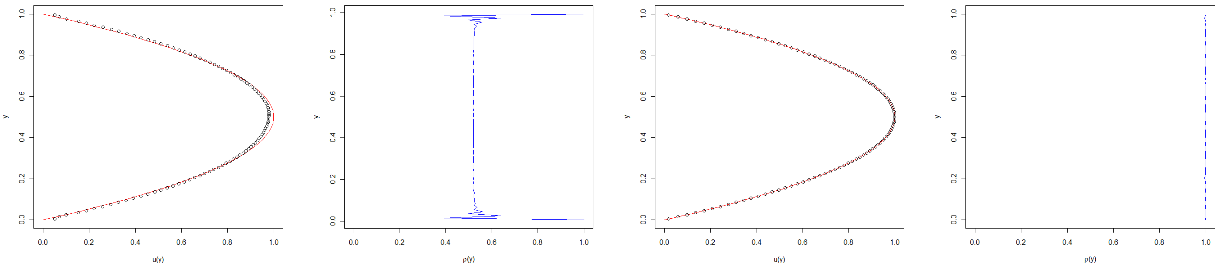

Figure 1: Distortion in velocity and density profiles (left two figures), correct velocity and density profiles (right two figures). Here and in the following figures dimensionless parameters are used.

important characteristics of the fluid is its dynamic viscosity. We can easily obtain its value from the velocity profile of Poiseuille flow simulations.

The Poiseuille flow can be organized with a body force applied to the fluid or with a gradient of pressure. In our simulations we have used the body force. This rises the question about the viscosity dependence on the body force. For the purposes of our work we simulate blood plasma, which can be considered as a Newtonian fluid. Therefore, dynamic viscosity should not depend on the body force. In the MD method we observe that this is not the case, and that velocity depends of body force (Figure 2, left). However, in the DPD method the viscosity remains practically the same (Figure 2, right), except for very small value of body force. This indicates that the DPD method can be more suitable for simulations of Newtonian fluids.

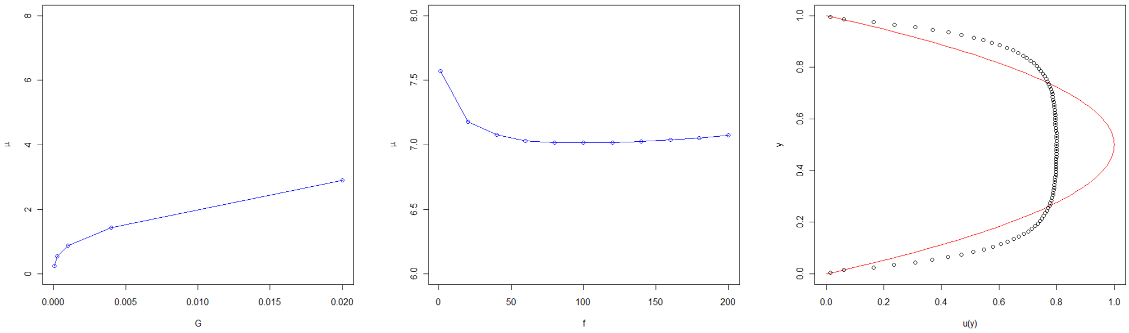

Figure 2: Viscosity dependance on body force in MD (left) and DPD (middle), and MD velocity profile with too large artificial viscosity (right).

Another important question is how to choose parameters of the model which provide realistic physical values of density and viscosity of the fluid and the size of the domain. If we consider for example blood flows in narrow capillaries, then we know the size of the domain, the velocity of the flow, the density and the viscosity of the fluid. First three properties can be easily entered as parameters of the model, but in order to get realistic viscosity we need to adjust all remaining parameters.

In MD we can vary the cutoff radius (and/or the number density), the amplitude of the random force and the value of artificial viscosity. Cutoff radius and artificial viscosity can be used to make large changes in the viscosity of the fluid (Figure 3). However, increasing the artificial viscosity too much results in velocity profiles different from parabolic (Figure 2). On the other hand, random force can be used to scale the dynamic viscosity on a finer scale.

