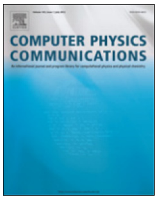
Computer Physics Communications 207 (2016) 1–23

Contents lists available at ScienceDirect

Computer Physics Communications

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



On the bending algorithms for soft objects in flows

Achim Guckenberger[a,*], Marcel P. Schraml[a], Paul G. Chen[b], Marc Leonetti[c], Stephan Gekle[a]

[a] *Biofluid Simulation and Modeling, Fachbereich Physik, Universität Bayreuth, Bayreuth, Germany*
[b] *Aix-Marseille Université, CNRS, Centrale Marseille, M2P2, UMR7340, Marseille, France*
[c] *Aix-Marseille Université, CNRS, Centrale Marseille, IRPHE, UMR7342, Marseille, France*

ARTICLE INFO

Article history:
Received 1 March 2016
Received in revised form
27 April 2016
Accepted 30 April 2016
Available online 11 May 2016

Keywords:
Helfrich bending
Laplace–Beltrami operator
Mean curvature
Capsule deformation in shear flow

ABSTRACT

One of the most challenging aspects in the accurate simulation of three-dimensional soft objects such as vesicles or biological cells is the computation of membrane bending forces. The origin of this difficulty stems from the need to numerically evaluate a fourth order derivative on the discretized surface geometry. Here we investigate six different algorithms to compute membrane bending forces, including regularly used methods as well as novel ones. All are based on the same physical model (due to Canham and Helfrich) and start from a surface discretization with flat triangles. At the same time, they differ substantially in their numerical approach. We start by comparing the numerically obtained mean curvature, the Laplace–Beltrami operator of the mean curvature and finally the surface force density to analytical results for the discocyte resting shape of a red blood cell. We find that *none* of the considered algorithms converges to zero error at all nodes and that for some algorithms the error even diverges. There is furthermore a pronounced influence of the mesh structure: Discretizations with more irregular triangles and node connectivity present serious difficulties for most investigated methods.

To assess the behavior of the algorithms in a realistic physical application, we investigate the deformation of an initially spherical capsule in a linear shear flow at small Reynolds numbers. To exclude any influence of the flow solver, two conceptually very different solvers are employed: the Lattice–Boltzmann and the Boundary Integral Method. Despite the largely different quality of the bending algorithms when applied to the static red blood cell, we find that in the actual flow situation most algorithms give consistent results for both hydrodynamic solvers. Even so, a short review of earlier works reveals a wide scattering of reported results for, e.g., the Taylor deformation parameter.

Besides the presented application to biofluidic systems, the investigated algorithms are also of high relevance to the computer graphics and numerical mathematics communities.

1. Introduction

The computer simulation of soft deformable objects such as cells, synthetic capsules or vesicles in three-dimensional (3D) hydrodynamic flows is a rapidly increasing field in computational physics. The smallest investigated systems consider the dynamic motion of a single object in shear or channel flow [1–15], in a gravitational field [16–19], through narrow constrictions [20–22], or the diffusion of small particles near elastic membranes [23]. On a larger scale, a number of studies focus on the effective viscosity of dense suspensions [24–29] which is closely connected to the formation of cell-free layers near the channel walls in case of blood flow [30–33]. The investigation of suspensions containing two or more types of particles is another important field in which usually one focuses on the cross-streamline migration of the particles [30,32,34–41]. From a computational perspective, an adequate method for the above problems requires two ingredients: Solution of a hydrodynamic problem for the flow for which a variety of methods such as Boundary Integral [42–44], Lattice–Boltzmann [28,45–48] or particle methods [49–52] are available, and solution of a solid mechanics problem for the objects' interfaces.

The investigated objects are filled with fluid, separated from the outside by a membrane which is typically modeled as an infinitely thin elastic sheet. Forces originating from the linearized deformation of such a sheet can be split into in-plane elasticity (shear and area dilatation) and out-of-plane (bending)

[*] Corresponding author.
E-mail address: achim.guckenberger@uni-bayreuth.de (A. Guckenberger).

http://dx.doi.org/10.1016/j.cpc.2016.04.018
0010-4655/© 2016 Elsevier B.V. All rights reserved.

components. For the former a number of elastic laws such as neo-Hookean (e.g. [3,53]) or Skalak [54] have been proposed, depending on the physical properties of the studied object, and recently different numerical modeling approaches have been compared [55]. Bending contributions are very often described via a simple law proposed by Canham and Helfrich [56,57], stating that the local bending energy density is proportional to the square of the local mean curvature. Depending on the type of object, different contributions may dominate the total force. Vesicles, for example, lack shear elasticity and are thus entirely dominated by bending forces [5–7,11,58]. For elastic capsules, on the other hand, the elasticity governs most of the behavior, with bending causing mostly secondary effects [3,28,53,58]. However, in certain situations it can become the dominating factor. For instance, it defines the wavelength of local wrinkles appearing for capsules especially at low shear rates due to local compressive forces [3,53,59–63]. Neglecting the bending rigidity in this case reduces not only the numerical stability but also the physical reliance greatly, making realistic simulations practically impossible [3,28,53,61,64]. For red blood cells, both elasticity and bending are relevant, where the latter determines the equilibrium shape [65]. Hence, the accuracy of the employed bending algorithm is of major concern.

