Computational haemodynamics of small vessels using the Moving Particle Semi-implicit (MPS) method

Alberto M. Gambaruto

alberto.gambaruto@bsc.es
Department of Computer Applications in Science and Engineering (CASE),
Barcelona Supercomputing Center, Barcelona, Spain

Abstract

The simulation of whole blood stands as a complex multi-body problem. The Moving Particle Semi-implicit method, a Lagrangian particle method to solve the incompressible Navier-Stokes (NS) equations, is developed to perform simulations in complex periodic domains. Red blood cells are modelled using the spring network approach, that act as body force terms in the NS equations. Detailed presentation and derivation of both the MPS method and different spring network models is given. An adaptive time step and an implicit scheme are adopted, improving the stability and overall computational efficiency.

The findings from the simulations show evidence that in proximity to the vessel wall, the red blood cells expose a larger surface area by orientation and deformation, due to the presence of a high velocity gradient. The greatest membrane internal stresses occur in the core region of the flow. The intra-cell interaction is driven by a complex flow field that can be visualised in a Lagrangian framework, and highlights vortex structures in the wakes and in between the cells. The stresses the blood exerts on the vessel wall is influenced by this complex flow field and by the presence of red blood cells.

Keywords:
micro-circulation, moving particle semi-implicit (MPS) method, spring network, red blood cells

1. Introduction

Computational haemodynamics in large vessels relies on solving the incompressible Navier-Stokes equations, often employing non-Newtonian rheological models. Such simulations serve to understand the large scale description of the flow field, and have supported studies including hemolysis [1], thrombosis [2], convective transport and vascular remodelling [3], amongst many others. In actual fact, blood is composed of plasma (~55% by volume) that is mostly water, and holds in suspension several types of cells: red blood cells (erythrocytes), white blood cells (leukocytes) and platelets (thrombocytes). The proportion of blood occupied by red blood cells is referred to as the hematocrit (HCT), and is normally about 40-45% by volume [4]. As such, when simulating blood flow in micro-sized vessels, the fluid medium can no longer be adequately described through a continuum model (homogenisation), but instead requires the consideration of each individual cell in suspension in order to accurately model and understand the physical and biological processes [4, 5, 6].

Modelling and simulating blood flow at the micro-scale [7, 8, 9], has important implications in understanding, for example, thrombosis [10] in far greater detail. Furthermore, it opens possibilities to study diseases such as malaria [11, 12], the development of micro-channel devices for efficient cell separation [13, 14], understanding the mechanisms behind the rheological behaviour [15], and drug delivery [16], to name but a few applications. Most of the numerical simulations of blood micro-circulation rely on Lagrangian particle or immersed boundary type methods. In this work, we employ the *moving particle semi-implicit* (MPS) method [17], and develop this numerical scheme further with the scope of simulating blood micro-circulation. In doing so we build on the works of [7, 12, 18, 19, 20, 21] that employ MPS for simulating blood micro-circulation.

The MPS method was introduced in [17] to solve the incompressible Navier-Stokes equations. The method has since developed greatly, both to improve stability and accuracy, and has been used to solve a variety of problems including wave impacts on floating bodies [22, 23, 24], sprays dynamics [25], jet and droplet breakup [26, 27, 28, 29], mixing [30], fragmentation and vapour explosions [31], drop impact and deposition [32, 33], heat transfer [34, 35], turbulence modelling [36], and coupling with the finite-element method [37]. The MPS method is similar to other methods that have successfully simulated blood micro-circulation, such as *smoothed particle hydrodynamics* (SPH) [38], *multiparticle collision dynamics* (MPC) [39, 40] and *dissipative particle dynamics* (DPD) [8, 9]. In these methods the particles commonly represent lumped volumes of fluid that carry, in a Lagrangian framework, all the information of the fluid along their trajectory. An alternative Lagrangian particle approach is to consider the particles as collocation points [34, 41, 42]. A Hamiltonian MPS method has been proposed in [46], with advantages of accurate conservation properties and symplectic integration.

The MPS method solves the incompressible Navier-Stokes equations directly with a strong formulation, through the substitution of the spatial derivatives with discrete differential operators. The incompressible SPH (ISPH) method [43] was developed subsequently and shares several aspects, however the spatial discretisations differ considerably. The SPH method uses a kernel function to interpolate unknowns, and favours a weak formulation of the governing equations. The DPD and MPC methods are mesoscopic stochastic simulation techniques, with consideration of thermal fluctuations and hydrodynamic interactions, that do not attempt to solve differential equations directly but instead achieve appreciable results by careful selection of the technique's various parameters and coefficients. The DPD, MPC and several variants of the SPH methods involve explicit time integration which may result in efficient parallel software implementations. While this simulation technique perspective may be attractive to model complex fluids, the inability to directly solve a differential equation can present obstacles in using the DPD or MPC methods.

The MPS method relies on a fractional-step approach to solve the irrotational and solenoidal fields separately. In the original form presented in [17], the viscous forces applied to the velocity field (irrotational field) are solved explicitly while the pressure (solenoidal field) is solved implicitly, hence giving rise to the name of the method. While this approach is generally preferred in the literature, the pressure can also be solved explicitly by using an equation of state [47, 48], thus improving computational performance in certain classes of problems. In the present study a fully implicit approach is developed, which is favourable in cases of small spatial discretisations since the limiting stability criteria (viscous terms) on the time step is related to h^{-2}, where h is the spatial discretisation size.

In this work the MPS method is developed in the context of modelling blood micro-circulation in periodic geometries, though the developments are naturally pertinent to other problems. Three

important additions to the classical MPS method are presented in detail: i) cell membranes are modelled as a network of springs that act on the Navier-Stokes equations through body force terms; ii) a pressure drop boundary condition is formulated to drive the flow; iii) an implicit scheme on the irrotational field is used to improve stability and allow for larger time steps. In this study the original discrete differential operators are used, as these are more familiar in the literature and avoids unnecessary detail of the various developments. Some key developments are nevertheless cited and discussed where relevant. In doing so, much of the available literature is concisely presented to a wider audience.

The papers is organised as follows. The MPS method is presented in detail in Section 2, and in Section 3 the spring network model to represent the cell membranes is outlined. Simulations of red blood cell micro-circulation are presented and discussed in Section 4, in incremental level of complexity. Conclusions about the numerical methods and the findings from the simulations are summarised in Section 5. An Appendix is also provided with details of the derivation of the discrete differential operators, linearised implicit scheme for the spring membrane, and a geometry reconstruction approach using radial basis functions.

2. Moving particle semi-implicit (MPS) method

The following outline of the MPS method follows closely the presentation of the standard method [17], together with the proposed developments for the simulation of blood circulation as a multi-body medium. Other advances present in the literature are cited and briefly discussed.

2.1. Particle discretisation and function approximation

The domain is initially discretised uniformly, using a Cartesian grid for simplicity (though this is not mandatory), to comprise of particles of three types: fluid, wall and ghost. The set of all particles $\{\mathcal{P}\}$ can therefore be grouped respectively into disjoint sets $\{\mathcal{P}_\mathcal{F}\}$, $\{\mathcal{P}_\mathcal{W}\}$, and $\{\mathcal{P}_\mathcal{G}\}$; the size of these is denoted by $|\{\mathcal{P}_\mathcal{F}\}| = N_F$, $|\{\mathcal{P}_\mathcal{W}\}| = N_W$ and $|\{\mathcal{P}_\mathcal{G}\}| = N_G$, and the total number of particles is $|\{\mathcal{P}\}| = N = N_F + N_W + N_G$. In this work we will consider periodic inflow/outflow sections, and subsequently can divide the sets into *real* particles that sample the spatial domain, and *periodic* particles that are mapped real particles at the periodic sections. Subsequently, the sets are expressed as $\{\mathcal{P}_\mathcal{F}\} = \{\mathcal{P}_{\mathcal{F}_r} \dot\cup \mathcal{P}_{\mathcal{F}_p}\}$, $\{\mathcal{P}_\mathcal{W}\} = \{\mathcal{P}_{\mathcal{W}_r} \dot\cup \mathcal{P}_{\mathcal{W}_p}\}$, and $\{\mathcal{P}_\mathcal{G}\} = \{\mathcal{P}_{\mathcal{G}_r} \dot\cup \mathcal{P}_{\mathcal{G}_p}\}$, with $N_F = N_{Fr} + N_{Fp}$, $N_W = N_{Wr} + N_{Wp}$, and $N_G = N_{Gr} + N_{Gp}$. A simple pictorial representation of the particle sets is provided in Fig. 1.

Due to the Cartesian discretisation, the fluid and wall particles uniformly sample the domain, resulting in a step-like wall representation. The use of ghost particles to pad the external domain is a common technique to ensure that the *particle number density* (see below) does not drop near the wall. An alternative to using ghost nodes is to provide the necessary pressure gradient at the wall by assuming a lumped wall weight function [50, 51], which allows for greater flexibility at a reduced accuracy.

In mesh-free methods, function approximation relies on the information available in a local neighbourhood, and is often based on a weighted radial interaction, hence proximity, such that closer neighbouring points have a greater influence than further ones. In the classical MPS method the weight function is based on the inverse distance (Shepherd interpolant), given by:

$$w(r) = \begin{cases} \frac{r_e}{r} - 1 & 0 < r < r_e \\ 0 & r_e < r \end{cases} \tag{1}$$

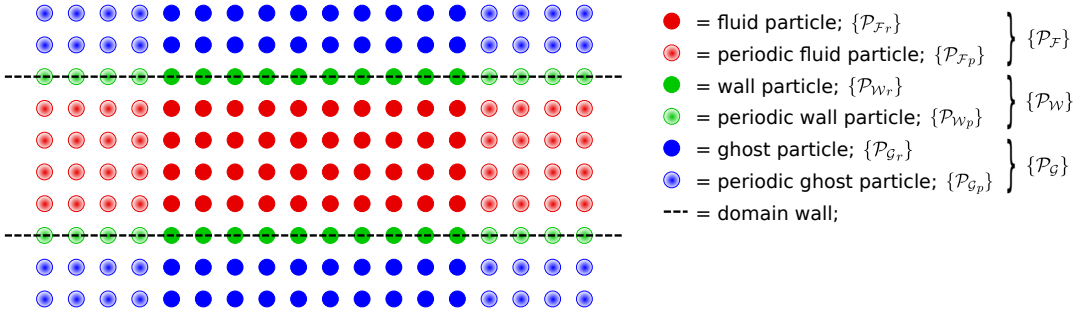


Figure 1: Pictorial representation of the particle types in a periodic domain.

where r_e is the radius of compact support, and r is the radial distance.

Consider a scattered set of points, where locally the central particle is located at $\mathbf{x}_i$ and M neighbouring particles have locations $\mathbf{x}_j$ and lie within the compact support r_e. The distance between these particles is given by $r = |\mathbf{x}_j - \mathbf{x}_i|$. Similar to above, we denote by M_F, M_W and M_G the number of neighbouring particles that belong to the respective sets $\{\mathcal{P}_\mathcal{F}\}$, $\{\mathcal{P}_\mathcal{W}\}$, and $\{\mathcal{P}_\mathcal{G}\}$, hence $M = M_F + M_W + M_G$. From these considerations we interpolate the value of a scalar ϕ at an arbitrary location i by:

$$\phi_i = \frac{1}{\sum_{j\neq i}^{M} w(r)} \sum_{j\neq i}^{M} \left(w(r)\, \phi_j\right) \tag{2}$$

A typical value for the radius of compact support is $r_e = 2.1h_0$, where h_0 is the particle spacing for a Cartesian grid discretisation of the domain [17]. The volume associated to each particle is therefore given by $V_{particle} = h_0^3$, and translating this to a spherical volume representation $V_{particle} = (4/3)\pi(l_0/2)^3$, where l_0 is the 'expected' particle spacing during simulations.

Other weighting functions have been tested in order to improve the stability of the MPS method in [52]; however it should be noted that a change in weighting function cannot be the critical solution to stabilise simulations nor achieve greatly improved accuracy [53].

2.2. Particle number density

An important measure in MPS is the particle number density, which can be directly derived from Eq. 2 by considering $\phi_j = 1$. The particle number density at particle i is given by:

$$n_i = \sum_{j\neq i}^{M} w(r) \tag{3}$$

It should be noted that the central part occupied by particle i is excluded. Since particles represent lumped volumes of fluid, the density is proportional to the particle number, $\rho_i \propto n_i$. For incompressible flows this particle number density should be constant n^0, indicating a uniform (albeit scattered) distribution of the particles. Note that by considering $\phi = 1$ we have assumed that the particles have the same mass, and therefore discretise the domain at a constant resolution, though this need not be the case and can act as a means of adaptive spatial discretisation.

2.3. Time integration

Consider the mass continuity and Navier-Stokes equations in the Lagrangian frame of reference:

$$\begin{aligned} \frac{1}{\rho}\frac{D\rho}{Dt} + \nabla \cdot \mathbf{u} &= 0 \\ \frac{D\mathbf{u}}{Dt} &= -\frac{\nabla p}{\rho} + \nu\nabla^2\mathbf{u} + \mathbf{g} \end{aligned} \tag{4}$$

where $\frac{D}{Dt}$ denotes the material derivative with respect to time, $\mathbf{u}$ is the velocity, p is the pressure, $\mathbf{g}$ is the external body force per unit mass, and ρ and ν are respectively the fluid density and kinematic viscosity. As a fractional step method, the MPS method is in essence a predictor-corrector method for the velocity. Consequently the particle velocities are given by $\mathbf{u}_i^{n+1} = \mathbf{u}_i^* + \mathbf{u}_i^{**}$ and the location of the particles is advanced by $\mathbf{x}_i^{n+1} = \mathbf{x}_i^n + \Delta t\, \mathbf{u}_i^{n+1}$. Here the superscripts * and ** respectively denote the predictor and corrector stages, and superscripts $n+1$ and n are consecutive time steps.

The temporal derivatives in these equations can be discretised using the forward Euler scheme as:

$$\frac{D\mathbf{u}}{Dt} = \frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} = \frac{\mathbf{u}^{**}}{\Delta t} + \frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} \quad ; \quad \frac{D\rho}{Dt} = \frac{\rho^{**}}{\Delta t} + \frac{\rho^* - \rho^n}{\Delta t} \tag{5}$$

It follows that the conservation of momentum equation becomes

$$\mathbf{u}^{**} + (\mathbf{u}^* - \mathbf{u}^n) = -\frac{\Delta t}{\rho}\nabla p + \Delta t\,(\mathbf{g} + \nu\nabla^2\mathbf{u}) \tag{6}$$

which can be split in the two time steps of predictor and corrector:

$$\mathbf{u}^* = \mathbf{u}^n + \Delta t\,(\nu\nabla^2\mathbf{u}^* + \mathbf{g}^\mathbf{n}) \tag{7a}$$

$$\mathbf{u}^{**} = -\frac{\Delta t}{\rho}\nabla p^{n+1} \tag{7b}$$

It is important to note in Eq. 6 we have not specified the time instance of the variables on the right hand side, while in Eqs. 7 we have specified these steps as implicit, with explicit treatment of the body force term only. The body force term, $\mathbf{g}$, includes: i) pressure gradient from boundary conditions, $\mathbf{f}_p$; ii) external body forces, such as gravity, $\mathbf{f}_e$; iii) the structural forces, which in our case are the cell membrane forces $\mathbf{f}_s$; hence $\mathbf{g} = (\mathbf{f}_s + \mathbf{f}_p + \mathbf{f}_e)$. Additional forces may be considered to model further physical processes with ease. Of these, the structural forces depend on the position of the particles, such that an explicit treatment is simpler at the current stage though an iterative approach may be employed at a greater computational cost. In Appendix B a linearised implicit formulation for the spring forces is presented, thereupon the body and viscous forces can eventually be included as a single implicit step. In this work the term $\mathbf{f}_e$ has been neglected. The above Eqs. 7 are rewritten to clearly outline the splitting to facilitate later discussion

$$\mathbf{u}^* = \mathbf{u}^n + \Delta t\,(\mathbf{f}_s^n + \mathbf{f}_p^n) + \Delta t\,(\nu\nabla^2\mathbf{u}^*) \tag{8a}$$

$$\mathbf{u}^{**} = -\frac{\Delta t}{\rho}\nabla p^{n+1} \tag{8b}$$

In a similar way, the continuity equation can be re-written as:

$$(\rho^{**} + (\rho^{*} - \rho^{n}))\,\frac{1}{\rho \Delta t} = -\nabla \cdot (\mathbf{u}^{**} + \mathbf{u}^{*}) \tag{9}$$

once again splitting into the two time steps of predictor and corrector:

$$\frac{1}{\rho}\frac{\rho^{*} - \rho^{n}}{\Delta t} = -\nabla \cdot \mathbf{u}^{*} \tag{10a}$$

$$\frac{1}{\rho}\frac{\rho^{**}}{\Delta t} = -\nabla \cdot \mathbf{u}^{**} \tag{10b}$$

Since $\rho \propto n$ we rewrite Eq. 10b as:

$$\frac{1}{n^{0}}\frac{n^{**}}{\Delta t} = -\nabla \cdot \mathbf{u}^{**} \tag{11}$$

We note that to enforce incompressibility ($D\rho/Dt = 0$) we require $n^{*} + n^{**} = n^{0}$, and substituting Eq. 8b into Eq. 11, and after rearranging we obtain the pressure Poisson equation:

$$\nabla^{2} p^{n+1} = \frac{-\rho}{\Delta t^{2}}\frac{(n^{*} - n^{0})}{n^{0}} \tag{12}$$

The solution of the pressure is then used to compute the pressure gradient, and subsequently the corrector velocity from Eq. 8b. The pressure Poisson equation above relates the pressure to the deficit in the particle number density, hence the local density variation. This approach ensures the conservation of mass even after simulating many time steps, however the system is very stiff and can cause pressure oscillations and induce instabilities. This is often reported as the main cause of simulation breakdown.

Alternative source terms to the pressure Poisson equation have been put forward to improve stability and accuracy of the method, and it is relevant to discuss these briefly now. One of these is clearly to consider the particles as collocation points [42, 34, 53, 54], and we rewrite Eq. 8b as:

$$-\frac{\Delta t}{\rho}\nabla p^{n+1} = \mathbf{u}^{**} = \mathbf{u}^{n} - \mathbf{u}^{*} \tag{13}$$

and taking the divergence on both sides, using $\nabla \cdot \mathbf{u}^{n} = \nabla \cdot \mathbf{u}^{n+1} = 0$ as the equation for mass conservation for an incompressible fluid and rearranging, we obtain an alternative pressure Poisson equation:

$$\nabla^{2} p^{n+1} = \frac{\rho}{\Delta t}\nabla \cdot \mathbf{u}^{*} \tag{14}$$

The resulting pressure field is smoother and the method is more robust. The disadvantage of this approach is that round-off errors will accumulate and mass conservation is no longer maintained with long simulations involving many time integration steps. A mixed approach, based on a weighted average from solutions to both Eq. 12 and Eq. 14, has been suggested [53, 54] to obtain a more accurate pressure distribution and empirically adjust for the mass conservation. Given the importance to conserve the red blood cell volumes, the pressure Poisson equation given by Eq. 12 is adopted. A further alternative source term for the pressure Poisson equation employs a higher order discretisations of the continuity equation, considering the substitution of Eq. 3 such that $\frac{Dn}{Dt} = \sum_{i \neq j}^{M} \frac{Dw(r)}{Dt}$, and differentiating using the product rule [55, 56]. Another source term proposed in [57] and discussed further in [58] to stabilise the MPS method, involves the addition of *error-compensating terms* in small amounts, which are based on the temporal gradient and Laplacian of the particle number density.

