Computers and Mathematics with Applications 64 (2012) 278–288

Contents lists available at SciVerse ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

Modelling and simulation of processes in microfluidic devices for biomedical applications

I. Cimrák [a,*], M. Gusenbauer [b], T. Schrefl [b]

[a] Dep. Soft. Technologies, Faculty of Management Science and Informatics, University of Zilina, 01026 Zilina, Slovakia

[b] St. Poelten University of Applied Sciences, St. Poelten, Austria

ARTICLE INFO

Keywords:
Microfluidic devices
Blood flow
Blood cell modelling
Circulating tumour cells
ESPResSo

ABSTRACT

We investigate a mathematical model describing the flow of a liquid in a microchannel. The model incorporates immersed objects in the fluid as well as fixed obstacles and boundaries of the microchannel. Objects can have different elastic properties, including solid objects and deformable objects. The flow description accounts for all types of mechanical interactions: fluid–object, object–object, fluid–walls, and object–walls interactions.

1. Introduction

Microfluidics appeared during the rapid development of inkjet printheads in the 1980s. Since then, the area of microfluidic applications has substantially expanded [1]. Nowadays, advances in microfluidics technology are reforming molecular biology procedures. They enable new methods in enzymatic analysis, DNA analysis, and proteomics [2]. An emerging application area for biochips is clinical pathology, especially the immediate point-of-care diagnosis of diseases. In addition, microfluidics-based devices, capable of continuous sampling and real-time testing of air/water samples for biochemical toxins and other dangerous pathogens, can serve as an always-on ''bio-smoke alarm'' for early warning. Further applications of microfluidic devices are in optofluidics, evolutionary biology [3] and cell biological research [4].

Microfluidic devices enable the precise control of the decreasing fluid volumes on the one hand and the miniaturization of the size of a fluid handling system on the other hand. The behaviour of fluids at the microscale can differ from 'macrofluidic' characteristics because the system starts to be dominated by factors such as surface tension, energy dissipation, and fluidic resistance.

The computer simulations of microfluidic devices significantly improve the design process. The simulations are essential to minimize development time and costs and help the designers get from concept to prototype quickly and efficiently.

For example, a biologist may want to separate deformable cells of type *A* from the suspension containing cells of two types *A* and *B*. He knows that *A* cells are a little bit larger and stiffer than *B* cells. Therefore he wants to use a filter with fixed-sized holes. To determine the size of the holes, he sets up a simulation toolbox and lets the computer decide which hole size leads to best sorting results.

Another example is the simulation procedure telling the scientist how the flow will be affected when a micropost is placed into a microchannel and thus how the capture efficiency of the device will be changed.

We aim at developing a simulation environment that tracks the movement of cells or other immersed objects as they move in the microchannel. The software calculates the interactions with each other, the channel walls, the fluid, and external forces that may be applied to manipulate the immersed objects. The simulation tracks the time evolution of both the fluid

* Corresponding author.
E-mail address: icimrak@gmail.com (I. Cimrák).

0898-1221/$ – see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2012.01.062

and suspended objects. The mathematical algorithms used by the software tend to be readily applied, allowing calculations in a straightforward manner and making it easy to incorporate new forces.

We extend the existing software package ESPResSo [5] (An Extensible Simulation Package for Research on Soft Matter Systems). This package is primarily used for particle movement simulations with a broad range of applications, such as dynamics of copolymers [6], DNA translocation [7], and other. We implement routines that allow a flexible description of immersed objects together with all necessary elastic and mechanical interactions. In this way, the simulation framework will be able to cover a broad range of microfluidic applications with different flowing objects, e.g. blood cells in blood flow, bacteria in water, microorganisms in drinks, and other.

In this research we focus on biomedical applications in the isolation of tumour cells. Circulating tumour cells (CTCs) are disseminated from the site of disease in metastatic or primary cancers, including breast, prostate, lung and other types of cancer. CTCs can be identified and counted in the peripheral blood of patients. The biological analysis of CTCs using lab-on-chip technologies effectively diagnoses the disease, determines personalized therapies and adjusts treatments in real time. Because of their rare occurrence (a few CTCs per 1 mL of blood) CTCs must be isolated from the blood sample. Recent developments and research of microfluidic devices made a significant breakthrough in the detection and filtration of CTCs from blood. One of the isolation approaches is based on a filtration by size of the cells. Generally, CTCs are larger and stiffer than healthy red blood cells (RBCs), and white blood cells are even larger. Therefore a series of microfilters can be used to subsequently filter first the largest white blood cells and then mid-sized RBCs. The isolation of the RBCs based on the size is addressed in [8]. We are in the early stage of this research.

Content of the paper

The paper is organized as follows. In Section 2 we describe the general model for fluid motion, for description of immersed objects and for the coupling of the fluid and immersed objects. We provide a series of tests that calibrate the parameters for fluid–structure coupling involving the study of drag coefficients for oblate and prolate ellipsoids.

