8. Advanced model discussions

In this chapter, advanced aspects of the computational model are presented which have not been addressed in one of the former chapters. A short overview of the sub-steps of the combined algorithm is given in section 8.1. The unit conversion between physical and lattice units is explained in section 8.2, followed by the description of the membrane mesh generation in section 8.3. The behavior of a single spherical, elastic capsule in shear flow is studied in detail in section 8.4 as a benchmark test. An efficient approach for initializing a dense suspension of deformable particles is discussed in section 8.5. Shortcomings and restrictions of the presented model are identified in section 8.6. The need for additional membrane-membrane interaction forces is motivated in section 8.7. Finally, wall slip at smooth walls and its circumvention is discussed in section 8.8.

8.1. Overview of the combined simulation algorithm

After the simulation has been initialized (sections 5.3.1 and 8.5), each time step of the combined algorithm as described in chapters 5, 6, and 7 consists of the sub-steps mentioned below. $\boldsymbol{x}_i$ denotes the position of the membrane node with index i, and $\boldsymbol{X}$ is the position of a fluid lattice node. The time step Δt is set to unity.

1. At the beginning of time step t, the membrane node positions $\boldsymbol{x}_i(t)$ and the fluid state $\boldsymbol{u}(\boldsymbol{X}, t)$, $\rho(\boldsymbol{X}, t)$ are known. From the configuration of the membranes, the forces $\boldsymbol{F}_i(t)$ acting on the membrane nodes in the Lagrangian frame are computed using the membrane model (chap. 7).
2. The membrane forces $\boldsymbol{F}_i(t)$ are spread to the Eulerian grid via immersed boundary method (IBM), cf. eq. (6.5), and the body force density $\boldsymbol{f}(\boldsymbol{X}, t)$ is obtained.
3. The body force $\boldsymbol{f}(\boldsymbol{X}, t)$ is used as input for the lattice Boltzmann method (LBM) which provides the new state of the fluid, $\boldsymbol{u}(\boldsymbol{X}, t+1)$, $\rho(\boldsymbol{X}, t+1)$ (chap. 5).
4. The new velocities of the membrane nodes, $\dot{\boldsymbol{u}}_i(t+1)$, are computed via IBM, cf. eq. (6.6).
5. The new positions of the membrane nodes, $\boldsymbol{x}_i(t+1)$, are found by evaluating eq. (6.12).
6. Information about the membrane and fluid states may be written to the disk for post-processing.
7. Go to sub-step 1 and proceed with time step $t+1$.

The sub-steps are also illustrated in fig. 8.1.

8.2. Conversion between physical and lattice units

A computer does not have the ability to compute dimensional quantities. Unit conversions are necessary in order to input physical data into a simulation and extract physical results again.

The first step is to realize that all mechanical quantities have a unit which can be decomposed into powers of length, time, and density. In the following, it is assumed that only mechanical quantities are required. Temperature and electric charges do not play a role in the present thesis.

T. Krüger, *Computer Simulation Study of Collective Phenomena in Dense Suspensions of Red Blood Cells under Shear,* DOI 10.1007/978-3-8348-2376-2_8,

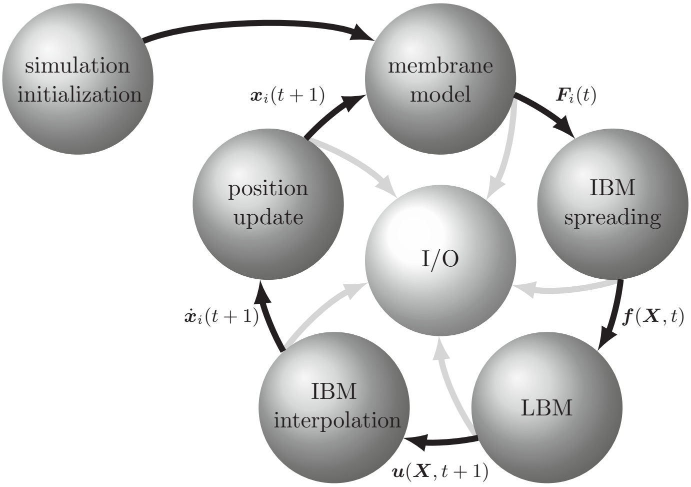

simulation initialization
membrane model
$\boldsymbol{x}_i(t+1)$
$\boldsymbol{F}_i(t)$
position update
IBM spreading
I/O
$\dot{\boldsymbol{x}}_i(t+1)$
$\boldsymbol{f}(\boldsymbol{X},t)$
IBM interpolation
LBM
$\boldsymbol{u}(\boldsymbol{X},t+1)$

Fig. 8.1.: Scheme of the combined simulation algorithm. It proceeds along the black arrows. Data of each sub-step may be written to the disk (gray arrows) for post-processing. More details are given in section 8.1.

Each physical quantity Q may then be written in the form

$$Q = \tilde{Q} C_x^{q_x} C_t^{q_t} C_\rho^{q_\rho} \tag{8.1}$$

where $\tilde{Q}$ is a number (the value of Q in lattice units), and C_x, C_t, and C_ρ are the conversion factors for length, time, and density (with units m, s, and $\mathrm{kg\,m^{-3}}$), respectively. The three numbers q_x, q_y, and q_ρ are the according and uniquely defined powers required to construct the correct unit for Q. Once the three conversion factors C_x, C_t, and C_ρ are known, units can be bidirectionally converted between the lattice and the physical system. It is straightforward to define derived conversion factors, e.g., for the pressure p. The unit of pressure is $\mathrm{Pa} = \mathrm{kg\,m^{-1}\,s^{-2}}$ and thus $C_p = C_\rho C_x^2 C_s^{-2}$. The lattice and physical values of the pressure are then connected via $p = \tilde{p} C_p$. This example can be generalized to any other dimensional mechanical quantity. Numbers have the same values in lattice and physical units, i.e., the conversion factor is unity.

In lattice Boltzmann simulations, C_x is usually the physical length associated with the distance between two neighboring lattice nodes (the lattice constant Δx), $C_x = \Delta x$ and $\Delta\tilde{x} = 1$. This will always be the case in the present thesis. Since the lattice value for the density is commonly set to unity, $\tilde{\rho} = 1$, C_ρ automatically equals the density of the fluid, $C_\rho = \rho$. It remains the identification of the time conversion factor C_t.

If the Reynolds number is the only relevant dimensionless parameter in a lattice Boltzmann simulation, the time conversion factor is computed in the following way: Since the Reynolds number is dimensionless, its value must be the same in the lattice and physical systems. Else, the systems would not be equivalent. The kinematic viscosity of the fluid can then be written as

$$\nu = \tilde{\nu} \frac{\Delta x^2}{\Delta t} = \tilde{\nu} \frac{C_x^2}{C_t} \tag{8.2}$$

where $\Delta x = C_x$ has been used and $\Delta t \overset{!}{=} C_t$ (i.e., $\Delta\tilde{t} = 1$) is the unknown time step. A comparison with eq. (5.7) reveals that

$$\tilde{\nu} = \frac{\tau - \frac{1}{2}}{3} \tag{8.3}$$

is the viscosity in lattice units for a lattice with $c_{\mathrm{s}} = \sqrt{1/3}\Delta x/\Delta t$. If the viscosity of the fluid, ν, the lattice constant, Δx, and the relaxation parameter, τ are known, the time step can be computed,

$$C_t = \Delta t = \frac{\tau - \frac{1}{2}}{3}\frac{\Delta x^2}{\nu}. \tag{8.4}$$

Eq. (8.4) can also be exploited to set up a simulation if another quantity than Δt is initially unknown.

As already mentioned in section 5.5, the LBM is not well suited to simulate low Reynolds number flows since the time step usually becomes small and a large number of time steps is required. This becomes also clear from eq. (8.4): A large viscosity ν leads to a small time step Δt. However, this raises the question whether the exact value of the Reynolds number (as long as it is small) is important after all. The time scale in highly viscous flows is arbitrary and does not depend on the Reynolds number [194, 209]. If the numerical Reynolds number was increased by a factor n, the time step would be increased by the same factor (if Δx and τ are not changed) and the number of required time steps and the related computing time would both be reduced by n. Of course, such an approach is only admissible if the physical results are not significantly compromised. Yet, a similar approach is constantly used in lattice Boltzmann simulations: The lattice Mach number is usually much larger than in reality. As long as the lattice Mach number is small, no *significant* effects are expected. It is generally not affordable to simulate fluids with the correct Mach number. Cates et al. [209] state that ' "fully" realistic simulations (in which lattice parameter values map directly onto those of the real world) are not the goal of mesoscale lattice Boltzmann.' In the present work, suspensions in the viscous regime are simulated, thus, the Reynolds number is small, and inertia effects are not relevant. For that reason, the time step may be increased as much as it is still compatible with the small Reynolds number assumption and as long as numerical stability is not endangered.

It is important to recognize that another dimensionless number is more important than the (small) Reynolds number when deformable particles in an external flow field are considered. The capillary number

$$\mathrm{Ca} := \frac{\rho\nu\dot{\gamma}r}{\kappa_{\mathrm{S}}} \tag{8.5}$$

quantifies the ratio of the viscous shear force of the fluid (density ρ, kinematic viscosity ν, and shear rate $\dot{\gamma}$) and the elastic shear force of the immersed particle (radius r, elastic shear modulus κ_{S} as introduced in section 7.1). The time scale of the physical problem is, therefore, defined by the capillary number, and the time conversion factor can be computed from

$$C_t = \frac{1}{C_{\dot{\gamma}}} = \frac{\tilde{\dot{\gamma}}}{\dot{\gamma}} = \frac{\mathrm{Ca}}{\dot{\gamma}}\frac{\tilde{\kappa}_{\mathrm{S}}}{\tilde{\rho}\tilde{\nu}\tilde{r}} \tag{8.6}$$

when the remaining parameters have already been chosen.