2.4. Outline of scheme

The method, based on the classical MPS scheme but with implicit treatment of the viscous terms and explicit forcing terms, can be concisely detailed in order of computation steps as follows (boundary conditions are stated but discussed in greater depth in section 2.6):

Step.1 pressure field for pressure boundary conditions (implicit step),
$\nabla^2 p^* = 0$, with BCs: Δp for periodic sections, $\nabla p \cdot e_n = 0$ at walls;

Step.2 body forces (pressure gradient, spring forces),
$\mathbf{f}_p^n = \frac{-1}{\rho}\nabla p^*$, $\mathbf{f}_s^n$ = spring forces;

Step.3 solve for the viscous forces and increment velocity (implicit step),
$\mathbf{u}^* = \mathbf{u}^n + \Delta t(\mathbf{f}_s^n + \mathbf{f}_p^n) + \Delta t(\nu\nabla^2\mathbf{u}^*)$, with BCs: $\mathbf{u}_i = 0$ at walls

Step.4 check CFL condition is satisfied, if not go to Step.3 with smaller Δt;

Step.5 move particles,
$\mathbf{x}^* = \mathbf{x}^n + \Delta t\ \mathbf{u}^*$;

Step.6 pressure Poisson equation to satisfy incompressibility (implicit step),
$\nabla^2 p^{n+1} = \frac{-\rho}{\Delta t^2}\frac{(n^*-n^0)}{n^0}$, with BCs: $\nabla p \cdot e_n = 0$ at walls;

Step.7 update the velocity,
$\mathbf{u}^{n+1} = \mathbf{u}^* + \mathbf{u}^{**} = \mathbf{u}^* - \frac{\Delta t}{\rho}\nabla p^{n+1}$;

Step.8 move particles,
$\mathbf{x}^{n+1} = \mathbf{x}^n + \Delta t\ \mathbf{u}^{n+1}$.

2.5. Discrete differential operators

The above scheme requires numerical approximations for the gradient and Laplacian operators. The discrete operators are given here, however the derivation and additional discussion is provided in Appendix A.

Let us consider an arbitrary scalar quantity ϕ given at each particle. The discrete gradient operator is computed as a weighted average over neighbouring particles as:

$$\langle\nabla\phi\rangle_i = \frac{d}{n_i}\sum_{j\neq i}^{M} w(r)\frac{(\phi_j - \phi_i)(\mathbf{x}_j - \mathbf{x}_i)}{|\mathbf{x}_j - \mathbf{x}_i|^2} \tag{15}$$

where d is the number of space dimensions, thus $d = 3$ in 3D. From the above scheme, we note that the gradient operator is used only on the pressure in Step.2 and Step.7, and therefore our scalar quantity $\phi \equiv p$. From Eq. 15 we note that the term $(p_j - p_i)$ may assume both positive and negative values, resulting in a repulsion or attraction. This pressure gradient would give rise to a fluctuating velocity correction field, that can degrade both accuracy and stability of the method. An alternative is to consider the following formulation [17]

$$\langle\nabla\phi\rangle_i = \frac{d}{n_i}\sum_{j\neq i}^{M} w(r)\frac{(\phi_j - \min(\phi_i))(\mathbf{x}_j - \mathbf{x}_i)}{|\mathbf{x}_j - \mathbf{x}_i|^2} \tag{16}$$

where $\min(\phi_i)$ signifies the minimum value of ϕ among the neighbouring M particles of particle i and itself. We can now see that the term $(p_j - \min(p_i))$ results exclusively in positive values, hence only repulsive forces, resulting in a smoother velocity correction field. The formulation is however now no longer symmetric, and consequently the conservation of momentum is no longer

respected. To solve this a corrected gradient operator was proposed [55, 56], which maintains symmetry and results only in repulsive forces

$$\langle \nabla \phi \rangle_i = \frac{d}{n_i} \sum_{j \neq i}^{M} w(r) \frac{\Big((\phi_i + \phi_j) - (\min(\phi_i) + \min(\phi_j))\Big)(\mathbf{x}_j - \mathbf{x}_i)}{|\mathbf{x}_j - \mathbf{x}_i|^2} \tag{17}$$

Due to its symmetric properties, this formulation however is incompatible with applying pressure boundary conditions. For example, in the case of a linear pressure gradient field, we erroneously obtain $\nabla p = 0$ when computed with Eq. 17. As a result of these observations, the discrete gradient operator used to compute ∇p^* (Step.2) is given by Eq. 16, while Eq. 17 is employed to compute ∇p^{n+1} (Step.7). The basic form (given by Eq. 15) is adopted in the outline of the discretised scheme in Section 2.8 for clarity, and it should be remembered that the alternative derivations are used in its place.

The Laplacian is derived from the solution of the diffusion equation [59]. For a scalar quantity ϕ, it is given by:

$$\langle \nabla^2 \phi \rangle_i = \frac{2d}{\lambda n_i} \sum_{j \neq i}^{M} w(r)(\phi_j - \phi_i) \tag{18}$$

where the normalisation coefficient serves to convert the analytic continuous solution to discrete points, and is given by

$$\lambda = \frac{\sum_{j \neq i}^{M} (w(r)\, r^2)}{\sum_{j \neq i}^{M} w(r)} \tag{19}$$

Alternative discrete operators for the Laplacian have been proposed in [56] as the divergence of the gradient operator, thus maintaining the order of the discrete differential operators consistent.

2.6. Boundary conditions and linear systems

The boundary conditions for the system given by Eq. 8a (Step.3) are periodic inflow/outflow sections, and Dirichlet condition on the wall for which $\mathbf{u} = 0$ (no-slip) is chosen here. For simplicity the ghost particles are given the same velocity as the wall, which is a reasonable approximation given the inverse distance weight function (Eq. 1) used and noting that the ghost particle are distant from the central particle. It has been reported in [32, 54] that mirroring the fluid velocities at the ghost particles based on the wall location will enforce the prescribed wall velocity more accurately. Starting from Eq. 8a and using the discrete Laplace operator in Eq. 18, we obtain

$$\mathbf{u}_i^* = \mathbf{u}_i^n + \Delta t\, (\mathbf{f}_s^n + \mathbf{f}_p^n) + \frac{\nu 2d\Delta t}{\lambda n_i^n} \sum_{j \neq i}^{M} \Big(w(r)(\mathbf{u}_j^* - \mathbf{u}_i^*)\Big) \qquad \text{for } \{i \in \mathcal{P}_{\mathcal{F}_r}\} \tag{20}$$

and considering the Dirichlet boundary conditions we expand to obtain:

$$\mathbf{u}_i^* \left(\frac{\lambda n_i^n}{\nu 2d\Delta t} + n_i^n \right) - \sum_{j \neq i}^{M_F} \Big(w(r)\, \mathbf{u}_j^*\Big) = \frac{\lambda n_i^n}{\nu 2d\Delta t} (\mathbf{u}_i^n + \Delta t(\mathbf{f}_s^n + \mathbf{f}_p^n)) + \sum_{j \neq i}^{M_W + M_G} \Big(w(r)\, \mathbf{u}_j^*\Big) \tag{21}$$

The resulting linear system is square (dimension N_F), symmetric, strictly diagonally dominant, hence positive definite. The left hand side represents the unknown variables and the right hand side the known values.

Due to the splitting scheme (which relies on the Helmholtz-Hodge decomposition), the linear system to solve the pressure Poisson equation (Step.6) employs the $\partial\, p/\partial\, e_n = 0$ Neumann condition on the wall, where e_n is the surface normal unit vector. In order to reduce the non-physical boundary layers due to this condition [60, 61], the Neumann condition is imposed at the wall considering the ghost particles to form the gradient, hence $(p_j - p_i) = 0$ for $\{i \in \mathcal{P}_{\mathcal{F}} \,\dot\cup\, \mathcal{P}_{\mathcal{W}}\,,\ j \in \mathcal{P}_{\mathcal{G}}\}$ is implied for the weighted average. Starting from Eq. 12 and using the discrete Laplace operator in Eq. 18, we obtain

$$\frac{2d}{\lambda n_i^*}\sum_{j\neq i}^{M} w(r)\left(p_j^{n+1} - p_i^{n+1}\right) = \frac{-\rho}{\Delta t^2}\frac{n_i^* - n^0}{n^0} \qquad \text{for } \{i \in \mathcal{P}_{\mathcal{F}_r} \,\dot\cup\, \mathcal{P}_{\mathcal{W}_r}\} \tag{22}$$

and considering the Neumann boundary conditions we obtain:

$$\sum_{j\neq i}^{M_F+M_W}\left(w(r)\; p_j^{n+1}\right) - p_i^{n+1}\left(\sum_{j\neq i}^{M_F+M_W} w(r)\right) = \frac{-\rho\lambda n_i^*}{\Delta t^2 2d}\frac{(n_i^* - n^0)}{n^0} \tag{23}$$

The matrix resulting from the linear system can be easily written to have unitary diagonal, equivalent to a Jacobi preconditioning.

$$\frac{1}{n_i^{\#}}\sum_{j\neq i}^{M_F+M_W}\left(w(r)\; p_j^{n+1}\right) - p_i^{n+1} = \frac{-\rho\lambda n_i^*}{n_i^{\#}\Delta t^2 2d}\frac{(n_i^* - n^0)}{n^0} \tag{24}$$

where $n_i^{\#} = \left(\sum_{j\neq i}^{M_F+M_W} w(r)\right)$. The linear system given by Eq. 23 is square (dimension $N_{Fr} + N_{Wr}$), symmetric weakly diagonally dominant, hence positive semi-definite. Since the solution of the pressure is not unique, the pressure should be fixed in some form. An option is to fix the pressure at given particles, which is seen to work well in cases when the pressure is set for several particles, as in free-surface problems [17]. However, if a single particle pressure is fixed at an arbitrary value, it is found that the round-off errors can be detrimental to the accuracy of the solution. The preferred approach in these cases is to set an average pressure through a Lagrange multiplier, and the system given by Eq. 12 can be written as:

$$\frac{2d}{\lambda n_i^*}\sum_{j\neq i}^{M} w(r)\left(p_j^{n+1} - (p_i^{n+1} + p_\lambda^{n+1})\right) = \frac{-\rho}{\Delta t^2}\frac{n_i^* - n^0}{n^0} \qquad \text{for } \{i \in \mathcal{P}_{\mathcal{F}_r} \,\dot\cup\, \mathcal{P}_{\mathcal{W}_r}\}\,. \tag{25}$$

Considering the Neumann boundary conditions and the average pressure condition, we obtain

$$\frac{1}{n_i^{\#}}\sum_{j\neq i}^{M_F+M_W}\left(w(r)\; p_j^{n+1}\right) - p_i^{n+1} - p_\lambda^{n+1} = \frac{-\rho\lambda n_i^*}{n_i^{\#}\Delta t^2 2d}\frac{(n_i^* - n^0)}{n^0} \tag{26a}$$

$$\frac{1}{N_F + N_W}\sum_{i}^{N_F+N_W} p_i^{n+1} = \langle p\rangle = 0 \tag{26b}$$

where p_λ is the Lagrange multiplier pressure, and $\langle p\rangle$ is the average pressure. The linear system formed by Eqs. 26 is square (dimension $N_{Fr} + N_{Wr} + 1$), and is no longer symmetric. The left hand side represents the unknown variables and the right hand side the known values.

The simulations considered involve periodic conditions to the fluid, and in order to simulate complex geometries a pressure drop provides the necessary flexibility. The pressure drop boundary condition may be derived by taking the pressure Poisson equation given by Eq. 12, and note that $n_i^n = n^0$ at the start of each time integration step (we are not solving for the pressure correction fractional step). We consequently arrive at the pressure Poisson equation with a zero source term (Step.1). Considering once again the average pressure Lagrange multiplier, and introducing Δp as the pressure drop across the periodic sections, hence $\Delta p_j \neq 0$ for $\{j \in \mathcal{P}_{\mathcal{F}_p} \dot{\cup} \mathcal{P}_{\mathcal{W}_p}\}$, we obtain:

$$\frac{2d}{\lambda n_i^n} \sum_{j \neq i}^{M} w(r) \Big((p_j^* + \Delta p_j) - (p_i^* + p_\lambda^*) \Big) = 0 \qquad \text{for } \{i \in \mathcal{P}_{\mathcal{F}_r} \dot{\cup} \mathcal{P}_{\mathcal{W}_r}\} \tag{27}$$

A pressure drop between any particles may be prescribed, however in fact we wish only to impose a pressure gradient across the periodic inflow/outflow sections, and subsequently $\Delta p_j \neq 0$ only on the periodic particles (sets $\{j \in \mathcal{P}_{\mathcal{F}_p} \dot{\cup} \mathcal{P}_{\mathcal{W}_p}\}$). By expanding and considering the Neumann boundary condition on the wall, we arrive at

$$\frac{1}{n_i^\#} \sum_{j \neq i}^{M_F + M_W} (w(r)\, p_j^*) - p_i^* - p_\lambda^* = \frac{-1}{n_i^\#} \sum_{j \neq i}^{N_{F_p} + N_{W_p}} (w(r)\, \Delta p_j) \tag{28a}$$

$$\frac{1}{N_F + N_W} \sum_{i}^{N_F + N_W} p_i^* = \langle p \rangle = 0 \tag{28b}$$

We clarify that Δp is the pressure drop imposed on the periodic particles only, such that periodic particles have the same index but have an incremented pressure at mapped locations. The linear system formed by Eqs. 28 is again square (dimension $N_{Fr} + N_{Wr} + 1$), and not symmetric. The left hand side represents the unknown variables and the right hand side the known values. It is important to note that Eqs. 28 are independent of the time step, which facilitates the implementation of an adaptive time step size.

We direct the interested reader to [18, 34, 20, 42, 62] for discussion of non-periodic boundary conditions for the inflow/outflow sections.

2.7. Additional stabilisation as local weak compressibility

The linear systems arising from the pressure calculation are stiff, and it has been suggested in [63, 64] to introduce a weak compressibility. The method is briefly presented here for completeness and clarity, following closely [64].

Consider the equation of state $c^2 = \partial p / \partial \rho$, where c^2 is the speed of sound. The derivative can be approximated to yield $\rho_i^{n+1} - \rho^0 = (1/c^2)(p_i^{n+1} - p^0)$, where ρ^0 and p^0 are the base density and pressure. Dividing both sides by ρ^0 and since $\rho \propto n$ we obtain

$$\frac{n_i^{n+1} - n^0}{n^0} = \frac{1}{\rho^0 c^2} (p_i^{n+1} - p^0) \tag{29}$$

For an incompressible fluid the base pressure can take any value and for convenience it is set to $p^0 = 0$. Let us also denote a small difference in the particle number density as $\Delta n = n_i^{n+1} - n^0$. Finally, for convenience we rename $\rho^0 = \rho$ to be consistent with the above notation. The resulting equation states that a difference in particle number density is proportional to a perturbation about the base pressure value:

$$\frac{\Delta n}{n^0} = \frac{1}{\rho c^2} p_i^{n+1} \tag{30}$$

To introduce weak compressibility we increment n^0 by Δn in Eq. 12:

$$\nabla^2 p^{n+1} = \frac{-\rho}{\Delta t^2}\left(\frac{n^* - (n^0 + \Delta n)}{n^0}\right) = \frac{-\rho}{\Delta t^2}\left(\frac{n^* - n^0}{n^0} - \frac{1}{\rho c^2} p_i^{n+1}\right) \tag{31}$$

Taking $\kappa = 1/\rho c^2$ to be a small constant and substituting in Eq. 18, this leads to

$$\frac{2d}{\lambda n_i^*} \sum_{j \neq i}^{M} w(r)\left(p_j^{n+1} - p_i^{n+1}\right) = \frac{-\rho}{\Delta t^2}\left(\frac{n^* - n^0}{n_i^0} - \kappa p_i^{n+1}\right) \qquad \text{for } \{i \in \mathcal{P}_{\mathcal{F}_r} \dot{\cup} \mathcal{P}_{\mathcal{W}_r}\} \tag{32}$$

and considering the boundary conditions as above we obtain

$$\sum_{j \neq i}^{M_F + M_W} \left(w(r) p_j^{n+1}\right) - p_i^{n+1}\left(n_i^{\#} + \kappa \frac{\rho \lambda n_i^*}{2d\Delta t^2}\right) = \frac{-\rho \lambda n_i^*}{\Delta t^2 2d} \frac{(n^* - n^0)}{n^0} \tag{33}$$

It is evident that the diagonal term of the resulting linear system is greater in this formulation of weak compressibility in comparison to Eq. 23. The resulting matrix from Eq. 33 is now strictly diagonally dominant, is better conditioned, and benefits a faster solution time when using iterative solvers. The pressure Lagrangian multiplier can be added in a similar fashion to Eqs. 26. Furthermore the weak compressibility may be used in conjunction to the pressure boundary condition Eqs. 28, though care must be taken not to obtain standing waved as a result.

The term $\kappa \frac{\rho \lambda n_i^*}{2d\Delta t^2}$ is however too small to have any practical effect. As the speed of sound is very large in relation to the actual maximum particle speed during the simulations, $|\mathbf{u}|_{max}$, it has been suggested to use $c = 10|\mathbf{u}|_{max}$ [47, 48]. This weak compressibility term may be further simplified either as a constant, hence $\left(n_i^{\#} + \kappa \frac{\rho \lambda n_i^*}{2d\Delta t^2}\right) \approx \left(n_i^{\#} + \kappa \frac{\rho \lambda n^0}{2d\Delta t^2}\right) = n_i^{\#} + \gamma$, or alternatively as a function of the particle number density (though the summations to compute $n^{\#}$ and n^* are on different sets of particles), hence $\left(n_i^{\#} + \kappa \frac{\rho \lambda n_i^*}{2d\Delta t^2}\right) \approx \left(n_i^{\#} + \kappa \frac{\rho \lambda n_i^{\#}}{2d\Delta t^2}\right) = \beta\, n_i^{\#}$. The first approach suggests a uniform weak compressibility in the domain, while the latter serves as a local percentage compressibility. Here the latter approach is used, however further investigation is necessary to evaluate the preferred approach. Rewriting Eq. 33 to include this approximation we obtain

$$\frac{1}{n_i^{\#}} \sum_{j \neq i}^{M_F + M_W} \left(w(r_{ij}) p_j^{n+1}\right) - \beta\, p_i^{n+1} = \frac{-\rho \lambda n_i^*}{n_i^{\#} \Delta t^2 2d} \frac{(n^* - n^0)}{n^0} \tag{34}$$

We set $\beta = 1.0001$, that is small enough not to affect the solution accuracy but improves the stability and reduces the computational time required to solve the linear systems with Krylov subspace iterative methods. Errors introduced by this weak compressibility are not cumulative as the simulation progresses, in the sense that the fluid volume and n^0 remain unaltered.

The stability of the MPS method has been analysed in [58], showing the method suffers from tensile instability. The suggested approach to stabilising the MPS method is to improve the order of discrete differential operators of the Laplacian and gradient, as well as including error-compensating source terms in the pressure Poisson equation as mentioned above. In this study, no instability was observed numerically. The procedural steps of the MPS method, taking into consideration the boundary conditions and discrete operators, are now presented.

2.8. Outline of discretised scheme

The discretised MPS method can be concisely detailed in order of computation steps as detailed below. For each step the left hand side represents the unknown variables and the right hand side the known values.