The concrete model of a red blood cell is presented in Section 3. All elastic properties are properly described. The core of the paper is presented in Section 4. Here, the coupling of the fluid and the immersed objects is calibrated and the proper friction coefficient is determined. Next, the RBC model parameters are calibrated using the experimental data from literature.

In this paper we focus on building the model of an RBC, its calibration and implementation in ESPResSo. General features of the simulation environment will be explained on the example of cancer cell isolation. We show an example of a simulation of a cell passage through a narrow channel. Such kind of simulations will be used for determination of the minimal gap sizes through which a healthy RBC can pass while a CTC is blocked. These results will be used in the design process of new devices in our upcoming works.

2. Model

To describe the mechanical processes inside a microfluidic device, we need to take into account the following phenomena:

Fluid dynamics. The dynamics of the fluid will be governed by the lattice–Boltzmann method (LBM). The LBM is fast and easy parallelizable, which will be necessary for simulations including high number of immersed objects.

Immersed objects. Immersed objects will be represented by the immersed boundary method (IBM), which is broadly used to describe the boundary of an object without the necessity to change the discretization mesh. The boundary of an object is represented by a triangular mesh containing points on the surface of the object. These points are moving in space under the influence of fluid–object interaction forces, as well as mechanic, elastic, and magnetic forces. For each type of immersed object, different forces are applicable. For example, a healthy RBC is highly deformable, so the stiffness contribution will be low, whereas the cancer cells are more rigid so the stiffness contribution will be higher.

Coupling of the fluid and the immersed objects. Interactions between fluid and objects will be simulated using the drag force acting on an obstacle moving in the fluid. This principle is well-established and already implemented in ESPResSo.

2.1. Fluid dynamics by the lattice–Boltzmann method

Instead of solving the Navier–Stokes equations, which solve the conservation equations of macroscopic properties, the LBM models [9] the fluid consisting of fictive particles. Such particles perform consecutive propagation and collision processes over a discrete lattice mesh. The unknown in the LBM is the distribution function for fictive particles. Macroscopic properties can be recovered by explicit formulas involving the unknown distribution function n.

Consider a lattice placed over the three-dimensional domain and consisting of cubic cells. This lattice creates an Eulerian grid which is fixed over the entire simulation. The variable of interest in the LBM is $n_i(x, t)$ which is the particle density function for the lattice point x, discrete velocity vector e_i, and time t. We use the D3Q19 version of the LBM (three

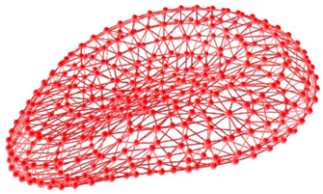

Fig. 1. Triangular mesh representing the boundary of an RBC deformed in the fluid flow.

dimensions with 19 discrete velocities, so $i = 1, \ldots, 19$). The governing equations for the LBM, in the presence of external forces, are

$$n_i(x + \delta_t, t + \delta_t) = n_i(x, t) - \frac{1}{\tau}(n_i(x, t) - n_i^{eq}(x, t)) + f_i(x, t), \tag{1}$$

where δ_t is the time step, τ denotes the relaxation time, n_i^{eq} is the equilibrium function depending on macroscopic variables velocity u and density ρ, and f_i is the external force exerted on the fluid. The macroscopic quantities such as velocity u and density ρ are evaluated from

$$\rho(x, t) = \sum_i n_i(x, t) \quad \text{and} \quad \rho(x, t)u = \sum_i n_i(x, t)e_i.$$

2.2. Immersed objects

The immersed objects are characterized by their boundaries. The boundaries are represented with a set of immersed boundary (IB) points, which may be advected by the fluid interaction. This method is especially suitable for the simulation of the deformation of immersed boundaries by fluid–structure interaction, and it has been widely used in biological fluid dynamics [10].

To take the mechano-elastic properties of the immersed objects into account, a triangular mesh is created on top of the IB points, see Fig. 1. Geometrical entities in this mesh (edges, faces, angles between two faces, ...) are used to model stretching, bending, stiffness, and other properties of the boundary.

For the motion of the IB points we use the Newton equation of motion

$$m_{ib}\frac{d^2X_j}{dt^2} = F_j, \tag{2}$$

where m_{ib} is the mass of the IB point, X_j is the position and F_j is the force exerted on the particular IB point. The source of F_j is twofold: fluid–structure interaction, which is described in Section 2.3, and elastomechanical properties of immersed objects. The latter are described in Section 3 for the case of red blood cells.

The mass of the IB point is a parameter that does not have any particular physical meaning. It is a free parameter that must be calibrated to fit experimental data. This parameter is the same for all IB points of one immersed object, however it will differ for different immersed objects.

2.3. Coupling of the fluid and the immersed objects

Eqs. (1) and (2) describe the motion of the fluid and of the immersed objects, respectively. Both motions influence each other so we need to couple the two equations. To do so we use an approach from [9] with a drag force that is exerted on an object moving in the fluid. Analogous to the Stokes formula for a sphere in a viscous fluid we assume the force exerted by the fluid on one IB point to be proportional to the difference of the velocity v of the IB point and the fluid velocity u at the same position, the vectorial equation for this relation reads as

$$F_j = \xi(v - u). \tag{3}$$

Here ξ is a proportionality coefficient which we will refer to as the friction coefficient. It is given in N s m^{-1} units. In the previous expression, the velocities v and u are computed at the same spatial location, whereas we possess u in fixed Eulerian grid points and v in moving Lagrangian IB points. Therefore for computation of u in the IB point, we use linear interpolation of the u values from nearby fixed grid points, see Fig. 2.

