

Three-dimensional boundary integral modeling of viscous drops and capsules

Citation for published version (APA):

Ahmadi Joneidi, A. (2014). *Three-dimensional boundary integral modeling of viscous drops and capsules*. [Phd Thesis 1 (Research TU/e / Graduation TU/e), Mechanical Engineering]. Technische Universiteit Eindhoven. <https://doi.org/10.6100/IR769556>

DOI:

[10.6100/IR769556](https://doi.org/10.6100/IR769556)

Document status and date:

Published: 01/01/2014

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

**Three-dimensional boundary integral modeling of
viscous drops and capsules**

Three-dimensional boundary integral modeling of viscous drops and capsules /
by Amin Ahmadi Joneidi.

Technische Universiteit Eindhoven, 2014.

A catalogue record is available from the Eindhoven University of Technology
Library

ISBN: 978-90-386-3570-5

Subject headings: Boundary Integral Method, Isogeometric Analysis, Drop
Deformation, Inextensible Membrane, Red Blood Cell

Copyright ©2014 by Amin Ahmadi Joneidi. All rights reserved.

Cover design: Paul Verspaget, Neunen (Verspaget & Bruinink)

Reproduction: University Press Facilities, Eindhoven, The Netherlands.

Three-dimensional boundary integral modeling of viscous drops and capsules

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Technische Universiteit Eindhoven, op gezag van de
rector magnificus, prof.dr.ir. C.J. van Duijn, voor een
commissie aangewezen door het College voor
Promoties in het openbaar te verdedigen
op donderdag 27 februari 2014 om 16.00 uur

door

Amin Ahmadi Joneidi

geboren te Ghaem Shahr, Iran

Dit proefschrift is goedgekeurd door de promotoren en de samenstelling van de promotiecommissie is als volgt:

voorzitter:	prof.dr. L.P.H. de Goey
promoter:	prof.dr.ir. P.D. Anderson
copromoter:	dr.ir. C.V. Verhoosel
leden:	prof.dr. D. Bothe (Technische Universität Darmstadt)
	dr. A. Reali (Università degli Studi di Pavia)
	prof.dr.ir. F.N. van de Vosse
	prof.dr F. Toschi

Contents

1	Introduction	1
1.1	Drops, vesicles and red blood cells.....	1
1.2	Numerical simulation using the boundary integral formulation	2
1.3	Research objectives	4
1.4	Outline.....	5
2	Boundary integral formulation for deformable objects in Stokes flow	7
2.1	Boundary integral method	8
2.2	Deformable objects	15
2.2.1	A droplet	16
2.2.2	An inextensible membrane	18
2.2.3	A vesicle	19
2.2.4	A red blood cell.....	20
2.3	Conclusions	21
2.A	Variation of the Stokes theorem.....	21
3	Interface discretization	23
3.1	Introduction.....	23
3.2	Triangular mesh representation.....	24
3.2.1	Icosahedron-based triangular mesh	24
3.2.2	Octahedron-based triangular mesh	25
3.2.3	Curvature evaluation	25
3.2.4	Gradient of the mean curvature	32
3.2.5	Difficulties of the traditional methods to find the mean curvature	34

3.3	Isogeometric analysis.....	38
3.3.1	B-spline surface parametrization	39
3.3.2	Initial geometry representation	42
3.3.3	Approximation of the velocity field	44
3.4	Conclusions	48
4	Isogeometric analysis for drop deformation	49
4.1	Introduction.....	49
4.2	Mathematical formulation	50
4.3	Numerical results	52
4.3.1	Verification	53
4.3.2	Spatial discretization aspects	58
4.3.3	Comparison of the L^2 -projection and collocation method	62
4.4	Conclusions	65
5	Inextensible membrane	67
5.1	Introduction.....	67
5.2	Mathematical formulation	69
5.3	Solution algorithm	71
5.3.1	Surface tension discretization	71
5.3.2	Isogeometric collocation	74
5.4	Numerical simulations	77
5.4.1	The incompressibility and inextensibility conditions.....	80
5.4.2	The surface tension approximation	82
5.4.3	The evolution of the inclination angle	86
5.5	Conclusions	89
5.A	Non-dimensionalization of the inextensible membrane formulation .	91
5.B	Surface gradient of the scalar-valued field	92
5.C	Verification of the surface gradient of a scalar-valued field formulation	93

6	Hybrid spline geometry	97
6.1	Introduction.....	97
6.2	The hybrid spline geometry formulation.....	99
6.2.1	The weighted-average hybrid geometry	99
6.2.2	Numerical evaluation of the parametric coordinate mapping	103
6.2.3	Evaluation of parametric gradients	105
6.3	Algorithmic aspects.....	106
6.3.1	Hybrid spline basis functions	107
6.3.2	Weight function definition	109
6.4	Approximation properties	111
6.4.1	Normal vector	111
6.4.2	Mean curvature.....	113
6.5	Numerical results	119
6.5.1	Drop deformation	119
6.5.2	Inextensible membrane deformation.....	121
6.5.3	Vesicle deformation	124
6.6	Conclusions	124
7	Conclusions and recommendations	125
7.1	Conclusions	125
7.2	Recommendations	127
	References	129
	Summary	137
	Samenvatting	141
	Acknowledgements	145
	Curriculum Vitae	147

1.1 Drops, vesicles and red blood cells

Deformable objects such as drops, vesicles and red blood cells have received considerable research interest due to their industrial applications as well as their biomedical relevance. For instance, to understand the properties of polymer blends intended for fiber-, film- or bulk-plastic production, formation of emulsions or water-in-water biopolymer mixtures, it is important to study the dynamics and deformation of drops in a continuous phase. As such, the research on drop deformation in multiphase systems has been studied extensively and remains a topic of active research. Single drop deformation studies provide valuable insights into the behavior of systems with multiple drops. The research on vesicles and red blood cells is motivated by the need for getting a better understanding of *e.g.* thrombus formation, diabetes and thalassemia. Increasing the understanding of the motion of such deformable objects is key to the development of more effective treatment strategies.

The motion of drops is generally governed by low Reynolds number flows. Think for example of the formation of (most) emulsions and deformable objects in micro-fluidic devices, for which the particles are sufficiently small to result in low Reynolds numbers. In such situations, inertia effect can be neglected. Drop deformation was initially studied experimentally by Taylor [1], who observed that a drop deforming in shear flow orients in a fixed direction until either a balance between the viscous and interfacial forces is obtained or drop breakup occurs. Torza *et al.* [2] experimentally observed that the drop breakup mechanism is dependent on the shear rate of the surrounding flow and the viscosity ratio between the internal and external fluids. Bartok and Mason [3] experimentally investigated the coalescence of colliding viscous drops. Bentley and Leal considered the deformation and breakup of drops and found the critical shear rate for different viscosity ratios [4]. Stone extended this research to more

complicated imposed flows, such as uniaxial extensional flows and time-periodic flows [5].

The physical structure of the membrane of a vesicle – which exists naturally in the human body and can also be created artificially (*e.g.* for drug delivery systems) – is considerably more complex than that of a drop. Such a membrane is formed by amphiphilic molecules which form a bilayer that encapsulates the internal substances [6, 7]. Of even higher complexity is the membrane of a red blood cell, for which a cytoskeleton network is attached to a phospholipid membrane. In contrast to the vesicle, the membrane of a red blood cell carries shear forces in addition to bending forces [8]. In many circumstances, however, the shear resistance of the membrane is negligible compared to its bending resistance, which, from the modeling perspective, allows for the interpretation of a red blood cell as a vesicle.

An essential physical characteristic of a vesicle (or red blood cell) is that, unlike for the drop, its membrane keeps its surface area unchanged during deformation. Depending on the ratio of their volume and surface area, vesicles and red blood cells can have several shapes in unstressed condition [9, 10]. The behavior of a vesicle has been studied by *e.g.* Fischer *et al.* [11], who observed elongation and orientation of a vesicle in a shear flow while the membrane was rotating around the interior of the vesicle (tank-treading behavior). Skalak and Branemark [12] observed the tumbling motion of a vesicle, in which case the vesicle rotates like a rigid body. Kantsler and Steinberg presented experimental observations on the transition between tank-treading and tumbling motion in the dynamics of a vesicle subjected to a shear flow [13]. Abkarian *et al.* investigated the deformation of a vesicle close to a wall and computed the lift force in a shear flow [14].

1.2 Numerical simulation using the boundary integral formulation

Although experimental research on the motion of deformable objects is indispensable, the possibilities for such research are restricted. On one hand this is caused by the small size of the objects under consideration. Moreover, it is evident that experimental studies of vesicles and red blood cells are not easily conducted. In contrast to experimental studies, analytical methods can be exploited without fundamental difficulties. However, the applicability of such methods is generally limited to simplified situations which restricts the validity of the models under consideration. Numerical methods extend the range of applicability of the models used in conjunction with the analytical methods, which renders these models useful for the simulation of problems of practical

interest. For that reason, numerical analysis is of great importance in the study of drops, vesicles and red blood cells.

Numerical studies on the motion of deformable objects are primarily concerned with the simulation of the dynamic behavior of the interfaces. From the vantage point of numerical modeling this raises the question whether the motion of an interface can be simulated without explicitly representing its surrounding (both interior and exterior) fluid flow? A positive answer to this question is provided by the Boundary Integral Method (BIM) [15, 16]. The central idea in the BIM is that under certain conditions a volumetric multiphase flow formulation can be cast into a formulation that involves the interface between the various fluids only. The fact that in this resulting formulation there is no need for the flow to be explicitly represented can be very advantageous, particularly for complex geometries for which volumetric meshing (as would be necessary for applying *e.g.* the finite element method) can be cumbersome. The above-mentioned condition under which the volumetric problem can be reduced to a surface problem mainly revolves around the availability of fundamental solutions [15]. Such fundamental solutions, or Green's functions, in practice restrict the applicability of the BIM to linear time-independent problems [15, 17]. Although these restrictions are severe, the BIM is still very well applicable to a wide range of practical problems. Examples of applications in which BIM is successfully employed are multiphase flows, acoustics and soil-structure interactions, just to name a few [17].

As a consequence of the low Reynolds numbers encountered in the applications under consideration in this thesis, the motion of the interfaces can be well described by the steady Stokes equations. The availability of fundamental solutions for these problems, known as Stokeslets, makes application of the BIM possible. In fact, over the past decades, the BIM has been found to provide an accurate and reliable way of dealing with interfacial flow problems, which has made this method widely applied in this field. For example, the BIM has been used for the simulation of drop deformation and breakup [18, 19], drop coalescence [20, 21], drop sliding on an inclined plane [22], drop-drop interaction [23], drop deformation in confined geometries [24], inextensible capsule deformation [25], vesicle migration in Poiseuille flow [26], and many other multiphase flow phenomena.

An essential ingredient of the BIM for the simulation of deformable objects suspended in Stokes flow is the evaluation of the stress jump across an interface. In general, this stress jump is a function of the geometry of the interface through quantities such as the surface normal vector, curvature, and, in the case of a vesicle or red blood cell, surface gradients of these geometric quantities. It is self-evident that successful application of the BIM to such problems requires the

evaluation of these surface stress jumps to be accurate and efficient.

1.3 Research objectives

The evaluation of the stress jump across the interface of a drop, vesicle or red blood cell is non-trivial when using non-smooth continuous surface meshes, as is common in boundary integral methods [15, 16]. On such meshes, the curvature – which appears in the stress jump of all above-mentioned deformable objects – is not defined along the element boundaries. More severe problems occur for the evaluation of the derivatives of the curvature, which appear in the stress jump for the vesicle or red blood cell. Proper numerical treatment of these difficulties is required to obtain a robust boundary integral formulation for the simulation of the motion of these deformable objects.