Step.1 pressure field for pressure boundary conditions (implicit step),

$$\frac{1}{n_i^{\#}} \sum_{j\neq i}^{M_F+M_W} \left(w(r)\ p_j^*\right) - \beta p_i^* - p_\lambda^* = \frac{-1}{n_i^{\#}} \sum_{j\neq i}^{N_{Fp}+N_{Wp}} \left(w(r)\ \Delta p_j\right) \qquad \text{for } \{i \in \mathcal{P}_{\mathcal{F}_r} \dot\cup \mathcal{P}_{\mathcal{W}_r}\}$$

$$\frac{1}{N_F+N_W} \sum_i^{N_F+N_W} p_i^* = \langle p \rangle = 0 \qquad \text{for } \{i \in \mathcal{P}_{\mathcal{F}_r} \dot\cup \mathcal{P}_{\mathcal{W}_r}\}$$

Step.2 body forces (pressure gradient, spring forces),

$$\nabla p_i^* = \frac{d}{n_i^{\#}} \sum_{j\neq i}^{M_F+M_W} \left(w(r)(p_j^* - p_i^*)\frac{(\mathbf{x}_j-\mathbf{x}_i)}{|r|^2}\right) \qquad \text{for } \{i \in \mathcal{P}_{\mathcal{F}_r}\}$$

$$\mathbf{f}_p^n = -\frac{\nabla p^*}{\rho}$$

$\mathbf{f}_s^n$ = spring forces

Step.3 solve for the viscous forces and increment velocity (implicit step),

$$\mathbf{u}_i^* \left(\frac{\lambda n_i^n}{\nu 2d\Delta t} + n_i^n\right) - \sum_{j\neq i}^{M_F} \left(w(r)\ \mathbf{u}_j^*\right) =$$

$$= \frac{\lambda n_i^n}{\nu 2d\Delta t}(\mathbf{u}_i^n + \Delta t(\mathbf{f}_s^n + \mathbf{f}_p^n)) + \sum_{j\neq i}^{M_W+M_G} \left(w(r)\ \mathbf{u}_j^*\right) \qquad \text{for } \{i \in \mathcal{P}_{\mathcal{F}_r}\}$$

Step.4 check CFL condition is satisfied, if not go to Step.3 with smaller Δt

Step.5 move particles,

$$\mathbf{x}_i^* = \mathbf{x}_i^n + \Delta t\ \mathbf{u}_i^* \qquad \text{for } \{i \in \mathcal{P}_{\mathcal{F}_r}\}$$

Step.6 pressure Poisson equation to satisfy incompressibility (implicit step),

$$\frac{1}{n_i^{\#}} \sum_{j\neq i}^{M_F+M_W} \left(w(r) p_j^{n+1}\right) - \beta p_i^{n+1} - p_\lambda^{n+1} = \frac{-\rho\lambda n_i^*}{n_i^{\#}\Delta t^2 2d} \frac{(n_i^*-n^0)}{n^0} \qquad \text{for } \{i \in \mathcal{P}_{\mathcal{F}_r} \dot\cup \mathcal{P}_{\mathcal{W}_r}\}$$

$$\frac{1}{N_F+N_W} \sum_i^{N_F+N_W} p_i^{n+1} = \langle p \rangle = 0 \qquad \text{for } \{i \in \mathcal{P}_{\mathcal{F}_r} \dot\cup \mathcal{P}_{\mathcal{W}_r}\}$$

Step.7 update the velocity,

$$\nabla p_i^{n+1} = \frac{d}{n_i^{\#}} \sum_{j\neq i}^{M_F+M_W} \left(w(r)(p_j^{n+1} - p_i^{n+1})\frac{(\mathbf{x}_j^*-\mathbf{x}_i^*)}{|r|^2}\right) \qquad \text{for } \{i \in \mathcal{P}_{\mathcal{F}_r}\}$$

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^* - \Delta t \frac{1}{\rho} \nabla p_i^{n+1} \qquad \text{for } \{i \in \mathcal{P}_{\mathcal{F}_r}\}$$

Step.8 move particles,

$$\mathbf{x}_i^{n+1} = \mathbf{x}_i^n + \Delta t\ \mathbf{u}_i^{n+1} \qquad \text{for } \{i \in \mathcal{P}_{\mathcal{F}_r}\}.$$

2.9. Adaptive time integration

There is currently no complete set of stability conditions on the time step for the MPS method, for a general scattered particle distribution. It is however straightforward to obtain guidelines by considering both the Courant number (CFL condition) for the advection term and a diffusion number for the viscous forces. We write these two conditions for MPS as:

$$advection: \qquad \Delta t < \frac{|\tilde{r}|_{min}}{|\tilde{\mathbf{u}}|_{max}} \tag{35a}$$

$$diffusion: \qquad \Delta t < \frac{\lambda}{2d\nu} \tag{35b}$$

where $|\tilde{\mathbf{u}}|_{max} = \max(|\mathbf{u}_i - \mathbf{u}_j|)$ is the maximum relative velocity magnitude of neighbouring particles, and $|\tilde{r}|_{min} = \min(|\mathbf{x}_i - \mathbf{x}_j|)$ is the minimum distance between neighbouring particles, where max and min are over the entire set of particles $\mathcal{P}$ at the given time instance. We note that the diffusion term is limited by a proportionality to h_0^2 (from λ) while for the advection the proportionality is to h_0 (from $|\tilde{r}|_{min}$). For micro-circulation therefore, the diffusion process requires a smaller time step as the spatial discretisation h_0 becomes smaller. For the parameters chosen for

the results presented below, the diffusion time step is $\Delta t \sim 5\times10^{-8}$, while the advection time step in the case of the flow in retina capillary model was found to fluctuate around $\Delta t \sim 1.5 \times 10^{-4}$. It is consequently beneficial to use an implicit time integration for the diffusion term of Eq. 8a (alternatively Eq. 21, Step.3), as suggested in [12]. We note also that an explicit sub-iteration scheme was proposed in [12, 19] for solving the diffusion term, however these will ultimately be more computationally intensive and less stable.

The CFL condition represents an important guideline for stability in the MPS method. The pressure correction term is related to differences in particle number density $n_i^* - n^0$, as described by Eq. 12 (alternatively Eq. 26, Step.6). In order to maintain stability and avoid non-physical behaviour, the CFL condition is adopted as

$$\Delta t \leq \alpha \frac{|\tilde{r}|_{min}}{|\tilde{\mathbf{u}}|_{max}} \tag{36}$$

with $\alpha = 0.1$ as suggested by [43] (while other works suggest values $\alpha = 0.2 - 0.25$ [47, 48, 49]).

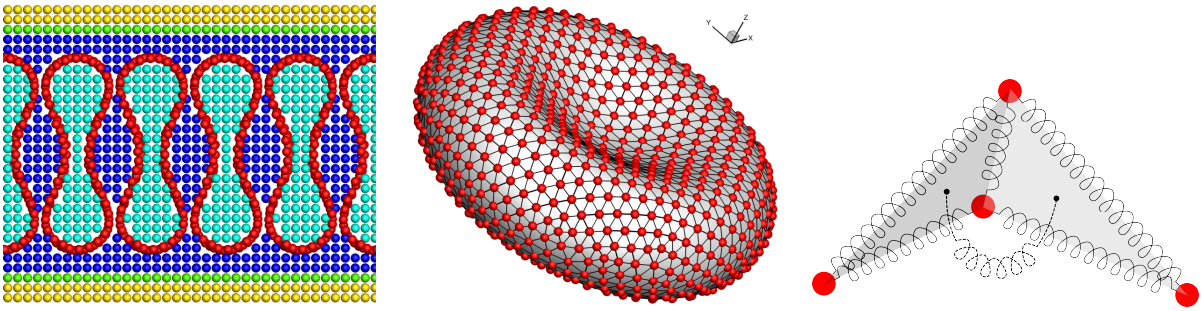
Despite the implicit formulation for both the predictor and corrector steps, the adaptive time step is based on the advection of particles that occurs at the end of the predictor stage (Step.5). The procedure to set the time step involves firstly choosing a tentative value of Δt (based on inter-particle distances and velocities) in order to solve for the velocities in the predictor stage (Step.3), and secondly the CFL condition is verified (Step.4). In the case that the CFL condition is not satisfied, a new, smaller Δt is estimated and the predictor stage is computed iteratively until the CFL condition is met. This iterative correction to reduce Δt to satisfy the CFL condition in practice occurs rarely, and may be furthermore shortened by employing a conservative scaling factor in the estimation of Δt.

2.10. Further remarks

We first remark that while many authors assume constant values of the Laplacian normalisation coefficient λ (Eq. 19), strictly speaking this is incorrect. This parameter, as well as the particle number density n (Eq. 3), should be recomputed for each particle every time the particles are moved.

The linear systems have been solved using the CG or BiCGStab iterative methods, depending if the matrix is symmetric or otherwise. The number of non-zero elements per row in the matrices is influenced by the number of neighbourhood particles M, which is found to be $M \sim 50$ for 3-dimensional simulations and $r_e = 2.1h_0$. This is larger than low-order mesh-based methods, such that in comparison to these the solution of the linear systems is more computationally intensive in the MPS method. The solution of the pressure Poisson equation, given by Step.6 is the stiffest of the linear systems to be solved due to the particle distribution, and requires the most number of solution iterations to solve in comparison to Step.1 or Step.3. Consequently, the adaptive time step discussed in section 2.9, that may require repeated solution of Step.3 to satisfy the CFL condition based on the choice of Δt, does not affect substantially the simulation time. Consequently, the adaptive time step results in a stability control and effectively a faster overall algorithm.

It is important to note that despite the implicit solution of the pressure Poisson equation to correct for incompressibility (Step.6), at the end of each time integration step we observe in general that $n_i^{n+1} \neq n^0$. The discrepancy is small, and the effects on mass conservation are not cumulative as the simulation time proceeds. This discrepancy is not effectively caused by the weak compressibility added by Eq. 34. As noted in [53], the reason is related instead to the fact



cross-section of a domain
at the start of the simulation

detail of red blood cell
membrane spring network

detail of spring elements
connecting the particles

Figure 2: Left: cross-section of example domain at the start of the simulation. The particles are arranged in a Cartesian fashion and the red blood cell membranes are then inserted, subsequently removing abutting particles. The different particle colouring highlights the different classification of the particles types. Centre: detail of a single red blood cell, showing the particle locations and the spring network connectivity. Right: pictorial representation of two triangle mesh elements sharing a common edge, showing the spring elements resisting both tension/compression and bending.

that the particle number density is non-linear with respect to the coordinates. Furthermore we note that the discrete Laplacian operator (used to compute the corrective pressure field) and the discrete gradient operator (used to compute the corrective velocity) are independently derived approximations, and may be inconsistent for general particle arrangement.

3. Modelling cell membranes as spring networks

The structure of biological cell membranes is typically a lipid bilayer, possibly under pre-stress, which has been previously modelled in particle methods by a spring network [7, 8, 9, 12, 65]. The spring network approach utilises a mesh to join fluid particles of the MPS method, and the connectivity of the mesh describes the spring elements. The conversion of material properties between a spring network model and continuum description can be approximated, as reported in [7, 66, 67, 68]. In the present work the mesh consists of piecewise-linear triangle elements. The internal forces generated by the springs act on the fluid particles as body forces in Eq. 8a (Step.3). Detail of a discretised red blood cell membrane is shown in Fig. 2.

The spring network approach is well suited to particle-based methods, due to their comparable accuracy and simplicity. Various constitutive models have been used for the springs to model cell membranes: spectrin-link [8], non-linear [9, 69], linear [7, 12]. Spring elements based on particle proximity have also been used to model thrombus formation [18] and adhesion/tethering [70]. While the spring network approach can provide sufficient accuracy in practice, it should be noted that a continuum model cannot be represented [66, 67], and if more accuracy in modelling is required then alternative approaches should be used [71]. One of the main difficulties in the spring network approach is to ensure independence of the mesh anisotropy [66, 68, 69, 72].

The spring network typically consists of two types of springs, those that resist tension/compression and bending, respectively generating forces $\mathbf{f}^t$ and $\mathbf{f}^b$. The network may be modelled such that these forces are either orthogonal or alternatively coupled, for which a variety of strategies exist. Here both orthogonal and a coupled network (constructed only with tension/compression springs) are presented. Additional penalty forces may also be used, such as to constrain the

cell volume and membrane area [7, 8, 9]. Due to the incompressibility property of the MPS method the penalty force to constrain of the cell volume is not required. The membrane area penalty forces may be difficult to implement without deteriorating the simulation, due to the mesh anisotropy present, and can be circumvented by the use of rigid struts [73] or through developing the spring constitutive models accordingly, and are not considered further in this work. Changes in membrane area are quoted in the results.

An additional repulsive surface penalty force, $\mathbf{f}^r$, is used to refrained particles from crossing the cell membranes [9]. In numerical tests, membrane crossing was not observed even without the repulsive surface penalty force, however this force was adopted nevertheless to cover any eventuality. Importantly, repulsive surface penalty force is also used to provide a simple lubrication force for the unresolved hydrodynamic interactions between solid bodies at very short distances. Overall, the resulting spring force (per unit mass) acting on a particle is given by $\mathbf{f}_s = \mathbf{f}^t + \mathbf{f}^b + \mathbf{f}^r$ for the displacements, and $\mathbf{f}_{s,D} = \mathbf{f}^{t,D} + \mathbf{f}^{b,D} + \mathbf{f}^{r,D}$ for the damping. In this study the damping is not considered in the results but the theory will be detailed. As noted above, the time integration for the spring forces is explicit, while in Appendix B we present a linearised implicit approach.

3.1. Tension/compression springs

The forces for tension/compression springs act to resist stresses in the plane of the triangle elements, and are given as [7]:

$$\mathbf{f}_i^t = \kappa_t \, \frac{(|\mathbf{x}_{ij}| - L_{ij,0})}{L_{ij,0}} \, \frac{\mathbf{x}_{ij}}{|\mathbf{x}_{ij}|} \tag{37}$$

where κ_t is the spring stiffness, $\mathbf{x}_{ij} = \mathbf{x}_j - \mathbf{x}_i$ is the vector connecting particles i and j, and $L_{ij,0}$ is the spring rest length for spring element ij. A pictorial representation of these springs is given in Fig. 2. We assume the membrane is under a very small pre-stress such that the rest spring length is $L_0 = 0.9\, l_0$ for all elements, based on the 'expected' particle spacing [12]. As a consequence numerically, the spring network tends to be more uniformly arranged, and also the particles joined by the network are on the whole closer than the expected inter-particle distance to avoid other particles from crossing the membrane.

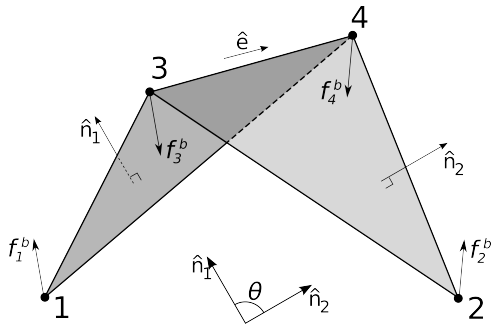
Let us consider two triangle elements of the spring network, connected along a common edge as shown in Fig. 3a. The tension/compression springs are the connections between particles 1-3, 1-4, 3-4, 2-3 and 2-4. The force acts symmetrically along the vector $\mathbf{x}_{ij}$, and both linear and angular moments are conserved. For a suitably uniform mesh, the force as given by Eq. 37 provides good results with a constant value of κ_t, and this approach has been considered in the subsequent simulations. In the case of a highly anisotropic mesh it is necessary to employ an anisotropic spring stiffness [66, 72]. The damping force is then given by:

$$\mathbf{f}_i^{t,D} = \kappa_{t,D} \left(\frac{\mathbf{x}_{ij}}{|\mathbf{x}_{ij}|} \cdot \mathbf{u}_{ij} \right) \frac{\mathbf{x}_{ij}}{|\mathbf{x}_{ij}|} \tag{38}$$

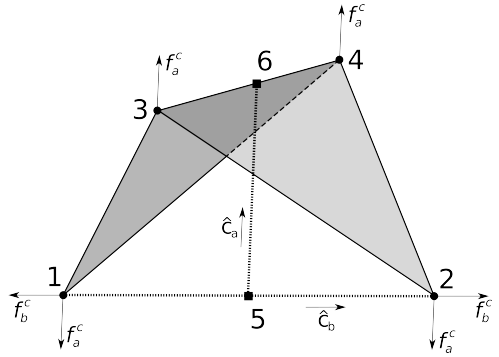
where $\mathbf{u}_{ij} = \mathbf{u}_i - \mathbf{u}_j$ and is the relative velocity between particles i and j. The damping force is projected onto the unit vector $\mathbf{x}_{ij}/|\mathbf{x}_{ij}|$, ensuring again the conservation of the linear and angular momentum [74].

3.2. Orthogonal torsion bending springs (uncoupled)

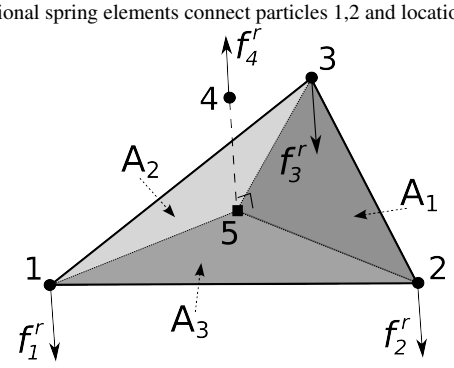
The torsion bending forces are constructed to act orthogonal to the tension/compression springs modes. By uncoupling these two sets of forces, the material properties of in-plane and



a) Configuration of torsion bending element (uncoupled).



b) Configuration of cross-linked bending element (coupled).
Additional spring elements connect particles 1,2 and locations 5,6.



c) Configuration of repulsive forces.
The abutting particle is number 4.

Figure 3: Detail of spring network for a) torsion bending elements, b) cross-linked bending elements, c) repulsive forces. Particles are represented by circles and other locations by squares.

flexural stresses can be considered independently. Following the work of [75, 76], the modes can

be elegantly presented as (see Fig. 3a):

$$\begin{aligned}
\eta_1 &= |E|\frac{N_1}{|N_1|^2} \\
\eta_2 &= |E|\frac{N_2}{|N_2|^2} \\
\eta_3 &= \frac{(\mathbf{x}_1 - \mathbf{x}_4)\cdot E}{|E|}\cdot\frac{N_1}{|N_1|^2} + \frac{(\mathbf{x}_2 - \mathbf{x}_4)\cdot E}{|E|}\cdot\frac{N_2}{|N_2|^2} \\
\eta_4 &= -\frac{(\mathbf{x}_1 - \mathbf{x}_3)\cdot E}{|E|}\cdot\frac{N_1}{|N_1|^2} - \frac{(\mathbf{x}_2 - \mathbf{x}_3)\cdot E}{|E|}\cdot\frac{N_2}{|N_2|^2}
\end{aligned} \tag{39}$$

where $N_1 = (\mathbf{x}_1 - \mathbf{x}_3)\times(\mathbf{x}_1 - \mathbf{x}_4)$, $N_2 = (\mathbf{x}_2 - \mathbf{x}_4)\times(\mathbf{x}_2 - \mathbf{x}_3)$, $E = (\mathbf{x}_4 - \mathbf{x}_3)$. Note that these modes have units $distance^{-1}$, such that for a given torque τ, the force at each node is given by $\mathbf{f}_j^b = \tau\,\eta_j$, for $j = 1, \ldots, 4$. These relations can be obtained by considering orthogonality of bending modes [75, 76], or simply by taking moments about the different edges. The torsion bending forces must be proportional to these modes to ensure they will not couple with the tension/compression springs. These modes are independent of the particle masses, such that non-uniform particles may be used if desired without changing these modes. We note in passing that an efficient means of computing the bending angle is $\theta = \text{acos}(\hat{n_1}\cdot\hat{n_2})$, where the correct sign is given by $\hat{e}\cdot(\hat{n_1}\times\hat{n_2})$.