There is also an opposite effect: not only does fluid act on the IB point, but also an IB point acts on the fluid. Therefore we need to transfer the opposite force $-F_j$ back to the fluid. We distribute F_j from the location of the IB point to the nearby grid points. Distribution is inversely proportional to the cuboidal volumes with opposite corners being the IB point and the grid point. In other words, if the IB point is closer to the grid point, then a larger contribution will be assigned to that grid point [9].

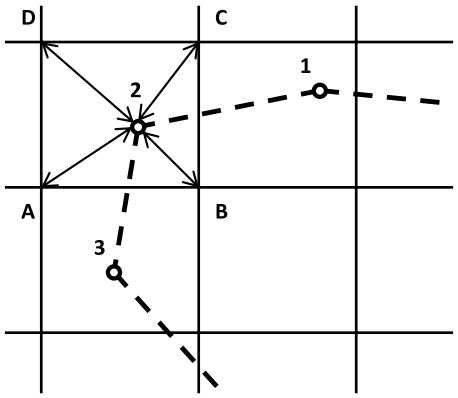

Fig. 2. Two dimensional case: Immersed boundary (dashed line) consists of IB points (denoted by numbers 1, 2, 3,...) creating a non-structured flexible mesh. This mesh is placed over the fixed rectangular fluid grid (with grid nodes denoted by letters A, B, C, D, ...). Once the IB point 2 is located in the rectangle $ABCD$, the fluid–structure interaction takes place between four grid points $ABCD$ and the IB boundary point 2.

The overall behaviour of an immersed object will also be influenced by the number (or the density) of the IB points on its boundary. The influence of the liquid on the movement of the object is through the drag force exerted on the boundary. This drag force should be distributed uniformly across the surface and therefore the density of IB points over the surface should be approximately fixed. So if we double the surface of the immersed object we should also double the number of IB points. Therefore we must keep the surface-to-nodes ratio constant when generating meshes for immersed objects with different size.

Another issue influencing the behaviour is the mass of the IB points. When the fluid pushes the object and tries to change its trajectory, the effect must be dependent on the weight of the object. So if we double the volume of the object, we should double also the mass of individual IB points.

3. Blood cell model

We demonstrate the IB method on a red blood cell model. The RBC consists of a cell membrane and a liquid cytoplasm. The RBC membrane consists of two layers, the plasma phospholipidic bilayer and the cytoskeletal spectrin network. The plasma bilayer is believed to be responsible for the constraints of constant area. It can rearrange itself very easily, and is often referred to as a fluid membrane. The network of proteins attached underneath the plasma layer is responsible for the shear resistance (stretching) and bending. The whole membrane also contains ionic pumps maintaining the inner volume of the RBC constant.

The RBC membrane is only few molecules thick, and hence it is frequently modelled as a two dimensional sheet with some thickness. As we have mentioned, we model the membrane with triangular mesh. The elastic properties of the membrane can be described in terms of four moduli: stretching modulus, bending modulus, area expansion modulus and volume expansion modulus.

The relaxed shape of an RBC has a biconcave shape that can be explicitly expressed by

$$y = 0.5(1 - x^2)^{1/2}(c_0 + c_1x^2 + c_2x^4), \quad -1 \le x \le 1,$$

where $c_0 = 0.207$, $c_1 = 2.002$, $c_2 = 1.122$, equation acquired from [11]. We denote L^0_{AB} the distance between two IB points A and B in the relaxed state. The following expressions for elastic forces are from [12].

Stretching modulus

The membrane of an RBC has a hyperelastic neo-Hookean behaviour. This behaviour is reproduced by the nonlinear spring model. For each edge between two IB points A and B we define L_{AB} the distance between A and B, by ΔL_{AB} we denote deviation from the relaxed state, that is $\Delta L_{AB} = L_{AB} - L^0_{AB}$. The stretching force in the IB points is computed from

$$F_s(A, B) = k_s\kappa(\lambda_{AB})\frac{\Delta L_{AB}}{L^0_{AB}}n_{AB}.$$

Here, n_{AB} is the unit vector pointing from A to B, k_s is the stretching constant, $\lambda_{AB} = L_{AB}/L^0_{AB}$, and κ is a nonlinear function that resembles neo-Hookean behaviour

$$\kappa(\lambda_{AB}) = \frac{\lambda^{0.5}_{AB} + \lambda^{-2.5}_{AB}}{\lambda_{AB} + \lambda^{-3}_{AB}}.$$

Bending modulus

The tendency of an RBC to maintain the resting shape is governed by prescribing the preferred angles between the neighbouring triangles of the mesh. One can argue, that a scaled cell will have all angles between the triangles preserved. However, the scaling of the cell will be prevented by e.g. the volume constraint or, by imposing preferred edge lengths by the stretching modulus.