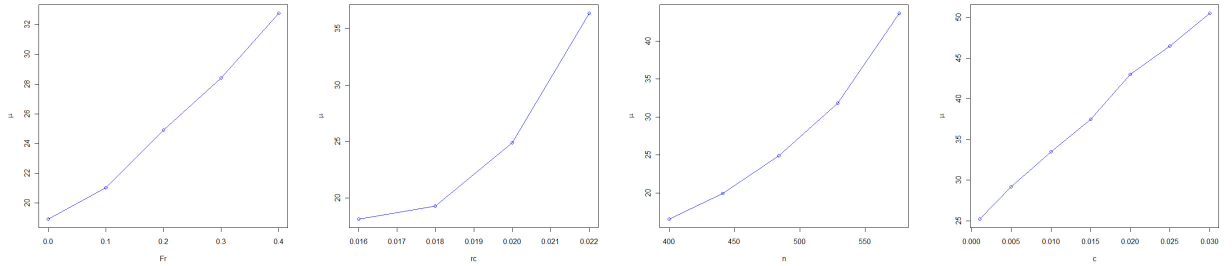

Figure 3: MD (from left to right): dynamic viscosity dependance on random force, cutoff radius, number density and artificial viscosity.

In the DPD method, there are more parameters which have to be adjusted: coefficients of conservative a_{ij} and dissipative force γ (random force coefficient σ is determined via equation (2.8)), exponent k in the equation (2.9), number density n and cutoff radius r_c. A large viscosity variation can be obtained by variation of all parameters, except for the coefficient of the conservative force which can be used to adjust viscosity more precisely. However, for very large changes in viscosity, it is better not to use extreme values of parameters because it can influence the velocity profile, prolong the period of time needed to achieve the stable flow, or just make the simulation more complex to compute. Viscosity dependence on parameters is presented in Figure 4.

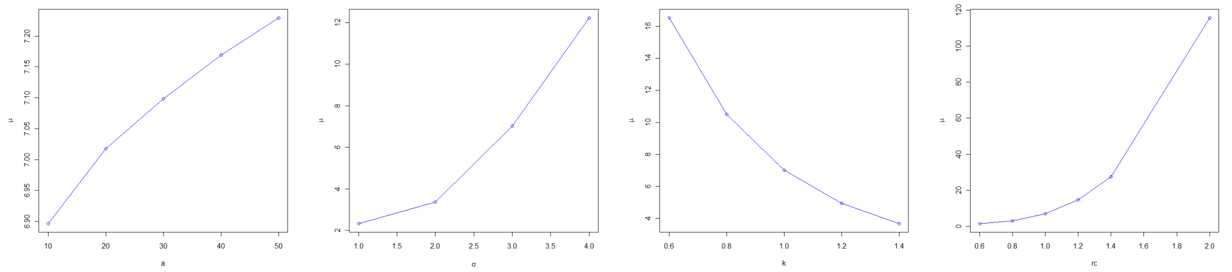

Figure 4: DPD (from left to right): dynamic viscosity dependance on conservative force coefficient a_{ij}, coefficient of dissipative force σ, exponent k and cutoff radius r_c.

One more significant difference between MD and DPD is that DPD converges faster to a stable flow.

4. Erythrocyte in capillary flows

4.1. Erythrocyte model

After simulating blood plasma with particle dynamics, the next step is to introduce erythrocytes in the model. It is known that erythrocytes have a viscoelastic membrane which can have a specific biconcave shape. This shape is a result of minimisation of surface energy. In 2D, we can take a membrane with perimeter $2r\pi$ surrounding an area $\frac{3}{5}r^2\pi$, and we will obtain a 2D ∞-shape. In order to model this behaviour, we take a n-sided regular polygon with particles at its vertices, and define three equations which govern the forces acting between vertices of the polygon. All three equations are based on the Hook's law (Figure 5). The first one defines the forces between each two neighbouring vertices and describes the membrane elongation:

$$F_i^L = k_L\left(1 - \frac{l_i}{l_0}\right), \tag{4.1}$$

where k_L is force strength coefficient, l_0 is equilibrium distance between two neighbouring vertices, and l_i is the distance between vertices i and $i+1$. The second equation describes the forces originating from local bending of the membrane:

$$F_i^B = k_B\left(1 - \frac{\alpha_i}{\alpha_0}\right), \tag{4.2}$$

where k_B is force strength coefficient, α_0 is the equilibrium angle between two neighbouring sides of the polygon, and α_i is the angle between vertices $i-1$, i and $i+1$. Since the membrane tends to become flat, the equilibrium angle α_0 is set to π. The third equation describes the pressure and it is responsible for preservation of the polygon area:

$$F^P = k_P\left(1 - \frac{A}{A_0}\right), \tag{4.3}$$

where k_P is force strength coefficient, A_0 is the equilibrium and A the current area of the polygon. As a result, particle i representing one polygon vertex feels the following forces:

$$\mathbf{F}_i^L = \frac{1}{2}\left(F_{i-1}^L\hat{\mathbf{r}}_{i-1,i} - F_i^L\hat{\mathbf{r}}_{i,i+1}\right), \tag{4.4}$$

$$\mathbf{F}_i^B = \frac{F_{i-1}^B - F_i^B}{l_{i-1}}\hat{\mathbf{n}}_{i-1,i} + \frac{F_{i+1}^B - F_i^B}{l_i}\hat{\mathbf{n}}_{i,i+1}, \tag{4.5}$$

$$\mathbf{F}_i^P = \frac{1}{2}F^P\left(l_{i-1}\hat{\mathbf{n}}_{i-1,i} + l_i\hat{\mathbf{n}}_{i,i+1}\right), \tag{4.6}$$

where $\mathbf{r}_{i-1,i} = \mathbf{r}_i - \mathbf{r}_{i-1}$, $r_{i-1,i} = |\mathbf{r}_{i-1,i}|$, $\hat{\mathbf{r}}_{i-1,i} = \mathbf{r}_{i-1,i}/r_{i-1,i}$, and $\hat{\mathbf{n}}_{i-1,i}$ is the unit vector normal to $\hat{\mathbf{r}}_{i-1,i}$ in the direction outside of the polygon. Then the movement of each particle is simply determined by the sum of these forces according to the Newton's second law of motion. In simulations we used large coefficients k_L and k_P to preserve polygon perimeter and area, while k_B was significantly lower. Starting from a regular polygon of area A, for the choice of parameters so that the perimeter is preserved, the equilibrium angle equals π and the equilibrium area equals $0.6A$, we obtain the characteristic biconcave shape of the membrane (Figure 5). However, because the model is basically the system of springs, to stabilise this system dampening has to be applied. This can be done directly, by introducing dampening expression in each of equations (4.1), (4.2) and (4.3). But once we put the erythrocytes in a flow simulated by MD or DPD it is not necessary to apply any additional dampening, because the artificial viscosity in MD or dissipative forces acting between particles in DPD will stabilise the membrane. Interaction between particles in MD or DPD with the erythrocyte is again point-wise, defined via vertices, and obeys the rules of MD or DPD.

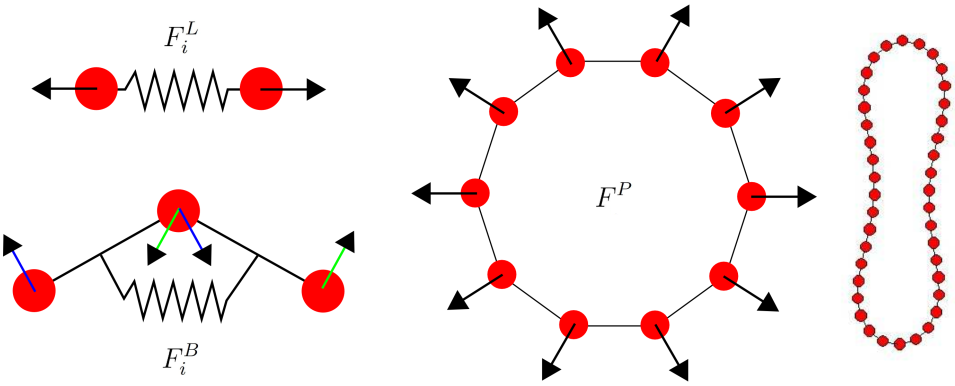

Figure 5: Erythrocyte: scheme of membrane elongation and bending forces (left), pressure (middle), and the obtained stable ∞-shape (right)

4.2. Capillary flow

In order to test the behavior of the erythrocyte model coupled with MD and DPD we simulated flow in a capillary of diameter 10μm with one erythrocyte starting in a ∞-shape turned orthogonally to the flow. It is known ([10]) that in such situations erythrocyte goes from its natural ∞-shape to a so called parachute shape following the parabolic velocity profile. With our model we have obtained the described behavior (Figure 6) and the erythrocyte remained in the parachute shape and position orthogonal to the flow for a quite long time. However this was not the stable state of the erythrocyte, and if the simulations are ran for long enough time, the erythrocyte would eventually turn in the direction of the flow and regain its ∞-shape.