It is easy to see that the capillary number and the Reynolds number can be controlled independently. For example, a proportional increase of the lattice shear rate $\tilde{\dot{\gamma}}$ and the lattice shear elasticity $\tilde{\kappa}_{\mathrm{S}}$ leaves the capillary number invariant, but the Reynolds number is increased since $\mathrm{Re} \propto \dot{\gamma}$ but $\mathrm{Re} \neq \mathrm{Re}(\kappa_{\mathrm{S}})$. This freedom will be used to increase the time step Δt for a fixed and well-defined capillary number in chap. 10.

For the sake of simplicity, the tilde for indicating the dimensionless value of a quantity will be dropped in the remainder of this thesis, except explicitly noted otherwise. Unit conversions in LB simulations are also discussed by Ding and Aidun [154] and Feng et al. [166].

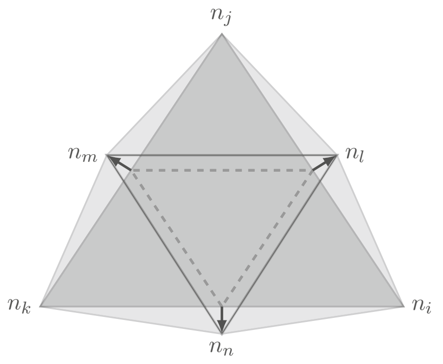
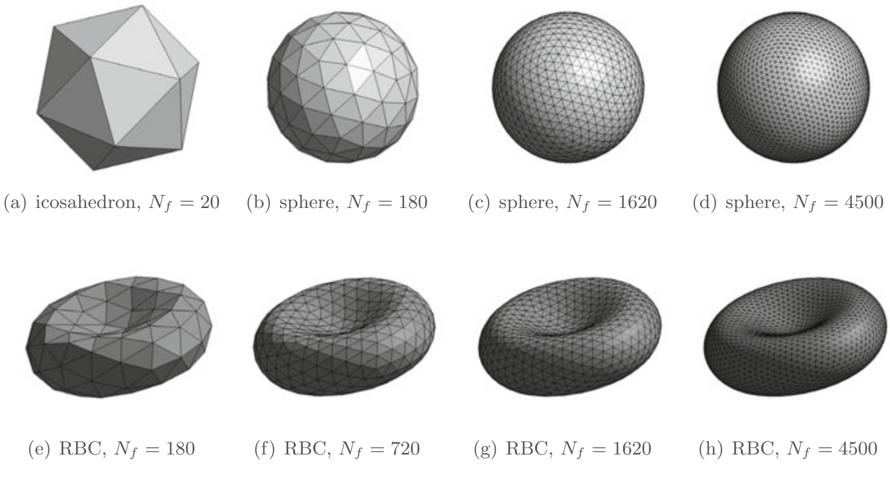


Fig. 8.2.: Membrane mesh generation by subdivision. Each face of the original icosahedron which is defined by three nodes n_i, n_j, and n_k (dark-gray) is subdivided into N^2 elements of equal size ($N = 2$ in this example). New nodes (here: n_l, n_m, n_n) are created and connected in such a way that N^2 faces of equal area are produced (dashed lines). Finally, the new nodes are radially shifted (out of the plane, black arrows) until they are located on the circumsphere of the icosahedron. The N^2 final faces are shown in light gray.



(a) icosahedron, $N_f = 20$ (b) sphere, $N_f = 180$ (c) sphere, $N_f = 1620$ (d) sphere, $N_f = 4500$

(e) RBC, $N_f = 180$ (f) RBC, $N_f = 720$ (g) RBC, $N_f = 1620$ (h) RBC, $N_f = 4500$

Fig. 8.3.: Examples of sphere and red blood cell meshes. The original icosahedron mesh with 20 faces ($N_f = 20$) is shown in (a). (b)–(d) Spherical meshes for various values of N_f. (e)–(h) Red blood cell meshes for various values of N_f.

8.3. Membrane mesh generation

There are basically two different ways to treat the Lagrangian mesh. First, one may introduce a structured, curvilinear grid with an intrinsic coordinate system (e.g., polar coordinates). Second, an unstructured grid (irregular decomposition of the surface into small patches whose connectivity must be specified explicitly) may be employed [210]. While the first approach usually leads to coordinate singularities at the poles, gradients have to be approximated in the second method. Both approaches are commonly used to describe deformable particles immersed in fluids. For example, Diaz et al. [211] and Lac et al. [212] have used structured meshes whereas Kraus et al. [76], Navot [192], and Ramanujan and Pozrikidis [191] have employed unstructured grids. In the present model, an unstructured mesh with triangular face elements is used as shown in fig. 8.3.

Tab. 8.1.: Properties of the sphere and red blood cell (RBC) meshes. The meshes for a sphere and a RBC for $N_f = 1620$ as shown in fig. 8.3(c) and fig. 8.3(g) are compared. A is the face area, l is the edge length (distance between neighboring nodes), φ is the interior angle in a face. The bar indicates the average of a quantity taken over the entire mesh, and σ denotes its standard deviation. The listed values are obtained for a spherical mesh with radius 7.5 and a RBC mesh with large radius 9 (both leading to $\bar{l} \approx 1$). All lengths and areas are given in lattice units.

quantity	sphere ($r = 7.5$)	RBC ($r = 9$)
range of face areas	$[0.40 \ldots 0.46]$	$[0.37 \ldots 0.50]$
average face area $\bar{A}$	0.43	0.44
average area deviation $\sigma_A/\bar{A}$	3.0%	7.5%
range of edge lengths	$[0.93 \ldots 1.08]$	$[0.80 \ldots 1.27]$
average edge length $\bar{l}$	1.01	1.01
average length deviation $\sigma_l/\bar{l}$	4.8%	9.2%
range of edge angles	$[53^\circ \ldots 72^\circ]$	$[41^\circ \ldots 84^\circ]$
average angle deviation $\sigma_\varphi/\bar{\varphi}$	8.0%	13.7%

Creating an unstructured mesh for a given surface (e.g., for a RBC or a sphere) is a non-trivial task. It is desirable to design meshes which are as homogeneous and isotropic as possible in order to minimize potential discretization artifacts. Feng and Michaelides [213] use a minimum potential approach, i.e., membrane nodes can move freely on the surface and interact via a repulsive pair potential. After some time, a node configuration is reached which corresponds to a local energy minimum. This configuration is then saved and used for later simulations. It is also possible to create triangular meshes directly with software tools like CGAL [214] or Gmsh [215]. Although simply obtained, these meshes seem to lack the desired homogeneity and isotropy (Krüger et al. [187]).

There is an approach to create high quality meshes for spherical surfaces, similar to that presented in [191]. One starts from a highly symmetric Platonic solid with N_f triangular faces. Here, an icosahedron with $N_f = 20$ is taken, cf. fig. 8.3(a). Each flat triangular surface element is then subdivided into N^2 equisized triangular sub-elements as indicated in fig. 8.2. The new nodes are radially shifted to the circumsphere of the icosahedron. This approach guarantees a surpassing homogeneity and isotropy of the mesh which is of large importance for the model, as discussed in section 8.6. Any closed triangular mesh with N_f faces has N_n nodes where $N_f = 2N_n - 4$. For a mesh created from an icosahedron as stated above, each node is member of five or six faces: The original 20 icosahedron nodes are member of five faces, all remaining nodes of six faces.

For an existing spherical mesh, it is straightforward to obtain the corresponding mesh for a RBC. The average shape of a RBC under physiological conditions can be parameterized by [216]

$$z(\varrho) = \pm\sqrt{1 - \left(\frac{\varrho}{r}\right)^2} \left(C_0 + C_2 \left(\frac{\varrho}{r}\right)^2 + C_4 \left(\frac{\varrho}{r}\right)^4\right). \tag{8.7}$$

The rotational symmetry axis of the RBC is along the z-axis ($\varrho = \sqrt{x^2 + y^2}$), and the parameters read $r = 3.91\,\mu\text{m}$ (large radius of a RBC), $C_0 = 0.81\,\mu\text{m}$, $C_2 = 7.83\,\mu\text{m}$, and $C_4 = -4.39\,\mu\text{m}$. Each point $\boldsymbol{x}'$ of the spherical mesh is then shifted to the point $\boldsymbol{x}$ of the RBC mesh according to $\boldsymbol{x}' = (x', y', z') \to \boldsymbol{x} = (x = x', y = y', z = z(\varrho'))$. Some exemplary meshes are shown in fig. 8.3.

Meshes are created in advance and saved as input data files for the simulations. Since the topology of the Lagrangian meshes and the connectivity of the membrane nodes never change, there is no need for any remeshing during a simulation. The high quality of the produced meshes can be inferred from fig. 8.3 and tab. 8.1: All face areas are of comparable size, and the edge length distribution has only a small width. Neither extremely small nor large face interior angles appear. This is important for the stability and a reduction of numerical artifacts as explained in section 8.6.

8.4. Benchmark test: single capsule in shear flow

This section bases on the investigations published in Krüger et al. [187]. In order to benchmark the combined algorithm for the fluid, the membrane, and their mutual coupling, a series of test simulations has been performed. The main intention of this benchmark is to understand (i) the numerical effect of the IBM interpolation stencil and (ii) the significance of the ratio $\bar{l}/\Delta x$ of capsule mesh and fluid lattice resolutions. Here, $\bar{l}$ is the average distance between neighboring membrane nodes (section 8.3), and Δx is the lattice constant (section 5.2). It is of primary interest to reveal how large the uncertainties for small or intermediate spatial resolutions are. Since a large number of particles shall be simulated eventually, the affordable resolution is restricted.