The goal of this thesis is to develop robust numerical models for the simulation of deformable objects of increasing complexity, and, as such, to work toward a numerical tool for the simulation of red blood cells. In the first part of this thesis, this goal is pursued by enhancing a non-smooth geometry mesh with advanced stress jump evaluation strategies, which permit for the consistent evaluation of the higher-order surface gradients that appear in the BIM. In the second part of this thesis, an alternative route to accomplishing the main research goal is followed by the introduction of a smooth surface parametrization by means of B-splines, which facilitates the direct evaluation of the higher-order stress jump terms. This idea is motivated by the recent developments on isogeometric analysis (IGA) [27], which has been applied to a wide variety of fields [28] including boundary element/integral methods [29–32].

As indicated above, in this thesis a series of objects with stress jump terms of increasing complexity is considered. The starting point will be a drop, for which the stress jump is governed by isotropic and homogeneous surface tension only, which results in the stress jump across the interface given by

$$\mathbf{f} = 2\sigma H\mathbf{n},$$

where σ is the interface tension, H is the mean curvature of the interface and \mathbf{n} is the unit outward-pointing normal vector to the interface. Subsequently, an inextensible membrane is considered as a prototypical model for the vesicle or red blood cell. In this case, the stress jump is given by

$$\mathbf{f} = 2\tau H\mathbf{n} - \nabla_s \tau,$$

where ∇_s is the surface gradient operator and τ is the surface tension that

enforces the membrane to keep its surface area fixed. This inextensibility condition results in an implicit definition of the surface tension. Finally, building on top of the inextensible membrane formulation, the stress jump in a vesicle or red blood cell is studied:

$$\mathbf{f} = \kappa \left(\frac{H^3}{2} - 2KH + \Delta_s H \right) \mathbf{n}.$$

In this expression, K is the Gaussian curvature and κ is the membrane rigidity. Evidently, compared to the inextensible membrane this stress jump poses the additional challenge of accurately evaluating the surface Laplacian of the mean curvature.

1.4 Outline

This thesis consists of seven chapters. In Chapter 2, the problem setup for a deformable object in a background flow is introduced along with the boundary integral formulation for a two-fluid Stokes flow. The increasingly complex stress jump terms are derived for the drop, the inextensible membrane, the vesicle and the red blood cell.

Chapter 3 consists of two parts. In the first part, the traditional boundary integral method, which is based on a non-smooth surface triangulation, is discussed. Various methods to create the triangular meshes are considered along with two strategies for the evaluation of the curvature and its surface gradients. The accuracy of both these strategies is assessed in the first part of this chapter. In the second part, isogeometric analysis is introduced as an alternative method to parametrize the interface, which result in the isogeometric boundary integral formulation. The construction of a B-spline parametrization along with the corresponding control net is discussed. Furthermore, two solution methods for obtaining the control point velocities are proposed.

In Chapter 4, the isogeometric boundary integral formulation is applied to the simulation of a drop (mainly) in shear flow. The drop simulations are used to extensively verify the isogeometric approach by studying the convergence behavior of the method under both mesh refinement and order elevation. Some algorithmic details such as the numerical integration schemes are studied in detail in this chapter.

In Chapter 5, the deformation of an inextensible membrane – which can be considered as a prototypical model for a red blood cell – is simulated using the isogeometric boundary integral formulation. An algorithm that enforces

the inextensibility condition, thereby implicitly defining the surface tension, is outlined. The isogeometric approach is verified using a series of simulations of inextensible membranes suspended in shear flow.

In Chapter 6, the difficulties – in evaluating higher order parametric derivatives at the poles of the parametrized geometry – associated with applying the bending force involved at the stress discontinuity over the interface of a vesicle or red blood cell are discussed. To alleviate these problems, the use of a hybrid geometry representation is proposed with the idea of defining two parametrizations of the same geometry and creating the hybrid geometry as the weighted average of these two parametrizations. An extensive comparison between the range of different quantities (*e.g.* the normal vector and mean curvature) is presented and the deformation of a drop, an inextensible membrane and finally a vesicle is simulated using the isogeometric boundary integral formulation applying the new hybrid geometry.

Finally conclusions and some recommendations are given in Chapter 7.

Boundary integral formulation for deformable objects in Stokes flow

The flow problem studied in this thesis is schematically shown in Figure 2.1, which shows an object (domain $\Omega^{(2)}$) suspended in a matrix fluid (domain $\Omega^{(1)}$), while the common interface between these two phases ($\Omega^{(1)} \cap \Omega^{(2)}$) is denoted by Γ . The object and matrix fluids are considered incompressible and Newtonian. In addition, inertia effects are neglected and no body force is assumed. Considering these conditions, the motion of the fluid is governed by the Stokes equations [33]

$$\begin{cases} -\nabla p^{(i)} + \mu^{(i)} \nabla^2 \mathbf{u}^{(i)} = \mathbf{0} \\ \nabla \cdot \mathbf{u}^{(i)} = 0 \end{cases} \quad \text{on } \Omega^{(i)}, \text{ with } i = 1 \text{ or } 2, \quad (2.1)$$

where $\mathbf{u}^{(i)}$, $p^{(i)}$ and $\mu^{(i)}$ denote the velocity field, pressure field and the viscosity of the i^{th} phase, respectively.

The stress jump and the velocities across the interface serve as boundary conditions, and are expressed as

$$\begin{cases} (\boldsymbol{\Pi}^{(1)} - \boldsymbol{\Pi}^{(2)}) \cdot \mathbf{n}(\mathbf{x}) = \mathbf{f}(\mathbf{x}) \\ \mathbf{u}^{(1)} = \mathbf{u}^{(2)} \end{cases} \quad \text{on } \Gamma, \quad (2.2)$$

where $\mathbf{n}(\mathbf{x})$ is the unit outward normal vector to Γ , $\mathbf{f}(\mathbf{x})$ is a surface force and $\boldsymbol{\Pi}^{(i)} = -p^{(i)} \mathbf{I} + \mu^{(i)} (\nabla \mathbf{u}^{(i)} + (\nabla \mathbf{u}^{(i)})^T)$ is the stress tensor at the surface with \mathbf{I} the identity tensor.

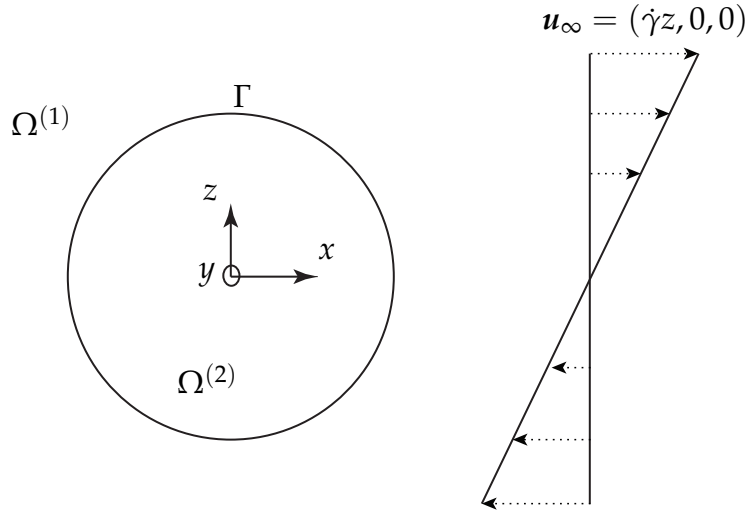


Figure 2.1 Schematic representation of the fluidic domains $\Omega^{(1)}$ and $\Omega^{(2)}$. Γ denotes the common interface between two phases ($\Omega^{(1)} \cap \Omega^{(2)}$). $\dot{\gamma}$ and \mathbf{u}_∞ are the shear rate and the far-field imposed velocity, respectively.

2.1 Boundary integral method

Rheology of the fluids and multiphase flows are part of fluid mechanics which have been studied using numerous numerical methods and attracted considerable interest in the past decades. One of the most attractive and well-known example of these sort of investigations is the blood rheology study which has a vital role and direct relation to the human life. Since red blood cells make nearly 45% of the human blood, RBC deformation and behavior occupied a considerable part of research interest in this field. Although the complication of the RBCs results further simplifications of its modeling, this kind of numerical simulations remained expensive and needs many parallel supercomputers. These numerical studies on massive number of RBCs are possible when the nature of blood is kept at the minimum level of the physical details of each cells. Lattice Boltzmann method(LBM) [34] has been recently used as a good tool to simulate thousands or millions of RBCs *e.g.* blood flow in plane cylindrical [35], 3D simulation of large number of deformable particles suspended in viscous flow [36] and blood cell dynamics [37]. In addition, the smoothed particle hydrodynamics (SPH) method has been implemented by Tsubota *et al.* [38, 39] to represent the cell membrane by particles connected by springs that produce the resistance features of the RBCs. Next to this, to catch the more complex properties of the RBCs or the other deformable objects and visualizing the complicated characteristics of one cell deformation, cell-cell interaction, cell-wall interactions and in general, reducing

the simplifications, the other numerical methods are presented and implemented. Most of these numerical techniques can be categorized as either front-capturing or front-tracking methods.

Volume-of-fluid (VOF) [40–43], level set [44, 45], combination of level set and VOF [46], continuum surface force [47] and the diffusive-interface method [48] are some examples of the capturing techniques which are referred also as the Eulerian approaches. This approach is the way of looking at the fluid motion or a certain location in the space through the time [33]. These methods consider the interface often without explicitly tracking the interface by conforming elements. A continuous scalar variable is defined with value 1 and 0 in the first and second fluid phases, respectively, and the interface is defined by the phase transition function. The flow convects the phase field function to mimic the interface movement. Thus, these methods are capable to determining the interface deformation, without explicitly tracking the interface. Although, topological changes such as merging, breaking, etc can straight forwardly be modeled by capturing techniques, they have disadvantages as well. More importantly, interface diffusion, maintaining a sharp boundary between the phases and mass conservation problems result in a loss accuracy.

The second approach is based on tracking the movement of the interface and is usually referred to as the Lagrangian approach. In these methods, the interface is explicitly represented by elements and the movement of the interface is tracked in time [33, 49–51]. Marker and cell method [52, 53] and finite element method [54–56] are used to track the interface and are such examples of this category.

The boundary integral method is a particular example of this category [15, 16]. The considerable advantage of this method is that a mesh is only needed at the surface, and the problem becomes two dimensional instead of fully three dimensional. Also, the benefits of the boundary integral method compared to all of the volumetric mesh methods are not only limited to the simplified mesh representation. Recent advances in boundary integral formulations allow the thin region in cell-cell and wall-cell interactions to be better resolved. This permits the simulations *e.g.* drop coalescence [20, 57] or drop sliding down an inclined plane [22]. Furthermore, this method is developed, coupled with and implemented in several applications such as spectral boundary integral method for RBCs flowing in complex geometries [58], a cross-flow migration of rigid particles [59], motion of an array of drops through a cylindrical tube [60], red blood cell simulation by spherical harmonic BIM [61], drop deformation in confined geometry [24], non-singular boundary integral method [23] and several more researches [22, 62–66].

A limitation of the boundary integral method is that only Newtonian fluids in creeping flow (no inertia condition) can be described. More specifically, only

linear time-independent equations for which the Green's functions are available can be modeled and solved by this method.

It is common to display the solution of linear and homogeneous boundary value problems in terms of boundary integrals [67]. Harmonic functions [68] and displacement fields in linear elastostatics [69] are some examples of boundary integral representations. For the Stokes flow problem, Equation (2.1), the integral form involves the boundary values of the velocity and surface stress jump.