Denoting by θ^0 the angle between two triangles in the resting shape, we compute the deviation of this angle $\Delta\theta = \theta - \theta^0$ and define the bending force for a triangle ABC

$$F_b(ABC) = k_b \frac{\Delta\theta}{\theta^0} n_{ABC}.$$

Here, n_{ABC} is the unit normal vector to the triangle ABC. This force is assigned to the vertex not belonging to the common edge. The opposite force divided by two is assigned to the two vertexes lying on the common edge.

Area constraint modulus

We compute the deviation of the triangle surface S_{ABC} from the triangle surface in the resting shape $\Delta S_{ABC} = S_{ABC} - S^0_{ABC}$. We impose the area constraint by assigning a shrinking/expanding force for every vertex

$$F_a(A) = -k_a \frac{\Delta S_{ABC}}{S_{ABC}} w_A,$$

where k_a is the area constraint coefficient, and w_A is the unit vector pointing from the centroid of triangle ABC to the vertex A. Similarly we assign the analogical forces to vertexes B and C.

The conservation of local area is too restrictive and we add the global area constraint by introducing a global term

$$F_a(A) = -\left(k_{al} \frac{\Delta S_{ABC}}{S_{ABC}} + k_{ag} \frac{\Delta S}{S}\right) w_A.$$

Here, the area constraint coefficient k_a is split into two coefficients, the global area coefficient k_{ag} and the local area constraint k_{al}.

Volume constraint modulus

We compute the deviation of the global volume of the cell V from the volume in the resting shape $\Delta V = V - V^0$. For each triangle we compute the following force

$$F_v(ABC) = -k_v \frac{\Delta V}{V^0} S_{ABC}\, n_{ABC},$$

where S_{ABC} is the area of triangle ABC, n_{ABC} is the normal unit vector of plane ABC, and k_v is the volume constraint coefficient. The volume of one cell is computed from

$$V = \sum_{ABC} S_{ABC}\, n_{ABC} \cdot h_{ABC},$$

where the sum is computed over all triangles of the mesh and h_{ABC} is the normal vector from the centre of triangle ABC to any plane which does not cross the cell. The force $F_v(ABC)$ is equally distributed to all three vertexes A, B, C.

The theoretical expressions for all elastic moduli are from [12]. Practically, we needed to include them in the simulation environment. We implemented five different interactions (the area constraint has been split into two separate constraints, the local and the global area constraint) in the ESPResSo software. Our implementation however had to conform the parallel structures of ESPResSo. The stretching force was already included. The implementation of the bending and the local area constraints are novel, nevertheless the technique was straightforward. The global area constraint and the volume constraint, however require a different approach. The significant difference is in that those two constraints use global information about the whole blood cell, whereas all the computations are performed locally on different computer nodes, when run in parallel. It can thus occur, that the mesh nodes of one blood cell are maintained by two or more computer nodes. Therefore, a two-step approach had to be used, first, each computer node computes the partial volume. After all computer nodes added their contribution, the complete volume of the cell can be distributed to each computer node and the local forces can be calculated. This solution has been designed with the help of ESPResSo developers [13].

The implemented elastic moduli are specific for the red blood cells. Some of them can be used for characterization of other types of immersed objects. For example, vesicles have only three constraints (volume, area and bending) and capsules only two constraints (volume and stretching). There are also other types of objects for which another specific interactions have to be implemented, for example rigid objects, or solid but deformable objects, all these objects need special treatment.

Finally we emphasize that the presented model of an RBC contains five parameters that influence the actual behaviour of the cell:

k_s—stretching coefficient
k_b—bending coefficient
k_{al}—local area constraint
k_{ag}—global area constraint, and
k_v—volume constraint.

Each of these parameters influences the elastic behaviour of the cell. The stretching and bending coefficients are biologically justified and there exists an approximate value for each parameter: k_s is of the order $5 \times 10^5 \text{pN}\,\mu\text{m}^{-1}$ and k_b is of the order $10^{-1}\text{pN}\,\mu\text{m}$ (parameters from [12,14]). These approximate values in lattice–Boltzmann units are $k_s = 0.005$ and $k_b = 0.001$. However the approximate value gives us only the order of magnitude and the actual value needs to be identified from real experiments. Further, the other three parameters have the following approximate values: $k_v = 0.33\ \mu$ Ns cm^{-1}, $k_{ag} = 5 \times 10^5 \text{pN}\,\mu\text{m}^{-1}$, $k_{al} = 5 \times 10^3 \text{pN}\,\mu\text{m}^{-1}$. They are more phenomenological and also need to be calibrated.

To calibrate our model we need to perform calibration tests with experimental data. These tests are described in Section 4.