Effects of mesh irregularity and non-uniformity are more pronounced and we follow [75, 76] to normalise these forces by the ratio of $|E|^2/(|N_1| + |N_2|)$, which is the ratio of common edge length to bending element span. Here a linearly increasing bending force is considered, hence $\tau = \kappa_b\,(\theta - \theta_0)$, where κ_b is the angular spring stiffness and θ_0 is the rest angle. The final form of the force resisting bending is:

$$\mathbf{f}_j^b = \left(\frac{|E|^2}{|N_1| + |N_2|}\right)(\theta - \theta_0)\,\kappa_b\,\eta_j \qquad ; \quad \text{for } j = 1, \ldots, 4 \tag{40}$$

We have chosen $\theta_0 = 0$ for all the bending elements, thus assuming that the membrane relaxes to a flat configuration, and effectively prescribing a pre-stress of the membrane in bending.

The damping force for these bending modes is given by:

$$\mathbf{f}_j^{b,D} = -\kappa_{b,D}\,|E|\,(d\theta/dt)\,\eta_j \qquad ; \quad \text{for } j = 1, \ldots, 4 \tag{41}$$

where $\kappa_{b,D}$ is the damping coefficient. The term $|E|$ is used for considerations of mesh scaling [75, 76]. Since $\eta_j = \partial\,\theta/\partial\,\mathbf{x}_j$ [76], the rate of change of the angle is given by:

$$d\theta/dt = \sum_j^4 \eta_j \cdot \mathbf{u}_j \tag{42}$$

where $\mathbf{u}_j$ is the velocity vectors at node j.

3.3. Cross-linked bending springs (coupled)

Forces that resist bending may also be modelled by cross-linked tension/compression springs [73, 77, 78, 79, 80]. The forces generated by the cross-links will not be orthogonal to the in-plane membrane stresses, effectively coupling the two sets of forces. These cross-linked springs will therefore resist both bending and in-place stresses [80].

There are several advantages to using this cross-linked approach to account for the bending. Firstly, the spring forces are simpler and allow for easier software implementation. Secondly, we note that for the torsion bending forces the modes have units $distance^{-1}$, such that distortions to the mesh elements resulting in skewed triangles, may generate large bending forces during the simulation. These large forces will result in smaller integration times and may even jeopardise the stability of the method, though in practice this was not seen for the simulations effected in the current study. Thirdly, a linearised implicit time integration scheme is easily obtained, as detailed in the Appendix B. A final remark is that due to the coupling of the cross-linked bending springs to the tension/compression springs that resist in-plane stresses, the resulting material will have mild non-linear properties [80], which may be exploited to better describe the lipid bi-layer membranes.

Let us consider Fig. 3b, where the bending element is now endowed with two extra springs. The first spring connects the particles at the span extremities, acting only on the particles 1 and 2

$$\mathbf{f}^c_{b,1} = \kappa_{c2}\,\frac{(|\mathbf{c}_b| - L_{b,0})}{L_{b,0}}\,\hat{\mathbf{c}}_b \;\;;\quad \mathbf{f}^c_{b,2} = -\mathbf{f}^c_{b,1} \;\;;\quad \mathbf{f}^c_{b,3} = 0 \;\;;\quad \mathbf{f}^c_{b,4} = 0 \tag{43}$$

where $\mathbf{c}_b = \mathbf{x}_2 - \mathbf{x}_1$ and $\hat{\mathbf{c}}_b = \mathbf{c}_b/|\mathbf{c}_b|$. This spring stiffness is given by κ_{c2}.

The second spring connects marker locations 5 and 6, which are located at the mid-points along the common edge (particles 3,4) and the span extremities (particles 1,2). The force generated does not act on these marker locations, but is imposed directly on the particles:

$$\begin{aligned}
\mathbf{f}^c_{a,j} &= \kappa_{c1}\delta_j\,\frac{(|\mathbf{c}_a| - L_{a,0})}{L_{a,0}}\,\hat{\mathbf{c}}_a \qquad ;\quad \text{for } j = 1,2 \\
\mathbf{f}^c_{a,j} &= -\,\kappa_{c1}\delta_j\,\frac{(|\mathbf{c}_a| - L_{a,0})}{L_{a,0}}\,\hat{\mathbf{c}}_a \qquad ;\quad \text{for } j = 3,4
\end{aligned} \tag{44}$$

where $\mathbf{c}_a = \delta_1\mathbf{x}_1 + \delta_2\mathbf{x}_2 + \delta_3\mathbf{x}_3 + \delta_4\mathbf{x}_4$, and $\hat{\mathbf{c}}_a = \mathbf{c}_a/|\mathbf{c}_a|$. In the present work we choose points 5 and 6 to be mid-points, hence $\delta_1 = \delta_2 = -1/2$ and $\delta_3 = \delta_4 = 1/2$, and $\sum \delta_i = 0$. As noted in [80], different choices for δ_i and consequently the locations of 5 and 6, are also possible. These forces conserve both linear and angular momentum.

The rest lengths of the springs are set to $L_{a,0} = 0$, and $L_{b,0} = L_0\sqrt{3} = 0.9\,l_0\sqrt{3}$ (assuming equilateral triangle elements). Hence, as for the torsion bending spring, we assume that the membrane relaxes to a flat configuration. The damping forces for these cross-linked springs are trivially derived similarly to Eq. 38, and are denoted by $\mathbf{f}^{c,D}_a$ and $\mathbf{f}^{c,D}_b$. Finally, we state that the bending springs using this cross-linked approach are given by $\mathbf{f}^b = \mathbf{f}^c_a + \mathbf{f}^c_b$ and $\mathbf{f}^{b,D} = \mathbf{f}^{c,D}_a + \mathbf{f}^{c,D}_b$.

3.4. Surface-crossing repulsive springs

The cell membranes are modelled as a spring network, connecting specific particles in the MPS discretisation. The spring network does not explicitly act as a barrier and in order to avoid particles crossing through the triangular mesh elements in either direction, repulsive forces are applied to push away any abutting particle, and consequently also the membrane particles in order to conserve moments. Following the notation in Fig. 3c the repulsive forces are given by:

$$\begin{aligned}
\mathbf{f}^r_j &= -\kappa_r\,\hat{\mathbf{n}}\left(\frac{d_e}{r_{4,5}} - 1\right)\frac{A_j}{A} \qquad 0 < r_{4,5} < d_e \quad ;\quad \text{for } j = 1,\ldots,3 \\
\mathbf{f}^r_4 &= -\sum_{j=1}^{3}\mathbf{f}^r_j
\end{aligned} \tag{45}$$

where $r_{4,5} = |\mathbf{x}_4 - \mathbf{x}_5|$, $\hat{\mathbf{n}} = (\mathbf{x}_4 - \mathbf{x}_5)/r_{4,5}$ is the unit normal in the direction of the abutting particle, and d_e is the radius of compact support for the repulsive forces.

Point 5 is the closest location to particle 4 on the triangle element. If point 5 lies outside the triangle element, no repulsive force is employed. Moreover it may be the case that repulsive forces may arise from two triangle elements if the membrane is locally non-planar, in which case the forces are summed over all affected elements. While this approach has been successful in the current work, we indicate to the interested reader that more complete repulsive models that check also for proximity to edges and vertices can be found in [81].

The repulsive force weight function in Eq. 45 is the same as that in Eq. 1, and is aimed at smoothly enforcing no element crossing but limiting its radius of influence to very short distances. The radius of compact support is chosen to be small $d_e = 0.8 l_0$, and the spring stiffness was set to be small $\kappa_r = 10^4$ in comparison to the other membrane forces, as detailed below. In practice, these forces are small, infrequently formed, and are used as a precaution measure rather than a necessary approach to avoid particles crossing the membrane. Importantly, the repulsive surface force is used to provide a simple lubrication force to capture the unresolved hydrodynamic interactions of the solid bodies at very short distances.

4. Results and Discussion

For all the results presented, the material properties of the fluid (plasma and red blood cells) were set as the density $\rho = 1000$ kg m^{-3} and the dynamic viscosity $\mu = 0.001$ Pa s. The cells are assumed to have the same fluid properties as the plasma. In order to impose different density and viscosity, a local weighted averaging can be used as a first approximation [41]. The particle discretisation was set to $h_0 = 0.4$ μm following [12], while other works have used slightly coarser ($h_0 = 0.46$ μm [21]) or finer ($h_0 = 0.26$ μm [7]) discretisations. In the present study only red blood cells were considered in the flow, and the initial biconcave shape was given as an analytic function proposed in [69]. Finally, we assume that the spring network modelling the cell membrane has a rest bending angle $\theta_0 = 0$ and a rest spring length $L_0 = 0.9\, l_0$, which effectively results in a very small pre-stress. Different cases are simulated, aimed at initially benchmarking results and subsequently increasing in the complexity of the simulations.

The computational domain was initially set using a Cartesian particle distribution, discretising the fluid domain, the wall and the ghost particles. Example cross-sections of the domain are shown in Figs. 2 and 4. This results in a step-like definition of the wall particles, which were chosen by a least-squares fit to the geometry. Ghost particles were then used to pad the domain by at least a distance of r_e. Subsequently the cell membranes, given as a triangular mesh (991 vertices, 1978 triangle faces), were inserted in the fluid domain and neighbouring fluid particles lying too close to the mesh elements were then incrementally removed until the total number of particles was approximately restored to that of the initial Cartesian discretisation. This procedure for setting up the particle layout results in the red blood cells all being slightly different due to the disparate number particles lying internal to the membranes, as can be seen in the example in Fig 2. A fast and robust approach to compute the location of particles as internal or external to the triangular mesh surfaces is given as the signed volume of a tetrahedron [82].

The stresses that the fluid exerts on the vessel wall are reported, which are computed from the Cauchy stress tensor $\boldsymbol{\sigma} = -p\mathbf{I} + \boldsymbol{\tau}$, where $\boldsymbol{\tau}$ is the extra stress and is computed from the strain rate tensor assuming the fluid is Newtonian as $\boldsymbol{\tau} = \mu\left(\nabla\mathbf{u} + \nabla\mathbf{u}^T\right)$. Here the pressure, p, is given by the implicit pressure calculation of Eq. 12 (Step.6), and does not include the pressure drop

imposed as boundary conditions to drive the flow (Step.1). The traction at the wall is given by

$$\mathbf{t} = \boldsymbol{\sigma} \cdot \mathbf{e}_n \tag{46}$$

and the wall shear stress (WSS) is then computed as

$$\boldsymbol{\tau}_w = \mathbf{t} - (\mathbf{t} \cdot \mathbf{e}_n)\mathbf{e}_n \tag{47}$$

where $\mathbf{e}_n$ is the unit outward normal to the wall surface, which was estimated using a least-squares approximation. We remark that when computing the velocity gradient tensor $\nabla\mathbf{u}$, one can directly use the discrete differential operator in Eq. 15, where the summation is over particles in sets $\{\mathcal{P}_\mathcal{F} \dot\cup \mathcal{P}_\mathcal{W}\}$. On computing these stresses we have neglected the presence of the glycocalyx and the endothelial cell layer, which can affect both the motion of the red blood cells and the wall stresses [83, 84, 85].

The orientation and shape of the cells is also reported, and this is given by the gyration tensor [21, 86] which is in effect a proper orthogonal decomposition of points discretising the cell membrane:

$$G = \frac{1}{N_v} \sum_{i=1}^{N_v} (\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}}) \tag{48}$$

where N_v is the number of particles discretising the cell membrane, and $\overline{\mathbf{x}} = \frac{1}{N_v}\sum_{i=1}^{N_v} \mathbf{x}_i$ is the centre of mass of the cell membrane points. The gyration tensor G is a 3×3 symmetric positive definite matrix, with eigenvalues $\lambda_1 \geq \lambda_2 \geq \lambda_3$ that define energy associated to the principal directions, which are the respective orthonormal eigenvectors ξ_1, ξ_2, ξ_3. For red blood cells at rest $\lambda_1 = \lambda_2 > \lambda_3$, with ξ_1, ξ_2 defining the biconcave plane of symmetry, while eigenvector ξ_3 identifies the shortest dimension of the cell, that is the axis of rotational symmetry. In effect this corresponds to an ellipsoidal best-fit to the membrane shape. Since the volume and surface area of the red blood cell are preserved, the ratios of the eigenvalues describe the extent of the deformed shape (as the ratio of the ellipsoid principal axes squared), and the eigenvectors define the orientation of the ellipsoid axes.

4.1. Poiseuille flow

As a simple test case, a periodic pipe of radius $R = 5\ \mu$m and length $L = 23.6\ \mu$m was considered, with a pressure drop of $\Delta p = 8$ Pa. No red blood cells were included in the simulation. For a Newtonian fluid in a pipe, the velocity profile for laminar flow is given by

$$u(r) = \frac{R^2 \Delta p}{4\mu L}\left(1 - \left(\frac{r}{R}\right)^2\right) \tag{49}$$

and $u_{max} = 2\langle u \rangle$, where $\langle u \rangle$ is the mean velocity. The results for this simulation are presented in Fig. 4 for a snapshot in time, where the values have been mapped to a Cartesian layout for ease of inspection. It should be noted that the colouring scheme has been chosen to highlight errors. A line plot shows that the velocity profile is seen to be smooth and captures well the parabolic shape.

Initially surprising is that the apparent maximum velocity is greater than the analytic solution. On closer inspection, by considering the wall particles to contribute by half to the fluid domain, we note that the discretised domain is in fact larger than the intended geometry. Taking this into consideration we find that the adjusted analytic solution shows the method to have numerical

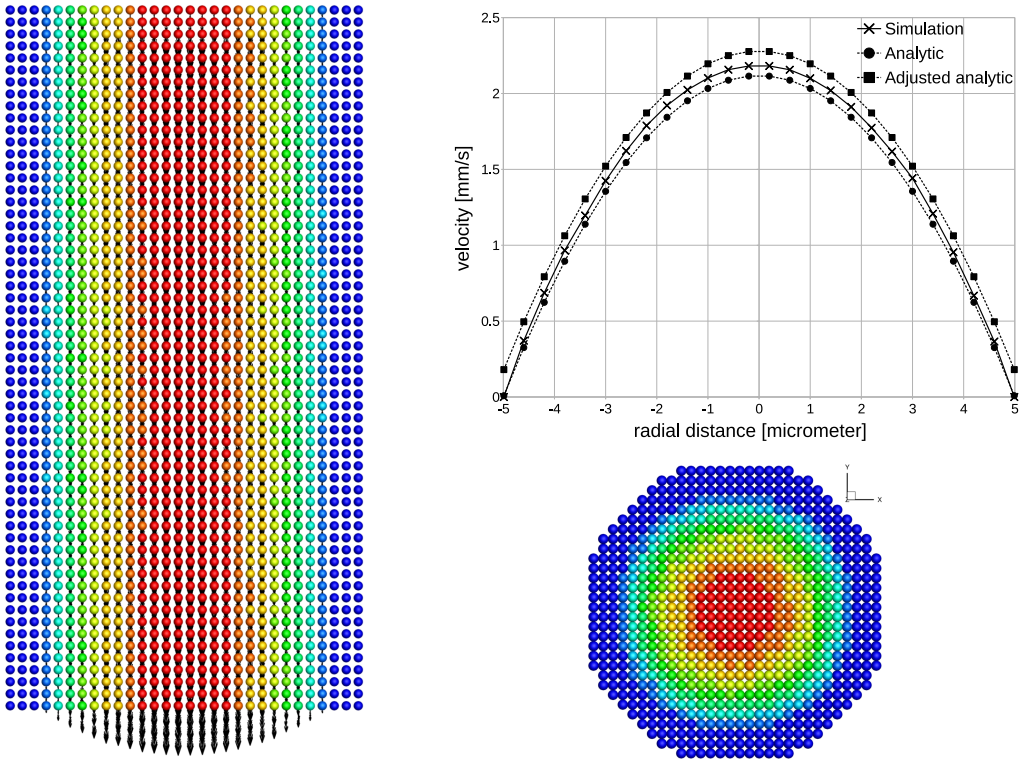
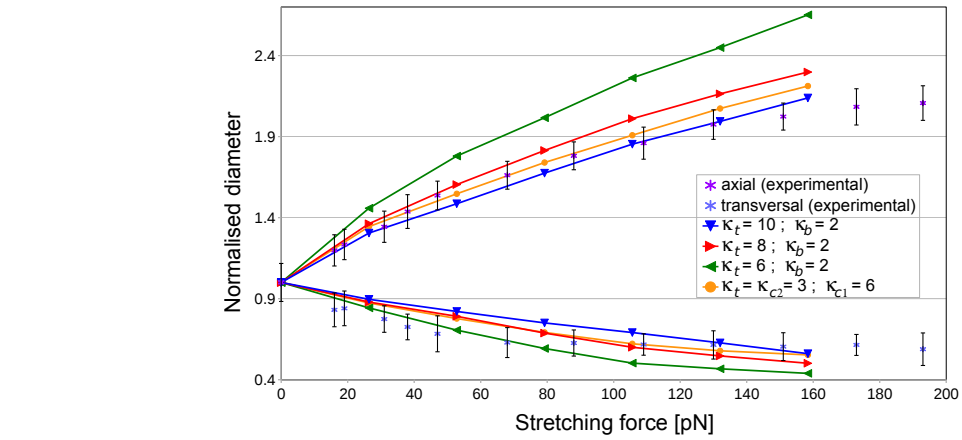


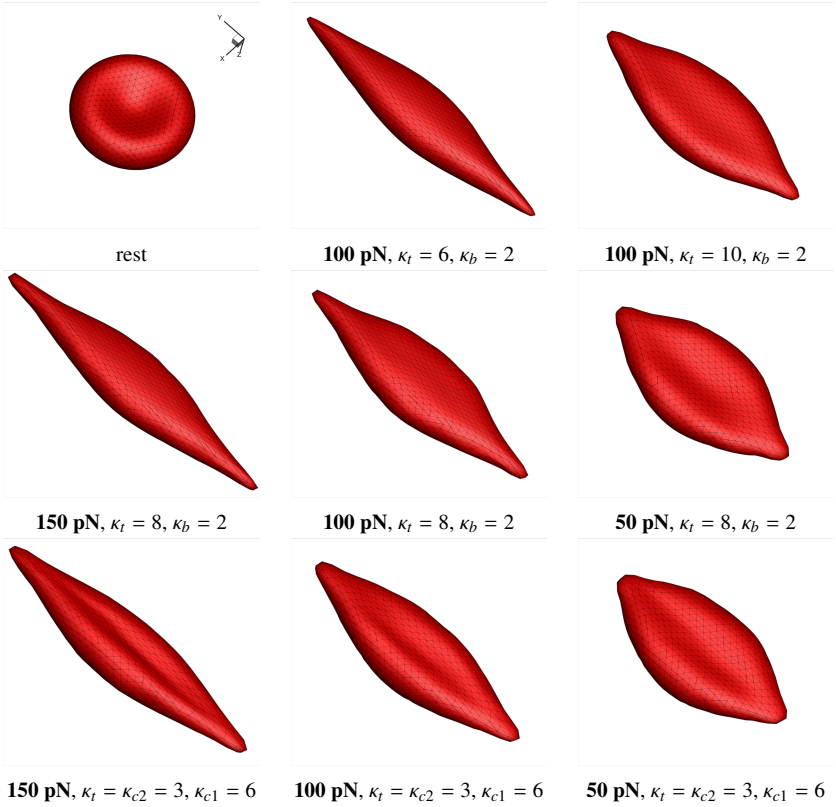
Figure 4: Results for Poiseuille flow in a circular pipe of diameter $D = 10\ \mu$m, taken as a snapshot of the simulation (no averaging involved). The colouring scheme is chosen to accentuate errors. The transversal line profile of velocity shows a smooth solution. The *adjusted analytic* profile takes into account the errors of geometry discretisation.

viscosity, resulting in an increased flow resistance and consequently a reduced flow rate. The numerical viscosity is expected due to the radial weighting function used for approximating values (Eq. 2). A 3% error is observed in the peak velocity between the simulation and the adjusted analytic solution.