As a start point to derive the boundary integral form of the Stokes equations, the Lorentz reciprocal identity is introduced [70]. Assuming \mathbf{u} and \mathbf{u}' are two fundamental solutions of the Stokes equations and $\mathbf{\Pi}$ and $\mathbf{\Pi}'$ are the associated stress tensors, the reciprocal identity is given by [70]

$$\frac{\partial}{\partial x_k} (u'_i \Pi_{ik} - u_i \Pi'_{ik}) = 0. \quad (2.3)$$

Identifying \mathbf{u}' with the flow due to a point force, \mathbf{g} , acting on the point \mathbf{x}_0 , it is obtained that [15]

$$u'_i(\mathbf{x}) = \frac{1}{8\pi\mu} G_{ij}(\mathbf{x}, \mathbf{x}_0) g_j, \quad \Pi'_{ik}(\mathbf{x}) = \frac{1}{8\pi} T_{ijk}(\mathbf{x}, \mathbf{x}_0) g_j. \quad (2.4)$$

G and T are the Green's functions and are found to be different for various cases. For example, the free space Green's functions are

$$G_{ij} = \frac{\delta_{ij}}{r} + \frac{\hat{x}_i \hat{x}_j}{r^3}, \quad T_{ijk} = \frac{\hat{x}_i \hat{x}_j \hat{x}_k}{r^5}, \quad (2.5)$$

where $r = |\hat{\mathbf{x}}|$ and $\hat{\mathbf{x}} = \mathbf{x} - \mathbf{x}_0$. For flow bounded by an infinite plane wall [71]

$$\mathbf{G}^w = \mathbf{Y}(\hat{\mathbf{x}}) - \mathbf{Y}(\hat{\mathbf{X}}) + 2h_0^2 \mathbf{G}^D(\hat{\mathbf{X}}) - 2h_0^2 \mathbf{G}^{SD}(\hat{\mathbf{X}}), \quad (2.6)$$

where \mathbf{Y} is the Stokeslet, $h_0 = x_0 - w$, $\hat{\mathbf{x}} = \mathbf{x} - \mathbf{x}_0$, $\hat{\mathbf{X}} = \mathbf{x} - \mathbf{x}_0^{IM}$ and \mathbf{x}_0^{IM} is the image of \mathbf{x}_0 with respect to the wall. \mathbf{G}^D and \mathbf{G}^{SD} are also defined as

$$G_{ij}^D(\mathbf{x}) = \pm \left(\frac{\delta_{ij}}{|\mathbf{x}|^3} - 3 \frac{x_i x_j}{|\mathbf{x}|^5} \right), \quad (2.7)$$

and

$$G_{ij}^{SD}(\mathbf{x}) = x_1 G_{ij}^D(\mathbf{x}) \pm \frac{\delta_{j1}x_i - \delta_{i1}x_j}{|\mathbf{x}|^3}, \quad (2.8)$$

where minus and plus signs depend on the geometrical axes. For the parallel wall configuration, the Green's functions are obtained by Liron and Mochon [72] and Jones [73] which involve the integration of Bessel functions.

In continuing with the free space, substituting Equation (2.4) into Equation (2.3) leads to

$$\frac{\partial}{\partial x_k} [G_{ij}(\mathbf{x}, \mathbf{x}_0)\Pi_{ik}(\mathbf{x}) - \mu u_i(\mathbf{x})T_{ijk}(\mathbf{x}, \mathbf{x}_0)] = 0, \quad (2.9)$$

where this is an equation that can be solved for \mathbf{u} . Consider a control volume, V , enclosed by a surface, D . Now, a point \mathbf{x}_0 which is the center of a small sphere, referred as V_ϵ with radius of ϵ , encircled by S_ϵ is considered inside of V (see Figure 2.2).

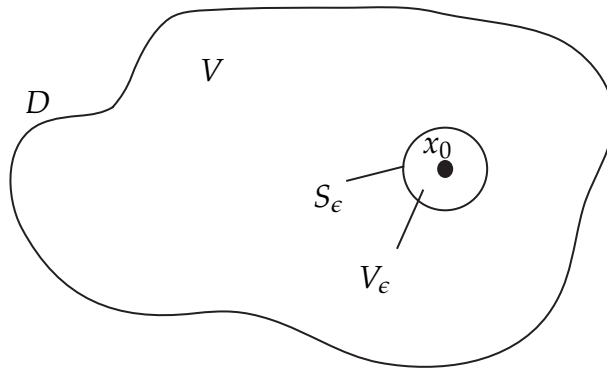


Figure 2.2 A control volume, V , enclosed by a surface, D , within the domain of a flow.

Taking the integral of Equation (2.9) over $V - V_\epsilon$ and converting that into a surface integral using the divergence theorem leads to

$$\int_{\Omega^*} [G_{ij}(\mathbf{x}, \mathbf{x}_0)\Pi_{ik}(\mathbf{x}) - \mu u_i(\mathbf{x})T_{ijk}(\mathbf{x}, \mathbf{x}_0)] n_k(\mathbf{x})dS(\mathbf{x}) = 0, \quad (2.10)$$

where Ω^* could be either D or S_ϵ . Over ϵ which is the sphere with radius S_ϵ , the

tensors, \mathbf{G} and \mathbf{T} are

$$G_{ij} = \frac{\delta_{ij}}{\epsilon} + \frac{\hat{x}_i \hat{x}_j}{\epsilon^3}, \quad T_{ijk} = \frac{\hat{x}_i \hat{x}_j \hat{x}_k}{\epsilon^5}. \quad (2.11)$$

Also, over S_ϵ , $\mathbf{n} = \hat{\mathbf{x}}/\epsilon$ and $dS = \epsilon^2 d\Omega$ where dS is infinitesimal surface element and Ω is the differential solid angle (the area element on the sphere is given in spherical coordinate (r, θ, ϕ) by $dA = r^2 \sin(\theta) d\theta d\phi$ where the differential solid angle $d\Omega = \sin(\theta) d\theta d\phi$). Substituting these expressions and Equation (2.11) into Equation (2.10), gives

$$\begin{aligned} \int_D [G_{ij}(\mathbf{x}, \mathbf{x}_0) \Pi_{ik}(\mathbf{x}) - \mu u_i(\mathbf{x}) T_{ijk}(\mathbf{x}, \mathbf{x}_0)] n_k(\mathbf{x}) dS(\mathbf{x}) = \\ - \int_{S_\epsilon} \left[\left(\delta_{ij} + \frac{\hat{x}_i \hat{x}_j}{\epsilon^2} \right) \Pi_{ik}(\mathbf{x}) + 6\mu u_i(\mathbf{x}) \frac{\hat{x}_i \hat{x}_j \hat{x}_k}{\epsilon^4} \right] \hat{x}_k d\Omega. \end{aligned} \quad (2.12)$$

\mathbf{u} and $\mathbf{\Pi}$ over S_ϵ tend to $\mathbf{u}(\mathbf{x}_0)$ and $\mathbf{\Pi}(\mathbf{x}_0)$ as $\epsilon \rightarrow 0$. Since $\hat{\mathbf{x}}$ scales linearly with ϵ (since \mathbf{x} is on S_ϵ), the stress term in the right hand side of Equation (2.12) changes linearly while the term associated with the velocity tends to a constant value. Hence, in the limit of $\epsilon \rightarrow 0$, Equation (2.12) can be rewritten as

$$\begin{aligned} \int_D [G_{ij}(\mathbf{x}, \mathbf{x}_0) \Pi_{ik}(\mathbf{x}) - \mu u_i(\mathbf{x}) T_{ijk}(\mathbf{x}, \mathbf{x}_0)] n_k(\mathbf{x}) dS(\mathbf{x}) = \\ \lim_{\epsilon \rightarrow 0} - 6\mu u_i(\mathbf{x}_0) \frac{1}{\epsilon^4} \int_{S_\epsilon} \hat{x}_i \hat{x}_j dS(\mathbf{x}). \end{aligned} \quad (2.13)$$

From $\mathbf{n} = \hat{\mathbf{x}}/\epsilon$, it is obtained that

$$\int_{S_\epsilon} \hat{x}_i \hat{x}_j dS(\mathbf{x}) = \epsilon \int_{S_\epsilon} \hat{x}_i n_j dS(\mathbf{x}). \quad (2.14)$$

Applying the divergence theorem to the right hand side of Equation (2.14) gives

$$\epsilon \int_{S_\epsilon} \hat{x}_i n_j dS(\mathbf{x}) = \epsilon \int_{V_\epsilon} \frac{\partial \hat{x}_i}{\partial \hat{x}_j} dV(\mathbf{x}) = \frac{4}{3} \pi \epsilon^4 \delta_{ij}. \quad (2.15)$$

Substituting Equation (2.15) into Equation (2.13) then yields

$$\begin{aligned}
u_j(\mathbf{x}_0) &= -\frac{1}{8\pi\mu} \int_D \Pi_{ik}(\mathbf{x}) n_k(\mathbf{x}) G_{ij}(\mathbf{x}, \mathbf{x}_0) dS(\mathbf{x}) \\
&\quad + \frac{1}{8\pi} \int_D u_i(\mathbf{x}) T_{ijk}(\mathbf{x}, \mathbf{x}_0) n_k(\mathbf{x}) dS(\mathbf{x}).
\end{aligned} \tag{2.16}$$

By considering $\boldsymbol{\sigma} = \mathbf{\Pi} \mathbf{n}$, Equation (2.16) can be rewritten as

$$\begin{aligned}
u_j(\mathbf{x}_0) &= -\frac{1}{8\pi\mu} \int_D \sigma_i(\mathbf{x}) G_{ij}(\mathbf{x}, \mathbf{x}_0) dS(\mathbf{x}) \\
&\quad + \frac{1}{8\pi} \int_D u_i(\mathbf{x}) T_{ijk}(\mathbf{x}, \mathbf{x}_0) n_k(\mathbf{x}) dS(\mathbf{x}).
\end{aligned} \tag{2.17}$$

Let the surface force over the external side of D be indicated by the superscript (1) . Then Equation (2.17) can be written as

$$\begin{aligned}
u_j^{(1)}(\mathbf{x}_0) &= -\frac{1}{8\pi\mu_1} \int_D \sigma_i^{(1)}(\mathbf{x}) G_{ij}(\mathbf{x}, \mathbf{x}_0) dS(\mathbf{x}) \\
&\quad + \frac{1}{8\pi} \int_D u_i(\mathbf{x}) T_{ijk}(\mathbf{x}, \mathbf{x}_0) n_k(\mathbf{x}) dS(\mathbf{x}).
\end{aligned} \tag{2.18}$$

Applying the reciprocal identity for the internal flow $\mathbf{u}^{(2)}$ at the point \mathbf{x}_0 located outside of V , gives

$$\int_D \sigma_i^{(2)}(\mathbf{x}) G_{ij}(\mathbf{x}, \mathbf{x}_0) dS(\mathbf{x}) - \mu_2 \int_D u_i(\mathbf{x}) T_{ijk}(\mathbf{x}, \mathbf{x}_0) n_k(\mathbf{x}) dS(\mathbf{x}). \tag{2.19}$$

Combining Equations (2.18) and (2.19) gives

$$\begin{aligned}
u_j^{(1)}(\mathbf{x}_0) &= -\frac{1}{8\pi\mu_1} \int_D f_i(\mathbf{x}) G_{ij}(\mathbf{x}, \mathbf{x}_0) dS(\mathbf{x}) \\
&\quad + \frac{1-\lambda}{8\pi} \int_D u_i(\mathbf{x}) T_{ijk}(\mathbf{x}, \mathbf{x}_0) n_k(\mathbf{x}) dS(\mathbf{x}),
\end{aligned} \tag{2.20}$$

where λ is the viscosity ratio between the internal and external fluids and the stress discontinuity at the interface is indicated by \mathbf{f} where $\mathbf{f} = \boldsymbol{\sigma}^{(1)} - \boldsymbol{\sigma}^{(2)}$. Now,

consider \mathbf{u}^∞ past the object (the imposed velocity), according to the superposition principle ($\mathbf{u} = \mathbf{u}_\infty + \mathbf{u}^D$ where \mathbf{u}_∞ is the far-field imposed velocity and \mathbf{u}^D is the velocity given by (2.20)), the velocity of \mathbf{x}_0 when it approaches from the exterior, can be written as:

$$\begin{aligned} u_j^{(1)}(\mathbf{x}_0) &= u_{\infty_j}(\mathbf{x}_0) - \frac{1}{8\pi\mu_1} \int_D f_i(\mathbf{x}) G_{ij}(\mathbf{x}, \mathbf{x}_0) dS(\mathbf{x}) \\ &\quad + \frac{1-\lambda}{8\pi} \int_D u_i(\mathbf{x}) T_{ijk}(\mathbf{x}, \mathbf{x}_0) n_k(\mathbf{x}) dS(\mathbf{x}). \end{aligned} \quad (2.21)$$

