Scalability of forces in mesh-based models of elastic objects*

Iveta Jančigová, Renáta Tóthová
Cell-in-fluid Research Group
Faculty of Management Science and Informatics, University of Žilina
Univerzitná 8215/1, 010 26 Žilina, Slovakia
Email: {iveta.jancigova, renata.tothova}@fri.uniza.sk

***Abstract*—Recently, there has been a lot of progress in many areas of biomedical engineering achieved as a result of modeling the objects of interest and then performing in-silico experiments with them. One of these areas is modeling blood and its components for various applications. We present a short description of such application and a mesh-based approach to modeling. We discuss which elastic forces in this model are scalable, which are not and demonstrate that mesh differences and characteristics of simulation experiments have to be taken into account when calibrating elastic coefficients. The corresponding simulations are performed using our object-in-fluid framework as a part of extensible simulation package ESPResSo.**

I. Introduction

Blood is a dense suspension of cells. The viscous blood plasma that comprises about 55% of blood volume is the fluid in which the cells are suspended. The cells are mostly red blood cells (about 40% of total blood volume) and then white blood cells, platelets, etc. Models of blood flow are widely used, for example in development of microfluidic devices able to filter special cells from sample [1]. There have been various approaches to this problem: using filtration by size, using magnetic beads as obstacles [2], etc. All of these require a proper model of blood and for this model it is necessary to capture the essential characteristics of blood plasma and red blood cells.

The cells have a biconcave disc shape with diameter $7-8\mu m$ and thickness of about $2\mu m$. Their elasticity allows them to significantly change this shape in the flow (for example, in human body they are able to pass through capillaries with $3\mu m$ diameter). In this paper, we focus on the modeling of this elasticity. One of the desired qualities of the model is scalability - when the size of the model increases (this increase will be specified more precisely later), one would like to see comparable behavior.

Overview of the article

In section II we cover the individual parts of our model with the focus on the elastic forces. Next, in section III we explain the scalability in context of our model, first by suggesting a new definition of forces in III-A and showing that they are preferable to the original ones. Then in subsection III-B we seek further improvement by fitting the elastic coefficient and in subsections III-C and III-D we explain how mesh geometry and design of simulation experiment influence these results. We summarize our conclusions in section IV.

*This work was supported by the Slovak Research and Development Agency under the contract No. APVV-0441-11.

II. Model

We start with a model proposed in [3] and implemented in [4]. The model comprises two main parts - the fluid and the elastic object. For blood plasma we use the Lattice-Boltzmann method (as discussed [5]), however, in simulations described in this paper, the fluid has always been stationary (up to the damping effect where it absorbs the energy transferred by the object).

For elastic objects we use a spring network model, in which the spring network forms a surface triangulation of the object. Our motivation is to model red blood cells and therefore the elastic properties that we need captured in the model are resistance of the membrane to surface dilation, viscoelastic resistance to bending and stretching of the membrane and total volume conservation. The original forces used to model this elasticity were the following (taken from [3]). Stretching force:

$$\mathbf{F}_s(AB) = k_s\kappa(\lambda)\frac{\Delta L}{L}\mathbf{n}_{AB} \tag{1}$$

where k_s is the stretching coefficient, $\kappa(\lambda)$ allows for nonlinearity of the stretching force (by making λ dependent on L), however for the purposes of this paper, we will keep $\kappa = 1$. L is the relaxed length of the edge AB, ΔL is the prolongation of this edge and $\mathbf{n}_{AB}$ is the unit vector pointing from A to B. Bending force:

$$\mathbf{F}_b(ABC) = k_b\frac{\Delta\theta}{\theta}\mathbf{n}_{ABC} \tag{2}$$

where k_b is the bending coefficient, θ is the resting angle between two triangles that have common edge AB, $\Delta\theta$ is the deviation from this angle and $\mathbf{n}_{ABC}$ is the unit normal vector to the triangle ABC. Local area force:

$$\mathbf{F}_a(A) = -k_a\frac{\Delta S_{ABC}}{S_{ABC}}\mathbf{w}_A \tag{3}$$

where k_a is the local area coefficient, S_{ABC} is the relaxed area of triangle ABC, ΔS_{ABC} is the deviation form this resting state and $\mathbf{w}_A$ is the unit vector pointing from the centroid of the triangle ABC to the vertex A. (Similar forces are assigned to vertices B and C). Global area force:

$$\mathbf{F}_A(A) = -k_A\frac{\Delta S}{S}\mathbf{w}_A \tag{4}$$

where k_A is the global area coefficient, S is the relaxed area of the whole object, ΔS is the deviation from this area and $\mathbf{w}_A$ is again the unit vector pointing from the centroid of the triangle ABC to the vertex A. Volume force:

$$\mathbf{F}_V(ABC) = -k_V \frac{\Delta V}{V} S_{ABC} \mathbf{n}_{ABC} \tag{5}$$

where k_V is the volume coefficient, S_{ABC} is the area of triangle ABC, V is the volume of the whole object, ΔV is the deviation from this relaxed volume and $\mathbf{n}_{ABC}$ is the unit normal vector to the plane ABC. Details of this model can be found in [6].