4.2. Comparison of spring network models

Different spring stiffness coefficients were tested, simulating a stretch test as reported in experimental work [11]. Note that the stretching force was shared between several particles (15 particles on either side) in order to reflect the experimental results, and avoid pinching the membrane in a single point. Results for the two configurations of springs to resist bending (coupled or uncoupled), are presented in Fig. 5 for a representative selection of parameter variation. The spring damping coefficients are set to zero, since this experimental test is a static test. To the author's knowledge, there is currently no clear experimental data in the literature which can serve to set and benchmark the viscous spring coefficients, though some studies in this direction can be found in [87] and references therein. As a consequence all the spring damping coefficients are set to zero throughout.





rest | **100 pN**, $\kappa_t = 6$, $\kappa_b = 2$ | **100 pN**, $\kappa_t = 10$, $\kappa_b = 2$

150 pN, $\kappa_t = 8$, $\kappa_b = 2$ | **100 pN**, $\kappa_t = 8$, $\kappa_b = 2$ | **50 pN**, $\kappa_t = 8$, $\kappa_b = 2$

150 pN, $\kappa_t = \kappa_{c2} = 3$, $\kappa_{c1} = 6$ | **100 pN**, $\kappa_t = \kappa_{c2} = 3$, $\kappa_{c1} = 6$ | **50 pN**, $\kappa_t = \kappa_{c2} = 3$, $\kappa_{c1} = 6$

Figure 5: Red blood cell static stretch test, comparison with experimental work [11]. $\kappa_r = 4 \times 10^4$ m s^{-2}, $\kappa_t = \kappa_{c1} = \kappa_{c2} = \ldots \times 10^4$ m s^{-2}, $\kappa_b = \ldots \times 10^{-2}$ m^2 s^{-2}.

Since elastic spring models are used, the forces imposed on the red blood cells result in an approximately linear stretching response. Non-linear models have also been proposed in the literature [8, 9, 69], with greater accuracy in fitting simulations to the experimental data. While Cosserat nets described in [88] have so far not been used to model cell membranes, they may provide suitable results also. Here we focus our attention to describe the resistance to bending in a fashion that is either coupled or uncoupled to the in-plane forces. The resulting shapes of the red blood cells after stretching differ slightly between the two bending models: the torsion springs approach (uncoupled) yields a rounded shape, while the cross-linked springs approach (coupled) maintains the biconcave shape to a certain extent. From the experimental work of [11], it is not clear as to which of the representations closer mirrors the more physiological configuration.

The spring stiffness coefficients chosen to be the best fit to the experimental data are the results of exhaustive tests, and will be used for all subsequent simulations. The parameters can be summarised as

- Configuration 1 (uncoupled):

  $\kappa_t = 8 \times 10^4$ m s^{-2}, $\kappa_b = 2 \times 10^{-2}$ m^2 s^{-2}, $\kappa_r = 4 \times 10^4$ m s^{-2}

- Configuration 2 (coupled):

  $\kappa_t = 3 \times 10^4$ m s^{-2}, $\kappa_{c1} = 6 \times 10^4$ m s^{-2}, $\kappa_{c2} = 3 \times 10^4$ m s^{-2}, $\kappa_r = 4 \times 10^4$ m s^{-2}

The coefficients reported in [12] (after conversion of units) are $\kappa_t = 6.97 \times 10^4$ m s^{-2}, $\kappa_b = 14.5 \times 10^{-2}$ m^2 s^{-2}. On comparing these to the parameters chosen in Configuration 1 we note a greater tension/compression spring stiffness and smaller bending stiffness. Since the spring network model used in [12] was not uncoupled as is Configuration 1, the bending force in their work also influenced the in-plane stresses. By decoupling the forces, the tension/compression spring stiffness must be higher as no longer supported by the bending springs, and furthermore the bending spring now acts only in the out of plane modes, requiring less resistance. It should be stressed that, while the above configurations provide good fit to the experimental data and between themselves, it is not possible to ensure they have the same macroscopic membrane properties due to the discrete modelling approximation of the spring network technique.

We note in passing that the cell membrane surface area change is 4% when subjected to the more emphatic stretching force of 150 pN and for the parameter choice of Configuration 1. The spring coefficients adopted in Configurations 1 and 2 may be converted to alternative and more conventional units to compare with values proposed in [7, 12], obtaining $\kappa_t = 1.15 \times 10^{-5}$ N/m, $\kappa_b = 0.33 \times 10^{-11}$ N. Approximate conversions to continuum model descriptions may be obtained following [7, 66, 67, 68].

4.3. Constriction

We next observe the dynamic response of the red blood cells given by Configuration 1 and 2, as they relax after passing through a periodic circular constriction. The diameter of the constriction is $D = 6$ μm and of the larger conduit $D = 24.8$ μm, while the length of the domain is 49.6 μm. The boundary condition to drive the flow was imposed as a pressure drop $\Delta p = 20$ Pa across the domain. The results for the two spring models are shown in Fig. 6 for a set of snapshots in time as the red blood cell traverses the domain on the fourth periodic cycle. The results for Configuration 2 had on average a greater time step in comparison to Configuration 1, resulting in $\sim 10\%$ reduction of total time integration steps.

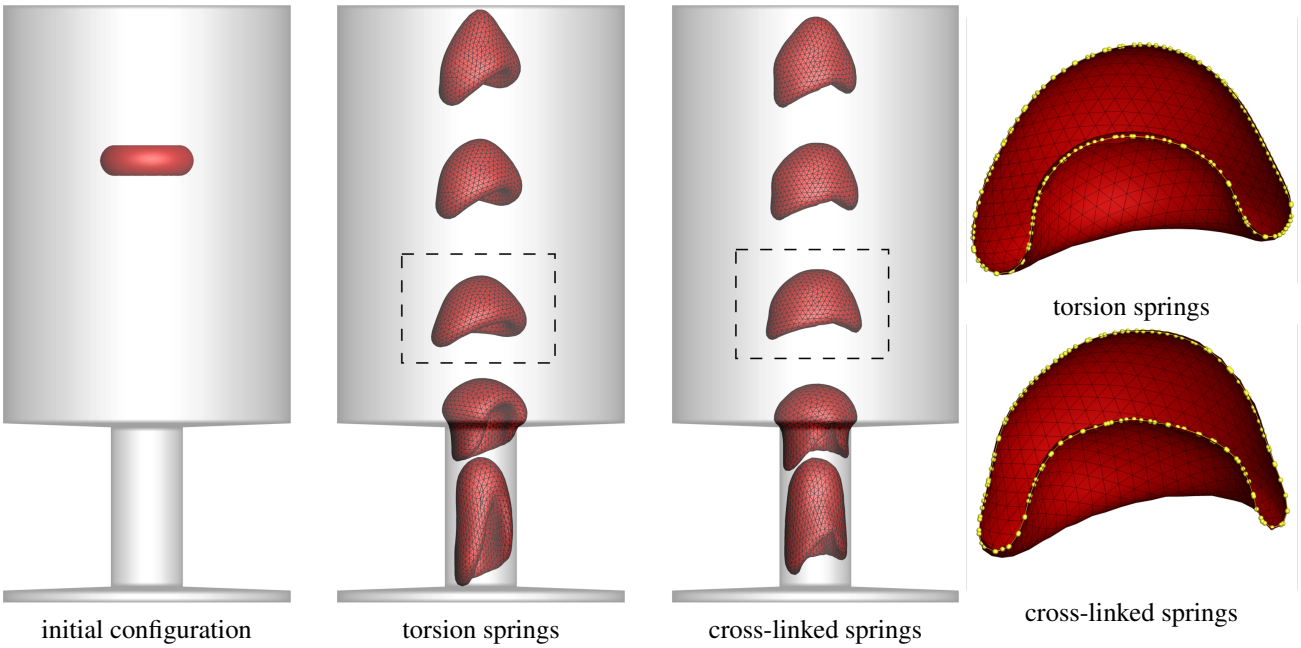
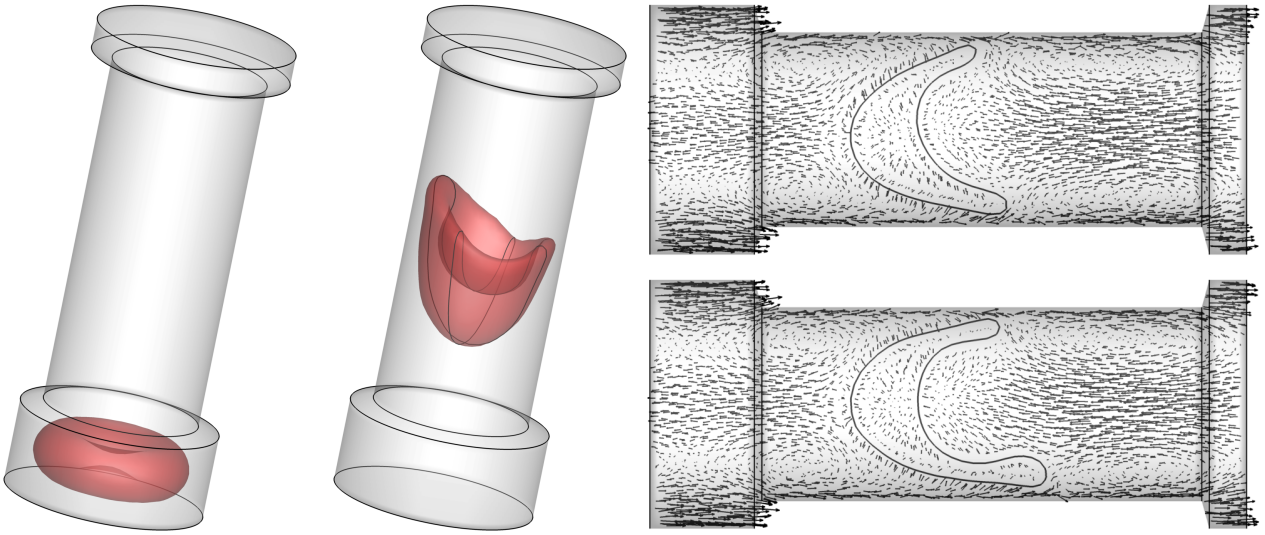


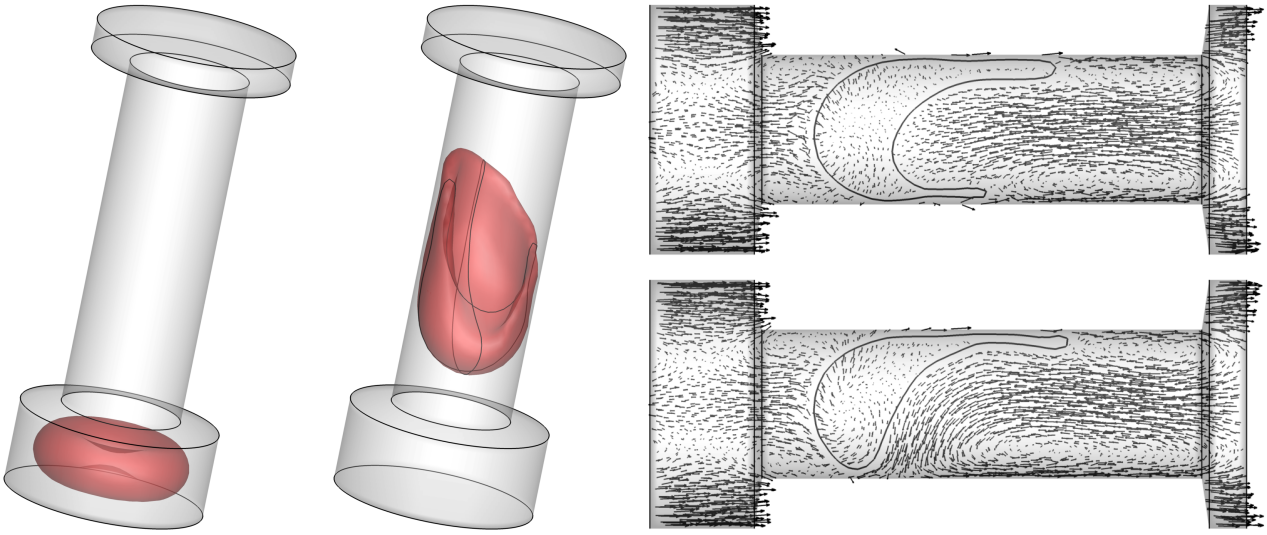
Figure 6: Left: initial configuration. Middle: single red blood cell passing through a constriction (diameter= 6 μm), at different time snapshots. Two bending spring models are employed. Right: detail of middle frame (dashed box), with the red blood cell cut in half to show the cross-section. Flow is bottom to top.

While the response is remarkably similar, especially since the results shown are for the fourth traversing cycle, we note that the use of the torsion springs tend to round the domain more than for the cross-linked springs. This reflects the findings of Fig. 5 for the stretch test. While the damping springs have been set to zero, experimental investigation from such a dynamic test is well suited to estimate the viscous properties of the membranes. For the remainder of this work the uncoupled spring network model will be employed, using the coefficients for Configuration 1.

We next investigate in greater detail the flow of a single red blood cell in two constrictions, of diameters $D = 7.8$ μm and $D = 6$ μm, with pressure difference boundary conditions respectively of 12.1 Pa and 20 Pa. The resulting red blood cell shapes observed, presented in Fig. 7, show similarities reported in both numerical [40] and experimental [13, 89] works. In order to study the disturbance of the flow due to the red blood cell motion, the mean co-axial velocity of the cell is subtracted from the fluid domain. By doing so we observe the flow in a Lagrangian frame relative to the red blood cell [40]. We note that the wakes in both constrictions are composed of a vortex ring. In the case of the constriction with diameter $D = 7.8$ μm, the cell deformation is approximately rotationally symmetric and the vortex ring is also uniform. In the constriction of diameter $D = 6$ μm, the cell deformation has a leading edge and the vortex ring has a greater diameter on one side to fill the resulting concave cell shape. This relative motion of the flow plays an important role in the intra-cell interactions, resulting in the complex migration trajectories and non-Newtonian rheology as reported in [15].



Δp = 12.1 Pa, diameter=7.8 μm, $\langle u \rangle$ = 1.2 mm/s



Δp = 20 Pa, diameter=6 μm, $\langle u \rangle$ = 1.1 mm/s

Figure 7: Left: initial configuration. Middle: view of red blood cell during the twentieth crossing of the domain (flow is top to bottom). Right: cross-section in two orthogonal planes, showing relative velocity vector field. $\langle u \rangle$ is the mean (co-axial) velocity of the red blood cell (flow is right to left).

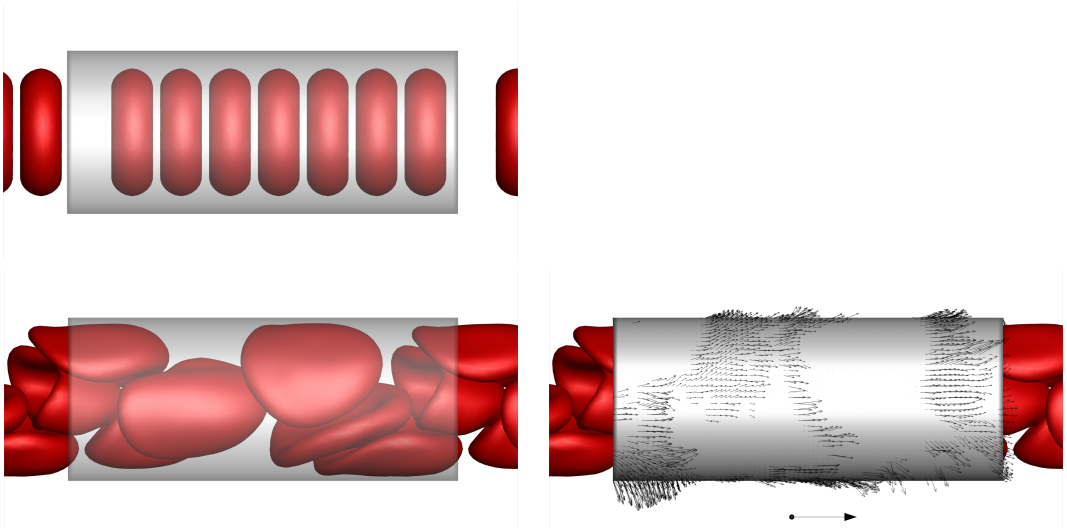


Figure 8: Pipe 1, at snapshot T=4.3 s. Top left: initial configuration. Bottom left: red blood cell distribution. Bottom right: stresses exerted on the conduit wall. Vector length scale is referenced to 10 Pa, given below the figure. Flow is left to right.

4.4. Straight pipe

Two periodic straight pipes of circular cross-section were considered to investigate simple flow characteristics of the cells. The test cases are summarised as follows:

- Pipe 1:

  diameter = 10 μm, length = 23.6 μm, pressure drop = 8 Pa, number of RBCs = 7, hematocrit = 50%, total simulation time = 4.3 s

- Pipe 2:

  diameter = 22.2 μm, length = 49.6 μm, pressure drop = 10 Pa, number of RBCs = 36, hematocrit = 24%, total simulation time = 1.2 s

Results at the final time instance for the two simulations are shown in Figs. 8 and 9. Transients due to the initial simulation setup are washed out after $\sim$ 0.2 s. Statistics of these simulations, excluding transients, are presented in Tables 1 and 2.