Similarly, the boundary integral formulation for the internal flow can be found

$$\begin{aligned} u_j^{(2)}(\mathbf{x}_0) &= -\frac{1}{8\pi\mu_1\lambda} \int_D f_i(\mathbf{x}) G_{ij}(\mathbf{x}, \mathbf{x}_0) dS(\mathbf{x}) \\ &\quad + \frac{1-\lambda}{8\pi\lambda} \int_D u_i(\mathbf{x}) T_{ijk}(\mathbf{x}, \mathbf{x}_0) n_k(\mathbf{x}) dS(\mathbf{x}). \end{aligned} \quad (2.22)$$

Now, the velocity in both Equations (2.21) and (2.22) can be generalized when the point \mathbf{x}_0 approaches the interface either from the interior or exterior as:

$$\begin{aligned} u_j(\mathbf{x}_0) &= 2u_{\infty_j}(\mathbf{x}_0) - \frac{1}{(4\pi\mu_1)(\lambda+1)} \int_D f_i(\mathbf{x}) G_{ij}(\mathbf{x}, \mathbf{x}_0) dS(\mathbf{x}) \\ &\quad + \frac{1-\lambda}{4\pi(1+\lambda)} \int_D u_i(\mathbf{x}) T_{ijk}(\mathbf{x}, \mathbf{x}_0) n_k(\mathbf{x}) dS(\mathbf{x}). \end{aligned} \quad (2.23)$$

Eventually, the form of the integral formulation can be found as (for the sake of notational convenience, vector notation is used) [15, 16]:

$$\begin{aligned} (\lambda+1)\mathbf{u}(\mathbf{x}_0) &= 2\mathbf{u}_\infty(\mathbf{x}_0) - \frac{1}{4\pi\mu_1} \int_\Gamma \mathbf{G}(\mathbf{x}, \mathbf{x}_0) \mathbf{f}(\mathbf{x}) d\Gamma(\mathbf{x}) \\ &\quad - \frac{\lambda-1}{4\pi} \int_\Gamma \mathbf{u}(\mathbf{x}) \mathbf{T}(\mathbf{x}, \mathbf{x}_0) \mathbf{n}(\mathbf{x}) d\Gamma(\mathbf{x}), \end{aligned} \quad (2.24)$$

where $\mathbf{u}_\infty(\mathbf{x}_0)$ is the imposed velocity and is interpreted as the velocity of the matrix background flow which the object is suspended in it. Since there is unknown $\mathbf{u}(\mathbf{x})$ in the second integral of Equation (2.24), an iterative method is

required to find the velocity. In the iso-viscous case ($\lambda = 1$), considered in this thesis, Equation (2.24) simplifies to

$$\mathbf{u}(\mathbf{x}_0) = \mathbf{u}_\infty(\mathbf{x}_0) - \frac{1}{8\pi\mu_1} \int_{\Gamma} \mathbf{G}(\mathbf{x}, \mathbf{x}_0) \mathbf{f}(\mathbf{x}) d\Gamma(\mathbf{x}), \quad (2.25)$$

and $\mathbf{f}(\mathbf{x})$ is the stress discontinuity across the interface of the deformable object. In this case, the second integral of Equation (2.24) vanishes and computational time of velocity calculation reduces drastically.

Since \mathbf{u} is the time derivative of \mathbf{x} , Equation (2.25) is in essence an integro-differential equation for the interface position $\mathbf{x}(t)$. Therefore, the object interface evolution in time is governed by

$$\mathbf{x}(t) = \mathbf{x}(0) + \int_0^t \mathbf{u}(\tau) d\tau, \quad (2.26)$$

where in this thesis, forward Euler time stepping integration is used to evaluate the time integral in (2.26) according to

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \Delta t \mathbf{u}(t), \quad (2.27)$$

where Δt is the (dimensionless) time step. $\mathbf{u}(t)$ depends on $\mathbf{x}(t)$ and follows from (2.25).

2.2 Deformable objects

Non-rigid objects will deform under the action of externally applied forces resulting from the motion of a matrix fluid. Various kind of deformable objects are considered in this thesis, each of which is characterized by a specific stress jump across the interface. The most prominent of such objects are listed and explained in following. The surface differential operators used in this thesis are defined as:

$$\nabla_s v = \nabla v - (\mathbf{n} \cdot \nabla v) \mathbf{n}, \quad (2.28)$$

$$\nabla_s \cdot \mathbf{w} = \nabla \cdot \mathbf{w} - (\mathbf{n}(\nabla \mathbf{w})) \cdot \mathbf{n}, \quad (2.29)$$

$$\Delta_s v = \nabla_s \cdot (\nabla_s v), \quad (2.30)$$

where \mathbf{n} is outward pointing unit normal vector to the interface and v and w are arbitrary sufficiently smooth scalar- and vector-valued functions, respectively.

2.2.1 A droplet

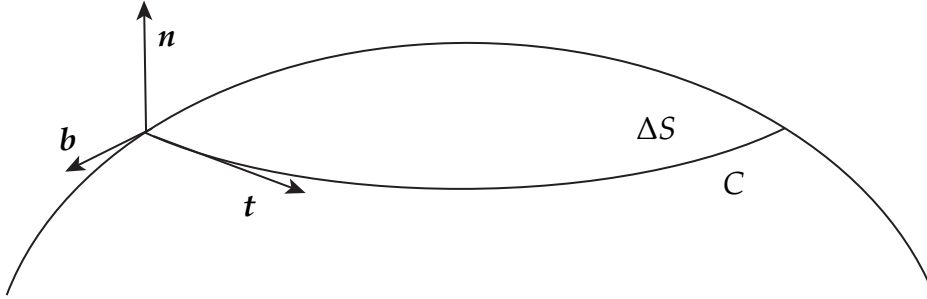


Figure 2.3 A surface piece, ΔS and bounding contour, C on a portion of the interface between two phases of fluids.

A drop has a simple membrane structure and the stress on the interface is mainly dominated by the surface or interfacial tension. A small surface portion of the interface, ΔS , that is enclosed by the contour, C , is considered in Figure 2.3. The force balance at the surface is written as [15]

$$\int_{\Delta S} \mathbf{f} dS + \int_C \sigma \mathbf{b} dl = \mathbf{0}, \quad (2.31)$$

where σ is the interfacial tension and \mathbf{b} is the unit vector normal to C . Using the identity $\mathbf{b} = \mathbf{t} \times \mathbf{n}$, the above equation can be rewritten as

$$\int_{\Delta S} \mathbf{f} dS = \int_C \sigma \mathbf{n} \times \mathbf{t} dl, \quad (2.32)$$

where \mathbf{n} is the unit vector normal to the interface and \mathbf{t} is the unit vector tangential to C as also shown in Figure 2.3. Using the variation of the Stokes's theorem given by (see Appendix 2.A):

$$\int_C \mathbf{w} \times \mathbf{t} \, dl = \int_{\Delta S} ((\nabla \cdot \mathbf{w})\mathbf{n} - (\nabla \mathbf{w})\mathbf{n}) \, dS, \quad (2.33)$$

where \mathbf{w} is a vector-valued differentiable function. The contour integral in the right hand side of Equation (2.32) can be converted into an area integral. In the current case, \mathbf{w} can be substituted by $\sigma\mathbf{n}$ to rewrite Equation (2.33) as

$$\int_C \sigma\mathbf{n} \times \mathbf{t} \, dl = \int_{\Delta S} ((\nabla \cdot (\sigma\mathbf{n}))\mathbf{n} - (\nabla(\sigma\mathbf{n}))\mathbf{n}) \, dS, \quad (2.34)$$

expansion of the right hand side yields

$$\int_C \sigma\mathbf{n} \times \mathbf{t} \, dl = \int_{\Delta S} (\mathbf{n} \cdot \nabla \sigma) \mathbf{n} + (\sigma \nabla \cdot \mathbf{n}) \mathbf{n} - (\nabla \sigma) \underbrace{\mathbf{n} \cdot \mathbf{n}}_{=1} - \sigma \underbrace{(\nabla \mathbf{n})\mathbf{n}}_{=0} \, dS, \quad (2.35)$$

where $(\nabla \mathbf{n})\mathbf{n} = (1/2)\nabla(\mathbf{n} \cdot \mathbf{n}) = (1/2)\nabla(1) = \mathbf{0}$. On the other side, the mean curvature is mathematically defined as:

$$H = \frac{1}{2} \nabla \cdot \mathbf{n}. \quad (2.36)$$

Combining Equations (2.32) and (2.35) and using (2.36), the stress discontinuity at the interface can be obtained as:

$$\mathbf{f} = 2\sigma H\mathbf{n} - \nabla_s \sigma. \quad (2.37)$$

where the surface gradient operator, ∇_s is defined in Equation (2.28). Note that, the mean curvature H is the local (scalar) mean curvature and physically defined as $H = \frac{1}{2}(1/R_1 + 1/R_2)$, where R_1 and R_2 are the radii of the curvature. The surface tension (σ) of a drop at a fixed temperature and with a clean interface can

be considered constant. Therefore, $\nabla_s \sigma = \mathbf{0}$ and Equation (2.37) reduces to

$$\mathbf{f} = 2\sigma H \mathbf{n}. \quad (2.38)$$

Hence, for a drop, the stress jump is fully governed by the constant surface tension, σ in the interface geometry.

2.2.2 An inextensible membrane

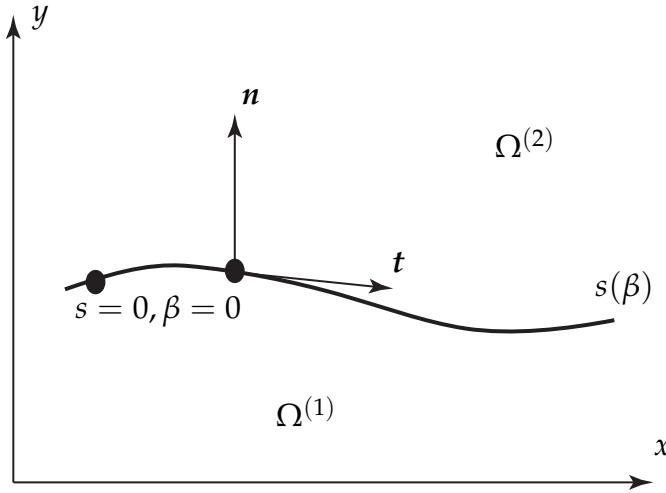


Figure 2.4 A parametrized segment of a 2D interface. s is the length along the segment of the interface and β is a parametric variable that increases monotonically along the interface.

Assume the interface ΔS shown in Figure 2.3 contains neither impurities nor surfactant, such that it can be characterized by an isotropic surface tension, τ [15]. The procedure to find the stress jump across the interface follows the derivations in Section 2.2.1 and yields

$$\mathbf{f} = 2\tau H \mathbf{n} - \nabla_s \tau, \quad (2.39)$$

where H is the mean curvature and τ is the isotropic tension. Two common ways of specifying the surface tension τ are provided by making the assumption of either elastic or inextensible interface behavior. Therefore, in principle, the surface tension of a drop is considered to be constant while this assumption

is not applicable for an inextensible or an elastic membrane which indicate the fundamental difference between these two objects. In this section, for the sake of simplicity, a 2D interface is assumed, but in the following chapter of the thesis (Chapter 5) a full 3D interface will be considered.