A single capsule with spherical rest shape and radius r is placed in the middle of a simple shear flow with external shear rate $\dot{\gamma}$ induced by moving walls at $z = \pm L_z/2$ with velocities $\pm u_\mathrm{w}$ along the x-axis (thus, $\dot{\gamma} = 2u_\mathrm{w}/L_z$). The system is cubic with size L_z^3. Due to the shear flow, the particle deforms and rotates about its stationary origin. After an initial transient, a steady configuration develops which is illustrated in fig. 8.4. The particle shape is a stationary, inclined ellipsoid whereas the membrane itself is 'tank-treading' with constant angular velocity ω about this shape. From this, two relevant observables can be extracted: the inclination angle θ and the deformation parameter D of the capsule. The inclination angle θ is defined as the angle between the flow axis and the largest semiaxis of the ellipsoid. The deformation parameter is

$$D := \frac{a - c}{a + c} \tag{8.8}$$

where a and c are the largest and smallest semiaxes of the deformed capsule, respectively. For a sphere, $D = 0$ holds. The angular velocity ω may also be computed (Krüger et al. [187]), but it is not in the focus of this benchmark.

Finding analytic solutions for problems involving deformable particles in external flows is not trivial. This makes it difficult to benchmark the IBM applied to deformable particles. However, for the above-mentioned problem, there exists an analytical solution [217] if (i) the constitutive elastic law of the membrane is known, (ii) the Reynolds number $\mathrm{Re} = \dot{\gamma} r^2/\nu$ of the flow can be neglected, (iii) the deformation of the capsule is small, $D \ll 1$, and (iv) the external flow is unbounded shear flow, i.e., the walls are far away, $L_z \gg r$. In the present case, the internal and external Newtonian fluids have the same density and viscosity. The capsule is only subject to the in-plane shear forces as described in section 7.1 because the theory does not include bending forces or non-local surface and volume forces.

When Skalak's constitutive law with $\kappa_\mathrm{S} = \kappa_\alpha$ is used (section 7.1), theory predicts $\theta_\mathrm{opp}/\pi := (\pi/4 - \theta)/\pi = 15\mathrm{Ca}/8$ for the inclination angle and $D = 25\mathrm{Ca}/4$ for the deformation parameter in steady state [187, 217] where

$$\mathrm{Ca} = \frac{\rho \nu \dot{\gamma} r}{\kappa_\mathrm{S}} \tag{8.9}$$

is the capillary number of the membrane, cf. eq. (8.5). For convenience, the angle $\theta_\mathrm{opp} \propto \mathrm{Ca}$ has been defined. It is the angle between the largest semiaxis of the capsule and the main diagonal in the shear plane. Obviously, in the limit of vanishing deformation, the inclination angle θ is $\pi/4$, i.e., 45°.

Preparations

Before the actual benchmark simulations can be performed, reasonable simulation parameter values have to be found. In particular, it has to be investigated (i) up to which Reynolds number the Stokes approximation is satisfied, (ii) which ratio of box size and particle radius H/r is

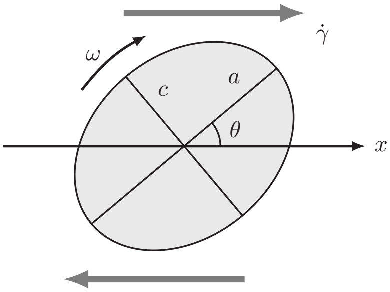


Fig. 8.4.: Tank-treading capsule. The capsule cross-section is shown in the xz-plane. It is deformed with major and minor semiaxes a and c. The inclination angle θ is taken between the major semiaxis and the x-axis (velocity direction of the external flow, dark gray arrows). The membrane rotates about its spatially fixed shape with angular velocity ω ('tank-treading').

required for *unbounded* shear flow, (iii) which deformation parameter D still is 'small', and (iv) which value of the LBM relaxation parameter τ is reasonable. The quadratic equilibrium distribution, eq. (5.6), is taken for all simulations in this section albeit the linearized equilibrium may have been used since the Reynolds number is small. The numerical fluid density ρ is set to unity throughout this section.

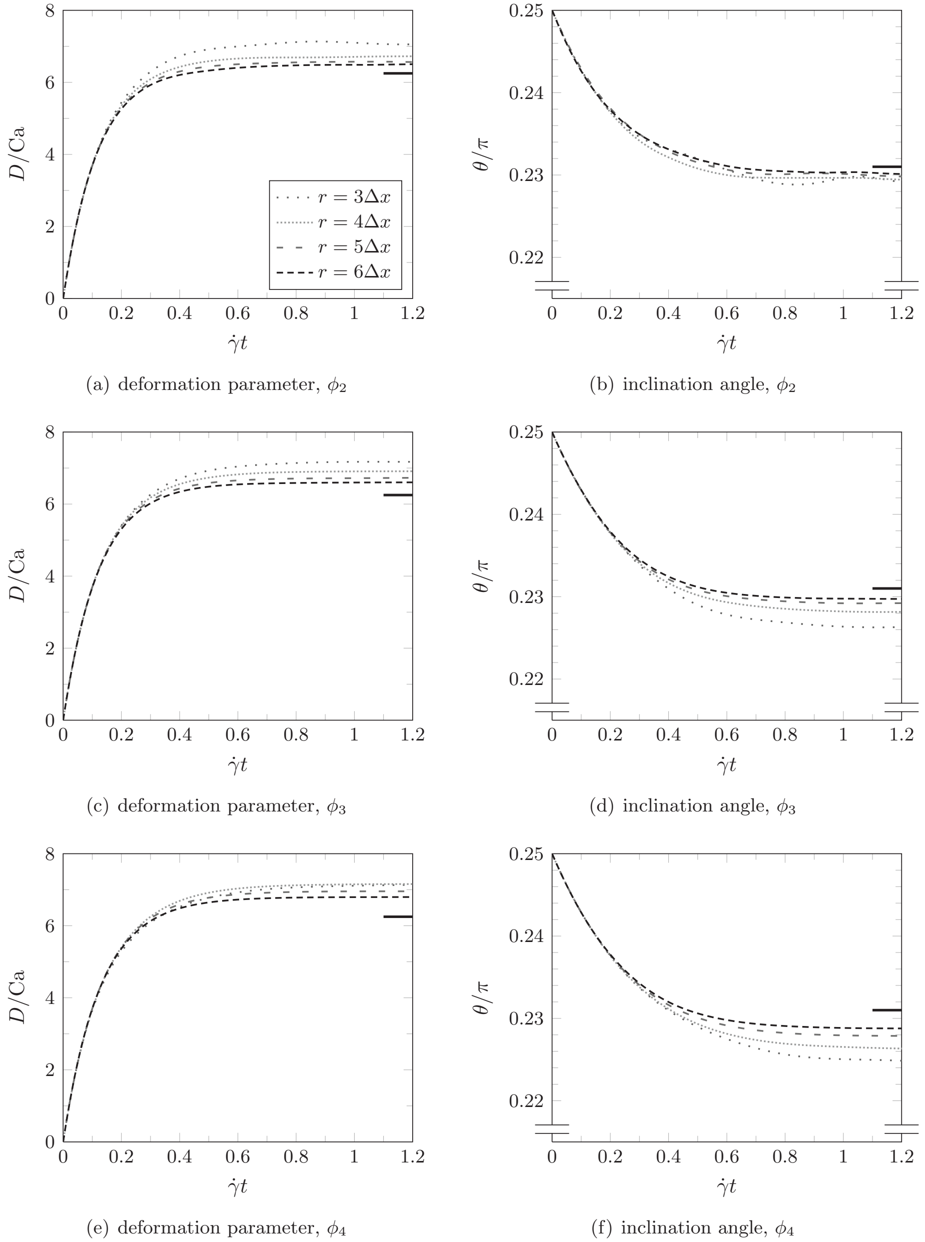
All of the preliminary tests have been thoroughly performed and discussed in Krüger et al. [187]. Only the results shall be given in the following. For these simulations, the 4-point interpolation stencil ϕ_4, eq. (6.8), has been used. The capsule radius is $r = 5\Delta x$, and the spherical, icosahedron-based mesh, cf. section 8.3, with $N_f = 1280$ faces ($\bar{l}/\Delta x = 0.76$) has been employed.

Although the transient behaves differently, the steady state value of the deformation parameter does not depend on the Reynolds number up to values of about 0.1 (larger values have not been tested). A relative box size of $L_z/r = 10$ was found to be sufficient to simulate unbounded shear flow. For Ca = 0.01 (corresponding to $D \approx 0.06$), the deformation is still 'small', and the linearized solution is sufficiently accurate for the problem description. Finally, it was found that the LBM relaxation parameter τ should not be larger than unity because higher values detrimentally affect the accuracy. Concluding, for the upcoming benchmark tests, the parameters have been fixed in the following way: Re = 0.02, $L_z/r = 10$, Ca = 0.01, $\tau = 1$. Having set these values, the only free parameters are the particle radius r in lattice units, the ratio $\bar{l}/\Delta x$ of membrane and lattice resolutions (and thus the capsule mesh size N_f), and the choice of the IBM interpolation stencil (ϕ_2, ϕ_3, or ϕ_4).