For simplicity and clarity, in this article our object of interest will be a sphere with a membrane that has the same characteristics modeled by a triangular spring network covering the sphere surface. The stretching elasticity of such sphere has already been discussed to some extent in [7].

The two models - Lattice-Boltzmann for the fluid and spring network for the elastic objects - are connected using the immersed boundary method.

III. Scalability of forces

As was demonstrated in [7], this definition of elastic forces might not be ideal. In a good model, the results should not depend on the triangulation of the object. Moreover, there should be a direct relationship between the physical moduli and the elastic parameters $(k_s, k_b, k_a, k_A, k_V)$ possibly dependent on the mesh density. Currently, this is not the case, since the normalization (e. g. division by L in the stretching force) implies dependence of elastic parameters on triangulation mesh.

A. New approach

To achieve this goal, we have tried to modify the stretching force (eq. 1) and the local area force (eq. 3) and remove the normalization. Thus the new formulas are:

$$\mathbf{F}_s(AB) = k_s \Delta L \mathbf{n}_{AB} \tag{6}$$

and

$$\mathbf{F}_a(A) = -k_a \Delta S_{ABC} \mathbf{w}_A \tag{7}$$

There is also a physical reason for this - the Hooke's law for simple springs states $F = k\Delta L$, so the new stretching force definition (eq. 6) corresponds to this formula.

The remaining three forces have not been modified in this fashion, since they are qualitatively different. Consider the stretching force. When the number of triangulation nodes (and thus also edges) increases, the length of the edges decreases - the dependence between number of edges and average edge length is depicted in fig. 1a). On the other hand, for bending force (eq. 2) and for angles between triangles we see a different behavior. In all meshes, angles are close to π radians and increase in nodes/edges implies only minimal change of the average angle - see fig. 1b). Thus, dividing by θ has the same effect as simply setting different bending coefficient.

Scalability is a property of a system that means the following. If we increase the size of the system - in our case number of surface triangulation nodes - the system behaves the same way as before. In this context, the stretching force is not scalable and the bending force is. Also, the local area force has the same property as the stretching force and thus is not scalable and

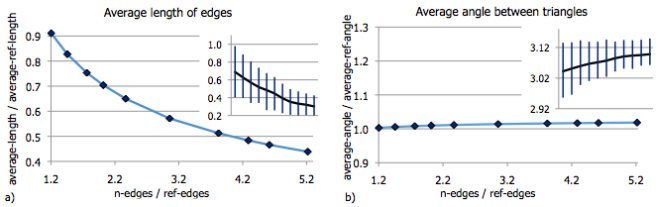

Fig. 1: Decrease of average edge length compared to length in reference mesh is dependent on increase in number of edges, again compared to reference mesh (fig. a)). Average length of edge (in μm) decreases with increased number of triangulation nodes (fig. a) inset) with error bars corresponding to minimum and maximum edge length. On the other hand, increase in average angle between triangles remains almost constant with respect to increase in number of edges (fig. b)), while the average angle (in radians) increases with increased number of triangulation nodes (fig. b) inset).

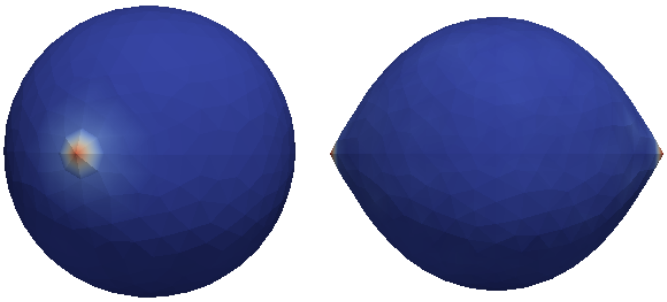

Fig. 2: An outward force of the same magnitude was applied at two opposite nodes of the triangulation (one of them depicted on the left). The stretched sphere was then left to relax. Once the stationary "lemon" state was reached (on the right), total prolongation and cross-sectional radius were measured. The displayed mesh has 601 surface nodes.

the global area force and volume force both behave similarly to the bending force and thus are scalable.