Figure 2.4 shows the parametrized segment of a 2D interface in the xy plane. s is the length along this segment and β is the variable along the interface. The extension ratio is defined as

$$\gamma = \frac{(\partial s / \partial \beta)_t}{(\partial s / \partial \beta)_{t=0}}, \quad (2.40)$$

where $t = 0$ denotes the moment that the interface is unstressed. In the elastic form of the interface, τ can be linearly set to

$$\tau = E(\gamma - 1), \quad (2.41)$$

where E is the modulus of the elasticity [15]. Note that when the interface is stretched, $\gamma > 1$ and when the interface is compressed, $\gamma < 1$.

On the other side, the membrane could be considered inextensible which means that the surface area is constant. In this case,

$$\frac{\partial \gamma}{\partial t} = 0, \quad (2.42)$$

which indicates that the variation of the interface with respect to β remains unchanged during the time.

2.2.3 A vesicle

Unlike the drop, a vesicle keeps its surface area constant. In fact, a high energy is required to deform a part of such a membrane. The general elastic bending energy, obtained from Hooke's law [74–79], is given by

$$E = \frac{\kappa}{2} \int_{\Gamma} H^2 d\Gamma + \kappa_G \int_{\Gamma} K d\Gamma, \quad (2.43)$$

where κ and κ_G indicate the membrane and stretching rigidity, respectively. In Equation (2.43) K is the Gaussian curvature defined by

$$K = H^2 - \nabla \mathbf{n} : \nabla \mathbf{n}^T. \quad (2.44)$$

Physically, the Gaussian curvature is defined as the multiplication of two principle curvatures $K = \frac{1}{R_1 R_2}$. In accordance with Gauss-Bonnet theorem, the second term in the integral is constant in the case that the topology of the interface remains unchanged [78, 80].

The stress discontinuity at the membrane can be obtained by taking the variational derivative of the bending energy, given in Equation (2.43) with respect to the membrane position [77]:

$$\mathbf{f} = \frac{\delta E}{\delta \mathbf{x}}. \quad (2.45)$$

Upon elaboration, it is found that (see Kaoui *et al.* [26] for derivation details).

$$\mathbf{f} = \kappa \left(\frac{H^3}{2} - 2KH + \Delta_s H \right) \mathbf{n}. \quad (2.46)$$

Remind that the second integral term of Equation (2.43) is constant and hence, the variational derivative of it with respect to the position is zero. Also, note that the term $\Delta_s H$ indicates the fourth order derivative of the position which express the fundamental difference between the stress discontinuity of the vesicle membrane (2.46) compared to that of the drop interface (2.38). In addition, because of the surface conservation at the vesicles membrane, the stress discontinuity in vesicles given by Equation (2.46) should be combined with the isotropic surface tension in Equation (2.39) and the inextensibility condition.

2.2.4 A red blood cell

In a red blood cell, a cytoskeleton network is attached to the membrane which gives it shear elasticity resistance. This leads to one additional interface energy

term for the red blood cell than for the vesicle, given by [81]

$$E_\mu = \frac{\mu}{2} \int (\lambda_1 - \lambda_2)^2 dA, \quad (2.47)$$

where μ is the shear modulus and λ_1 and λ_2 are the extensions of a relaxed skeleton patch in the two principal direction. In general, a vesicle can be supposed as a RBC without cytoskeleton since under certain circumstances, E_μ is quiet small compared to the other terms of the energy at the membrane and hence, the vesicle model closely resemble the RBC model.

2.3 Conclusions

The complexity of the stress jump across the interface of the deformable objects is discussed for a drop, vesicle and RBC. In addition, boundary integral method (BIM) is introduced as a powerful tool to approximate the velocity at the interface of these objects. The privilege of using BIM is that in this method, only the discretization of the surface is required, as the 3D volume problem is recasted in a 2D surface description. In the next chapters, a traditional discretization of the surface compared to a new discretization of the interface using isogeometric analysis is presented. This new spacial discretization is coupled by BIM to approximate the velocity of the mentioned objects suspended in a far-field imposed flow.

Appendix

2.A Variation of the Stokes theorem

The identity

$$\oint_C \mathbf{F} \times \mathbf{t} \, dl = \int_S ((\nabla \cdot \mathbf{F})\mathbf{n} - (\nabla \mathbf{F})\mathbf{n}) \, dS,$$

which in index notation is written as

$$\oint_C \epsilon_{ijk} F_j t_k \, dl = \int_S \left(\frac{\partial F_j}{\partial x_j} n_i - \frac{\partial F_j}{\partial x_i} n_j \right) \, dS, \quad (2.48)$$

where ϵ_{ijk} is the Levi-Civita symbol, is derived starting from Stokes' theorem

$$\oint_C u_j t_j \, dl = \int_S \epsilon_{jkl} \frac{\partial u_l}{\partial x_k} n_j \, dS.$$

Upon the substitution of $u_j = \epsilon_{jmn} \delta_{mi} F_n = \epsilon_{jin} F_n$ with $i = 1, \dots, 3$ in the above expression, one obtains

$$\oint_C \epsilon_{jin} F_n t_j \, dl = \int_S \epsilon_{jkl} \epsilon_{lin} \frac{\partial F_n}{\partial x_k} n_j \, dS.$$

Using the properties of the Levi-Civita symbol, in particular $\epsilon_{ljk} \epsilon_{lin} = \delta_{ji} \delta_{kn} - \delta_{jn} \delta_{ki}$, the above expression can be rewritten to obtain the identity (2.48).

3.1 Introduction

In the previous chapter, the equations that govern the motion of a deformable object under the influence of a background flow field have been introduced. Using the boundary integral method, the velocity of the interface of a deformable object at any point can be evaluated and upon integration in time the motion of the interface can be fully determined. Unfortunately, it is impossible to analytically evaluate the velocity field for arbitrary configurations. Therefore, the surface needs to be parametrized by a finite number of segments and proper numerical evaluation is required to find the velocity field.

The velocity at any given point on the interface of an iso-viscous deformable object is given by Equation (2.25)

$$\mathbf{u}(\mathbf{x}_0) = \mathbf{u}_\infty(\mathbf{x}_0) - \frac{1}{8\pi\mu_1} \int_{\Gamma} \mathbf{G}(\mathbf{x}, \mathbf{x}_0) \mathbf{f}(\mathbf{x}) d\Gamma(\mathbf{x}),$$

where $\mathbf{f}(\mathbf{x})$ indicates the stress discontinuity across the interface and $\mathbf{u}_\infty(\mathbf{x}_0)$ is the velocity of the matrix background flow, $\mathbf{G}(\mathbf{x}, \mathbf{x}_0)$ is the Green's function and μ_1 is the viscosity of the ambient fluid. Since \mathbf{u} is the time derivative of \mathbf{x} , in principle, Equation (2.25) is an integro-differential equation for the interface position $\mathbf{x}(t)$ and the interface evolution in time is given in Equation (2.26)

$$\mathbf{x}(t) = \mathbf{x}(0) + \int_0^t \mathbf{u}(\tau) d\tau.$$

The surface is parametrized by the finite number of segments, the number of

which is governed by Equation (2.25), which is generally of the form

$$\mathbf{x}^h(t) = \sum_{i=1}^n N_i \mathbf{X}_i(t), \quad (3.1)$$

with N_i a set of (scalar-valued) basis functions defined over the interface Γ and $\mathbf{X}_i(t)$ a set of (vector-valued) time-dependent coefficients.

In the current chapter, two different ways of discretization are introduced. The first one is based on the triangular discretization of the surface, which is here referred to as the traditional discretization. The traditional way to discretize the interface using triangular elements is introduced in Section 3.2. Two well-known methods to compute the mean curvature are presented in Section 3.2.3. In Section 3.2.4, a method to compute the higher-order derivatives of the mean curvature using the linear interpolation of the triangular elements is briefly explained. The inaccuracies of the presented methods in computing the mean curvature and its derivatives are demonstrated in Section 3.2.5. Isogeometric analysis (IGA) is presented in Section 3.3 as an alternative method to overcome the mentioned inaccuracies. In Section 3.3.3, different ways of approximating the velocity field using IGA are discussed. Finally, conclusions are drawn in Section 3.4.

3.2 Triangular mesh representation

To find the velocity at a given point on the interface in three-dimensional space using the boundary integral formulation (2.25), the surface is parametrized by the definition of piece-wise polynomials over a mesh partitioning the interface. This results in a parametrization of the form of Equation (3.1). One of the most common methods to parametrize the interface consists of a triangular mesh [23, 24, 82–85]. To construct a discretized interface using triangular meshes, two well-known possibilities exist: an icosahedron or an octahedron. Both methods follow the same procedure and are discussed below.

3.2.1 Icosahedron-based triangular mesh

An icosahedron can be used to represent the geometry of an object. The procedure to create a mesh of the initial spherical geometry is illustrated in Figure 3.1. Figure 3.1 (a) shows a regular icosahedron with 20 triangular faces and 12 nodes (or vertices). Next, in Figure 3.1 (b), each of the twenty triangles is

divided by l equal segments to partition each triangular face of the icosahedron into l^2 flat sub-triangles. Finally, as shown in Figure 3.1 (c), these sub-elements are projected radially outward onto the spherical surface. The final sphere obtained by this approach has $20l^2$ triangular elements and $10l^2 + 2$ nodes. Each node is connected to 6 other nodes except for 12 extraordinary nodes which are connected to 5 adjacent nodes (the nodes in Figure 3.1 (a)).

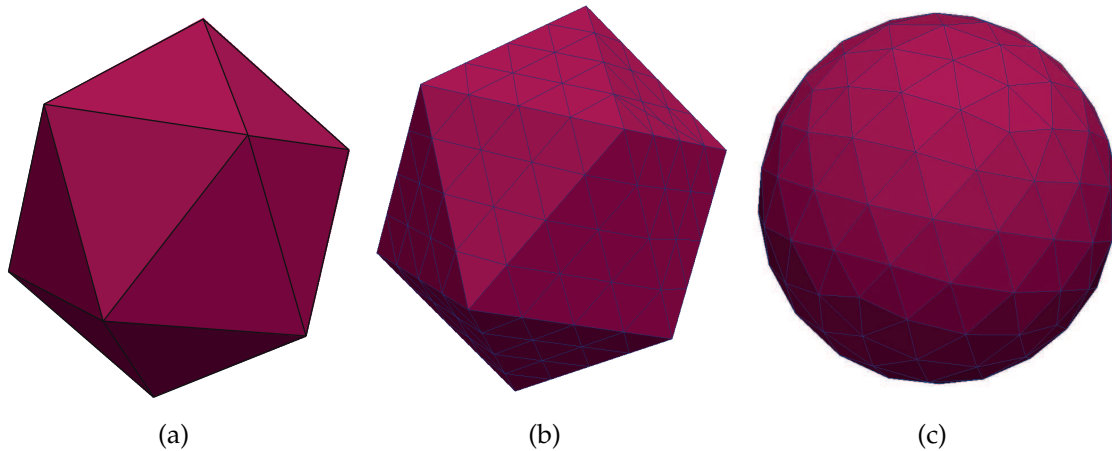


Figure 3.1 Icosahedron-based triangulation of a sphere: (a) An icosahedron with 20 triangular faces. (b) Subdividing each face of the icosahedron into l^2 triangular sub-elements (in this case, $l = 4$). (c) A complete discretized sphere obtained by projecting each element onto the sphere.