In the case of Pipe 1, we observe the behaviour of the cells in a state of high hematocrit in a small vessel. At the time instance T = 4.3 s, the mean co-axial velocity and the mean velocity magnitude, are $\langle u \rangle_{fluid}$ = 0.70 mm/s and $\langle |\mathbf{u}| \rangle_{fluid}$ = 0.82 mm/s for the entire fluid particles ($\{\mathcal{P}_{\mathcal{F}}\}$), and $\langle u \rangle_{RBC}$ = 0.76 mm/s and $\langle |\mathbf{u}| \rangle_{RBC}$ = 0.92 mm/s for the red blood cells only. For a Newtonian fluid the analytic solution is $\langle u \rangle_{analytic}$ = 1.06 mm/s. By considering the ratio $\langle u \rangle_{fluid} / \langle u \rangle_{analytic}$ = 0.66, it is evident that the presence of the red blood cells plays an important role in increasing the apparent viscosity, and hence the resistance. The difference between the velocity magnitude and the co-axial component indicates a considerable transversal flow, though the mean velocity of each red blood cell is found to be approximately co-axial. The transversal flow is seen to affect the stress distribution the blood exerts on the vessel wall, shown in Fig. 8,

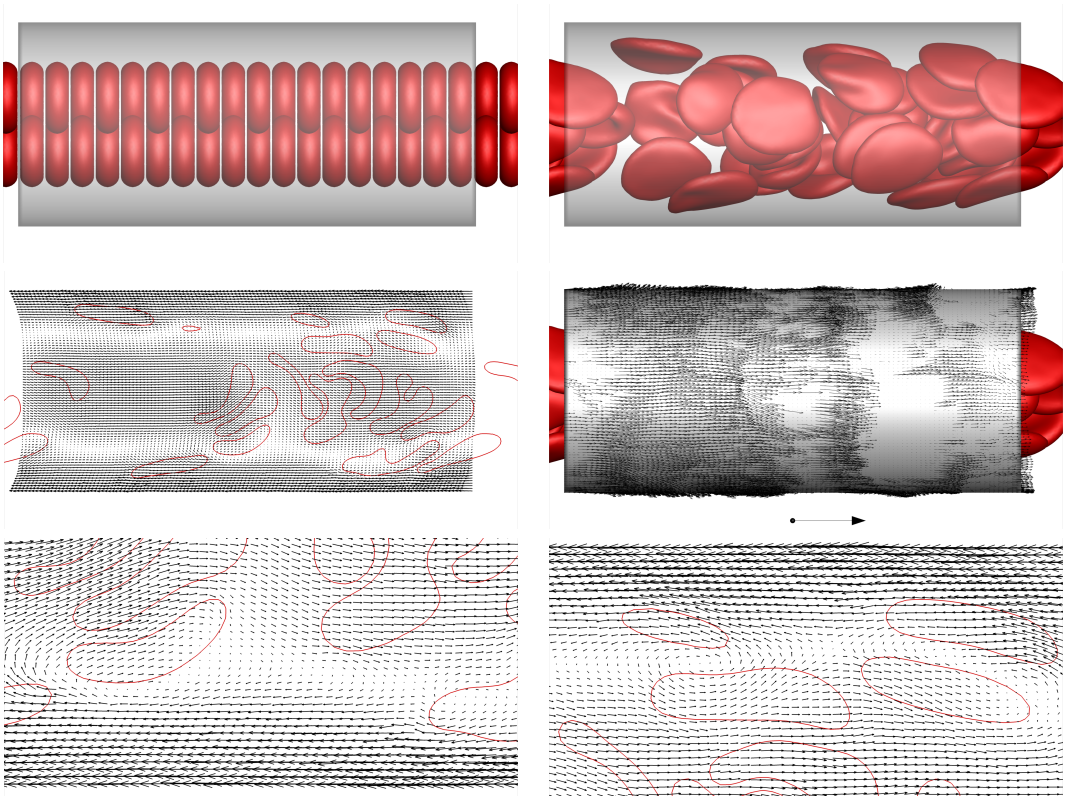


Figure 9: Pipe 2, at snapshot T=1.2 s. Top row: (left) initial configuration; (right) red blood cell distribution. Middle row: (left) cross-section of domain showing relative velocity vector field; (right) stresses exerted on the conduit wall. Bottom row: details of cross-section along bottom wall (left) and top wall (right). Vector length scale of stresses is referenced to 10 Pa, given below the figure. Flow is left to right.

		flux count (mean)	flux count (standard deviation)	residence time (%)
Pipe 1	core (0 – 1.7 μm)	10.4	2.9	85
	external (1.7 – 3.5 μm)	9.7	3.0	15
Pipe 2	core (0 – 2.9 μm)	5.2	4.2	18
	middle (2.9 – 5.8 μm)	10.0	4.2	38
	external (5.8 – 8.7 μm)	4.6	3.2	44

Table 1: Statistics of red blood cell radial migration, given by number of times a cell enters a different radial partition. Mean values and standard deviation of the flux count, as well as the mean residence time in each radial partition, are computed for the Pipe 1 and Pipe 2 simulations, excluding the transients at the start of the simulation. Note that Pipe 2's middle radial partition has two neighbouring partitions, and hence the flux count is (approximately) twice as large.

where we identify that the normal stresses on the surface are of the same order of magnitude as the tangential components. The stresses exerted on the vessel wall, which may play an important role in mechanotransduction signalling, are strongly influenced by the presence of the traversing cells. The stresses exerted on the vessel wall, as fluctuations about mean values, have been recently reported in [90].

In the simulation of Pipe 2, which has a reduced hematocrit value and a larger diameter, we observe that the behaviour of the cells is strikingly different. From Fig. 9 we note that a cell free layer is established, and furthermore we visually identify that the cells located closer to the walls are oriented to expose a greater surface area towards the walls. At the time instance T = 1.2 s, the mean co-axial velocity and the mean velocity magnitude, are $\langle u \rangle_{fluid}$ = 2.62 mm/s and $\langle |\mathbf{u}| \rangle_{fluid}$ = 2.62 mm/s for the fluid particles, and $\langle u \rangle_{RBC}$ = 3.66 mm/s and $\langle |\mathbf{u}| \rangle_{RBC}$ = 3.67 mm/s for the red blood cells only. For a Newtonian fluid the analytic solution is $\langle u \rangle_{analytic}$ = 3.11 mm/s. The ratio $\langle u \rangle_{fluid} / \langle u \rangle_{analytic}$ = 0.84 indicates a reduced apparent viscosity in Pipe 2 compared to Pipe 1. Furthermore, the velocities of both fluid and red blood cells are found to be effectively co-axial. We also note that the red blood cells have on average a higher velocity than the fluid for both cases studied, though this is more marked for Pipe 2 for which the cells lie in the core region of the pipe and establish a cell-free layer close to the conduit wall. This phenomenon corresponds to the Fåhræus effect. We also note that, in contrast to the results for Pipe 1, the stresses exerted on the vessel wall for Pipe 2 are seen to be predominantly tangential to the wall and are less affected by the flow of the cells. This is due to the higher flow rate as well as the presence of a cell free layer, that separates both the cells and the disturbed flow field from the vessel wall.

The migration of the red blood cells in the radial direction has important physiological implications, including exchange processes. With the inclusion of platelets and white blood cells, the radial migration is expected to differ, with implication especially to thrombosis and inflammation. In order to consider cell migration, the conduits are radially partitioned and the flux of the cells between these regions is reported in Table 1. The cell location is considered to be its centre of mass, and the radial partitions are chosen by considering the maximum radial location of red blood cells recorded during the simulation, and then dividing this maximum radial distance linearly. In order to evaluate the flux between regions, the standard deviation of the flux count and the residence times in the partitions are also presented to gauge the effect of small oscillations across the partitions with respect to longer period migrations. The mean flux count is found to be $\sim$ 10 for Pipe 1 and $\sim$ 5 for Pipe 2. Due to the different simulation times, we note that there

		$\langle \lvert \mathbf{u} \rvert \rangle$ (mm/s)	$\langle \lambda_1/\lambda_3 \rangle$	$\langle \measuredangle(\langle \mathbf{u} \rangle_{RBC} \cdot \xi_3) \rangle$ (degrees)	$\langle \lvert \mathbf{f}_s \rvert \rangle$ (pN)
Pipe 1	core ($0 - 1.7\ \mu$m)	0.96	2.0	62.3	2.8
	external ($1.7 - 3.5\ \mu$m)	0.78	2.6	73.7	2.6
Pipe 2	core ($0 - 2.9\ \mu$m)	4.3	2.3	65.2	2.6
	middle ($2.9 - 5.8\ \mu$m)	4.2	2.8	74.3	2.6
	external ($5.8 - 8.7\ \mu$m)	2.9	3.2	74.8	2.4
Capillary		0.5	2.6	60.8	2.7

Table 2: Mean values for: velocity magnitude of the red blood cells; stretch ratio of ellipsoid shape; angle between red blood cell mean velocity and ellipsoid minor axis; resulting internal membrane forces. The statistics exclude transients at the start of the simulation.

is in fact a greater radial flux (subsequently convective stir mixing) in the case of Pipe 2. In the smaller vessel the motion is caused by a greater intra-cell interaction resembling moving around obstacles, while in the large vessel the interactions are effected by the relative flow field and fluid mechanic forces. The reason for these differences includes vessel diameter, flow rate and hematocrit. The residence time of the red blood cells in each of the partitions is predominantly in the core of the conduit for Pipe 1, and towards the external region for Pipe 2.

Analysis of the red blood cell motion in the conduits is reported also in Table 2, and the key findings are now summarised. From the orientation of the cells in the different partitions, we note that for both simulations the minor axis ξ_3 is oriented more towards the radial direction in the external division, in comparison to the core flow region. Though not reported in the table, it was found that the principal orientation (ξ_1, or ξ_2 in the case $\xi_1 \approx \xi_2$) of each cell was closely aligned to mean velocity of the cell, which was overwhelmingly co-axial. In the case of Pipe 2, the mean velocity in the core and middle partitions are similar, indicating that the presence of the cells locally increases the apparent viscosity, giving rise to a shear-thinning non-Newtonian rheology. The stretch ratio λ_1/λ_3 was greater with increased radial distance, which highlights that the higher velocity gradient near the vessel walls was responsible for this unidirectional deformation. This high velocity gradient is also responsible for the more pronounced alignment of the minor axis ξ_3 to the radial direction. The mean membrane internal force, averaged over all the cells during the simulation, is noted to be greater in the core region due to increased membrane bending, curiously indicating that the intra-cell interaction causes a greater membrane strain than the higher velocity gradient experienced in the external region (which was noted to induce the greatest stretch ratio λ_1/λ_3).

Finally, a cross-section of the flow field is shown in Fig. 9, where the mean co-axial velocity of the red blood cells has been subtracted from the velocity field. A rather flat velocity profile at the centre of the conduit is evident. The flow field and interactions are now too complex to understand as a whole, however we take two detailed regions and identify clearly from this relative motion the vortex patterns in the wake and in between the cells. The motion and orientation of the cells resemble to some extent the findings reported as early back as [91].

4.5. Capillary section

The geometry used in this test case is that of a retina capillary, reconstructed based on the work of [92] and shown in Fig. 10. The reconstruction procedure makes use of radial basis

function interpolation, and is detailed in Appendix C. The scale of the vessels has been set to have a vessel diameter $D = 10\,\mu$m, and the geometry has been altered to ensure periodicity of the flow sections. The two periodic inflow/outflow sections are in the x and y-axis directions as seen in Fig. 11. The pressure drop across both periodic sections was chosen to be 20 Pa in the positive axis direction. The resulting mean velocity magnitude at time $T = 0.76$ s is $\langle|\mathbf{u}|\rangle_{fluid} \approx 0.42$ mm/s for all the fluid particles ($\{\mathcal{P}_{\mathcal{F}}\}$), and $\langle|\mathbf{u}|\rangle_{RBC} \approx 0.59$ mm/s for the red blood cells only. The choice of vessel dimension and pressure drop were chosen to reflect reasonable physiological conditions, though no experimental values were available for comparison. The total simulation time is 1.2 s, and the first 0.2 s are discarded from the analysis due to transients caused by the simulation initial setup. Details of the simulation are given below. We note in passing that the cell membrane area change recorded during the course of this simulation was 2% on average and 10% peak.

- Capillary:

  approximate diameter = 10 μm, planar dimensions = $74.8 \times 97.6\,\mu$m, number of RBCs = 85, hematocrit = 45%

Comparing the results of this more complex geometry to those of the straight periodic pipes, we notice from Table 2 that the mean properties are largely equivalent even if the average velocity magnitude is less. Of interest we see that the flow in the capillary shows greater similarity to the core flow regions of the straight pipes. From a snapshot of the simulation at $T = 0.76$ s, shown in Fig. 11, it is evident that there is no cell-free layer development as was also reported for Pipe 1. We do however note a scarcity of red blood cells in certain portions of the conduits, and this is caused by the reduced flow in these segments. The traction of the vessel walls appear to be principally due to the pressure in these regions of low flow velocity, while elsewhere the wall shear stress has a greater contribution. The patterns of traction exerted on the vessel wall are not as evident as in the case of the pipe geometries due to the complex shape of the vessel, however the influence of the cells is still apparent, especially at bifurcations.

A planar cross-section of the domain is presented in Fig. 11, where the velocity vectors are drawn. The flow distribution clearly shows the branches with higher flow rates. A detail of the central geometry region is also presented, in which the flow separation at the junction of two branches is seen. From this detail we can note that despite the lower resolution accuracy of the MPS method, such features are well captured and resolved. While it is not possible to provide a relative motion velocity field, due to the different orientation of the vessel segments, it is still possible to discern that the flow is locally affected by the presence of the cells.

5. Conclusions

The theory and implementation of the moving particle semi-implicit (MPS) method has been presented in great depth, with references to the main advances in the field. The original numerical scheme has been modified to include: i) pressure drop boundary conditions for periodic domains; ii) implicit solution of the viscous terms (irrotational field) to improve stability and computational time; iii) use of body force terms to account for red blood cell membranes. Stability criteria based on the particle advection through a Courant-Friedrichs-Lewy (CFL) condition, resulted in a stable algorithm for all simulations performed. These additions have successfully allowed for the simulations of red blood cells in complex periodic geometries, as an example of a multi-body flow problem.

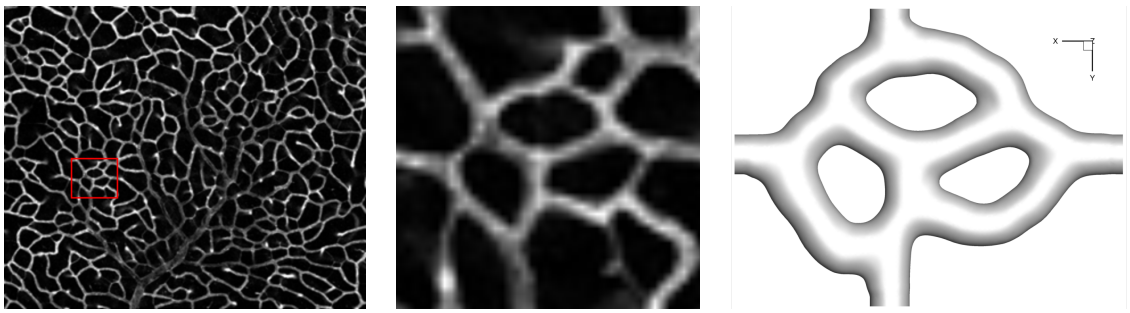


Figure 10: Detail of images of a capillary network, from [92], used as inspiration for the periodic capillary geometry used (diameter= 10 μm). Original image scale from experimental data is unknown.

The cell membranes were modelled using a spring network approach, and two approaches were detailed in great depth. The comparison of these two approaches was against experimental data of static stretch tests, and simulations of the post-constriction dynamic relaxation. It is not possible to set exactly the same macroscopic membrane properties for the two spring network approaches, however similar results were obtained with good comparison to the experimental results.

Results of the simulations show good comparison with reported phenomena, such as the presence of a cell free layer and shear-thinning non-Newtonian rheology. The use of a pressure drop boundary condition allows for the simulation of complex geometries in a simple manner. The orientation of the cells with respect to the bulk direction of motion indicates a tendency for the cells to expose a greater surface area to the vessel wall the closer they approach it. The elongation of the cells also increases with increasing redial position. Both phenomena are driven by the high fluid velocity gradient near the wall. The internal membrane forces are however seen to be greater within the core of the flow and are related to the membrane bending. Effects of stress memory, that will result in the cytoskeleton and membrane restructuring, both important to study flow-induced trauma to the blood cells [93], have not been included here.

Stresses exerted on the conduit wall by the flow field show that the suspended cells induce a complex pattern. This is seen more clearly in slower flow and small diameter vessels, and the presence of the cell free layer in larger vessels diminishes these non-uniform patterns. The presence of the glycocalyx and the endothelial cells, which will affect the transmission of forces onto the wall, have not been considered in this work.

Future work should include the presence of platelets and white blood cells in the simulations, since the transport and migration phenomena are expected to be greatly affected by the presence of cells with disparate characteristics and properties. Adhesion and tethering models (such as the Bell stochastic approach [94]) should also be considered, as these are known to to affect the both cell-cell and cell-wall interactions, as well as being important phenomena in processes such as thrombosis.

Acknowledgements

The author gratefully acknowledged the support of the Japanese Society for the Promotion of Science (JSPS) to visit the Biological Flow Studies Laboratory at Tohoku University, and

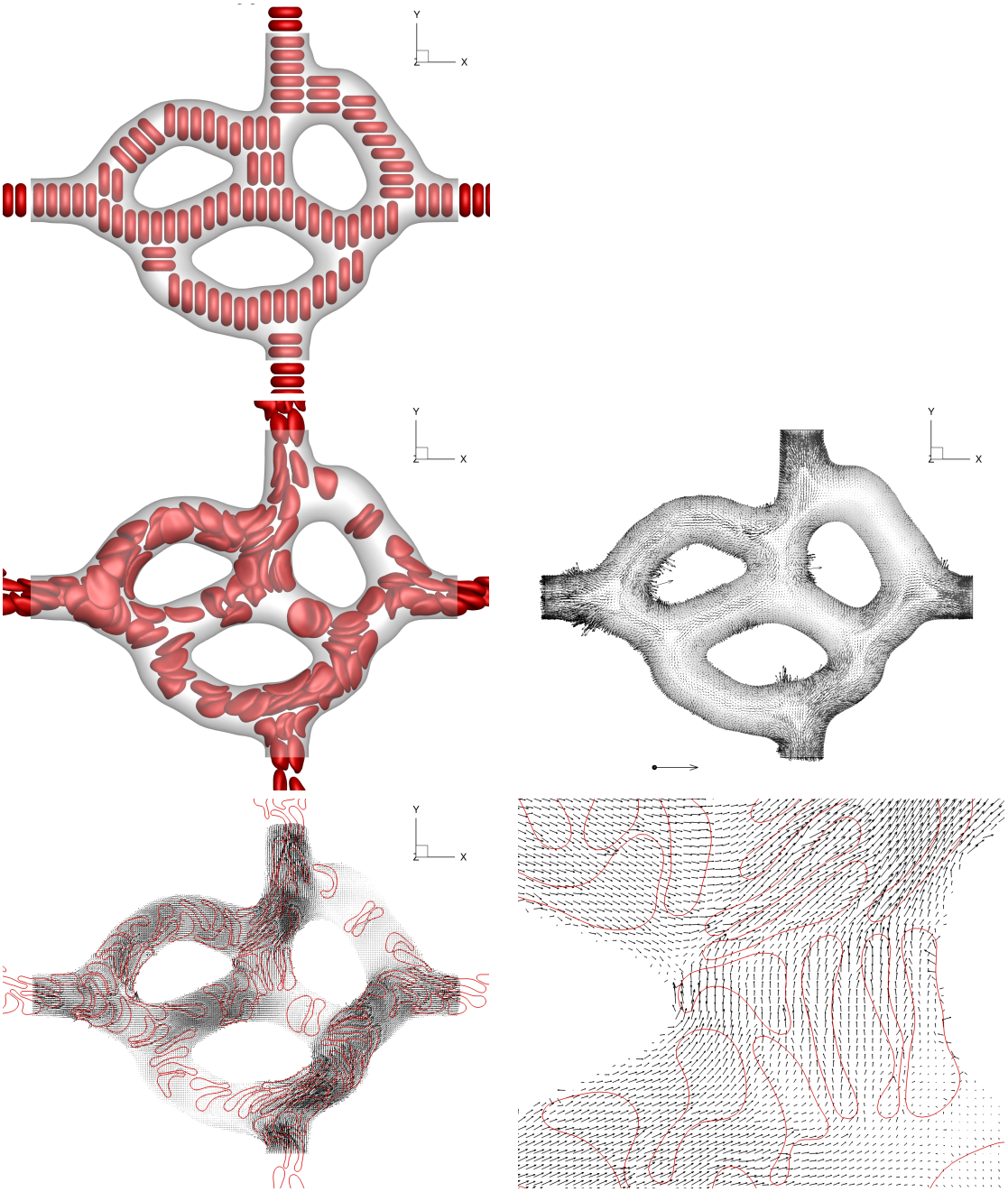


Figure 11: Capillary simulation, at snapshot T=0.76 s. Top row: (left) initial configuration. Middle row: (left) red blood cell distribution; (right) stresses exerted on the vessel wall. Vector length scale is referenced to 10 Pa, given below the figure. Bottom row: (left) mid-plane cross-section showing the velocity vector field; (right) detail of cross-section (centre region of the domain). Flow is left to right and bottom to top.

the support from project 'MatComPhys' under the European Research Executive Agency FP7-PEOPLE-2011-IEF framework.