To demonstrate the difference between these two types of forces, we have performed the following simulations using the object-in-fluid framework [4] that our group developed for simulations of objects (both elastic and stiff) moving in fluid. This framework is part of an open source molecular dynamics simulation package ESPResSo [8]. (Visualizations have been done using open source analysis and visualization application ParaView [9].)

We have used 11 different triangulations of a sphere with $4\mu m$ radius - with 500, 601, 727, 879, 1007, 1182, 1524, 1905, 2137, 2301, 2600 surface nodes. These are Delaunay triangulations and so the resulting triangles are fairly close to equilateral. These spheres were immersed in stationary fluid. In each simulation, we applied a constant outward force at two opposite points on the surface of the sphere (e. g. North and South pole) and then let the deformed sphere relax (fig. 2). Once the length of the object and cross-sectional radius reached steady state, we measured this length and the average cross-sectional radius. All simulations were run twice - once using the normalized (eq. 1 and 3) and once using the non-normalized forces (eq. 6 and 7). The differences are displayed in fig. 3. We see that the non-normalized forces result in

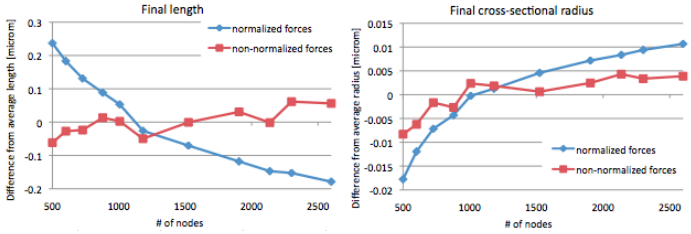

Fig. 3: Difference between simulations using normalized and non-normalized forces: On the left - variation of relaxed length from the average relaxed length over all simulations is larger using normalized forces compared to non-normalized. Similarly, on the right, the variation of final cross-sectional diameter is smaller when using the non-normalized forces compared to the normalized. All elastic parameters were the same across these simulations.

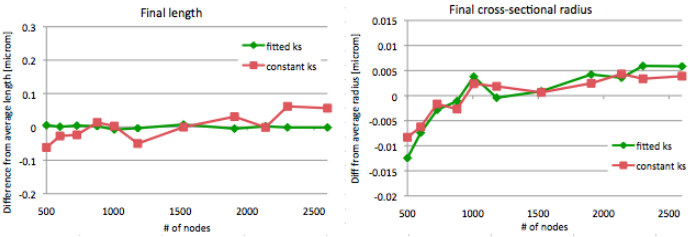

Fig. 4: Difference between simulations using constant and fitted k_s. On the left - variation of relaxed length from the average relaxed length over all simulations varies when using constant k_s and is held constant when fitting k_s. On the right, the variation of cross-sectional radius is similar when using the fitted coefficient compared to constant. All elastic parameters except k_s were the same across these simulations.

more similar behavior of the different-sized meshes while the normalized forces give more variation in both cross-sectional radius and total length of relaxed deformed sphere.
Therefore, we conclude that in order to better model elasticity of spring network triangulation of immersed objects, one should use non-normalized forces.

B. Looking for consistent elastic behavior

While we see that non-normalized forces decrease the variation in elastic behavior of differently triangulated spheres, we would like to achieve even more consistency across the mesh density. To achieve this, we have fitted the stretching coefficient for each mesh in such a way that applying the same force gives us the same prolongation of the stretched sphere. The red lines in fig. 4 represent the same data as the red lines in fig. 3. The green ones show that while the final prolongation was kept the same, the cross-sectional radius displayed roughly the same level of variation as in the constant k_s case. To determine whether this can be improved, we have looked at two types of variations of this simulation experiment described in the following two sections - mesh variations and different method of stretching the object.

C. Mesh variations

Firstly, we have repeated the simulation experiments with applying the same force in different directions (along the

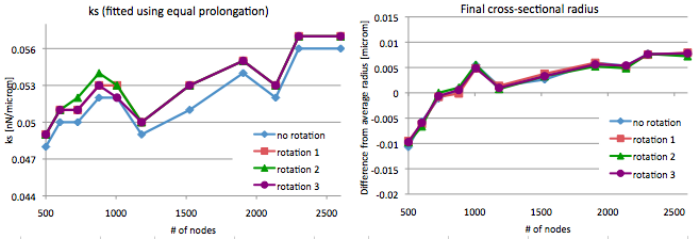

Fig. 5: Comparison of rotated meshes: On the left: stretching coefficient k_s fitted using the same force to achieve the same prolongation. On the right, corresponding relaxed cross-sectional radii.