4. Calibration

The first experiment described in Section 4.1 involves the calibration of general fluid–object interactions. As described in Section 2.3, the influence of the fluid on the immersed objects and vice versa is maintained through the drag force given by (3). The parameter describing the strength of this interaction is the friction coefficient ξ. However the friction coefficient is directly linked to another parameter, the mass of the IB points m_{ib} appearing in (2). For calibration of these two parameters we use an experiment involving the motion of a moving ellipsoid in a static liquid.

Next we calibrate our cell model in Section 4.2. The experiment involves the stretching of a red blood cell on its opposite sides with a known force. The cell elongates and the cell's horizontal and vertical diameters are measured.

4.1. Friction coefficient and mass calibration

We simulate the following experiment: We put a ball with some initial velocity v_0 to a static fluid. Fluid exerts a drag force on the ball. Under the influence of this force, the ball slows down. We compute the velocity as a function of time. We assume that the density of the ball is the same as that of the surrounding fluid.

Exact solution for velocity

According to the classical theory for the motion of a spherical object in the Stokes creeping flow, the drag force exerted on a ball can be expressed as

$$F_d = -6\pi \nu r v K, \tag{4}$$

where r is the radius of the ball, ν is the dynamic viscosity of the fluid, v is the relative velocity of the ball with respect to the fluid and K is the shape factor. For the case of a sphere $K = 1$, while K is different from one when a prolate or oblate ellipsoid is considered instead of a ball. This shape factor can be calculated from [15,16]. Taking the basic equation of motion of an object with mass m under the influence of force F we have

$$mx'' = F,$$

where x is the position of the ball. Setting $F = F_d$ we can compute the velocity of the ball $v = x'$ from

$$mv' + 6\pi \nu r v K = 0,$$

with the initial condition $v(0) = v_0$. The solution of the previous differential equation is a simple exponential function and thus the exactly computed velocity denoted by v_{ex} can be expressed as

$$v_{ex} = v_0 \exp\left(-\frac{6\pi \nu r K}{m} t\right). \tag{5}$$

Simulated velocity

We set the experiment using the real parameters for blood plasma taken from online resources [17,18]. The actual values are provided in lattice–Boltzmann units. The simulation is done over the finite time interval. The initial velocity v_0 is set such that the Reynolds number satisfies the condition for Stokes creeping flow. The remaining parameters for the experiment are

- channel dimensions $40 \times 25 \times 25$
- viscosity of the fluid $\nu = 1.5$
- density of fluid $\rho = 1.025$
- initial velocity of the ball $v_0 = 0.1$.

Fig. 3. Oblate (left) and prolate (right) ellipsoid and the corresponding radii. The movement of ellipsoids in the fluid is from left to right.

Table 1
Parameters of simulated ellipsoids. We show the axial radius a, the transversal radius b, the shape factor K, the surface and the volume of the ellipsoid, the number of nodes in the mesh, the surface-to-nodes ratio, the friction coefficient ξ, and the recovered value of the mass of the IB points m_{ib}.

a	b	K	Surface	Volume	Nodes	Surf./nodes	ξ	m_{ib}
2	4	0.90	138.75	134.04	431	0.322	0.01	0.0133
3	4	0.95	168.53	201.06	524	0.322	0.01	0.0190
4	4	1.00	201.06	268.08	624	0.322	0.01	0.0240
5	4	1.05	235.31	335.10	730	0.322	0.01	0.0285
6	4	1.10	270.69	402.12	840	0.322	0.01	0.0325
7	4	1.15	306.88	469.15	952	0.322	0.01	0.0365
8	4	1.20	343.65	536.17	1068	0.322	0.01	0.0400

We will simulate the movements of oblate and prolate ellipsoids. The transversal cross-section will always be a circle with radius 4 and the axial radius will be equal to 2 and to 3 for oblate ellipsoids, to 4 for a ball, and to 5, 6, 7 and 8 for prolate ellipsoids, see Fig. 3. Different values of a give different values of the shape coefficient K appearing in (4). The values of K are listed in the third column of Table 1.

In view of the remarks from the end of Section 2.3, we set the surface-to-nodes ratio to a fixed number. This number is chosen arbitrarily, however we have in mind that the triangular mesh should not be too sparse. Then we have generated seven different triangular meshes for ellipsoids with axial radius $a = 2, 3, 4, 5, 6, 7, 8$ and transversal radius $b = 4$. The corresponding surface-to-nodes ratios are listed in Table 1.

With this experiment we need to calibrate two parameters: the friction coefficient ξ and the mass of the IB points m_{ib}. These two parameters are however linked together by Eqs. (2) and (3). Within one simulation with a single object if we double both m_{ib} and ξ, simulation gives the same results. Therefore the friction coefficient and the mass of the IB points can be in the case of a single object rescaled by an arbitrary number. Software implementation is more favourable for flexible mass of the IB points, therefore we fix the friction coefficient to value 0.01 and we determine the correct value of m_{ib}.