Convergence for fixed mesh resolution N_f

In this simulation series, the influence of the hydrodynamic resolution is tested alone, i.e., the capsule mesh resolution N_f is kept constant. This way, it is possible to study the effect of a non-constant ratio $\bar{l}/\Delta x$ by varying the capsule radius r. The employed mesh resolutions are $N_f = 320$ and 1280.

For the mesh with 320 faces, the capsule radius has been set to $r/\Delta x = 3$, 4, 5, and 6, corresponding to $\bar{l}/\Delta x = 0.90$, 1.20, 1.50, and 1.80, respectively. The results are shown in fig. 8.5. Although the mesh resolution N_f is small, it can be seen that the physics of the system is roughly captured. The accuracy of the solutions increases when the radius r becomes larger. Furthermore, it can be inferred that the values obtained with the interpolation stencil ϕ_2 are closest to the expected values. The larger the range of the IBM interpolation, the larger the deformation



(a) deformation parameter, ϕ_2

(b) inclination angle, ϕ_2

(c) deformation parameter, ϕ_3

(d) inclination angle, ϕ_3

(e) deformation parameter, ϕ_4

(f) inclination angle, ϕ_4

Fig. 8.5.: Deformation parameter D and inclination angle θ for a tank-treading capsule. D/Ca and θ/π are shown for varying ratios $\bar{l}/\Delta x$ for a mesh with $N_f = 320$ faces and interpolation stencils ϕ_2 in (a) and (b), ϕ_3 in (c) and (d), and ϕ_4 in (e) and (f). $r/\Delta x = 3$, 4, 5, and 6 correspond to $\bar{l}/\Delta x = 0.90$, 1.20, 1.50, and 1.80, respectively. The expected values are shown as short black lines (6.25 for D/Ca and 0.231 for θ/π). The legend in (a) is valid for all subfigures.

parameter D and the larger (smaller) the angle θ_{opp} (θ). As detailedly discussed in Krüger et al. [187], this can be interpreted in the following way: The finite-range interpolations due to the IBM lead to an apparent growth of the capsule radius r, which increases the apparent capillary number Ca, cf. eq. (8.5). Thus, the deformation is slightly larger than expected. Consequently, the deformation excess is smallest for the narrow interpolation stencil ϕ_2. When the radius r is increased, the relative width of the interpolation decreases and the numerical results approach the analytical solution. The mesh with 1280 faces has been tested for radii $r/\Delta x = 3$, 5, 7, and 9, corresponding to $\bar{l}/\Delta x = 0.45$, 0.75, 1.06, and 1.36, respectively. The plots are not shown here separately[1] because the results are qualitatively similar to those obtained for $N_f = 320$: The results become more accurate when (i) the IBM interpolation stencil has a smaller support and (ii) when the radius r increases.

It has been observed that the 2-point interpolation stencil ϕ_2 produces significantly wrong solutions when $\bar{l}/\Delta x > 2$ (data not shown). At this point, the spacing between neighboring mesh nodes is so large that fluid can penetrate the capsule membrane without experiencing the no-slip condition. For ϕ_3 and ϕ_4, a similar behavior at $\bar{l}/\Delta x = 2$ has not been observed. The probable explanation is the larger range of the interpolations, still keeping the fluid from passing through the membrane. The exact value of the mesh ratio $\bar{l}/\Delta x$ seems to play only a minor role as long as it is not too small (< 0.5) or too large (> 1.5). This indicates that the resolution of the membrane—at least for small deformations—does not require an extremely fine mesh, and the mesh ratio $\bar{l}/\Delta x$ can be safely chosen somewhere between 0.5 and 1.5 without significantly influencing the physical results. This is an important result since it allows a certain flexibility in setting up the simulations, especially in view of efficiency. However, it has been seen that the ratio of interpolation width (caused by the IBM interpolation stencil) and the radius r of the capsule should be as small as possible. This point will also be discussed in the following.

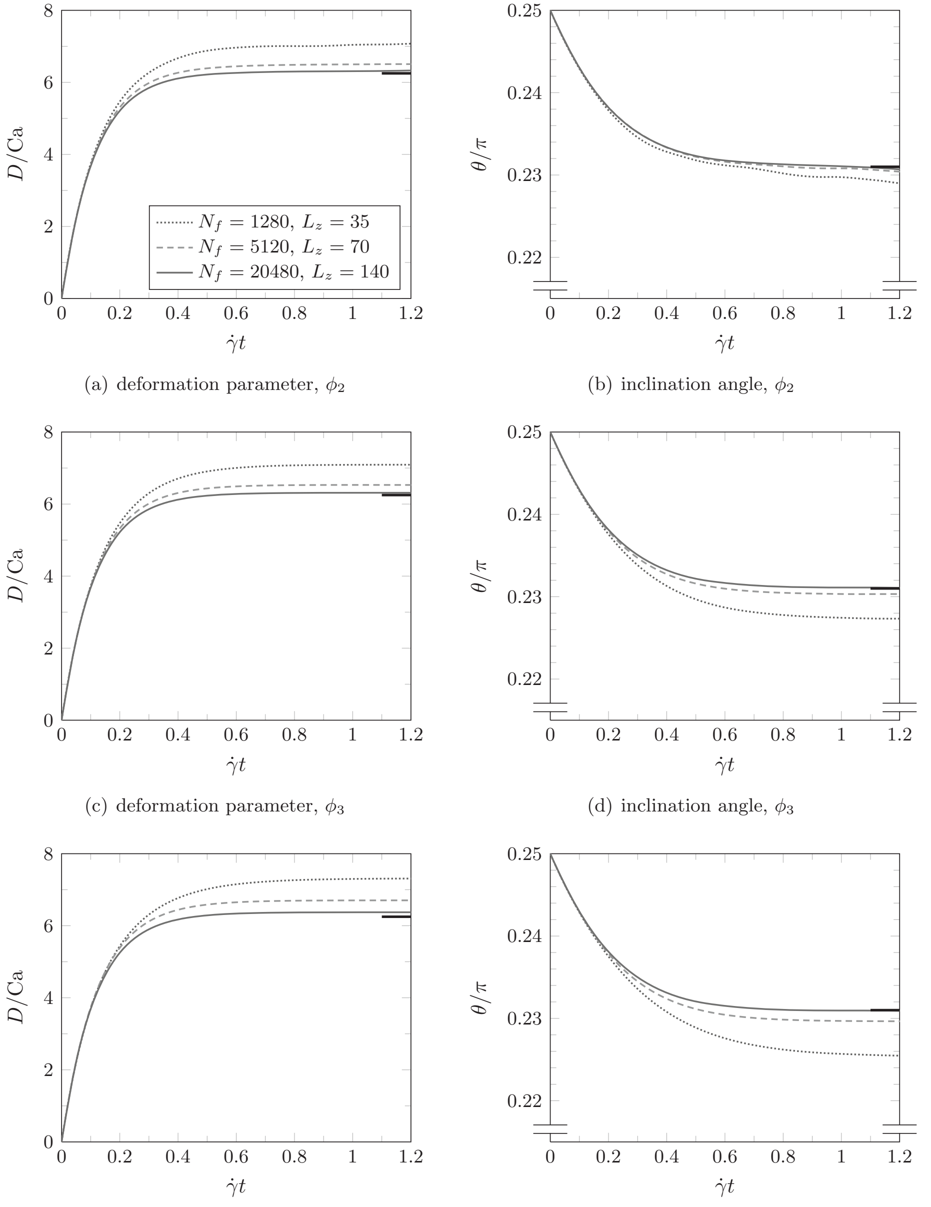
Convergence for fixed mesh ratio $\bar{l}/\Delta x$

In this second series, both the mesh and the hydrodynamic resolutions are increased by the same rate, i.e., the mesh ratio $\bar{l}/\Delta x$ is fixed. The mesh and hydrodynamic resolutions are $N_f = 1280$ and $L_z = 35$, $N_f = 5120$ and $L_z = 70$, and $N_f = 20480$ and $L_z = 140$, respectively. The mesh ratio is $\bar{l}/\Delta x = 0.53$ in all cases. It has to be noted that for $N_f = 320$, a mesh ratio of $\bar{l}/\Delta x = 0.53$ leads to quite unacceptable results since the capsule radius r becomes too small compared to the numerical width of the interpolation stencils. The results are shown in fig. 8.6. It is obvious that the steady state values of D/Ca and θ/π converge to their analytic values (6.25 and 0.231, respectively) when the resolution is refined. In order to quantify the results, the errors at $\dot{\gamma}t = 1.2$ (which is already in steady state) are listed in tab. 8.2 and shown in fig. 8.7. The convergence is close to second order. The only exception is the convergence of the inclination angle θ with the interpolation stencil ϕ_2. This deviation is caused by mesh degradation: Since the applied forces are merely in-plane forces, the capsule mesh starts to form ripples after some time. These ripples are caused by numerical artifacts which are most significant for ϕ_2 since this interpolation stencil is not as smooth as ϕ_3 or ϕ_4 (fig. 6.3). In-plane forces cannot counteract the ripples, and a bending force would be required to avoid them (section 7.2). However, as mentioned before, a bending force cannot be included since the analytical solution does not account for it. Fortunately, in other simulations within this work, the bending force is always included, and ripple formation has never been observed in these cases.

Conclusions

It has been found that the finite range of the IBM interpolations apparently increases the membrane radius, which causes numerical inaccuracies. Compared to these, fluid lattice and

[1] They can be found in Krüger et al. [187] instead.