x, y and z axis) to check that the outcome is independent of direction. There was virtually no difference between the three corresponding results, but we do not consider this check sufficient. Ideally, we would like to apply the forces along an arbitrary line that does not coincide with axes, but this was not possible due to the structure of our triangulations. While they have regularity in terms of "equator" and two main "meridians" offset by $\frac{\pi}{4}$, the triangles covering the resulting octants do not have vertices aligned on opposite sides of the sphere. Therefore we could not align the opposite forces properly.
To perform an alternative check, we introduced random perturbations of existing meshes. We kept the position of two opposite nodes fixed and independently randomly rotated all other vertices around the center of the sphere in such a way that the underlying triangulation remained reasonably regular. We have used three different rotations for each sphere triangulation and repeated the simulations with non-normalized forces and individual fitting of stretching coefficient k_s. The results are depicted in fig. 5. We see that there is a small difference between the original meshes and the rotated meshes as a group. The fitted stretching coefficients for the rotated meshes are slightly higher than those of the original meshes and while there is a small variation within the rotated group for meshes with fewer nodes, it disappears for denser meshes. The overall shape of the k_s curve remains the same, which leads to the conclusion that even the rotation was not quite able to remove the influence of mesh features on the result. The cross-sectional radius behaves consistently for all these meshes.
To demonstrate the scalability of stretching force, the relaxed cross-sectional radius should remain constant for all meshes in these simulation experiments. The rotation simulations show that the triangulation features manifest themselves at this level of detail.

D. Method variations

Secondly, we have modified the way we stretch the sphere. Since the application of force at the individual surface point puts a large amount of very localized strain on the object, we used an approach inspired by actual biological simulations [10], where silica beads are placed at the opposite sides of cells. The cells are then stretched by applying force on the beads which means that the forces act on a local ring area on the cell surface.
We selected triangulation points that roughly correspond to the circular rings with 5% offset from the poles - fig. 6. As expected, this approximation is more precise and results in more regular rings for denser meshes, i.e. larger amount of

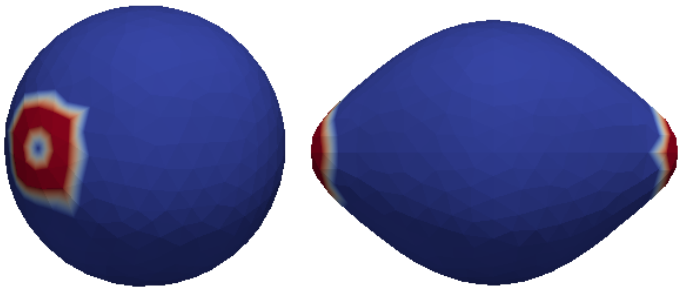

Fig. 6: A sphere that is stretched by applying forces along two rings. $t = 0$ on the left, on the right the sphere is after relaxation. The displayed mesh has 601 surface nodes.

surface nodes.

We repeated the simulations, but this time we divided the total applied force among the nodes that were part of the ring. Note, that while the magnitude of the total force was the same on both sides of the sphere, the individual forces had small differences in some cases, because the number of nodes in the two rings on the opposite sides of the object varied slightly (up to 5 nodes, with about 50 nodes per ring in the densest meshes).

With this type of simulation we can observe few things - fig. 7. Firstly, the coefficient k_s is decreasing here with increased mesh density compared to simulations where we applied force at single points. In the single point case, the spring network more resembled a set of serially connected springs. As a consequence, by increasing their number, the stretching coefficients had to increase too, in order to have the same elastic behavior. On the other hand, when the force is applied along the ring, we still have a set of serial springs, but now their parallel connections come more into play and in such case, we see k_s decreasing in order to maintain the same behavior.

Another thing we see is that while the variations in the relaxed cross-sectional radius are about the same magnitude in both types of experiments, they are much more scattered in the ring pulling case compared to the point pulling case. This is due to the irregularities in the ring selection. Since we cannot use perfectly circular rings, there are variations in acting forces resulting in non-smooth behavior of resulting cross-sectional radii.

Our final observation is that using the ring pulling method, one reaches the steady state in about half simulation time compared to the point pulling method. This is because the stopping criterion here involves average over more points, while in the point pulling method only the distance between the two furthest points was measured.

Overall, we can say that the ring pulling method is more physical, more computationally efficient, and gives results on the same scale as point pulling method. As far as the resulting stretching coefficient is concerned, we conclude that one cannot find a single formula that would give relation between the physical in-plain shear modulus and simulation model stretching coefficient, because the simulation method introduces its own variation whose magnitude is comparable to the data variation we are trying to compensate.