Appendix A. Differential operators

We maintain the notation presented in Section 2. The discrete differential operators presented reflect those more commonly used in the literature concerning the MPS method. The work of [95] provides additional discussion on these operators, and proposes higher order approximations that resemble the *meshless finite difference method* (MFDM) which employs a least-squares fit to a Taylor expansion on scattered modes [96, 97].

Appendix A.1. Discrete gradient operator

Given a scalar quantity ϕ, the gradient vector for particles at positions $\mathbf{x}_i$ and $\mathbf{x}_j$ is discretised as $\nabla\phi_{ij} = (\phi_j - \phi_i)(\mathbf{x}_j - \mathbf{x}_i)/(|\mathbf{x}_j - \mathbf{x}_i|^2)$. The gradient in MPS, where weighted averaging over the M neighbouring particles j is used, is written as:

$$\langle\nabla\phi\rangle_i = \frac{d}{\sum_{j\neq i}^{M} w(r)} \sum_{j\neq i}^{M} \left(w(r) \frac{(\phi_j - \phi_i)(\mathbf{x}_j - \mathbf{x}_i)}{|\mathbf{x}_j - \mathbf{x}_i|^2} \right) \tag{A.1}$$

where d is the number of space dimensions, thus $d = 3$ in 3D. The value of d is necessary due to the averaging process though initially not self evident. Consider an example in 3D and a uniform distribution of particles: $x_1 = (0,0,0), x_2 = (1,0,0), x_3 = (0,1,0), x_4 = (0,0,1)$ with scalars $\phi_1, \phi_3, \phi_4 = 0; \phi_2 = 1$. Using the above equation (hence multiplying by d) leads to the gradients between x_1 and its neighbourhood to be $\nabla\phi_{x_1-x_2} = (3,0,0), \nabla\phi_{x_1-x_3} = (0,0,0), \nabla\phi_{x_1-x_4} = (0,0,0)$. The average of these leads to $\nabla\phi_{x_1} = (1,0,0)$ which is correct, therefore the averaging process and imposing orthogonal gradients leads to the d appearing in the gradient formulation. From this simple example we can see that stating a gradient in one direction implicitly defines a zero gradient in the orthogonal directions, hence the requiring the scaling by d when averaging.

Appendix A.2. Discrete Laplacian operator

The Laplacian is modelled as a time-dependent diffusion problem in the standard MPS method. The formulation is derived by considering

$$\frac{d\phi}{dt} = \alpha\nabla^2\phi \tag{A.2}$$

where $\alpha > 0$ is the diffusion coefficient. The solution can be obtained using Fourier transforms and is a classical problem. In the case that a point source (delta function) of unit magnitude and located at the origin is applied as initial condition, for a d-dimensional problem the solution is given by

$$\phi(x,t) = \left(\frac{1}{\sqrt{4\pi\alpha t}}\right)^d \exp\left(-\frac{r^2}{4\alpha t}\right) \tag{A.3}$$

where r^2 denotes the distance from the origin: $r^2 = x^2 + y^2 + z^2$ in the case of $d = 3$. It is apparent that the solution is a linear addition in the spatial dimensions. This result corresponds to a normal probability distribution with mean = 0 and variance $\sigma^2 = 2d\alpha t$, hence the dependence of the variance on time is linear. The Gaussian function is the Green's function, the impulse

response, for the diffusion equation. Applying this result to the particles, a scalar ϕ_i at particle i is distributed to its neighbours j, such that in time Δt the variance of the distribution is $2d\alpha\Delta t$. The formula of the discrete variance must however take into consideration that the domain is compactly supported and discrete, and therefore a scaling coefficient should be considered. This is achieved by normalising the analytic solution with the weighted variance on the discrete points by a coefficient [59], denoted here as λ. Using the result from the analytic solution, the discrete model for diffusion of a scalar quantity in time Δt is proposed in the form:

$$\phi_i^{n+1} = \phi_i^n + \frac{2d\alpha\Delta t}{\lambda} \frac{1}{\sum_{j\neq i}^{M} w(r)} \sum_{j\neq i}^{M} \left((\phi_j^n - \phi_i^n)\, w(r) \right) \tag{A.4}$$

Similar to the above problem, consider point i to have a delta function of unit magnitude and the surrounding points to be zero (hence $\phi_i = 1$ and $\phi_j = 0$), the variance of the increment $(\phi_i^{n+1} - \phi_i^n)$ of Eq. A.4 is given by:

$$\sigma^2 = \frac{2d\alpha\Delta t}{\lambda} \frac{1}{\sum_{j\neq i}^{M} w(r)} \sum_{j\neq i}^{M} w(r)\, r^2 \tag{A.5}$$

Equality of this equation to the analytic solution of the variance yields:

$$2d\alpha\Delta t = \frac{2d\alpha\Delta t}{\lambda} \frac{1}{\sum_{j\neq i}^{M} w(r)} \sum_{j\neq i}^{M} w(r)\, r^2 \tag{A.6}$$

leading to evaluation of the normalisation coefficient as

$$\lambda = \frac{\sum_{j\neq i}^{M} w(r)\, r^2}{\sum_{j\neq i}^{M} w(r)} \tag{A.7}$$

Substituting into Eq. A.4, the resulting formulation is obtained to model the Laplacian:

$$\langle \nabla^2 \phi \rangle_i = \frac{2d}{\lambda n_i} \sum_{j\neq i}^{M} \Big(w(r)(\phi_j - \phi_i) \Big) \tag{A.8}$$

An alternative Laplacian operator, given as the divergence of the gradient operator, has been proposed in [56, 55].

Appendix A.3. Discrete divergence operator

The divergence operator, for an arbitrary vector Φ, is given by [54, 53]

$$\langle \nabla \cdot \Phi \rangle_i = \frac{d}{\sum_{j\neq i}^{M} w(r)} \sum_{j\neq i}^{M} \left(w(r) \frac{(\Phi_j - \Phi_i) \cdot (\mathbf{x}_j - \mathbf{x}_i)}{r^2} \right) \tag{A.9}$$

Though not used in the present work, it would be required to compute the pressure Poisson equation given by Eq. 14, as well as an alternative derivation of the Laplacian operator.

Appendix B. Spring networks

Much of the literature regarding spring networks for modelling thin material surfaces is discussed in the context of computer graphics and modelling cloth fabric. The interested reader is referred to review papers [98, 99] for a general presentation of deformable models, while other papers of interest regarding spring network models can be found in [100, 101, 102, 103, 104, 105, 106]. In this section we focus our attention to time integration strategies, which can affect stability of the method and the correct mechanical behaviour, and are important also with regards to computational cost. Semi-implicit schemes have been developed, based on linearisation of a second-order backward difference formula (BDF) stencil [79] or an implicit Euler stencil [107, 108], and it has been shown that the linearisation will effectively cause additional artificial viscosity [108, 109, 110]. A mix of implicit and explicit schemes is suggested in [75].

It has been noted with regards to Eqs. 8 that an explicit scheme to solve the viscous term can suffer from instability as well as high computational costs due to small time step requirements [12, 19]. An implicit treatment of the spring forces $\mathbf{f}_s$ would result in a fully implicit formulation of Eqs. 8, and is now briefly detailed following the work of [79, 107]. Let us define the net force vector $\mathbf{f}$. Then the acceleration $\ddot{\mathbf{x}}$ for a particle i is simply $\ddot{\mathbf{x}}_\mathbf{i} = \mathbf{f}_i/m_i$, and the particle velocity as $\mathbf{u} = \dot{\mathbf{x}}$. The mass m_i is determined by the MPS fluid discretisation, or alternatively/additionally by summing 1/3rd the mass of all the triangle mesh elements containing the particle, hence lumping the mass of the membrane at the vertices. Define also the diagonal mass matrix (of size $3N \times 3N$) by $\text{diag}(\mathbf{M}) = (m_1, m_1, m_1, m_2, m_2, m_2, \ldots, m_N, m_N, m_N)$ (where N is the total number of point mass particles, i.e. mesh vertices). Accordingly we obtain:

$$\ddot{\mathbf{x}} = \mathbf{M}^{-1}\mathbf{f}(\mathbf{x}, \dot{\mathbf{x}}) \tag{B.1}$$

Given the known position $\mathbf{x}(t_0)$ and velocity $\dot{\mathbf{x}}(t_0)$ of the system at time t_0, our goal is to determine a new position $\mathbf{x}(t_0 + h)$ and velocity $\dot{\mathbf{x}}(t_0 + h)$ at time $t_0 + h$. To compute the new state and velocity using an implicit technique, we must first transform equation (B.1) into a first-order differential equation [107]:

$$\frac{d}{dt}\begin{pmatrix} \mathbf{x} \\ \dot{\mathbf{x}} \end{pmatrix} = \frac{d}{dt}\begin{pmatrix} \mathbf{x} \\ \mathbf{u} \end{pmatrix} = \begin{pmatrix} \mathbf{u} \\ \mathbf{M}^{-1}\mathbf{f}(\mathbf{x}, \mathbf{u}) \end{pmatrix} \tag{B.2}$$

Let us denote $\Delta\mathbf{x} = \mathbf{x}(t_0+h) - \mathbf{x}(t_0)$ and $\Delta\mathbf{u} = \mathbf{u}(t_0+h) - \mathbf{u}(t_0)$, and for simplicity $\mathbf{x}(t_0) = \mathbf{x}_0$, $\dot{\mathbf{x}}(t_0) = \dot{\mathbf{x}}_0$ and $\mathbf{f}(t_0) = \mathbf{f}_0$. The explicit forward Euler method applied to equation (B.2) approximates $\Delta\mathbf{x}$ and $\Delta\mathbf{u}$ as:

$$\begin{pmatrix} \Delta\mathbf{x} \\ \Delta\mathbf{u} \end{pmatrix} = h\begin{pmatrix} \mathbf{u}_0 \\ \mathbf{M}^{-1}\mathbf{f}_0 \end{pmatrix} \tag{B.3}$$

The implicit backward Euler method is given by:

$$\begin{pmatrix} \Delta\mathbf{x} \\ \Delta\mathbf{u} \end{pmatrix} = h\begin{pmatrix} \mathbf{u}_0 + \Delta\mathbf{u} \\ \mathbf{M}^{-1}\mathbf{f}(\mathbf{x}_0 + \Delta\mathbf{x}, \mathbf{u}_0 + \Delta\mathbf{u}) \end{pmatrix} \tag{B.4}$$

This equation is non-linear, so it is linearised by applying a Taylor series expansion to $\mathbf{f}$ to first order:

$$\mathbf{f}(\mathbf{x}_0 + \Delta\mathbf{x}, \mathbf{u}_0 + \Delta\mathbf{u}) = \mathbf{f}_0 + \frac{\partial\mathbf{f}}{\partial\mathbf{x}}\Delta\mathbf{x} + \frac{\partial\mathbf{f}}{\partial\mathbf{u}}\Delta\mathbf{u} \tag{B.5}$$

where $\partial\mathbf{f}/\partial\mathbf{x}$ and $\partial\mathbf{f}/\partial\mathbf{u}$ are evaluated at $(\mathbf{x}_0, \mathbf{u}_0)$. This results in

$$\begin{pmatrix} \Delta\mathbf{x} \\ \Delta\mathbf{u} \end{pmatrix} = h\begin{pmatrix} \mathbf{u}_0 + \Delta\mathbf{u} \\ \mathbf{M}^{-1}(\mathbf{f}_0 + \frac{\partial\mathbf{f}}{\partial\mathbf{x}}\Delta\mathbf{x} + \frac{\partial\mathbf{f}}{\partial\mathbf{u}}\Delta\mathbf{u}) \end{pmatrix} \tag{B.6}$$

Taking the first row (i.e. $\Delta\mathbf{x} = h(\mathbf{u}_0 + \Delta\mathbf{u})$) and substituting into the second row, yields

$$\Delta\mathbf{u} = h\mathbf{M}^{-1}\left(\mathbf{f}_0 + \frac{\partial\mathbf{f}}{\partial\mathbf{x}}h(\mathbf{u}_0 + \Delta\mathbf{u}) + \frac{\partial\mathbf{f}}{\partial\mathbf{u}}\Delta\mathbf{u}\right) \tag{B.7}$$

Letting $\mathbf{I}$ denote the identity matrix, the above equation can be written as

$$\left(\mathbf{I} - h\mathbf{M}^{-1}\frac{\partial\mathbf{f}}{\partial\mathbf{u}} - h^2\mathbf{M}^{-1}\frac{\partial\mathbf{f}}{\partial\mathbf{x}}\right)\Delta\mathbf{u} = h\mathbf{M}^{-1}\left(\mathbf{f}_0 + h\frac{\partial\mathbf{f}}{\partial\mathbf{x}}\mathbf{u}_0\right) \tag{B.8}$$

for which we can solve for $\Delta\mathbf{u}$, and then compute $\Delta\mathbf{x} = h(\mathbf{u}_0 + \Delta\mathbf{u})$. The linear system is sparse and depends on evaluating $\mathbf{f}_0$, $\partial\mathbf{f}/\partial\mathbf{x}$, $\partial\mathbf{f}/\partial\mathbf{u}$.

We remind the reader that in Section 3 we wrote $\mathbf{x}_{ij} = \mathbf{x}_j - \mathbf{x}_i$, and $\mathbf{u}_{ij} = \mathbf{u}_i - \mathbf{u}_j$, and we now introduce $\hat{\mathbf{x}}_{ij} = \mathbf{x}_{ij}/|\mathbf{x}_{ij}|$. Then in the case of linear tension/compression springs given by Eq. 37

$$\begin{aligned} \frac{\partial\mathbf{f}}{\partial\mathbf{x}} &= \frac{\kappa_t}{L_{ij,0}}\left(I_3\left(1 - \frac{L_{ij,0}}{|\mathbf{x}_{ij}|}\right) + L_{ij,0}\frac{\mathbf{x}_{ij}\,\mathbf{x}_{ij}^T}{|\mathbf{x}_{ij}|^2}\right) \\ \frac{\partial\mathbf{f}}{\partial\mathbf{u}} &= 0 \end{aligned} \tag{B.9}$$

In the case of damped tension/compression springs, given by Eq. 38

$$\begin{aligned} \frac{\partial\mathbf{f}}{\partial\mathbf{x}} &= \frac{\kappa_{t,D}}{|\mathbf{x}_{ij}|}\left(\left(\hat{\mathbf{x}}_{ij}\cdot\mathbf{u}_{ij}\right)\left(I_3 - 2\,\hat{\mathbf{x}}_{ij}\,\hat{\mathbf{x}}_{ij}^T\right) + \left(\hat{\mathbf{x}}_{ij}\,\mathbf{u}_{ij}^T\right)\right) \\ \frac{\partial\mathbf{f}}{\partial\mathbf{u}} &= \kappa_{t,D}\,\left(\hat{\mathbf{x}}_{ij}\,\hat{\mathbf{x}}_{ij}^T\right) \end{aligned} \tag{B.10}$$

where I_3 is the 3×3 identity matrix.

Appendix C. Reconstructon of vessel from the medial axis

The reconstruction of the vessel geometry from the image is obtained in three steps [3, 111, 112]: i) definition of the vessel centreline and local radius R; ii) construct an implicit function such that an iso-surface defines the desired geometry, using radial basis function interpolation; iii) employ marching tetrahedra to extract a piecewise linear triangle mesh definition of the object surface.

The first step may be obtained from manual selection or employing erosion algorithms. Here a constant vessel diameter was used and the vessel was considered planar, as is common for retina capillaries. Starting from the centreline and the local vessel diameter, using in a Frenet frame of reference to compute the centreline tangent and normal direction, a point cloud that samples the vessel wall can be generated. The point cloud should aim to uniformly sample the domain and avoid both point clustering or gaps.

The second step is in effect the interpolation of this point cloud using the radial basis function approach. Each point is attributed to a scalar, $h(\mathbf{x})$, where $\mathbf{x}_i$, $i = 1, \ldots, N$, are the position vectors

of the points, and in specific $h(\mathbf{x}) = R$ for the points sampling the centreline, and $h(\mathbf{x}) = 0$ for the points on the surface.

A linear system is formed $h(\mathbf{x}_i) = \sum_{j=1}^{N} \mathbf{c}_j \phi(\mathbf{x}_i - \mathbf{x}_j)$, for $i = 1, \ldots, N$, and cubic radial basis function interpolation $\phi(\mathbf{x}_i - \mathbf{x}_j) = |\mathbf{x}_i - \mathbf{x}_j|^3$ (from the polyharmonic family of functions) is employed, from which we seek the solution of the unknown coefficients $\mathbf{c}$. Having computed the coefficients, the implicit function at any location is computed as $f(\mathbf{x}) = \sum_{j=1}^{N} \mathbf{c}_j \phi(\mathbf{x} - \mathbf{x}_j)$.

The final step in the procedure is to extract a surface mesh definition from the implicit function definitions. We therefore seek the iso-surface $f(\mathbf{x}) = 0$, which is extracted using the marching tetrahedra approach [113] with linear interpolation to give an initial triangulation

References

[1] Stewart SFC, Hariharan P, Paterson EG, Burgreen GW, Reddy V, Day SW, Giarra M, Manning KB, Deutsch A, Berman MR, Myers MR, Malinauskas RA. Results of FDAs First Interlaboratory Computational Study of a Nozzle with a Sudden Contraction and Conical Diffuser. *Cardiovascular Engineering and Technology*, **4**(4):374-391, 2013.

[2] Fasano A, Santos RF, Sequeira A. Blood coagulation: a puzzle for biologists, a maze for mathematicians. *Modeling of Physiological Flows, Springer*, pp. 41-75, 2012.

[3] Gambaruto AM, Doorly DJ, Yamaguchi T. Wall Shear Stress and Near-Wall Convective Transport: Comparisons with Vascular Remodelling in a Peripheral Graft Anastomosis. *Journal of Computational Physics*, **229**(14):5339-5356, 2010.

[4] Popel AS, Johnson PC. Microcirculation and hemorheology. *Annual review of fluid mechanics*, **37**:43-69, 2005.

[5] Lipowsky HH. Microvascular rheology and hemodynamics. *Microcirculation*, **12**(1):5-15, 2005.

[6] Pries AR, Neuhaus D, Gaehtgens P. Blood viscosity in tube flow: dependence on diameter and hematocrit. *American Journal of Physiology - Heart and Circulatory Physiology*, **263**(6):H1770-H1778, 1992.

[7] Tsubota K, Wada S. Elastic force of a red blood cell membrane during tank-treading motion: consideration of membrane's natural state. *International Journal of Mechanical Sciences*, **52**:356-364, 2010.

[8] Fedosov DA, Caswell B, Karniadakis GE. A Multiscale Red Blood Cell Model with Accurate Mechanics, Rheology, and Dynamics. *Biophysical Journal*, **98**:2215-2225, 2010.

[9] Bessonov N, Babushkina E, Golovashchenko SF, Tosenberger A, Ataullakhanov F, Panteleev M, Tokarev A, Volpert V. Numerical modelling of cell distribution in blood flow. *Mathematical Modelling of Natural Phenomena*, **9**(6):69-84, 2014.

[10] Kamada H, Tsubota K, Nakamura M, Wada S, Ishikawa T, Yamaguchi T. A threedimensional particle simulation of the formation and collapse of a primary thrombus. *International Journal for Numerical Methods in Biomedical Engineering* **26**(3-4):488-500, 2010.

[11] Suresh S, Spatz J, Mills JP, Micoulet A, Dao M, Lim CT, Beil M, Seufferlein T. Connections between single-cell biomechanics and human disease states: gastrointestinal cancer and malaria. *Acta Biomaterialia*, **1**(1):15-30, 2005.