We start with recovering m_{ib} for a sphere. The recovered value is $m_{ib} = 0.018$. This value was determined using the least square method by minimization of the distance function defined by

$$|v_{ex} - v_{sim}| = \left(\sum_i (v_{ex}(t_i) - v_{sim}(t_i))^2 \right)^{\frac{1}{2}},$$

where the summation is done over a finite number of time instances discretizing the time interval. We have discretized the time interval with 300 time steps, however in Fig. 4 we have displayed only a fraction of these values for better visibility. The dependence of the distance function on m_{ib} is a convex function with unique minimizer equal to 0.018. The minimizer was determined with the accuracy of two valid decimal places. We should emphasize that the correct mass of IB points is dependent on the viscosity of the liquid as well as on its density.

We further continue to check if the recovered value of m_{ib} is correct. We performed the determination of m_{ib} for six other objects varying from oblate ellipsoid to prolate ellipsoid. The results are summarized in Table 1. The recovered values of m_{ib} are in the last column.

In Fig. 4 we see the graph of simulated velocities v_{sim} for different ellipsoids from Table 1. For every ellipsoid we used a similar surface-to-node ratio.

In view of the remarks from the end of Section 2.3, we know that the increase of the volume should result in linear increase of m_{ib}. The dependence of recovered values for m_{ib} on the volume of the ellipsoids is depicted in Fig. 5. We can see that the dependence is approximately linear with the following formula

$$m_{ib}(\text{vol}) = 6.58 \times 10^{-5}\text{vol} + 5.61 \times 10^{-3},$$

which verifies our simulations.

4.2. Cell calibration

We would like to calibrate the parameters for a blood cell. In the previous section we concluded that the surface-to-nodes ratio will be a fixed number equal to 0.322. A blood cell with a typical dimension, depicted in Fig. 6, has the axial diameter

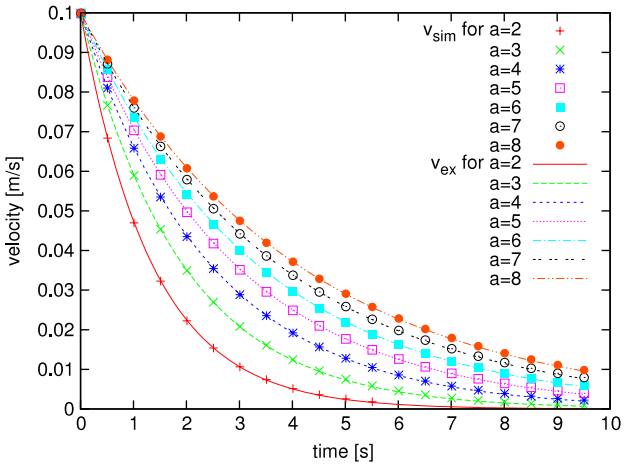

Fig. 4. The time evolution of simulated velocities v_{sim} (displayed with symbols) compared with exact velocities v_{ex} computed from (5) (displayed with lines). Velocities were simulated for different shapes corresponding to axial radii $a = 2, 3, 4, 5, 6, 7, 8$. Other parameters including the number of IB points and m_{ib} are listed in Table 1.

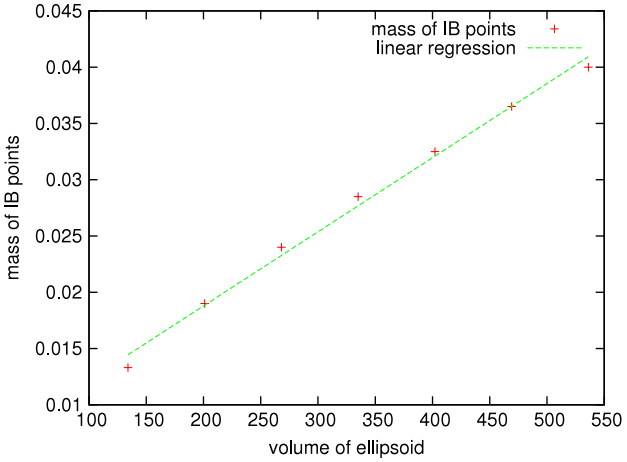

Fig. 5. The linear dependence of mass of IB points on volume of ellipsoids.

equal to 2.5 μm and transversal diameter to 7.92 μm. The surface is approximately 131 μm^2 and volume 90 μm^3. Therefore we generate a triangular mesh with 400 mesh nodes to discretize the surface of the blood cell keeping the surface-to-nodes ratio equal to 0.322. The volume is 90 μm^3 and thus to conform to a linear dependence of volume and mass of the IB points from Fig. 5 we must set $m_{ib} = 0.00005 \times 90 + 0.004 = 0.0085$ and the friction coefficient needs to be set to $\xi = 0.01$.

We have mentioned before that ξ and m_{ib} are linked by Eqs. (2) and (3) and that for the simulation with a single object they can be rescaled. We thus rescale these two coefficients such that $m_{ib} = 1$ and $\xi = 1.17$. These values are used in our further simulations.

To calibrate the elastic properties of the blood cell we simulate the stretching experiment from [14] involving optical tweezers. This laser beam technique is based on the manipulation resulting in the object being trapped by the laser beam. For example, a dielectric bead of silica when trapped by a laser beam can be physically moved as the laser beam is displaced. If such a bead is attached strongly to the surface of a cell, it serves as a handle or grip and displaces the cell membrane. The optical tweezers method can be used to stretch the cell directly in one or more directions by trapping beads that are strategically attached to the cell surface through specific or non-specific binding [14].