4.3. 3D simulations

In this section we briefly describe blood flow simulations in 3D. We use the same approaches as described above to model homogeneous Newtonian fluids. Erythrocyte model, though similar to

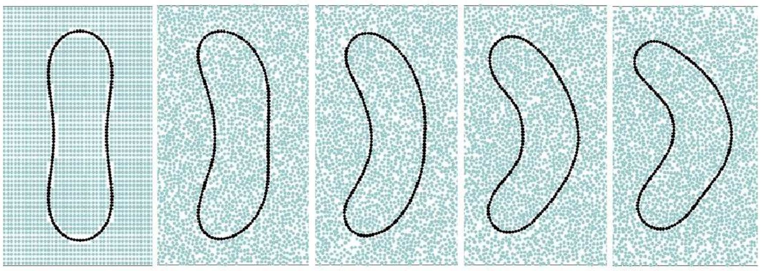

Figure 6: Single erythrocyte in a Poiseuille flow: development of the parachute shape.

the 2D case, becomes more complex in realization. The main difference is related to the triangulation of the surface. Four forces govern behavior of the membrane, instead of three forces in the 2D case. They depend on the volume of the whole body, on the surface area of each triangle, on the angle between neighboring triangles and on the distance between neighboring vertices. The detailed description of the algorithms will be presented elsewhere. Let us note that similar approaches are used in literature [6].

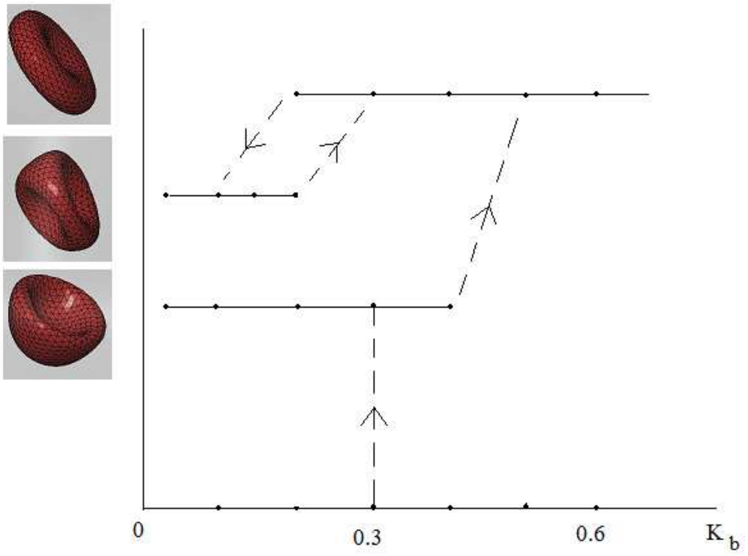

Figure 7: Bifurcation diagram. Existence of different forms of erythrocyte depending on the value of bending coefficient. Different shapes of membrane can coexist for the same values of parameters (see the explanation in the text).

Let us first study the form of the membrane. It is characterized by the following parameters: the number of points at the surface, the coefficients of four forces and the corresponding damping coefficients, the equilibrium volume and angles. Set the number of points $N = 642$ (there are 1280 triangles), the volume force coefficient $k_v = 100$, the side area force coefficient $k_a = 100$, the bending coefficient $k_b = 0.3$, and the stretching coefficient $k_s = 0.2$. The corresponding damping coefficients are $\sigma_v = 1000$, $\sigma_a = 1000$, $\sigma_b = 0.2$, $\sigma_s = 20$. The equilibrium angles equal π and the equilibrium volume 0.6 of the volume of the corresponding polyhedron obtained if the volume force is zero.