3.2.2 Octahedron-based triangular mesh

Another common technique to obtain a triangular discretization of a sphere is based on an octahedron [84, 85]. Figure 3.2 (a) shows an octahedron, which consists of 8 triangular faces and 6 vertices. Figure 3.2 shows the procedure of the construction of a discretized sphere that follows for the icosahedron (see Section 3.2.1). In this case, the final sphere has $8l^2$ triangular elements and $4l^2 + 2$ nodes. Each node is connected to 6 other nodes except for 6 specific nodes, which connected to only 4 adjacent nodes (the nodes in Figure 3.2 (a)).

3.2.3 Curvature evaluation

The mean curvature is always present in the stress discontinuity terms across the interface of the deformable objects (see Section 2.2). Finding the mean curvature in as accurate as possible way is essential for the robust numerical simulation

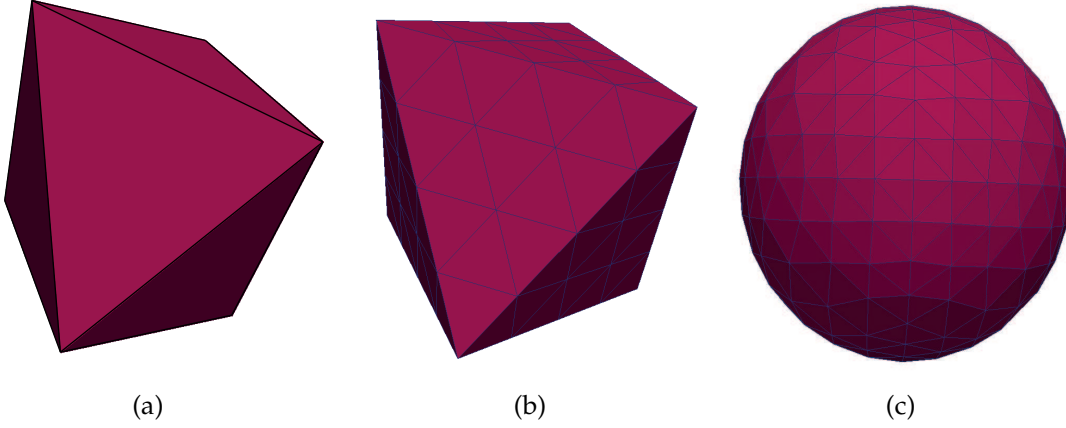


Figure 3.2 Octahedron-based triangulation of a sphere: (a) An octahedron with 8 triangular faces. (b) Subdividing each face of the octahedron into l^2 triangular sub-elements (in this case, $l = 4$) (c) A complete discretized sphere obtained by projecting each element onto the sphere.

of the motion of the interface. Since the mean curvature of the surface is not properly defined in the case of the non-smooth surface triangulations discussed here, indirect evaluation methods are required. The two most commonly used methods are discussed next.

Contour integration method

Consider S to be the surface encircled by the contour C as shown in Figure 3.3, where \mathbf{n} is the outward normal vector to the surface S , \mathbf{b} is the vector orthogonal to C and in S and \mathbf{t} is the tangential vector to the C and in S . The key concept of the contour integration method is using the variation of the Stokes theorem represented in Equation (2.33)

$$\int_C \mathbf{w} \times \mathbf{t} \, dl = \int_S ((\nabla \cdot \mathbf{w})\mathbf{n} - (\nabla \mathbf{w})\mathbf{n}) \, dS,$$

where \mathbf{w} is a differentiable vector-valued field. By assuming $\mathbf{w} = \mathbf{n}$, this theorem becomes

$$\int_C \mathbf{n} \times \mathbf{t} \, dl = \int_S ((\nabla \cdot \mathbf{n})\mathbf{n} - \underbrace{(\nabla \mathbf{n})\mathbf{n}}_{=0}) \, dS, \quad (3.2)$$

where $(\nabla \mathbf{n})\mathbf{n} = (1/2)\nabla(\mathbf{n} \cdot \mathbf{n}) = (1/2)\nabla(1) = \mathbf{0}$. Now, using the identities $-\mathbf{b} = \mathbf{n} \times \mathbf{t}$ and $H = (1/2)\nabla \cdot \mathbf{n}$, Equation (3.2) becomes

$$-\int_C \mathbf{b} \, dl = \int_S 2H\mathbf{n} \, dS. \quad (3.3)$$

From this, the mean curvature can be approximated through [82]

$$2HnS \approx 2 \int_S H\mathbf{n} \, dS = - \int_C \mathbf{b} \, dl, \quad (3.4)$$

which yields:

$$H \approx -\frac{1}{2S} \left| \int_C \mathbf{b} \, dl \right|. \quad (3.5)$$

Two common ways of constructing the contour C on a triangle are:

- The path that passes through the bisectors of the triangle edges and the edge midpoints surrounding any given point [82] (see Figure 3.3).
- The path that passes through the center of the mass of the triangle edges and the edge midpoints surrounding any given point [23, 24].

The most common way to examine the accuracy of the method for the computation of the mean curvature is to check the approximated curvatures with the analytical values taken from a unit sphere (radius equals one). In this case, the mean curvature in every point should be equal to 1. In Figure 3.4, the relative error of the field of the mean curvature is plotted versus the number of nodes, where the error is defined as

$$\epsilon_H = \frac{\|H_{\text{ana}} - H_{\text{num}}\|}{\|H_{\text{ana}}\|} \quad \text{with} \quad = \sum_{i=1}^n |\square_i|, \quad (3.6)$$

where H_{ana} and H_{num} are the exact analytical and the numerically computed values of H . From Figure 3.4, it is observed that this error measure converges linearly with the number of nodes.

Although this graph clearly shows convergence of the mean curvature in the considered global norm, inspection of the computed local curvatures reveals

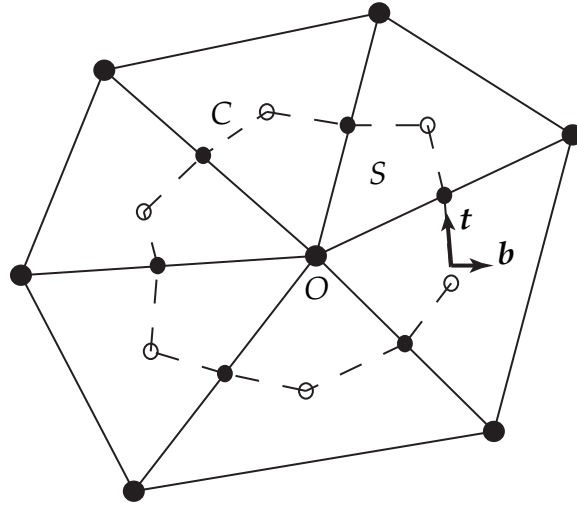


Figure 3.3 Schematic representation of the mean curvature calculation at point O , as presented in Equation (3.4). \mathbf{b} is the outward unit normal to the contour C and S is the surface surrounded by C .

that serious inaccuracies exist at the irregular nodes (12 nodes in the case of the icosahedron). In these nodes, independently of the number of the elements, the computed mean curvature is 1.14 (14% percent error).

The local inaccuracies in the computation of the mean curvature are not necessarily a problem when it comes to the simulation of the motion of a drop. As an example, a sphere with 2000 triangular elements is considered as the initial configuration of a drop. A simulation has been performed without background flow. Thus, \mathbf{u}_∞ in Equation (2.25) is equal to zero. Since the equilibrium shape of the relaxed drop is expected to be a sphere, no obvious deformation is observed. The 11th node, which is one of the irregular nodes, is considered and Figure 3.5 shows the variation of the mean curvature in this point over time. Evidently, the mean curvature at the initial moment has 14% inaccuracy (1.14 instead of 1). However, during the simulation, this inaccuracy gradually disappears and the mean curvature approaches to the accurate value (which is 1), which indicates that the interfacial forces stabilize the values of the mean curvature at these nodes. The same behaviour can be observed for the other irregular nodes and also in other types of the background flows [23, 24]. In fact, the force terms at the drop interface correct the computed mean curvature as computed by the contour integration method and have a stabilizing effect on the simulation.

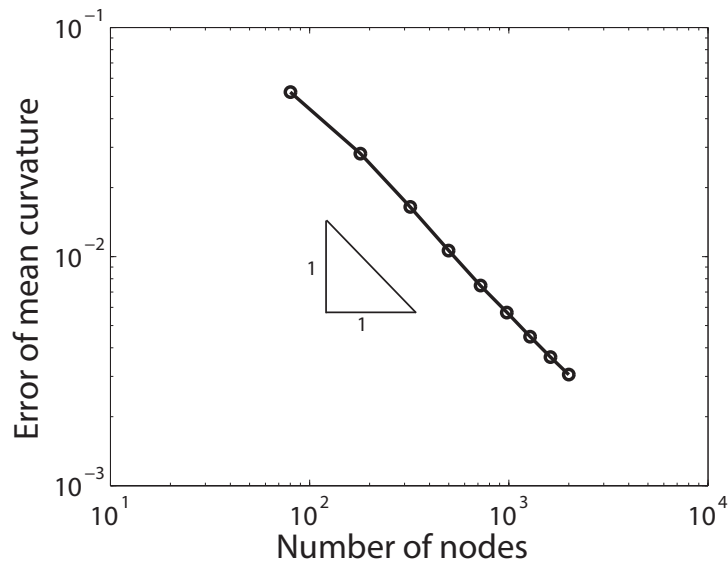


Figure 3.4 The relative error of the mean curvature, computed by the contour integration method for a triangulated sphere based on an icosahedron.

Paraboloid fitting method

The idea of approximating the surface by a paraboloid in each node is presented in Figure 3.6. The description of the procedure to approximate the best fitted paraboloid and find the mean curvature is introduced in Algorithm 1.

Algorithm 1 Curvature finding procedure using the paraboloid fitting method

Initialization

$$\mathbf{n}_O = \mathbf{n}_{O_{\text{init}}}$$

▷ $\mathbf{n}_{O_{\text{init}}}$: Normal vector on O

Approximate the function

repeat

Choose (x', y', z') with $O = (0, 0, 0)$

Approximate z' by a quadratic function

▷ See Eq. (3.7)

Find the coefficients A, B, C, D and E

▷ See Eq. (3.8)

Compute $\mathbf{n}_{\text{paraboloid}}$

▷ See Eq. (3.9)

until $\|\mathbf{n}_{\text{paraboloid}} - \mathbf{n}_O\| < \epsilon$

Compute the mean curvature

▷ See Eq. (3.11)

A local Cartesian coordinate system (x', y', z') can be defined in every node. By aligning z' with the outward normal of the point O at the surface, z' can be approximated by a quadratic function of x' and y' . The normal vector outward

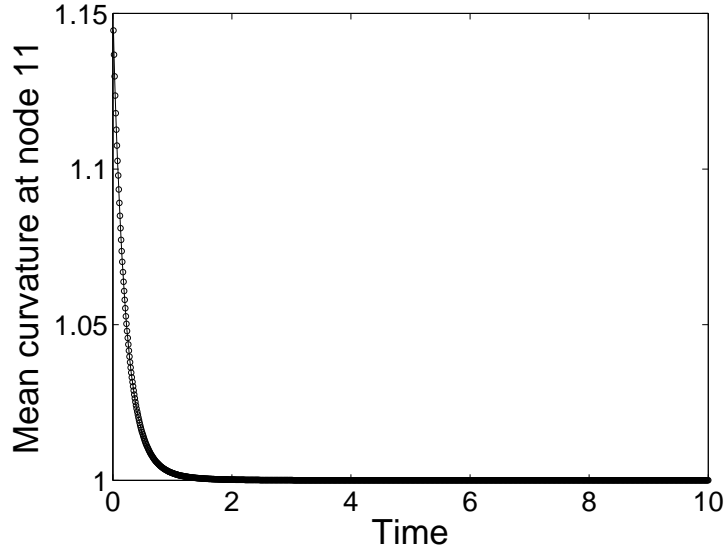


Figure 3.5 Variation of the values of the mean curvature over non-dimensional time for the 11th node, which is one of the irregular nodes with five elements connectivity. The simulation is performed with an initial spheroidal configuration, while there is no background flow.

to one of the elements can be chosen as $\mathbf{n}_{O\text{init}}$. To have the best fitted paraboloid (since z' is not aligned perfectly with \mathbf{n}_O at the first selection), an iteration loop is needed [83].