As depicted in Fig. 7, a force is being induced on the opposite sides of a cell through two silica beads. For different forces ranging from 67 pN up to 193 pN we compute the axial and transverse diameter of the deformed cell. In Table 2, we present values from [14] denoted by d^f_{ax} and d^f_{tr} with $f = 67$, 130 or 193 pN.

It is too optimistic to hope that we can recover the exact values from Table 2. Therefore we use a least square method for determination of k_s, k_b, k_{al}, k_{ag} and k_v. For simplicity $\mathbf{k}$ denotes the five-dimensional vector

$$\mathbf{k} = (k_s, k_v, k_{al}, k_{ag}, k_v).$$

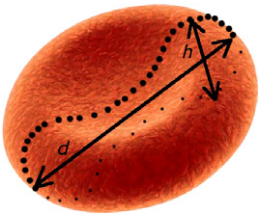

Fig. 6. Dimensions of a red blood cell: axial diameter $h = 2.5\ \mu\text{m}$ and transversal diameter $d = 7.8\ \mu\text{m}$.

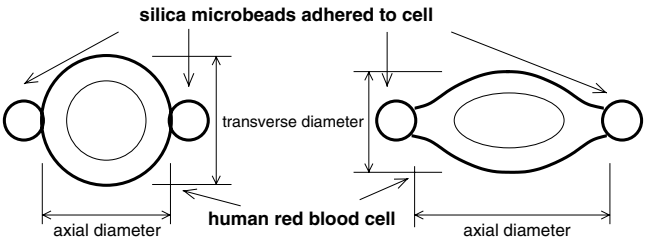

Fig. 7. Scheme of the stretching test.

Table 2
Measured dimensions of an RBC under the stretching test.

Force exerted on cell	67 pN	130 pN	193 pN
Axial diameter d^0_{ax}	12.34	14.17	15.3
Transverse diameter d^0_{tr}	5.05	4.53	4.29

Using our simulation tool we are able to compute the values of d_{ax} and d_{tr} for arbitrary $\mathbf{k}$ and arbitrary force f. Therefore the axial and transverse diameters become functions of $\mathbf{k}$ and f

$$d_{ax} = d_{ax}(\mathbf{k}, f), \qquad d_{tr} = d_{tr}(\mathbf{k}, f).$$

We can build up a cost function $\mathcal{F}$ computed as the square distance of simulated $d_{ax}(\mathbf{k}, f)$, $d_{tr}(\mathbf{k}, f)$ and the data given in Table 2

$$\mathcal{F}(\mathbf{k}) = \sum_{f=67,130,193} (d_{ax}(\mathbf{k}, f) - d^0_{ax}(f))^2.$$

In such a way we can assign a single numerical value for every set of parameters $\mathbf{k}$ that tells us how closely those parameters correspond to reality. If $\mathcal{F}$ is zero then we have found the correct parameters, which is however rarely true.

We started the determination of optimal $\mathbf{k}$ by noticing that the sensitivity of $\mathcal{F}$ to area coefficients and volume coefficient was low. Therefore we first subsequently determined the values of k_{al}, k_{ag}, k_v that kept constant the local and global surface, and the volume. For this we needed about 20 runs for each parameter. In this way we determined the optimal values of k_{al}, k_{ag}, k_v which were used in further determination of k_s and k_b.

Since the sensitivity to the stretching and the bending coefficients was high, the determination of these two parameters had to be performed more carefully. We first set 49 estimates of couple (k_s, k_b) where both parameters took seven values covering a large range of several orders of magnitude. In this manner we detected a few couples (k_s, k_v) that gave significantly lower values of $\mathcal{F}$. These couples are however only very coarse approximations of optimal values.

Then for each coarse approximation we set seven values around it. We have run again 49 simulations to tune the coarse approximation and we have thus obtained a medium-coarse approximation by identifying those couples giving significantly lower values of $\mathcal{F}$ than the others.

Finally we have run the final round. For each medium-coarse approximation we have again set seven values around it and we have run 49 simulations to tune the optimal values. In this way we have determined quite precise approximation of the optimal values. In the final round of simulations, the sensitivity of $\mathcal{F}$ was very low since the range of values was quite restricted.

In Table 3 we present optimal values of $\mathbf{k}$. The optimal values slightly differ from the approximate values $k_s = 0.005$, $k_b = 0.001$ obtained from [12,14].

Using the parameters obtained in the previous sections we simulated the passage of a cell through a narrow channel. In Fig. 8, a red blood cell is depicted on its way through the channel. In this example we used only one processor for the computations of the cell movement. The detailed analysis of parallel structures needed for cell description will be published in our future publications, as well as detailed results concerning minimal gap size through which the healthy red blood cell can pass.

Table 3
Optimal values of elasticity parameters for a healthy RBC.