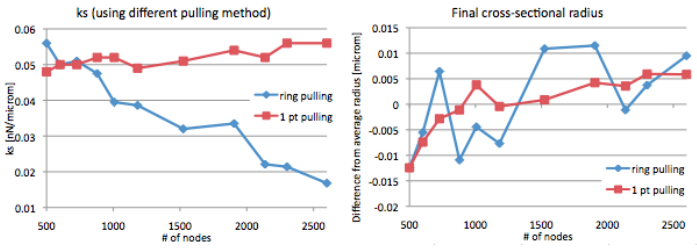

Fig. 7: Comparison of two stretching methods: On the left - fitted stretching coefficients, on the right - corresponding relaxed cross-sectional radii.

IV. Conclusion

In this work we have shown that when performing simulations of elastic objects modeled by surface spring network triangulations, it is preferable to use non-normalized elastic forces to reduce the effect of mesh dependence. Using the stretching force as an example, we have demonstrated that the complete mesh independence is difficult to achieve because the variations in elastic behavior that are due to local mesh geometry and the simulation experiment design are on the same scale as the variations one would like to remove. As a consequence, it is not possible to write a simple formula that would convert physical elastic moduli to simulation coefficients of elasticity. One direction for future work, where we see a possibility for improvement, is to redefine the elastic forces in such a way that one can look at the corresponding elastic energies and try to achieve scalability in terms of energy.

Acknowledgment

The authors would like to thank Ivan Cimrák for his time and valuable advice.

References

[1] S. Nagrath, L. V. Sequist, S. Maheswaran, D. W. Bell, D. Irimia, L. Ulkus, M. R. Smith, E. L. Kwak, S. Digumarthy, A. Muzikansky, P. Ryan, U. J. Balis, R. G. Tompkins, D. A. Haber, and M. Toner, "Isolation of rare circulating tumour cells in cancer patients by microchip technology," *Nature*, vol. 450, pp. 1235–1239, 2007.

[2] M. Gusenbauer, A. Kovacs, F. Reichel, L. Exl, S. Bance, H. Ozelt, and T. Schrefl, "Self-organizing magnetic beads for biomedical applications," *Journal of Magnetism and Magnetic Materials*, vol. 324, no. 6, pp. 977 – 982, 2012.

[3] M. Dupin, I. Halliday, C. Care, and L. Alboul, "Modeling the flow of dense suspensions of deformable particles in three dimensions," *Phys Rev E Stat Nonlin Soft Matter Phys.*, vol. 75, p. 066707, 2007.

[4] I. Cimrák, M. Gusenbauer, and I. Jančigová, "An ESPResSo implementation of elastic objects immersed in a fluid," *Computer Physics Communications*, vol. 185, no. 3, pp. 900 – 907, 2014.

[5] B. Dunweg and A. J. C. Ladd, "Lattice Boltzmann simulations of soft matter systems," *Advances in Polymer Science*, vol. 221, pp. 89–166, 2009.

[6] I. Cimrák, M. Gusenbauer, and T. Schrefl, "Modelling and simulation of processes in microfluidic devices for biomedical applications," *Computers and Mathematics with Applications*, vol. 64, no. 3, pp. 278 – 288, 2012.

[7] I. Cimrák, I. Jančigová, K. Bachratá, and H. Bachratý, "On elasticity of spring network models used in blood flow simulations in ESPResSo," in *III International Conference on Particle-based Methods – Fundamentals and Applications PARTICLES 2013*, M. Bisschoff, E. Oñate, D. Owen, E. Ramm, and P. Wriggers, Eds., 2013, pp. 133–144.

[8] A. Arnold, O. Lenz, S. Kesselheim, R. Weeber, F. Fahrenberger, D. Roehm, P. Košovan, and C. Holm, "ESPResSo 3.1 - molecular dynamics software for coarse–grained models," in *Meshfree Methods for Partial Differential Equations VI, Lecture Notes in Computational Science and Engineering*, M. Griebel and M. Schweitzer, Eds., vol. 89, 2013, pp. 1–23.

[9] A. Henderson, "ParaView guide, A Parallel Visualization Application," Kitware Inc., Tech. Rep., 2007.

[10] J. P. Mills, L. Qie, M. Dao, C. T. Lim, and S. Suresh, "Nonlinear elastic and viscoelastic deformation of the human red blood cell with optical tweezers," *Molecular & Cellular Biomechanics*, vol. 1, no. 3, pp. 169–180, 2004.