In the first step of this iterative procedure, consider

$$z' = Ax' + By' + Cx'^2 + Dx'y' + Ey'^2. \quad (3.7)$$

The coefficients of the paraboloid are then found by the minimization of

$$F = \sum_{i=1}^{N_O} \frac{(Ax'_i + By'_i + Cx_i'^2 + Dx'_iy'_i + Ey_i'^2 - z_i'^2)^2}{x_i'^2 + y_i'^2 + z_i'^2}, \quad (3.8)$$

where N_O is the number of nodes in the vicinity which for regular nodes equals $N_O = 6$ while for irregular nodes $N_O = 5$ (in an icosahedron-based discretization) or $N_O = 4$ (in an octahedron-based discretization). This minimization can be done by setting $\frac{\partial F}{\partial A} = \dots = \frac{\partial F}{\partial E} = 0$. The weights $(x_i'^2 + y_i'^2 + z_i'^2)$ are used to intensify the influence of the adjacent nodes [83]. An update normal vector can

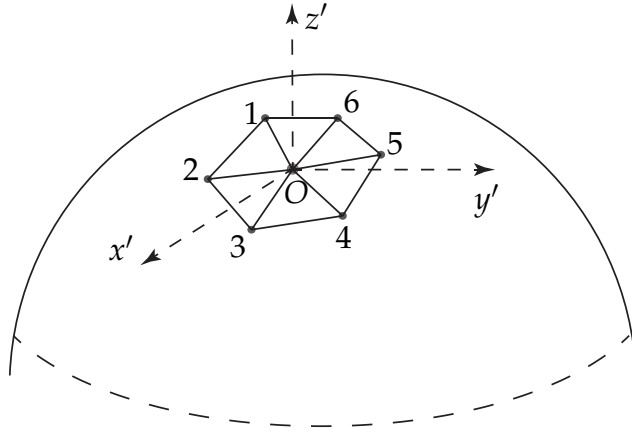


Figure 3.6 Schematic representation of paraboloid fitting for the calculation of the mean curvature.

now be computed by $\mathbf{n} = \frac{\nabla z'}{\|\nabla z'\|}$, which substantiates to

$$\mathbf{n}_{\text{paraboloid}} = \frac{1}{\sqrt{1 + A^2 + B^2}} (-A, -B, 1). \quad (3.9)$$

The above steps should be repeated until $\|\mathbf{n}_{\text{paraboloid}} - \mathbf{n}_O\| < \epsilon$. At the end, the best fitted paraboloid is found and according to the definition, $H = \frac{1}{2} \nabla \cdot \mathbf{n}$, the mean curvature is given by

$$H = -\frac{C + E}{\sqrt{A^2 + B^2 + 1}} + \frac{A^2C + ABD + B^2E}{\sqrt{(A^2 + B^2 + 1)^3}}. \quad (3.10)$$

Zinchenko *et al.* [83] presented an approximation for the computed mean curvature using the paraboloid fitting method as

$$H = -(C + E), \quad (3.11)$$

which is used in this work. In addition, note that in the current case, the typical number of the connected nodes in the neighborhood, N_O , is 6, which is more than the unknowns A , B , C , D and E (see Section 3.2.1), which is of course not a problem since the L^2 minimization is involved. It should be mentioned that in the irregular nodes which are connected to only 5 elements, the best paraboloid could be fitted. More attention should be taken in the case of using an octahedron to build the discretized mesh, since the irregular nodes are connected to only 4 adjacent nodes which is not sufficient to complete the analytical approximation for z' . In this case, one of the closest nodes (or any interpolated point in the

vicinity) could be used to build up the analytical approximation function for a paraboloid.

In Figure 3.7, the relative error in the mean curvature of an icosahedron-based triangulated sphere as computed by the paraboloid fitting method is shown. As for the contour integration method, this error quantity converges linearly with the number of nodes.

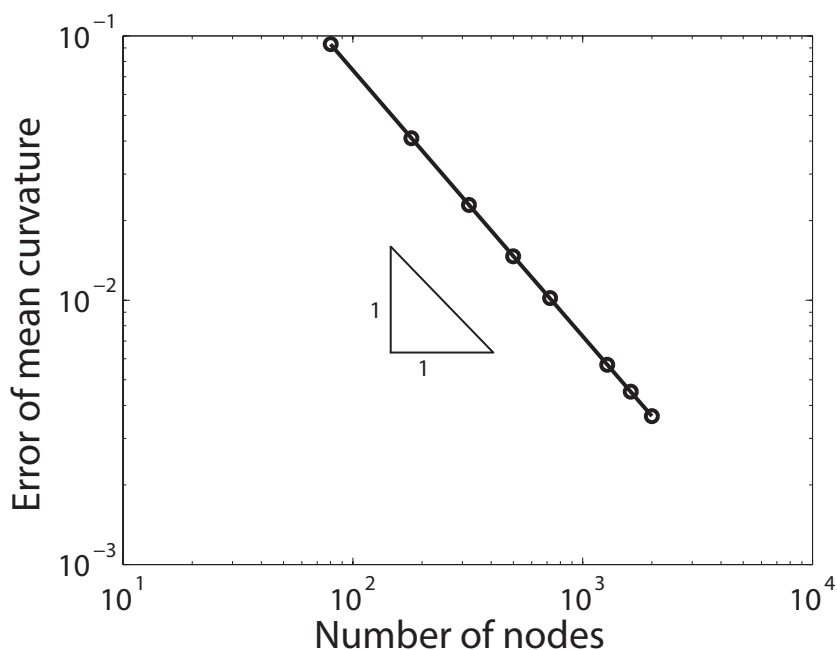


Figure 3.7 The relative error of the mean curvature, computed by the paraboloid fitting method for a triangulated sphere based on an icosahedron.

The mean curvature of the irregular nodes in this method, unlike the contour integration method, have shown no difference compared to the other regular nodes and the inaccuracy reported in Figure 3.5 no longer exist. During the time, it remains accurate when a simulation runs for a drop with no background flow (as is the case for the simulation in Figure 3.5).

3.2.4 Gradient of the mean curvature

Equation (2.46) shows the stress terms across a vesicle membrane, one of which includes the surface Laplacian of the mean curvature. To compute such higher-order gradients, the paraboloid fitting method as outlined above for the computation of the mean curvature must be generalized.

Figure 3.8 displays a regular node O , surrounded by 6 other triangular elements ($N_c = 6$). To take the gradient of the mean curvature at this node, a function which can be used to approximate the mean curvature is required. A linear approximation of the mean curvature for interpolation is assumed as

$$H = \sum_{i=1}^3 N_i(x, y, z) H_i, \quad (3.12)$$

where the shape functions are

$$\begin{aligned} N_1(x, y, z) &= \frac{1}{K} [(z_3 y_2 - z_2 y_3)x + (x_3 z_2 - x_2 z_3)y + (x_2 y_3 - x_3 y_2)z], \\ N_2(x, y, z) &= \frac{1}{K} [(x_3 z_1 - x_1 z_3)x + (x_1 z_3 - x_3 z_1)y + (x_3 y_1 - x_1 y_3)z], \\ N_3(x, y, z) &= \frac{1}{K} [(y_1 z_2 - y_2 z_1)x + (x_2 z_1 - x_1 z_2)y + (y_2 x_1 - y_1 x_2)z], \end{aligned} \quad (3.13)$$

whit K a constant given by

$$K = (y_2 z_3 - y_3 z_2)x_1 + (y_3 z_1 - y_1 z_3)x_2 + (y_1 z_2 - y_2 z_1)x_3. \quad (3.14)$$

Now, an analytical approximation function of the mean curvature for each triangular element is obtained. For the selected point O in Figure 3.8, the above procedure can be applied to all connected triangles. Taking the derivative of the analytical approximation function of the mean curvature yields the gradient of the mean curvature in each element. Eventually, taking the average (either arithmetic or weighted by the areas) of the obtained gradients, gives the approximated values of the gradient of the mean curvature (∇H) in each node.

To find the Laplacian of the mean curvature ΔH , the same procedure can be used and Equation (3.12) becomes

$$H_{,x} = \sum_{i=1}^3 N_i(x, y, z) H_{,x_i}, \quad (3.15)$$

where $H_{i,x} = \frac{\partial H_i}{\partial x}$. The same treatment should be done to find $H_{,y_i}$ and $H_{,z_i}$. Now, for all three components of the gradient of the mean curvature, an analytical

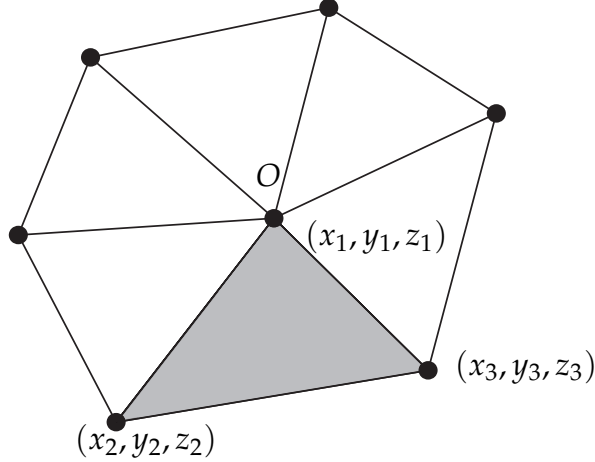


Figure 3.8 Schematic representation of a typical node O in triangular discretization.

relation exists. The same process explained above then yields the Laplacian of the mean curvature ΔH .

3.2.5 Difficulties of the traditional methods to find the mean curvature

In the previous sections, the applied methods to find the mean curvature and the gradients thereof on a triangulated surface have been introduced. These methods have been shown to be applicable for the simulation of drop deformation where the mean curvature is the dominant force term across the interface [23, 24]. However, in a vesicle and a red blood cell membrane, a higher-order gradient of the mean curvature exists. Using the linear interpolation of the mean curvature at each node (see Section 3.2.4), the Laplacian of the mean curvature is obtained. The computation of these higher-order terms using the methods elaborated in Sections 3.2.3 and 3.2.4 causes serious difficulties.

To check the accuracy of the higher order gradients of the mean curvature, an ellipsoid is considered as an initial configuration of a vesicle since the mean curvature of an ellipsoid can be evaluated analytically. An ellipsoid with the centroid at the origin is given by

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1, \quad (3.16)$$

from which the mean curvature follows as

$$H = \frac{1}{8} \frac{abc \left(3(a^2 + b^2) + 2c^2 + (a^2 + b^2 - 2c^2) \left(\frac{2z^2}{c^2} - 1 \right) - 2(a^2 - b^2) \left(\frac{x^2}{a^2} - \frac{y^2}{b^2} \right) \right)}{\left(\frac{a^2 b^2}{c^2} z^2 + c^2 \left(\frac{b^2}{a^2} x^2 + \frac{a^2}{b^2} y^2 \right) \right)^{\frac{3}{2}}}, \quad (3.17)$$

where a , b and c are the lengths of the axes of the ellipsoid. A comparison between the analytically and the numerically computed mean curvature, the surface gradient and the surface Laplacian of the mean curvature is discussed below.