With these values of parameters, we take the polyhedron as initial configuration. The form of the membrane converges to the parachute shape (Figure 7). We take this new shape as an initial condition and vary the value of bending coefficient. For each new simulation we take the result of the previous simulation as initial condition. The results are summarized in the diagram in Figure 7. If we decrease k_b, then the shape of the membrane remains qualitatively the same. For larger values of this coefficient we observe transition to the biconcave shape. It persists in a rather wide range of values of k_b. When it become sufficiently small, transition to asymmetric biconcave shape occurs. Taking this new shape as initial condition, we find the interval of values for which it exists. For a larger value of the coefficient, the shape changes back to symmetric biconcave.

Thus, three different shapes of the membrane are observed. For some values of parameters two of them or even all three can coexist.

Consider next behavior of a single erythrocyte in flow in the case where its equilibrium volume equals 0.6, when the shape of the membrane without flow can be either biconcave or parachute. It depends on various parameters including the radius of the cylinder R_c and the ball R_b (initial membrane shape in the simulations). Figure 8 shows the case where $R_c = 0.4$ and $R_b = 0.35$. When the flow change the elastic body to the parachute shape, its diameter becomes larger than that of the cylinder and it touches the walls. We impose slip boundary conditions for the elastic body and no-slip for the particles of the fluid (Section 2.3). After some time the elastic body reaches it steady shape and moves along the cylinder with a constant speed.

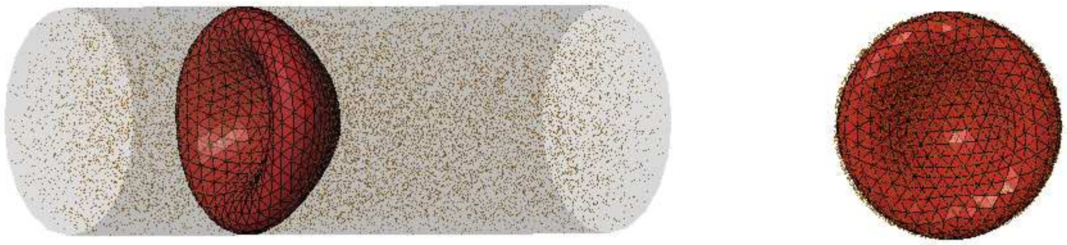

Figure 8: Single erythrocyte in a narrow Poiseuille flow. View from the side (left) and along the axis of the cylinder (right).

In a larger cylinder ($R_c = 0.5$), the erythrocyte behavior can be different. For other fixed parameters, it depends on the applied body force, that is on the velocity of the flow. If this velocity is large enough, then the erythrocyte moves along the cylinder and, at the same time, rotates along the internal surface of the cylinder in a periodic way (Figure 9). For a smaller flow velocity,