[12] Imai Y, Kondo H, Ishikawa T, Lim CT, Yamaguchi T. Modeling of haemodynamics arising from malaria infection. *Journal of Biomechanics*, **43**:1386-1393, 2010.

[13] Lima R, Ishikawa T, Imai Y, Yamaguchi T. Blood flow behavior in microchannels: past, current and future trends. *Single and two-Phase Flows on Chemical and Biomedical Engineering, Bentham Science Eds*. pp. 513-547, 2012.

[14] Tanaka T, Ishikawa T, Numayama-Tsuruta K, Imai Y, Ueno H, Matsuki N, Yamaguchi T. Separation of cancer cells from a red blood cell suspension using inertial force. *Lab on a Chip*, **12**(21):4336-4343, 2012.

[15] Fedosov DA, Noguchi H, Gompper G. Multiscale modeling of blood flow: from single cells to blood rheology. *Biomechanics and modeling in mechanobiology*, **13**(2):239-258, 2014.

[16] Müller K, Fedosov DA, Gompper G. Margination of micro-and nano-particles in blood flow and its effect on drug delivery. *Nature Scientific Reports*, **4**(4871), 2014.

[17] Koshizuka S, Oka Y. Moving particle semi-implicit method for fragmentation of incompressible fluid *Nuclear Science and Engineering*, **123**(3):421–434, 1996.

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

[19] Imai Y, Kondo H, Ishikawa T, Lim CT, Tsubota K, Yamaguchi T. Three-dimensional Simulation of Blood Flow in Malaria Infection. *13th International Conference on Biomedical Engineering IFMBE Proceedings*, **23**:2244-2247, 2009.

[20] Tsubota K, Wada S, Yamaguchi T. Simulation study on effects of hematocrit on blood flow properties using particle method. *Journal of Biomechanical Science and Engineering*, **1**(1):159-170, 2006.

[21] Alizadehrad D, Imai Y, Nakaaki K, Ishikawa T, Yamaguchi T. Quantification of red blood cell deformation at high-hematocrit blood flow in microvessels. *Journal of Biomechanics*, **45**(15):2684-2689, 2012.
[22] Sueyoshi M, Kashiwagi M, Naito S. Numerical simulation of wave-induced nonlinear motions of a two-dimensional floating body by the moving particle semi-implicit method. *Journal of Marine Science and Technology*, **13**(2):85–94, 2008.
[23] Shibata K, Koshizuka S. Numerical analysis of shipping water impact on a deck using a article method. *Ocean Engineering*, **34**:585-593, 2007.
[24] Koshizuka S, Nobe A, Oka Y. Numerical analysis of breaking waves using the moving particle semi-implicit method. *International Journal for Numerical Methods in Fluids*, **26**:751-769, 1998.
[25] Ishii E, Ishikawa T, Tanabe Y. Hybrid Particle/Grid Method for Predicting Motion of Micro- and Macrofree Surfaces. *Journal of Fluids Engineering*, **128**(5):921–930, 2006.
[26] Shibata K, Koshizuka S, Oka Y. Numerical analysis of jet breakup behaviour using particle method. *Journal of Nucelar Science and Technology*, **41**(7):715-722, 2004.
[27] Nomura K, Koshizuka S, Oka Y, Obata H. Numerical analysis of droplet breakup behaviour using particle method. *Journal of Nucelar Science and Technology*, **38**(12):1057-1064, 2001.
[28] Harada T, Suzuki Y, Koshizuka S, Arakawa T, Shoji S. Simulation of droplet generation in micro flow using MPS method. *The Japan Society of Mechanical Engineers, Series B*, **49**(3):731-736, 2006.
[29] Sun Z, Xi G, Chen X. Mechanism study of deformation and mass transfer for binary droplet collisions with particle method. *Physics of Fluids*, **21**(3):032106, 2009.
[30] Sun Z, Xi G, Chen X. A numerical study of stir mixing of liquids with particle method. *Chemical Engineering Science*, **64**:341-350, 2009.
[31] Koshizuka S, Ikeda H, Oka Y. Numerical analysis of fragmentation mechanics in vapor explosions. *Nuclear Engineering and Design*, **189**:423-433, 1999.
[32] Xie H, Koshizuka S, Oka Y. Modelling of a single drop impact onto liquid film using particle methods. *International Journal For Numerical Methods In Fluids*, **45**:1009-1023, 2004.
[33] Xie H, Koshizuka S, Oka Y. Simulation of drop deposition process in annular mist flow using three-dimensional particle method. *Nuclear Enginering and Design*, **235**:1687-1697, 2005.
[34] Heo S, Koshizuka S, Oka Y. Numerical analysis of boiling on high hat-flux and high subcooling condition using MPS-MAFL. *International Journal of Heat and Mass Transfer*, **45**:2633-2642, 2002.
[35] Zhang S, Morita K, Fukuda K, Shirakawa N. An improved MPS method for numerical simulations of convective heat transfer problems. *International Journal for Numerical Methods in Fluids*, **51**:31-47, 2006.
[36] Gotoh H,Sakai T. Key issues in the particle method for computation of wave breaking, *Coastal Engineering*, **52**(2-3):171–179, 2006.
[37] Lee CJK, Noguchi H, Koshizuka S. Fluid-shell structure interaction analysis by coupled particle and finite element method. *Computers and Structures*, **85**:688-697, 2007.
[38] Tanaka N, Takano T. Microscopic-scale simulation of blood flow using SPH method. *International Journal of Computational Methods*, **2**(4):555-568, 2005 .
[39] Gompper G, Ihle T, Kroll DM, Winkler RG. Multi-particle collision dynamics: a particle-based mesoscale simulation approach to the hydrodynamics of complex fluids. *Advanced Computer Simulation Approaches for Soft Matter Sciences III, Advances in Polymer Science, Springer*, **221**: 1-87, 2009.
[40] McWhirter JL, Noguchi H, Gompper G. Flow-induced clustering and alignment of vesicles and red blood cells in microcapillaries. *Proceedings of the National Academy of Sciences*, **106**(15):6039-6043, 2009.
[41] Tiwari S, Kuhnert J. Modeling of two-phase flows with surface tension by finite pointset method (FPM). *Journal of Computational and Applied Mathematics*, **203**(2):376-386, 2007.
[42] Yoon HY, Koshizuka S, Oka Y. A particle-gridless hybrid method for incompressible flows. *International Journal for Numerical Methods in Fluids*, **30**(4):407–424, 1999.
[43] Shao S, Lo EYM. Incompressible SPH method for simulating Newtonian and non-Newtonian flows with a free surface. *Advances in Water Resources*, **26**(7):787–800, 2003.
[44] Cummins SJ, Rudman M. An SPH projection method. *Journal of Computational Physics*, **152**(2):584-607, 1999.
[45] Xu R, Stansby P, Laurence D. Accuracy and stability in incompressible SPH (ISPH) based on the projection method and a new approach. *Journal of Computational Physics*, **228**(18):6703-6725, 2009.
[46] Suzuki Y, Koshizuka S, Oka Y. Hamiltonian moving-particle semi-implicit (HMPS) method for incompressible fluid flows. *Comput. Methods Appl. Mech. Engrg.*, **196**:2876-2894, 2007.
[47] Oochi M, Koshizuka S, Sakai M. Explicit MPS Algorithm for Free Surface Flow Analysis. *Transactions of JSCES, Paper o.20100013*, 2010.
[48] Shakibaeinia A, Jin Y-C. A weakly compressible MPS method for modeling of open-boundary free-surface flow. *International Journal for Numerical Methods in Fluids*, **63**(10):1208–1232, 2010.
[49] Hu XY, Adams NA. An incompressible multi-phase SPH method. *Journal of computational physics*, **227**(1):264-278, 2007.

[50] Harada T, Koshizuka S, Shimazaki K. Improvement of wall boundary calculation model for MPS method. *Transactions of Japan Society for Computational Engineering and Science (JSCES)*, Paper No. 20080006, 2008.
[51] Harada T, Koshizuka S, Kawaguchi Y. Smoothed Particle Hydrodynamics in Complex Shapes. *Proc. of Spring Conference on Computer Graphics*, pp. 26-28. 2007.
[52] Ataie-Ashtiani B, Farhadi L. A stable moving-particle semi-implicit method for free surface flows. *Fluid Dynamics Research*, **38**(4):241–256, 2006.
[53] Tanaka M, Masunaga T. Stabilization and smoothing of pressure in MPS method by quasi-compressibility. *Journal of Computational Physics*, **229**(11):4279–4290, 2010.
[54] Asai M, Fujimoto K, Tanabe S, Beppu M. Slip and No-Slip Boundary Treatment for Particle Simulation Model with Incompatible Step-Shaped Boundaries by Using a Virtual Maker. *Transactions of the Japan Society for Computational Engineering and Science*, **2013**:20130011, 2013.
[55] Khayyer A, Gotoh H. Modified Moving Particle Semi-implicit methods for the prediction of 2D wave impact pressure. *Coastal Engineering*, **56**(4):419–440, 2009.
[56] Khayyer A, Gotoh H. A higher order Laplacian model for enhancement and stabilization of pressure calculation by the MPS method. *Applied Ocean Research*, **32**(1):124–131, 2010.
[57] Kondo M, Koshizuka S. Improvement of stability in moving particle semi-implicit method. *International Journal for Numerical Methods in Fluids*, **65**(6):638-654, 2011.
[58] Khayyer A, Gotoh H. Enhancement of stability and accuracy of the moving particle semi-implicit method. *Journal of Computational Physics*, **230**(8): 3093-3118, 2011.
[59] Koshizuka S. Computational fluid dynamics. *Baifu-kan publishers*, 1997.
[60] Weinan E, Jian-Guo L. Projection Method I: Convergence and Numerical Boundary Layers. *SIAM Journal on Numerical Analysis*, **32**(4):1017-1057, 1995.
[61] Guermond JL, Minev P, Shen J. An overview of projection methods for incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, **195**:6011-6045, 2006.
[62] Lastiwka M, Basa M, Quinlan NJ. Permeable and non-reflecting boundary conditions in SPH. *International Journal for Numerical Methods in Fluids*, **61**(7)709-724, 2009.
[63] Ikedai H, Koshizuka S, Okai Y, Park HS, Sugimoto J. Numerical Analysis of Jet Injection Behavior for Fuel-Coolant Interaction using Particle Method, *Journal of Nuclear Science and Technology*, **38**(3):174–182, 2001.
[64] Koshizuka S. Particle Method. *The Japan Society for Computational Engineering and Science (JSCES)*, 2005.
[65] Tosenberger A, Salnikov V, Bessonov N, Babushkina E, Volpert V. Particle Dynamics Methods of Blood Flow Simulations. *Math. Model. Nat. Phenom.*, **6**(5):320-332, 2011.
[66] Van Gelder A. Approximate Simulation of Elastic Membranes by Triangulated Spring Meshes. *Journal of Graphics Tools*, **3**(2):21-41. 1998.
[67] Omori T, Ishikawa T, Barthès-Biesel D, Salsac AV, Walter J, Imai Y, Yamaguchi T. Comparison between spring network models and continuum constitutive laws: Application to the large deformation of a capsule in shear flow. *Physical Review E*, **83**:041918, 2011.
[68] Volino P, Courchesne M, Magnenat-Thalmann N. Versatile and efficient techniques for simulating cloth and other deformable objects. *SIGGRAPH '95 Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pp. 137-144, 1995.
[69] Fedosov DA, Caswell B, Karniadakis GE. Systematic coarse-graining of spectrin-level red blood cell models. *Computer Methods in Applied Mechanics and Engineering*, **199**:1937-1948, 2010.
[70] Kondo H, Imai Y, Ishikawa T, Tsubota K, Yamaguchi T. Hemodynamic Analysis of Microcirculation in Malaria Infection. *Annals of Biomedical Engineering*, **37**(4):702-709, 2009.
[71] Dupont C, Salsac A, Barthes-Biesel D, Vidrascu M, Le Tallec P. Coupling boundary integral and shell finite element methods to study the fluid structure interactions of a microcapsule in a simple shear flow. *International Conference on Boundary Element and Meshless Techniques (Beteq)*, Palaiseau, France, July 2013.
[72] Eck M, DeRose T, Duchamp T. Multiresolution Analysis of arbitrary meshes. *SIGGRAPH '95 Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pp. 173-182, 1995.
[73] Provot X. Deformation constraints in a mass-spring model to describe rigid cloth behaviour. *Graphics interface*, 1995.
[74] Schmedding R, Gissler M, Teschner M. Optimized damping for dynamic simulations. *Proceeding SCCG '09 Proceedings of the 25th Spring Conference on Computer Graphics*, pp. 189-196, 2009.
[75] Bridson R, Marino S, Fedkiw R. Simulation of cloth with folds and wrinkles. *Eurographics/SIGGRAPH Symposium on Computer Animation*, pp. 28-36, 2003.
[76] Bridson R. Computational aspects of dynamic surfaces. PhD thesis, Stanford University.
[77] Kelager M, Niebe S, Erleben K. A Triangle Bending Constraint Model for Position-Based Dynamics. *Workshop on Virtual Reality Interaction and Physical Simulation VRIPHYS*, pp. 31-37, 2010.
[78] Eberhardt B, Weber A, Strasser W. A fast, flexible, particle-system model for cloth draping. *Computer Graphics and Applications, IEEE*, **16**(5):52-59. 1996.

[79] Choi K, Ko H. Stable but responsive cloth. *ACM Trans. Graph. In SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, **21**(3):604-611,2002.
[80] Volino P, Magnenat-Thalmann N. Simple linear bending stiffness in particle systems. *Eurographics/SIGGRAPH Symposium on Computer Animation*, pp. 101-105, 2006.
[81] Bessonov NM, Golovashchenko SF, Volpert VA. Numerical Modelling of Contact Elastic-Plastic Flows. *Math. Model. Nat. Phenom.*, **4**(1):44-87, 2009.
[82] Aftosmis MJ, Berger MJ, Melton JE. Robust and efficient Cartesian mesh generation for component-based geometry. *AIAA journal*, **36**(6):952-960, 1998.
[83] Secomb TW, Hsu R, Pries AR. Motion of red blood cells in a capillary with an endothelial surface layer: effect of flow velocity. *American Journal of Physiology-Heart and Circulatory Physiology*, **281**(2):H629-H636, 2001.
[84] Secomb TW, Hsu R, Pries AR. Blood flow and red blood cell deformation in nonuniform capillaries: effects of the endothelial surface layer. *Microcirculation*, **9**(3):189-196, 2002.
[85] Secomb TW, Hsu R, Pries AR. Effect of the endothelial surface layer on transmission of fluid shear stress to endothelial cells. *Biorheology*, **38**(2):143-150, 2001.
[86] Pan W, Caswell B, Karniadakis GE. A low-dimensional model for the red blood cell. *Soft Matter*, **6**(18):4366-4376, 2010.
[87] Bronkhorst PJ, Streekstra GJ, Grimbergen J, Nijhof EJ, Sixma JJ, Brakenhoff GJ. A new method to study shape recovery of red blood cells using multiple optical trapping. *Biophysical journal*, **69**(5):1666-1673, 1995.
[88] Spillmann J, Teschner M. Cosserat Nets. *IEEE Transactions on visualization and computer graphics*, 2009.
[89] Pries AR, Secomb TW. Rheology of the microcirculation. *Clinical hemorheology and microcirculation*, **29**(3):143-148, 2003.
[90] Freund JB, Vermot J. The Wall-stress Footprint of Blood Cells Flowing in Microvessels. *Biophysical Journal*, **106**:752-762, 2014.
[91] Jeffery GB. The motion of ellipsoidal particles immersed in a viscous fluid. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, **102**(715):161-179, 1922.
[92] Fantin A, Schwarz Q, Davidson K, Normando EM, Denti L, Ruhrberg C. The cytoplasmic domain of neuropilin 1 is dispensable for angiogenesis, but promotes the spatial separation of retinal arteries and veins. *Development*, **138**(19):4185-4191, 2011. (Supplementary material: *supp_138.19.4185_DEV070037FigS1.jpg*)
[93] Sutera SP. Flow-induced trauma to blood cells. *Circulation research*, **41**(1):2-8, 1977.
[94] Pozrikidis C. Computational hydrodynamics of capsules and biological cells. *CRC Press*, 2010.
[95] Tamai T, Shibata K, Koshizuka S. Development of the Higher-order MPS Method Using the Taylor Expansion. *Transactions of the Japan Society for Computational Engineering and Science*, **2013**:20130003, 2013.
[96] Liszka T., Orkisz J. The finite difference method at arbitrary irregular grids and its application in applied mechanics. *Computers and structures*, 11:83-95, 1980.
[97] Liszka T.J., Duarte C.A.M., Tworzydlo W.W. Hp-meshless cloud method. *Computer Methods in Applied Mechanics and Engineering*, 139(1):263-288, 1996.
[98] Terzopoulos D, Fleicher K. Deformable Models. *The Visual Computer*, **4**:306-331,1988.
[99] Nealen A, Müller M, Keiser R, Boxerman E, Carlson M. Physically based deformable models in computer graphics. *Computer Graphics Forum*, **25**(4):809-836, 2005.
[100] Grinspun E, Hirani AN, Desburn M, Schröder P. Discrete shells. *Eurographics/SIGGRAPH Symposium on Computer Animation*, pp. 62-67, 2003.
[101] Selle A, Su J, Irving G, Fedkiw R. Robust high-resolution cloth using parallelism, history-based collisions, and accurate friction. *IEEE Transactions on Visualization and Computer Graphics*, **15**(2):339-350, 2009.
[102] Volino P, Magnenat-Thalmann N. Implementing fast cloth simulation with collision response. *Proceedings Computer Graphics International*, pp. 257-266, 2000.
[103] Volino P, Magnenat-Thalmann N. Comparing efficiency of integration methods for cloth simulation. *Proceedings Computer Graphics International*, pp. 265-272, 2001.
[104] Bridson R, Fedkiw R, Anderson J. Robust treatment of collisions, contact and friction for cloth animation. *ACM Transactions on Graphics, Proceedings of ACM SIGGRAPH 2002*, **21**(3):594-603, 2002.
[105] Hauth M, Etzmuss O. A High Performance Solver for the Animation of Deformable Objects using Advanced Numerical Methods. *Computer Graphics Forum*, **20**(3):319-328, 2001.
[106] Thomaszewski B, Wacker M, Straßer W. A consistent bending model for cloth simulation with corotational subdivision finite elements. *In Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 107-116, 2006.
[107] Baraff D, Witkin A. Large steps in cloth simulation. *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pp. 43-54, 1998.
[108] Oh S, Ahn J, Wohn K. Low damped cloth simulation. *The Visual Computer*, **22**(2):70-79, 2006.
[109] Desbrun M, Schröder P, Barr A. Interactive animation of structured deformable objects. *Proceedings of the 1999 conference on Graphics interface*, **99**(5):1-8, 1999.

[110] Kang Y, Choi J, Cho H, Park C. Fast and stable animation of cloth with an approximated implicit method. *Proceedings in Computer Graphics International*, pp. 247-255, 2000.

[111] Gambaruto AM, Taylor DJ, Doorly DJ. Decomposition and Description of the Nasal Cavity Form. Annals of Biomedical Engineering. 40(5):1142-1159, 2012.

[112] Gambaruto AM, Peiró J, Doorly DJ, Radaelli AG. Reconstruction of shape and its effect on flow in arterial conduits. *International Journal for Numerical Methods in Fluids*, **57**(5):495-517, 2008.

[113] Bloomenthal J. An implicit surface polygonizer. *Graphics gems IV*, pp. 324–349, 2004. Academic Press Professional Inc., San Diego, CA, USA.