(a) deformation parameter, ϕ_2

(b) inclination angle, ϕ_2

(c) deformation parameter, ϕ_3

(d) inclination angle, ϕ_3

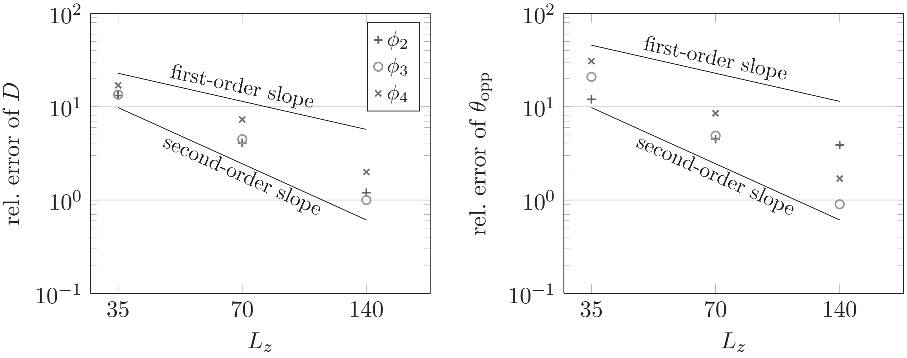
(e) deformation parameter, ϕ_4

(f) inclination angle, ϕ_4

Fig. 8.6.: Deformation parameter D and inclination angle θ for a tank-treading capsule. D/Ca and θ/π are shown for varying resolutions L_z and N_f for $\bar{l}/\Delta x = 0.53$ and interpolation stencils ϕ_2 in (a) and (b), ϕ_3 in (c) and (d), and ϕ_4 in (e) and (f). The mesh and fluid resolutions are $N_f = 1280$ and $L_z = 35$ (dotted lines), $N_f = 5120$ and $L_z = 70$ (dashed lines), and $N_f = 20480$ and $L_z = 140$ (solid lines), respectively. The expected values are shown as short black lines (6.25 for D/Ca and 0.231 for θ/π). The legend in (a) is valid for all subfigures.

Tab. 8.2.: Convergence of the capsule's deformation parameter D and the angle θ_{opp} for the interpolation stencils ϕ_2, ϕ_3, and ϕ_4. The relative deviations $\delta D/D^{\text{a}} := (D^{\text{s}} - D^{\text{a}})/D^{\text{a}}$ and $\delta\theta_{\text{opp}}/\theta^{\text{a}}_{\text{opp}} := (\theta^{\text{s}}_{\text{opp}} - \theta^{\text{a}}_{\text{opp}})/\theta^{\text{a}}_{\text{opp}}$ are shown at time $\dot{\gamma}t = 1.2$ (subscripts 's' and 'a' denote simulation and analytical, respectively). The convergence order α is taken from a fit to the functions $\delta D/D^{\text{a}}, \delta\theta_{\text{opp}}/\theta^{\text{a}}_{\text{opp}} \propto L_z^{-\alpha}$. For $\delta\theta_{\text{opp}}$ and ϕ_2, a meaningful convergence order could not be obtained due to mesh degradation. A graphic representation of this table is shown in fig. 8.7.

resolution		ϕ_2		ϕ_3		ϕ_4	
L_z	N_f	$\frac{\delta D}{D^{\text{a}}}$	$\frac{\delta\theta_{\text{opp}}}{\theta^{\text{a}}_{\text{opp}}}$	$\frac{\delta D}{D^{\text{a}}}$	$\frac{\delta\theta_{\text{opp}}}{\theta^{\text{a}}_{\text{opp}}}$	$\frac{\delta D}{D^{\text{a}}}$	$\frac{\delta\theta_{\text{opp}}}{\theta^{\text{a}}_{\text{opp}}}$
35	1280	13.2%	12.0%	13.5%	20.9%	17.0%	30.8%
70	5120	4.1%	4.5%	4.5%	4.9%	7.3%	8.5%
140	20480	1.2%	3.0%	1.0%	0.9%	2.0%	1.7%
convergence order α		1.7	N/A	1.9	2.2	1.5	2.1



(a) convergence of D for varying L_z

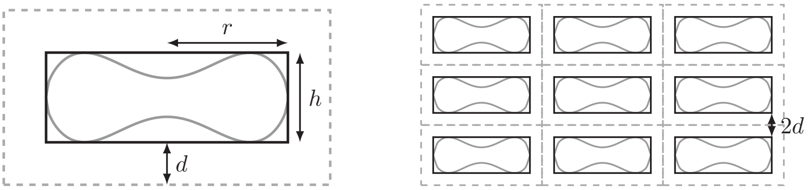
(b) convergence of θ_{opp} for varying L_z

Fig. 8.7.: Convergence of the deformation parameter D and the angle θ_{opp} for the interpolation stencils ϕ_2, ϕ_3, and ϕ_4. In (a), the relative error of the deformation parameter D is shown for increasing mesh and fluid resolutions with fixed $\bar{l}/\Delta x = 0.53$. The analog results for the relative error of the angle θ_{opp} are shown in (b). The data is taken from tab. 8.2. The legend in (a) is valid for (b) as well.

membrane meshing artifacts can be neglected, at least for small deformations. A large hydrodynamic resolution, i.e., a large value of $r/\Delta x$ is the only way to significantly reduce the numerical error. The computing time sets an upper bound for reasonable values of $r/\Delta x$. It is convenient to introduce a hydrodynamic radius r^* based on the true deformation D which is larger than the expected deformation. For the two-point interpolation stencil, the hydrodynamic radius is about $r^* \approx r + 0.3\Delta x$, which can be obtained from tab. 8.2. This observation will be important in chap. 10 where the simulation results for blood viscosity are compared to experimental data.

For a fixed mesh ratio $\bar{l}/\Delta x$, convergence to the analytical solution can be observed when r and N_f are simultaneously increased. The convergence order is close to two. This is convincing evidence that the presented numerical tool produces reliable results in the small deformation limit for which analytical solutions exist.

It has been seen that the resolution ratio $\bar{l}/\Delta x$ can be selected somewhere between 0.5 and 1.5 without compromising the physical results. This offers a flexible parameter choice with respect to the lattice and mesh resolutions. If $\bar{l}/\Delta x$ becomes smaller than 0.5, the IBM interpolations do not produce reliable results as will be explained in section 8.7. The present benchmarks are only valid for small deformations. For simulations with large deformations, however, the membrane mesh resolution must be sufficiently high to handle regions with large local curvature.



(a) safety layer for a single red blood cell

(b) sketch of the column structure of the cells

Fig. 8.8.: Column initialization of a red blood cell (RBC) suspension simulation. (a) Each RBC is bounded by a box with height h and length/width $2r$ (solid rectangle). The bounding box is surrounded by a safety layer of thickness d (dashed line). (b) The mutual distance between any two neighboring bounding boxes is $2d$.

8.5. Initialization of dense suspensions: distribution of the particles in the simulation box

It is a formidable task to initialize a simulation of a dense suspension of particles. Here, 'dense' means volume fractions above 30–40%. Due to the generally non-spherical shape of the particles, in particular for RBCs, it is hard to fill the volume efficiently. It is also of significant importance to initialize the particles randomly to promote mixing and to reduce the transient time at the beginning of the simulations. In the following, two approaches for the RBC positioning are presented: column initialization and random initialization. The discussions can be easily generalized to other particle shapes as well.

Initializing particle positions in columns

The simplest way to arrange the RBCs is to put them in simple rectangular columns with a given gap between them. One can show that the obtainable hematocrit is

$$\mathrm{Ht} = \frac{4r^2h}{(2r+2d)^2(h+2d)} \times 58\% \tag{8.10}$$

where r is the large radius of the RBC, $h \approx \frac{2}{3}r$ is its height, and d is a safety layer around each cell (fig. 8.8). The numerical value of 58% in eq. (8.10) is the ratio of the volumes of a RBC and the tight bounding box surrounding it (for a sphere, this value is 52%). The shortest distance between two adjacent cells is $2d$. This safety layer is important as will be discussed in section 8.6. The maximum hematocrit which can be reached this way is 58% when $d \to 0$. However, d should not fall much below 0.5 lattice units. For a typical radius of $r = 8$ and $d = 0.5$, the hematocrit would be 43% and close to the physiological value of about 45%. In principle, the hematocrit could be increased by going to higher resolutions (increasing r) while keeping d constant. This approach, however, is computationally expensive.

The above method is not a good choice for dense suspensions since it requires a long transient simulation time to get rid of the unphysical column information. In order to break the periodicity of the columns, each cell may be shifted and rotated by a random value. Due to the necessary presence of the safety distance, these random shifts and rotations must be small (a few 0.1 lattice nodes and a few degrees, respectively). Otherwise the cells would come too close or may even overlap. Additionally, the simulation box must have dimensions which are multiples of $(2r + 2d)$ and $(h + 2d)$, respectively, if the volume should be efficiently filled. This is not possible in general. Any deviation from the rectangular geometry would lead to a reduction of the available hematocrit. Another possible way to break the periodicity is to remove single cells

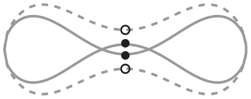


Fig. 8.9.: Self-intersection of a red blood cell (RBC) during initialization. If no internal fluid is used during the growth process, the top and bottom dimples (small circles) are not aware of each other, and the RBC may self-intersect (solid line, black circles). The undeformed shape is shown as comparison (dashed line, white circles). Introducing a repulsion force between the dimples solves this numerical problem.

from the lattice, i.e., to create gaps which accelerate randomization [87]. However, this reduces the hematocrit at the same time and is not recommendable for dense suspensions.

Due to the above-mentioned disadvantages, another approach must be considered for dense systems with Ht $> 30\%$.