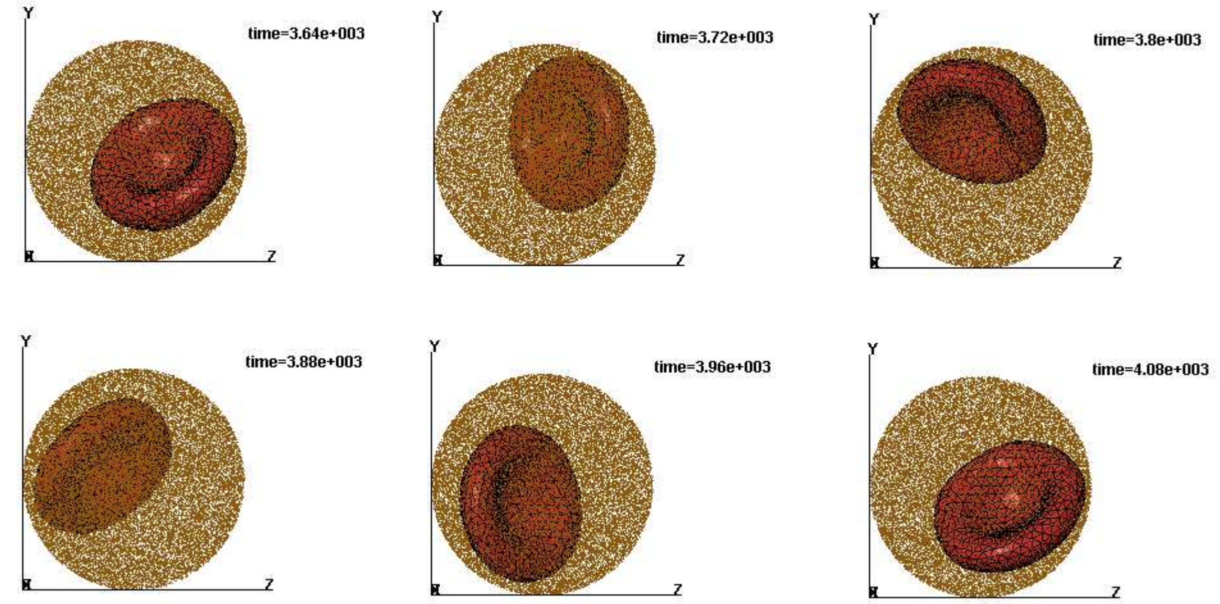

Figure 9: Single erythrocyte in the capillary flow. View along the axis. Erythrocyte rolls around the internal wall of the cylinder.

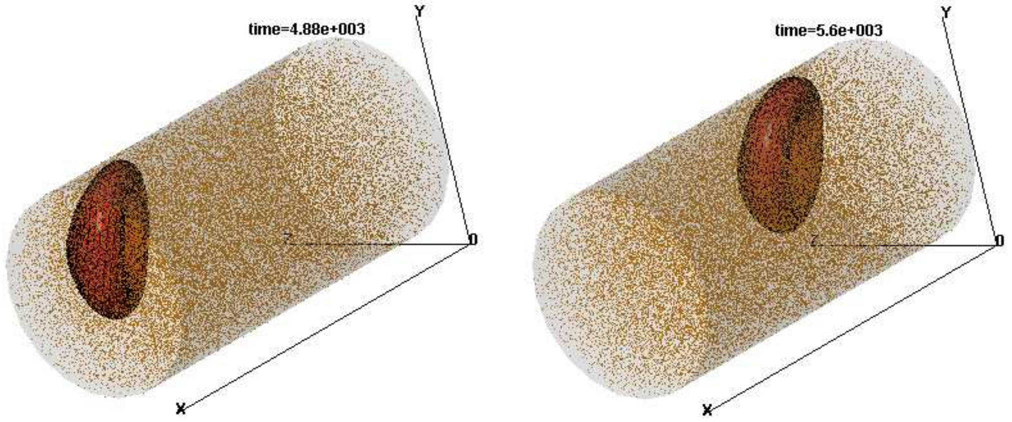

Figure 10: Single erythrocyte in the capillary flow. Erythrocyte slips along the internal wall of the cylinder.

erythrocyte can have a small oscillations near the axis of the cylinder or have a steady position. Finally, for even smaller flow velocity, it comes to the surface and slips along it parallel to the axis of the cylinder (Figure 10). We recall that it does not slip to the wall in our model.

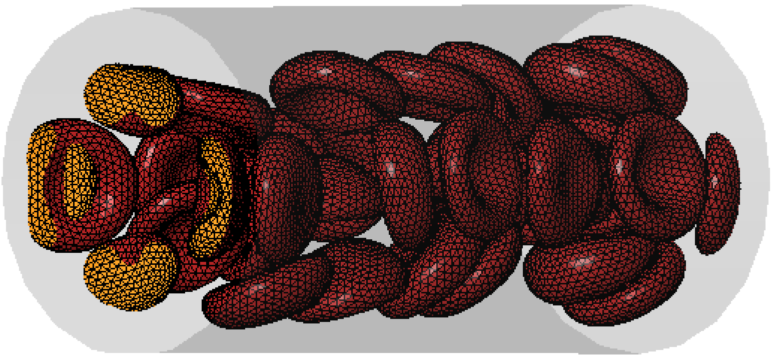

Figure 11: Flow with many erythrocytes.