Stretching k_s	0.008
Bending k_b	0.0016
Local area k_{al}	0.01
Global area k_{ag}	1.0
Volume k_v	10.0

Fig. 8. Simulation of an RBC passage through a narrow channel.

For modelling of more cells one needs to address the collisions between the cells. In our forthcoming research we will use the potential approach that defines a potential between the cells dependent on their position. In case of a collision the potential generates forces repelling two colliding cells.

5. Conclusions

We have investigated a mathematical model describing the flow of a liquid with immersed objects in a microchannel. The model is composed of two components—the flow model, governed by the lattice–Boltzmann method, and the immersed object model, governed by immersed boundary method. These two components interact via the drag force. The model for immersed objects is general, the objects can have different elastic properties.

Further we focus on blood flow modelling and we introduce a model of a triangulated red blood cell. We describe in detail the implementation of elastic properties of an RBC and we list five parameters of the model that can be tuned. Afterwards we calibrate these parameters to fit the experimental data.

The presented model is easy adaptable for other cells appearing in the blood flow. Different cells have different elastic properties and thus the proper parameters need to be calibrated. Further direction for the research will cover the adhesion properties of the cells. The adhesion, or affinity, is used when antibody-covered surfaces appear in the flow with the aim of capturing the CTCs. To model the adhesion process, one needs to "translate" the chemical process into representation by forces so that it can be implemented in the model. This is a challenging interdisciplinary task that will be addressed in the forthcoming research.

Acknowledgments

The authors gratefully acknowledge the financial support of Life Science Krems GmbH, the Research Association of Lower Austria. A large part of this work has been performed during the academic stay of I. Cimrak at the University of Applied Sciences, St. Poelten, Austria in the frame of a mobility allowance awarded by the Fund for Scientific Research—Flanders FWO, Belgium.

References

[1] S. Haeberle, R. Zengerle, Microfluidic platforms for lab-on-a-chip applications, Lab on a Chip 7 (2007) 1094–1110.
[2] K. Herold, A. Rasooly (Eds.), Lab-on-a-Chip Technology: Fabrication and Microfluidics, Caister Academic Press, 2009.
[3] J.E. Keymer, P. Galajda, C. Muldoon, S. Park, R. Austin, Bacterial metapopulations in nanofabricated landscapes, Proc. Natl. Acad. Sci. 103 (2006) 17290–17295.
[4] J. Berthier, P. Silberzan, Microfluidics for Biotechnology, Artech House, Boston, 2006.
[5] H. Limbach, A. Arnolda, B. Manna, C. Holma, ESPResSo- an extensible simulation package for research on soft matter systems, Comput. Phys. Comm. 174 (2006) 704–727.
[6] P. Kosovan, J. Kuldova, Z. Limpouchova, K. Prochazka, E. Zhulina, O. Borisov, Amphiphilic graft copolymers in selective solvents: molecular dynamics simulations and scaling theory, Macromolecules 42 (2009) 6748–6760.
[7] S. Kesselheim, M. Sega, C. Holm, Applying ICC* to DNA translocation. Effect of dielectric boundaries, Comput. Phys. Comm. 182 (2011) 33–35.
[8] S. Zheng, et al., Membrane microfilter device for selective capture, electrolysis and genomic analysis of human circulating tumor cells, J. Chromatogr. A 1162 (2007) 154–161.
[9] P. Ahlrichs, B. Dunweg, Lattice-Boltzmann simulation of polymer-solvent systems, Internat. J. Modern Phys. C 8 (1998) 1429–1438.
[10] Z. Feng, E. Michaelides, The immersed boundary-lattice Boltzmann method for solving fluid–particles interaction problems, J. Comput. Phys. 195 (2004) 602–628.
[11] E. Evans, Y. Fung, Improved measurements of the erythrocyte geometry, Microvasc. Res. 4 (1972) 335–347.
[12] M. Dupin, I. Halliday, C. Care, L. Alboul, Modeling the flow of dense suspensions of deformable particles in three dimensions, Phys. Rev. E (3) 75 (2007) 066707.

[13] S. Kesselheim, P. Kosovan, R. Weeber, F. Weik, Personal communication at ICT University of Stuttgart, Germany, 2011.
[14] J.P. Mills, L. Qie, M. Dao, C.T. Lim, S. Suresh, Nonlinear elastic and viscoelastic deformation of the human red blood cell with optical tweezers, Mol. Cell. Biol. 1 (2004) 169–180.
[15] S. Datta, D. Srivastava, Stokes drag on axially symmetric bodies: a new approach, Proc. Indian Acad. Sci. Math. Sci. 109 (1999) 441–452.
[16] A. Chwang, Y. Wu, Hydromechanics of low-Reynolds-number flow. Part 2. singularity method for Stokes flow, J. Fluid Mech. 67 (1975) 787–815.
[17] C.R. Nave, Hyperphysics, 2011. http://hyperphysics.phy-astr.gsu.edu/hbase/tables/viscosity.html.
[18] K. Benson, Introductory Physics, Lecture, 2011. http://www.emory.edu/PHYSICS/Faculty/Benson/141/141-99/revmcat9/revmcat9.html.