Initializing particle positions randomly

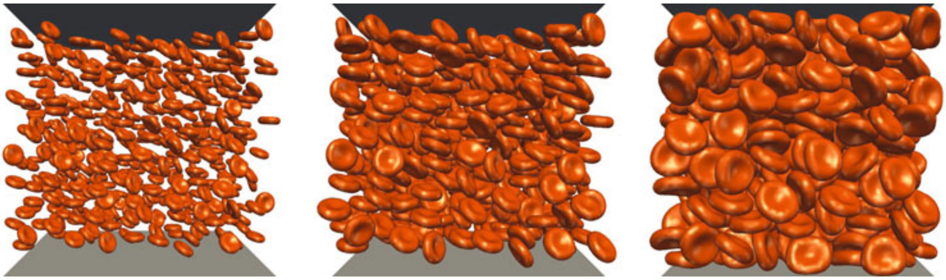
A promising way to position the RBCs randomly is pointed out by MacMeccan [194] and Clausen et al. [218]. The approach followed here is slightly different. It consists of two steps:

1. Position the cells with random locations and orientations. Use particles with 50% of their actual radius (initial hematocrit is one eighth of the desired value).
2. Grow the cells to their full size taking into account interaction forces to avoid jamming and overlap.

Randomly positioning particles with high volume fraction without overlap can be extremely time consuming since at some point most configurations do not allow of the addition of further particles. For this reason, the particles are first decreased in size and then randomly positioned throughout the simulation box. During positioning, it is checked whether any particle overlaps with a solid wall or any other particle which has already been created. Here, the cells are treated as spheres, and only the center-to-center distance is used for checking. If the new particle overlaps with a wall or any other particle already positioned, a new position vector is generated, and it is tested again. The positioning step is completed almost instantaneously since the effective volume fraction at this stage is typically only 5–8% and overlaps are rare.

The growing step is more challenging, and there are different approaches available. For the present thesis, the volume of the cells is increased from its reduced value V_0 (which is typically $V/8$) by a constant value V_+ at each time step. At each growth step, the membrane interaction forces as described in section 8.7 are evaluated. This way, overlap of the particles during growth is circumvented. The basic idea is to reach a high volume fraction by letting the particles deform during growth, i.e., they are allowed to change their shape in order to fill the available volume more efficiently. This approach differs from that presented in [194, 218] where the cells are rigid during growth. In order to maintain smooth membrane shapes, also the shear and bending forces (cf. sections 7.1 and 7.2) are computed at each growth step. It has turned out that 5000–10000 growth steps are sufficient for suspensions with volume fractions up to 70% and average particle radii of about 8 lattice units.

Hydrodynamic interactions are not required in this (unphysical) stage, and LBM and IBM are not used. Instead, a simple molecular dynamics model is employed for the integration of the equations of motion. The velocity and position of each node i at growth step $n + 1$ is computed



(a) after random positioning (b) after half of the process (c) after completed process

Fig. 8.10.: Random initialization of a dense red blood cell (RBC) suspension simulation. 520 RBCs with Ht = 46% are grown from their initial to their final size (initial radius is 50% of final radius). The cells are shown (a) directly after random positioning (Ht = 6%), (b) after half of the growth process (Ht = 23%), and (c) directly after the completed growth process.

from its velocity and position at step n and the total force it experiences at step $n+1$,

$$\begin{aligned} \dot{\boldsymbol{x}}_i(n+1) &= \dot{\boldsymbol{x}}_i(n) + \frac{1}{m}\boldsymbol{F}_i(n+1), \\ \boldsymbol{x}_i(n+1) &= \boldsymbol{x}_i(n) + \dot{\boldsymbol{x}}_i(n+1). \end{aligned} \tag{8.11}$$

This simplified dynamics has proven to be sufficient for the growth process. A typical value for the mass is $m = 100$ (lattice units). This value can be chosen arbitrarily as long as the growth process is stable. It is shown in fig. 8.10 for a RBC suspension with 46% volume fraction.

In order to improve the quality of the growth process, an additional force is required. Areas of the same membrane which are not directly connected are not aware of each other, i.e., the two dimples of a single RBC are mutually invisible. The reason is the locality of the shear and bending forces. In the full simulations, the dimples of a cell cannot touch each other since the interior fluid prevents them from doing so. In the growth process, due to the absence of the fluid coupling, the dimples may self-intersect (fig. 8.9). For that reason, an additional force is used during growth which introduces a repulsion of the nodes in the top and bottom nodes near the dimples. The exact form of this force is not relevant as long as it maintains the desired safety distance between the dimples.

8.6. Limitations and restrictions of the numerical model

In this section, the restrictions and limitations of the numerical model are presented. The focus lies on the issues related to the present work rather than on an exhaustive overview. The section is divided into three parts: The problems related to the LBM are briefly presented, followed by a more elaborate discussion about the IBM and some statements about the membrane model. The intention is to identify the simulation parameters and physical applications for which the model produces reliable results.

Restrictions of the lattice Boltzmann method

For the simulation parameters chosen in the present thesis, there are basically no numerical problems related to the LBM. Due to the small Reynolds numbers, the relaxation parameter τ is always chosen sufficiently large ($\tau \approx 1$), and stability problems do not occur. The lattice Mach number is usually kept below 0.1. A noticeable disadvantage of the LBM is the coupling of the discretizations of position and velocity space. As discussed in sections 5.5 and 8.2, the time step is already set when the Reynolds number and the spatial resolution have been chosen for a given value of τ. For small Reynolds number flows, this usually leads to extremely small time steps and long integration times. Hence, a redefinition of the time step as discussed in section 8.2 may be necessary.

Restrictions of the immersed boundary method

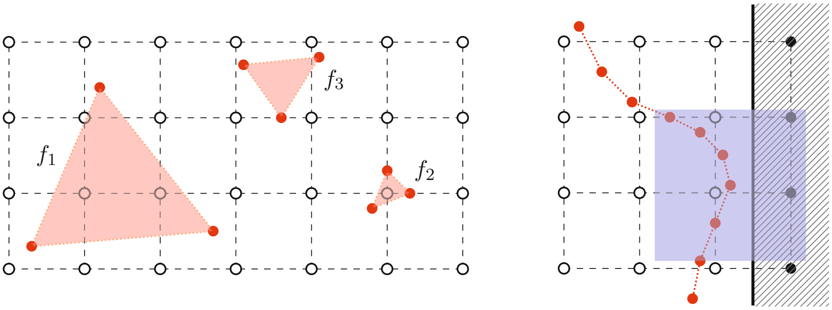
The simplicity of the IBM (both the concept and implementation) does not come without a price, and the devil is in the details. There are a few issues which have to be considered when the IBM is used for the simulation of dense suspensions.

It is commonly observed that the IBM does not properly work when membrane nodes (either belonging to the same membrane or to distinct membranes) come too close (e.g., [89]). The reason is the indirect position update: Membrane nodes are advected by the ambient fluid velocity only, and interpolations of the velocities are required. If the distance between two membrane nodes is significantly smaller than the lattice constant Δx, velocity gradients do not survive the interpolation, and the nodes move with similar velocities. Consequently, it is not possible to provide a large velocity gradient for nearby membrane nodes, and these nodes cannot be separated in a realistic time, even if there is a large repulsion force between them. Membranes, thus, can stick together, in the worst case for all times. This reasoning is explained in more detail in section 8.7. It is obvious that the above limitation is more visible in dense suspensions than in dilute systems since the membrane node density is larger then. A high mesh quality significantly helps to minimize this problem: If the mesh is designed in such a way that the node-node distance distribution is narrow and centered at about Δx (cf. section 8.3), nodes within a given membrane usually never come too close. It remains to deal with the problem of membrane-membrane collisions, as discussed in more detail in section 8.7.

The above problem gives a somewhat natural answer to the question how to choose the ratio of Eulerian grid and Lagrangian mesh resolutions. In principle, the average node-node distance in a membrane, $\bar{l}$, can be set independently of the lattice constant Δx. It is expected that there is a given ratio $\bar{l}/\Delta x$ for which the numerical model works most reliably. On the one hand, if $\bar{l}/\Delta x$ is too small, the nodes stick together since the Eulerian grid cannot provide a sufficient resolution for the velocity field (see above and section 8.7). On the other hand, too large a value for $\bar{l}/\Delta x$ will lead to 'holes' in the membrane, cf. fig. 8.11(a). Fluid may then penetrate the membrane without being forced by nearby membrane nodes. The no-slip condition will then be violated and the numerical results become less accurate. Thus, the distance between neighboring membrane nodes should not be larger than the range of the force spreading defined by the width of the interpolation stencils in section 6.3. Based on these considerations and the results in Krüger et al. [187], the average node-node distance is set to $\bar{l} \approx \Delta x$ in all upcoming simulations if not otherwise stated.

A problem similar to the node-node behavior at small distances appears when nodes come too close to a solid wall. If a wall node is within interpolation range of a membrane node, on the one hand, force is spread to the wall and therefore 'lost' with respect to the fluid. On the other hand, it is not generally clear which velocity the wall should be assigned to in order to obtain the correct interpolated node velocity, cf. fig. 8.11(b). Bagchi [88] has circumvented the problem by assuring that the cells always have a minimum distance from the walls. Feng and Michaelides [79] introduce a repulsion force so that nodes never come too close to the wall. A similar path is also followed in the present thesis (section 8.7). It is noteworthy that analog problems also arise in other numerical methods where Eulerian and Lagrangian systems are combined (e.g., [85, 219]).