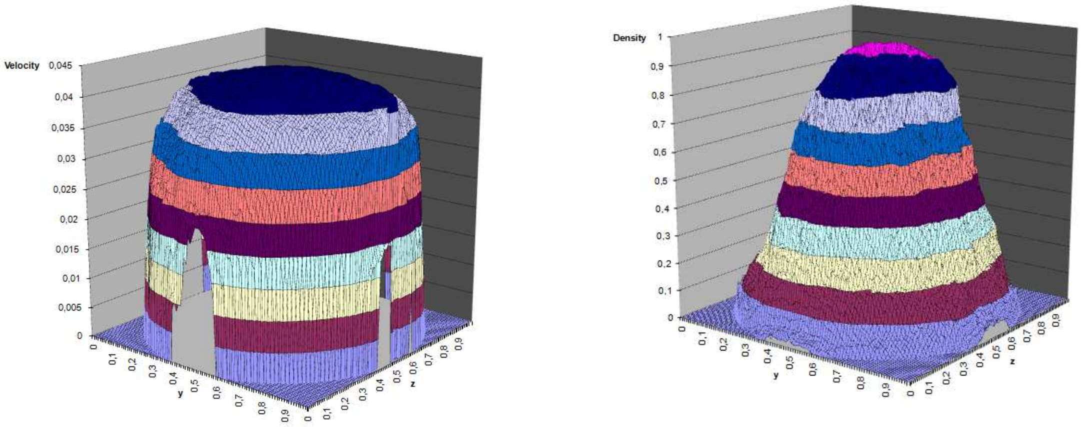

Figure 12: Velocity (left) and density (right) distributions in the flow with many erythrocytes.

We finish this section with an example of flow simulation with many erythrocytes (Figure 11). As it is known, they are not uniformly distributed in the cross section of the cylinder concentrating more close to the axis, and the velocity distribution becomes more flat (Figure 12).

Acknowledgements

This work was partially supported by the projects PICS “Mathematical modelling of blood diseases” (France-Russie), RFFI project No. 10-01-91055, ANR-10-BLAN 0107. The authors are grateful to F. Ataullakhanov, M. Panteleev and A. Tokarev for useful discussions.

References

[1] C. Bui, V. Lleras, O. Pantz, *Dynamics of red blood cells in 2D*, ESAIM Proceedings, 28 (2009), 182–194.

[2] M. M. Dupin, I. Halliday, C. M. Care, L. Alboul, L. L. Munn, *Modeling the flow of dense suspensions of deformable particles in three dimensions*, Physical Review E, 75 (2007), 066707.

[3] P. Espanol, P. Warren, *Statistical mechanics of dissipative particle dynamics*, Europhys. Lett., 30 (1995) (4), 191–196.

[4] D. A. Fedosov, I. V. Pivkin, G. E. Karniadakis, *Velocity limit in DPD simulations of wall-bounded flows*, Journal of Computational Physics, 227 (2008), 2540-2559.

[5] D. A. Fedosov, B. Caswell, G. E. Karniadakis, *A multiscale red blood cell model with accurate mechanics, rheology, and dynamics*, Biophysical Journal, 98 (2010), 2215-2225.

[6] D. A. Fedosov, *Multiscale Modeling of Blood Flow and Soft Matter*, PhD dissertation at Brown University, (2010).

[7] A.L. Fogelson, em Cell-based models of blood clotting, In: A.R.A Anderson, A.A.J. Chaplain, K.A. Rejniak (Eds). Single-cell-based models in biology and medicine. Birkauser, Basel, 2007, pp. 243–270.

[8] G.P. Galdi, R. Rannacher, A.M. Robertson, S. Turek, *Hemodynamics flow. Modeling, analysis, and simulations*, Birkhäuser, Basel, 2008.

[9] R. D. Groot, P. B. Warren, *Dissipative particle dynamics: Bridging the gap between atomistic and mesoscopic simulation*, J. Chem. Phys., 107 (1997) (11), 4423–4435.

[10] S. M. Hosseini, J. J. Feng, *A particle-based model for the transport of erythrocytes in capillaries*, Chem. Eng. Sci., 64 (2009), 4488–4497.

[11] M. Karttunen, I. Vattulainen, A. Lukkarinen, *A novel methods in soft matter simulations*, Springer, Berlin, 2004.

[12] K. Tsubota, S. Wada, H. Kamada, Y. Kitagawa, R. Lima, T. Yamaguchi, *A particle method for blood flow simulation, application to flowing red blood cells and platelets*, Journal of the Earth Simulator, Volume 5, March 2006, 2–7.