To compute the mechanics of the membrane, it is typically discretized via a set of marker points whose positions are advected with the hydrodynamic flow. The most flexible, most easy-to-implement and therefore also one of the most widely used methods to interpolate between the nodes is a discretization via flat triangles [18,43,58,63,66–69]. Recently, subdivision surface methods [4,70–74] are becoming increasingly popular, too. Other methods include curved triangles [2,75,76], B-Splines [3], or global approaches such as spherical harmonics [10,20,77]. The latter are most efficient for not too large deformations. Bending forces are computed as the derivative of the out-of-plane stress which, by the principle of virtual work, is the variational derivative of the bending energy [78,79]. Since the mean curvature already contains the second derivative, in total the fourth derivative of the surface geometry is required. This poses a severe algorithmic and numerical challenge because the surface discretizations are often not $\mathcal{C}^4$ smooth.

Here we study a set of six algorithms to compute the bending forces for the most common case of a membrane discretized via flat triangles. A major difference between the algorithms is their approach on the Laplace–Beltrami operator, a key component of the bending forces. Note that its discretization is subject to active research [80–87]. For this work, we employ methods that are devised by or based on principles of Kantor and Nelson [88, hereafter called Method A], Gompper and Kroll [89, Method B], Meyer et al. [90, Method C], Belkin et al. [84, Method D], Farutin et al. [68, Method E] and Loop and Cirak et al. [73,74, Method S]. The latter, albeit being a subdivision method, also departs from flat triangles. To the best of our knowledge, no publication has so far used Belkin et al.'s discretization (Method D) for the computation of bending forces. In a recent work, Tsubota [69] compared three different algorithms akin to Methods A and C. He considered a shear flow setup and the equilibrium shape of a red blood cell (RBC), finding that Method A shows notable deviations to C. No comparison with an analytically solvable reference shape or earlier simulation work was attempted.

As a start we calculate the discretization error for the analytically known surface of an RBC. We find a strong difference in the quality and robustness of the algorithms: Most are very sensitive to the surface discretization and *none* converge at all nodes as the resolution is increased. The results are summarized in Tables 2 and 3. To assess the performance of the bending algorithms in a typical flow setup, we then investigate the deformation of an initially spherical capsule in a viscous shear flow. The capsule is endowed with both shear and bending rigidity. To exclude any artifacts possibly arising from the flow solver, we use the Boundary Integral (BIM) as well as the Lattice–Boltzmann method (LBM). In general we find a very good agreement between both approaches. The deviations between the six bending algorithms are much less pronounced in this setup than in the analytical part. A comparison with the literature, however, reveals a wide scattering of reported values for the Taylor deformation parameter.

We finally note that our study may also be relevant in other areas where the numerical evaluation of the Laplace–Beltrami operator, which is a main focus of this work, plays an important role. In geometry processing, for example, it is often used for the visualization of high-curvature regimes, highlighting of surface details, or surface smoothing and reconstruction [81,84,90,91].

2. Computation of bending forces

2.1. The physical model of the bending energy

All bending algorithms used in the present work and in the majority of the literature depart from the seminal works of Canham [56] and Helfrich [57]. They considered a three-dimensional soft object with an infinitely thin interface endowed with bending resistance. They then proposed the following constitutive law for the bending energy:

$$E_{\mathrm{B}} = 2\kappa_{\mathrm{B}} \int_S [H(\boldsymbol{x})]^2 \,\mathrm{d}S(\boldsymbol{x}). \tag{1}$$

Henceforth, S is the instantaneous smooth surface of the object and κ_{B} is the bending modulus. The local mean curvature is given by $H = \frac{1}{2}(c_1 + c_2)$, where c_1 and c_2 are the local principal curvatures. H is taken to be positive for a sphere. In principle an additional term dependent on the Gaussian curvature appears in the bending energy. Fortunately, this term is constant if the topology of the object does not change [57,89]. Thus it is negligible for the purpose of force computations. A spontaneous (or reference) curvature can be included in Eq. (1) [65], but for simplicity we take the minimum energy reference state as a flat sheet.

For later convenience, we introduce an alternative expression for H [81]:

$$H(\boldsymbol{x}) = \frac{1}{2} \sum_{i=1}^{3} (\Delta_{\mathrm{S}} x_i)\, n_i(\boldsymbol{x}), \quad \boldsymbol{x} \in S. \tag{2}$$

$\boldsymbol{n}(\boldsymbol{x})$ is the outer normal vector of the membrane surface S at position $\boldsymbol{x}$ and $\Delta_{\mathrm{S}} = \nabla^{\mathrm{S}} \cdot \nabla^{\mathrm{S}}$ denotes the Laplace–Beltrami operator with ∇^{S} being the surface gradient. Subscripts indicate vector components.

2.2. Principles for the computation of bending forces

The general goal is to compute the forces from the bending energy (1) while using an approximation for the surface S. As outlined in the introduction, we approximate S via flat triangles, i.e. each surface element consists of three nodes (vertices) and three straight edges. The force is then required at each node $\boldsymbol{x}^{(i)}$ with $i = 1, \ldots, N$. We denote by N the number of nodes and by N_{T} the number of triangles.

To be more precise, the hydrodynamic simulations performed in Section 4 require either the force $\boldsymbol{F}^{\mathrm{h}}(\boldsymbol{x}^{(i)})$ (LBM) or the traction jump $\triangle \boldsymbol{f}^{\mathrm{h}}(\boldsymbol{x}^{(i)}) := (\sigma_+ - \sigma_-) \cdot \boldsymbol{n}$ (BIM). σ_+ and σ_- are the stress tensors at the outside and inside of S, respectively, and $\boldsymbol{n}$ again the outer normal vector. The force equilibrium conditions read [53,58]

$$\boldsymbol{F}^{\mathrm{h}} = -\boldsymbol{F} \quad \text{and} \tag{3}$$

$$\triangle \boldsymbol{f}^{\mathrm{h}} = -\boldsymbol{f}. \tag{4}$$