The time steps for the LBM and the membrane update are the same since they are directly



(a) ratio of mesh resolutions

(b) nodes close to a wall

Fig. 8.11.: Problems related to the immersed boundary method. (a) Ratio of mesh resolutions: Three faces (red triangles) with different node-node distances $\bar{l}$ are shown (face f_1: $\bar{l} = 2.4\Delta x$, face f_2: $\bar{l} = 0.5\Delta x$, face f_3: $\bar{l} = 0.9\Delta x$). While the node-node distances for face f_1 are so large that lattice nodes (white circles) may not feel a membrane force at all, the nodes in face f_3 are so close that the interpolation becomes inaccurate. For face f_3, both detrimental effects are minimized. (b) Nodes close to a wall: If nodes (red circles) of a membrane (curved dotted line) are in direct wall proximity, force has to be spread to wall nodes (black circles within square region). Conversely, the velocity interpolation requires velocity information from the wall.

coupled via IBM, cf. section 8.1. Polymers immersed in a fluid and subject to thermal fluctuations may be simulated via LBM, molecular dynamics (MD), and viscous coupling [182]. In this case, a separation of time steps for the polymer (MD) and the fluid (LBM) is required. It is not unusual to have more than 50 MD time steps within one LBM time step. In the present case, this is not necessary since thermal fluctuations are not considered and the membrane motion is overdamped. Additionally, it is often stated that the explicit time integration of the IBM leads to numerical instabilities, and improvements have been proposed [220, 221]. This problem, however, is usually observed in systems of rigid particles. These particles may be modeled as deformable objects with high rigidity. Any deformation is penalized by a strong response force trying to maintain the rigid shape, which can lead to unphysical oscillations. On the contrary, in the present case, the particles are designed to be deformable, i.e., the membrane moduli are small (except for the surface and the volume conservation, see below), and large deformations lead to manageable force magnitudes.

Similar to nearly all Eulerian-Lagrangian coupling methods, the IBM formally is only first order accurate in space when sharp interfaces (e.g., membranes) are simulated [78]. Peng and Luo [222] observed that that the IBM has a second-order convergence behavior for LBM fluid flow around a rigid cylinder in 2D. Contrarily, Zhu et al. [223] and Caiazzo and Maddu [224] find only first order convergence in space for the velocity field. In section 8.4, it has been observed that the IBM can be of second-order accuracy when the deformation of a capsule in shear flow is considered (additional information also provided in Krüger et al. [187]). One of the most important findings is that the hydrodynamic radius of the particle is larger than expected from the input mesh. The finite-range interpolations of the IBM are believed to be the reason for this behavior. Generally speaking, the accuracy of the IBM seems to depend strongly on the application and observables. Due to the complexity of the mathematical basis of the coupled IBM-LBM system, a thorough analysis of the spatial accuracy is not provided here, and the reader is referred to the literature, e.g., [224].

Even if the Navier-Stokes solver provides a divergence-free velocity field, the velocity field interpolated by IBM will not be divergence-free in general (cf. section 7.4). For this reason, one may track the membrane volumes explicitly and correct for volume drifts, e.g., by introducing a volume restoration force as in eq. (7.15). In the present case, the resulting volume deviations are below 1% and can be neglected.

Le and Zhang [225] reported an unphysical boundary slip velocity in a coupled IBM-LBM system when the relaxation parameter τ is significantly larger than unity. Practically, it should be < 2. Indeed, a similar observation has been pointed out in section 8.4 (and, in more detail, in Krüger et al. [187]). For that reason, the LBM relaxation parameter in the present thesis never exceeds unity except for benchmarking. This is also in line with the restrictions given by the bounce-back boundary conditions in section 5.4.

Restrictions of the membrane model

The membrane consists of flat triangular face elements which remain flat even after deformation. If the local curvature radius becomes comparable to the extension of a face, the mesh resolution is not sufficient any more, and the deformation state is not described in a reliable way. This sets an upper bound for the achievable shear rates which can be simulated. The smaller the membrane resolution (the less faces used), the smaller the shear rate which can be simulated. Moreover, higher particle volume fractions detrimentally decrease the maximum shear rate since membrane-membrane interactions lead to additional deformations. Numerically, too strong a deformation can lead to the collapse of faces or the folding of pairs of faces. Due to the shortcomings of the IBM, such an unphysical deformation usually cannot be reversed, and the simulation becomes either inaccurate or it even crashes[2]. Also Pozrikidis [84] and MacMeccan [194]—using different membrane models—observed that a coarse membrane resolution leads to numerical problems when large deformations are simulated. Local fluid velocities above ≈ 0.1 may lead to instabilities since this translates to a position shift of ≈ 0.1 (about 10% of the extension of a face) in one time step. Unphysical oscillations may therefore emerge and destroy the simulation in the worst case. The maximum velocity restriction ($u < 0.1$) is in line with the small Mach number premise of the LBM.

As explained before, the penalty moduli for the surface and volume conservations, κ_A and κ_V, (cf. sections 7.3 and 7.4) are not allowed to become too large. However, as commonly stated in the literature [31, 77, 79], the exact values of κ_A and κ_V are not relevant as long as the surface and volume are sufficiently well conserved (within 1%, say). It should be noted that the surface and volume deviations become smaller when the spatial resolution is increased. Reasonable values for κ_A and κ_V are not obvious a priori, but they can be inferred from a few test runs.

Due to the discretization of the mesh, the membrane forces may suffer from meshing artifacts which can be caused by extremely small face areas since these enter the bending force in the denominator, cf. appx. C.2. Membrane faces, therefore, should be of comparable size. Additionally, faces with large interior angles are of disadvantage because they cannot accurately capture the shear state of the faces. For this reason, it is desirable to have equisized faces with shapes close to equilateral triangles. This is ensured by the mesh generation algorithm presented in section 8.3. As the connectivity of the mesh is never changed in the present model, nodes cannot 'diffuse' in the mesh, i.e., the mesh has no fluidity properties. Physically, this means that the particles always require a finite *elastic* shear resistance. Vesicles—which are basically 2D fluids—cannot support elastic in-plane shear stresses, and membrane nodes would eventually diffuse. Consequently, the current model supports the simulation of capsules rather than vesicles.

[2] The reason for such a crash is the explicit time integration in combination with large forces which may drive individual membrane nodes out of the numerical grid.

Conclusions

Simulations at arbitrarily small shear rates are only restricted by the available computing time. For high shear rates, the mesh resolution and the lattice Mach number limitation define upper bounds. The present model is particularly suited for small Reynolds number simulations where $\tau \approx 1$ can be selected. The resolutions of the Lagrangian and Eulerian meshes should be similar, $\bar{l} \approx \Delta x$, in order to produce optimum results. High volume fraction simulations may be problematic due to the IBM weakness for small node-node separations. A repulsion force may be incorporated to remove the numerical problem. Yet, the consequences of the repulsion force on the physical results should be monitored. Due to the IBM interpolations, one has to consider potential accuracy issues which have to be checked. The current membrane model should only be applied to capsules (membranes with finite elastic shear resistance) and not to vesicles since the mesh connectivity is fixed.

8.7. Interactions between nearby membranes

It is a well-known problem in lattice computations that the hydrodynamic description breaks down if the distance d between two particles becomes smaller than a length comparable to the lattice constant Δx (e.g., [73, 79, 154, 226]). Even worse, due to the discrete time stepping, particles may eventually overlap if no countermeasures are taken.

For spherical particles, lubrication correction forces can be computed analytically and may be added to re-introduce the correct hydrodynamics at a sub-grid length scale [73, 110]. This approach, however, is difficult to employ in the present model since the particles generally have non-spherical shapes. Additionally, the IBM concept reveals a numerical weakness when particle nodes have a distance much smaller than the lattice resolution Δx (see below). For that reason, a pragmatic method is used in order to maintain a safety distance between the particles: a repulsion force for nodes being too close to each other.

Before details of the appropriated repulsion force are given, a qualitative analysis of the IBM at short distances should be performed. In the framework of the IBM, particle nodes can only translate by moving along with the ambient fluid, and velocity interpolations play an important role, cf. section 6.3. If two particle nodes are close to each other (d significantly smaller than Δx), the nodes basically see the same ambient velocity field, and it becomes more and more difficult to separate the nodes once they have approached each other. In order to understand this, it is instructive to imagine a 1D case with two particle nodes i and j at positions x_i and x_j close to each other, with a mutual distance $x_j - x_i = d < \Delta x$, cf. fig. 8.12. Both nodes are located between two adjacent fluid lattice nodes at positions X_l and X_r with $X_r - X_l = \Delta x$. For this particular case, the velocities of the two nodes can be written as, cf. eq. (6.6),

$$\begin{aligned} \dot{x}_i &= \left(1 - \frac{x_i - X_l}{\Delta x}\right) u(X_l) + \left(1 - \frac{X_r - x_i}{\Delta x}\right) u(X_r), \\ \dot{x}_j &= \left(1 - \frac{x_j - X_l}{\Delta x}\right) u(X_l) + \left(1 - \frac{X_r - x_j}{\Delta x}\right) u(X_r) \end{aligned} \tag{8.12}$$

when ϕ_2, eq. (6.11), is used as interpolation stencil. It is straightforward to show that the relative velocity of the two nodes is

$$\dot{x}_j - \dot{x}_i = \frac{d}{\Delta x} \left(u(X_r) - u(X_l)\right). \tag{8.13}$$

In other words, due to the linear interpolation, the velocity difference which can bring the nodes apart from each other is reduced by a factor $d/\Delta x$ if the particles are close. Although not exactly

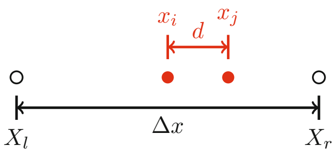


Fig. 8.12.: Immersed boundary method and small node distances. A simplified problem in 1D is shown: Two nodes (red circles) at positions x_i and x_j and mutual distance d are located between two adjacent fluid lattice nodes (white circles) at X_l and X_r with distance Δx.

the same, the results for the other interpolation stencils, ϕ_3 and ϕ_4 in eq. (6.8) and eq. (6.10), are comparable. A qualitatively similar result is also expected in 2D and 3D.

In order to hinder nodes to approach each other too closely, a repulsion force may be employed. It should act along the connection line of the nodes and fulfill $F_j = -F_i$ (in the 1D case) in order to obey momentum and angular momentum conservation. Applied to the above example, it is easy to see that the resulting force acting on lattice node X_l is

$$f(X_l)\Delta x = \left(1 - \frac{x_i - X_l}{\Delta x}\right) F_i + \left(1 - \frac{x_j - X_l}{\Delta x}\right) F_j = \frac{d}{\Delta x} F_i, \tag{8.14}$$

cf. eq. (6.5). The force acting on lattice node X_r is the same with different sign[3]. At this point, the important observation is that the effective repulsion force $f(X_l)\Delta x$ which survives the IBM spreading is reduced by a factor $d/\Delta x$ compared to the input force F_i. For that reason, the repulsion force should increase stronger than linearly when the node distance d goes to zero.

A simple repulsion force is a power-law of the distance d with a cut-off at a finite length R,

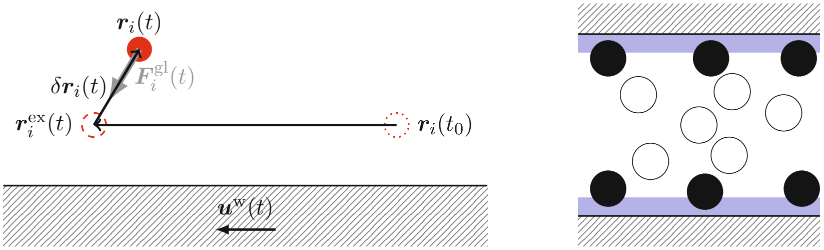
$$\boldsymbol{F}_{ij}(\boldsymbol{d}_{ij}) = \begin{cases} -\kappa_{\text{int}} \left[\left(\frac{\Delta x}{d_{ij}}\right)^k - \left(\frac{\Delta x}{R}\right)^k \right] \frac{\boldsymbol{d}_{ij}}{d_{ij}} & \text{for} \quad d_{ij} < R, \\ 0 & \text{for} \quad d_{ij} \geq R \end{cases} \tag{8.15}$$

where $\boldsymbol{F}_{ij} = -\boldsymbol{F}_{ji}$ is the force acting on node i given the distance $\boldsymbol{d}_{ij} = \boldsymbol{x}_j - \boldsymbol{x}_i$ between node i and a nearby node j. The power k has to be larger than unity in order to overcome the interpolation artifact in eq. (8.14). The magnitude of the repulsion is controlled via the parameter κ_{int}. The unphysical parameters k, R, and κ_{int} have to be chosen in such a way that (i) the force is only active at distances as small as possible, that (ii) the separation of nodes does not become too small ($0.5\Delta x$, say), and that (iii) the repulsion force does not lead to numerical instabilities. In the present thesis, $k = 2$ and $R = \Delta x$ are used. Feng and Michaelides [79], following a similar idea, state that details of the repulsive force do not influence the macroscopic behavior.

It has to be stated that a repulsive force increases the apparent radius of the particles if the particles are in close contact. On the one hand, as long as the particles have a distance larger than the interaction range, the interaction force is deactivated. On the other hand, for dense suspensions, particles are often in close proximity, and the repulsive force is acting. Still, the range of this force is comparable to the additional effective radius caused by the IBM interpolations (cf. section 8.4). The interaction force as defined in eq. (8.15), therefore, can be considered a 'natural' force avoiding overlap of the hydrodynamic particle volumes. It is an open question up to which volume fraction this interpretation is still valid. For large volume fractions, particles have to deform even in the absence of flow. However, in a quiescent situation, hydrodynamic effects are absent whereas the repulsion forces are still active.

It is important to stress that—in the present model—only nodes in different membranes interact via this kind of repulsion force. By design, the average distance of neighboring membrane nodes

[3] This must be the case since the IBM interpolation stencils obey momentum conservation.


$\boldsymbol{r}_i(t)$
$\delta\boldsymbol{r}_i(t)$
$\boldsymbol{F}_i^{\mathrm{gl}}(t)$
$\boldsymbol{r}_i^{\mathrm{ex}}(t)$
$\boldsymbol{r}_i(t_0)$
$\boldsymbol{u}^{\mathrm{w}}(t)$

(a) Computation of the penalty force

(b) Stick condition

Fig. 8.13.: Implementation of wall roughness. (a) A membrane node i initially located at position $\boldsymbol{r}_i(t_0)$ (dotted circle) close to a wall (dashed region) is expected to move along with the wall to position $\boldsymbol{r}_i^{\mathrm{ex}}(t)$ (dashed circle) after time $t - t_0$, but it is generally found at position $\boldsymbol{r}_i(t)$ (solid circle) instead. The stick force $\boldsymbol{F}_i^{\mathrm{gl}}(t)$ is proportional to the distance $\delta\boldsymbol{r}_i(t) = \boldsymbol{r}_i(t) - \boldsymbol{r}_i^{\mathrm{ex}}(t)$, cf. eq. (8.17). (b) After growing the particles to their full size, all particles which have at least one mesh node in the 'glue' region (light blue) near the walls (dashed regions) are stuck to the wall. Stuck particles are shown as black, free particles as white circles.

in the undeformed state is close to Δx, cf. section 8.3. Even for strong deformations, nodes in a membrane usually do not come closer than half a lattice spacing. This behavior is caused by the in-plane tensions related to the incompressibility constraints of the membrane, cf. section 7.1. The repulsion force, therefore, is only switched on when the distance between two distinct membranes falls below the range R, which is virtually never the case if the suspension is dilute. In these cases, hydrodynamic interactions usually maintain a given distance between the membranes. If the suspensions become denser and denser, the mutual distances between membranes decrease, and, eventually, numerical problems would arise without a repulsion force.

Nodes which come too close to a wall are subject to an additional force introducing a repulsion away from the wall. The functional form is the same as in eq. (8.15) with the difference that the distance $\boldsymbol{d}_{ij}$ is replaced by the shortest (i.e., normal) distance to the wall. Since the wall is not allowed to move in normal direction, it is not necessary to add the repulsion force to the wall. Such a force would merely increase the wall pressure which is not of significance in the present thesis.

8.8. Wall slip and roughness

In experiments, the rheological properties of a suspension strongly depend on the structure of the shearing surfaces. It is known that smooth surfaces promote a slip behavior which reduces the viscosity of the suspension at small shear stresses [227, 228]. The reason is a thin fluid lubrication film developing between the surface and the first layer of suspended particles. If the shear stress is close to or smaller than the yield stress of the suspension, the entire velocity gradient can drop off in the fluid film, resulting in locally high shear rates. Even though the suspension in the bulk is jammed, the overall suspension can flow, and a yield stress may be hidden from the observer. If slip effects are not desired, one may either subtract the slip velocity (e.g., by measuring the velocity profile locally) or prevent slippage in the first place.

The development of the lubrication layer can be avoided if rough (on the particle length scale) surfaces are used. For the present model, the wall roughness may be realized in the following way: The particles are randomly initialized in the available volume as described in section 8.5. After this, all particles in a given vicinity of the wall are 'stuck to the wall'. In order to do so, the nodes of these membranes experience an additional force trying to keep the nodes in a position where they would be located if they moved along with the wall. This concept is visualized in fig. 8.13(a). At the beginning of the simulation, after the membranes have grown to their full size and membranes close to the walls have been identified, each node of these membranes remembers its initial position $\boldsymbol{r}_i(t_0)$. At a later time, $t > t_0$, the wall has moved by a certain distance and with it the expected node position $\boldsymbol{r}_i^{\mathrm{ex}}(t)$ in such a way that

$$\boldsymbol{r}_i^{\mathrm{ex}}(t) = \boldsymbol{r}_i(t_0) + \int_{t_0}^{t} \mathrm{d}t' \, \boldsymbol{u}^{\mathrm{w}}(t') \tag{8.16}$$

where $\boldsymbol{u}^{\mathrm{w}}$ is the wall velocity. Generally, the node will be at a position $\boldsymbol{r}_i(t)$ instead, with relative distance $\delta\boldsymbol{r}_i(t) = \boldsymbol{r}_i(t) - \boldsymbol{r}_i^{\mathrm{ex}}(t)$ to its expected position. A Hookean penalty force for this node is defined according to

$$\boldsymbol{F}_i^{\mathrm{gl}}(t) = -\kappa_{\mathrm{gl}} \delta\boldsymbol{r}_i(t). \tag{8.17}$$

The strength of the force is given by the 'glue modulus' κ_{gl}. This force is added to the other forces acting on the nodes. An opposite force with the same magnitude is exerted on the corresponding wall in order to find the correct wall shear stress afterwards. This way, momentum conservation is not violated.

In practice, a membrane is stuck to a wall if at least one of its nodes has a wall distance not greater than the large radius r of the membranes. However, other conventions may be used as well. This can be interpreted as a glue film of thickness r covering both walls, cf. fig. 8.13(b). The above method bases on the approach presented by Feng and Michaelides [79] who use a Hookean penalty force to maintain the shape of the pseudo-rigid particles.