Difficulties of the contour integration method

In Table 3.1, a comparison between the analytically and the numerically calculated mean curvature, the surface gradient and the surface Laplacian of the curvature is reported for three regular nodes on an ellipsoid with $a = 1.35$, $b = 1/\sqrt{1.35}$ and $c = 1/\sqrt{1.35}$. Obviously, the mean curvature is sufficiently accurate (the error is always less than 1%). But in the surface gradient of the mean curvature, the error is significant. For the surface Laplacian of the mean curvature, the errors are even bigger. In Table 3.2, the same comparison is made for three irregular nodes where $N_c = 5$. It is clear that in this case, neither the mean curvature nor the surface gradient and the Laplacian of the mean curvature are accurate. It is found that the application of the contour integration method in the cases of a red blood cell or a vesicle deformation simulation leads to drastic instabilities, which render the method to be useless.

Difficulties of the paraboloid fitting method

In Table 3.3, a comparison between the analytically and the numerically calculated mean curvature, the surface gradient and the surface Laplacian of the mean curvature is reported for the same three nodes as used in Table 3.1. It is observed that the calculated mean curvature by the paraboloid fitting method is slightly less accurate than those calculated by the contour integration method, however, the accuracy is still reasonable. A comparison between the values of the surface gradient and the surface Laplacian of the mean curvature with the data in Table 3.1, shows fairly better performance of the paraboloid fitting method in finding the higher-order derivatives of the curvature. Although the numerically

Table 3.1 A comparison between the analytically and the numerically calculated mean curvature, the surface gradient and the surface Laplacian of the mean curvature in three arbitrarily selected regular nodes. All results are obtained using the contour integration method.

N_{cp}	H_{ana}	H_{num}	$\nabla_s H_{ana}$	$\nabla_s H_{num}$	$\Delta_s H_{ana}$	$\Delta_s H_{num}$
250	2.1411	2.1432	-0.13349	-1.1704	1.5407	-0.3788
			-0.4674	-0.4583		
			0.4964	0.5444		
600	1.7849	1.7817	0.5492	0.5446	1.1387	1.1384
			-0.1214	-7.72×10^{-2}		
			-0.1069	-0.1634		
900	1.9556	1.9557	0.8614	0.8725	1.4179	1.3569
			0.2002	0.2278		
			-0.3335	-0.3262		

Table 3.2 A comparison between the analytically and the numerically calculated mean curvature, the surface gradient and the surface Laplacian of the mean curvature in three irregular arbitrarily selected nodes with $N_c = 5$. All results are obtained using the contour integration method.

N_{cp}	H_{ana}	H_{num}	$\nabla_s H_{ana}$	$\nabla_s H_{num}$	$\Delta_s H_{ana}$	$\Delta_s H_{num}$
11	2.8442	3.2487	1.5189	1.7977	-2.6657	-6.5885
			6.14×10^{-15}	-1.08×10^{-7}		
			1.9367	1.5767		
66	1.6954	1.9420	0.3335	0.3536	0.9633	-4.1150
			0.0541	-8.23×10^{-2}		
			0.0284	-4.34×10^{-2}		
131	2.2249	2.5457	1.2366	1.3479	1.4959	-5.2850
			-0.6295	-0.4428		
			-0.5355	-0.3773		

computed surface Laplacian of the mean curvature is not very accurate, they are still appear to be acceptable.

The result for a vesicle deformation simulation, including all terms of the bending force across the membrane with an ellipsoidal initial configuration in shear flow is shown in Figure 3.9. Slightly after starting the simulation, some points begin to badly deform and finally an unexpected distortion occurs. This contrasts the observations reported in Table 3.3, which shows an acceptable accuracy of the gradient of the mean curvature for all regular nodes. By closer inspection, it is found that the wrinkles start from the irregular points and these wrinkles become

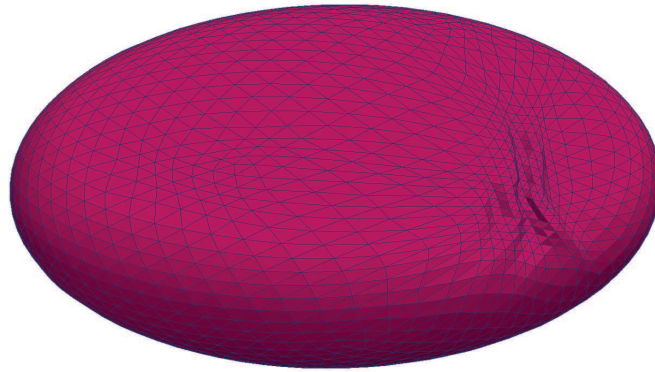


Figure 3.9 A snapshot of a simulation break down for a vesicle with an initial ellipsoidal configuration. All stress jump terms are calculated using the paraboloid fitting method.

bigger until the simulation breaks down. By looking at Table 3.4 and comparing the analytical with the numerically computed surface gradient and the surface Laplacian of the mean curvature, the reason of the unexpected distortion shown in Figure 3.9 is discovered. The Laplacian of the mean curvature at these nodes is still inaccurate and results small wrinkles, which are intensified in the time and subsequently the simulation cause to crash. Despite the fact that the paraboloid fitting method simplifies the vesicle bending force term, still it does not give the satisfactory results.

Table 3.3 A comparison between the analytically and the numerically calculated mean curvature, the surface gradient and the surface Laplacian of the mean curvature in three regular arbitrarily selected nodes. All the values are based on the paraboloid fitting.

N_{cp}	H_{ana}	H_{num}	$\nabla_s H_{ana}$	$\nabla_s H_{num}$	$\Delta_s H_{ana}$	$\Delta_s H_{num}$
250	2.1411	2.1410	-0.13349	-1.1538	1.5407	1.6194
			-0.4674	-0.4953		
			0.4964	0.5029		
600	1.7849	1.7908	0.5492	0.5461	1.1387	1.1306
			-0.1214	-0.1217		
			-0.1069	-0.1090		
900	1.9556	1.9626	0.8614	0.8635	1.4179	1.3163
			0.2002	0.2138		
			-0.3335	-0.3310		

The irregular nodes (with five element connectivity) effectively cause serious

Table 3.4 A comparison between the analytically and the numerically calculated mean curvature, the surface gradient and the surface Laplacian of the mean curvature in three irregular arbitrarily selected nodes with $N_c = 5$. All the values are based on the paraboloid fitting.

N_{cp}	H_{ana}	H_{num}	$\nabla_s H_{ana}$	$\nabla_s H_{num}$	$\Delta_s H_{ana}$	$\Delta_s H_{num}$
11	2.8442	2.8518	1.5189	1.5106	-2.6657	-0.6291
			6.14×10^{-15}	4.62×10^{-2}		
			1.9367	1.9250		
66	1.6954	1.6993	0.3335	0.3452	0.9633	1.0079
			0.0541	5.69×10^{-2}		
			0.0284	3.25×10^{-2}		
131	2.2249	2.2307	1.2366	1.2338	1.4959	1.3077
			-0.6295	-0.6324		
			-0.5355	-0.5332		

difficulties in the modeling of the vesicle deformation and lead to instabilities. These difficulties motivates the usage of an alternative discretization to overcome these problems. Therefore, in the next section, isogeometric analysis (IGA) is presented as a new discretization technique to create the interface using higher-order B-splines.

3.3 Isogeometric analysis

In the previous sections, triangulation-based surface discretization methods and the ways to compute the mean curvature have been introduced. To overcome the difficulties associated with the computational inaccuracies in the curvatures and derivatives thereof, an alternative method is proposed here, *viz.* Isogeometric Analysis. In combination with the boundary integral formulation, this method is referred to as the Isogeometric analysis boundary integral method (IGA-BIM). Recently, IGA has been successfully applied for the discretization of the boundary element and boundary integral method. Politis *et al.* have used IGA to solve potential-flow problems [29]. They have shown the improvement rate of convergence IGA-BIM compared to classical BIM. Takahashi and Matsumoto used IGA to solve the Laplace equation in 2D [30] and Simpson *et al.* used this method for elastostatic analysis [31]. Also, in 3D, Heltai *et al.* have applied nonsingular IGA for Stokes flows and have shown the better results with regard to accuracy [32]. Moreover, Lin *et al.* have presented the boundary integral-based IGA approach to shape optimization [86].

As discussed in the previous sections, in traditional boundary integral formulations, the surface is parametrized by the definition of piece-wise polynomials over an element discretizing the surface presented in (3.1). These basis functions are continuous (but not smooth) over the element boundaries and are interpolatory at the nodes. Consequently, the nodes serve as the coefficients in the discretization (3.1). The surface velocity vector \mathbf{u} , given by Equation (2.25) will be discretized as

$$\mathbf{u}^h(\boldsymbol{\zeta}, t) = \sum_{i=1}^n N_i(\boldsymbol{\zeta}) \mathbf{U}_i(t), \quad (3.18)$$

with $N_i(\boldsymbol{\zeta})$ a set of (scalar-valued) basis functions defined over a parameter domain $\Gamma \subset \mathbb{R}^2$ and $\mathbf{U}_i(t)$ the control point velocities. Note that the superscript h is used to indicate that \mathbf{x}^h is a finite dimensional approximation of the surface.

Here, isogeometric analysis [27] of the motion of the deformable objects under the influence of a background flow is performed. The key ingredient of this isogeometric analysis is that a B-spline representation of the interface is employed. In contrast to the traditional BIM, the spline basis functions are generally smooth over the parameter domain and can be used to construct a smooth surface using Equation (3.1). This smooth surface representation facilitates the direct evaluation of the surface normals and the mean curvatures and allows the higher-order surface derivatives to be incorporated in the formulation. These higher orders derivatives are particularly needed when the method is to be applied to *e.g.* a red blood cell or a vesicle. Therefore it is anticipated that Isogeometric analysis can remedy the difficulties with traditional BIM as explained in Section 3.2.5.

In the remainder of this section, the B-spline basis functions, which in combination with the control net define a B-spline parametrization of the interface, and further relations to obtain the normal vector and the mean curvature on this parametrized surface are introduced. The way of creating the initial configuration is presented and a convergence study is performed. Moreover, two approximation methods to determine the velocity of the control points are described.

3.3.1 B-spline surface parametrization

To define the B-spline parametrization of the interface, first, the definition of a univariate B-spline will be introduced. A univariate B-spline is defined by a knot vector $\Xi = [\tilde{\zeta}_1, \tilde{\zeta}_2, \dots, \tilde{\zeta}_{n+p+1}]$ where $\tilde{\zeta}_i \in \mathbb{R}$ is the i^{th} knot, and $i = 1, 2, \dots, n +$

$p + 1$, with n the number of basis functions and p the order ¹ of the B-spline. The knot vector is non-decreasing and is said to be open if the first and last knot values are repeated $p + 1$ times. The B-spline functions are defined recursively, starting with $p = 0$:

$$N_{i,0}(\xi) = \begin{cases} 1 & \xi_i \leq \xi < \xi_{i+1} \\ 0 & \text{otherwise.} \end{cases} \quad (3.19)$$

The higher degree B-spline shape function then follow as [87, 88]:

$$N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi). \quad (3.20)$$

In combination with a control net $\{\mathbf{X}_i\}_{i=1}^n$ these B-spline basis functions define a B-spline curve according to Equation (3.1).

A B-spline surface is constructed as the tensor product of two B-spline curves. In order to construct the bivariate basis functions, the parameter domain $\hat{\Gamma} = [\xi_1, \xi_{k_\xi}] \otimes [\eta_1, \eta_{k_\eta}]$ which is partitioned by the knot vectors Ξ and \mathcal{H} is considered. With $\{N_{i,p_\xi}(\xi)\}_{i=1}^{n_\xi}$ and $\{N_{i,p_\eta}(\eta)\}_{i=1}^{n_\eta}$ the univariate basis functions defined over Ξ and \mathcal{H} , respectively, the bivariate basis functions then follow as

$$N_i(\xi) = N_{i,p_\xi}(\xi) N_{j,p_\eta}(\eta), \quad (3.21)$$

with $i = \iota + (j - 1)n_\xi = 1, \dots, n (= n_\xi n_\eta)$. Note that $\xi = (\xi, \eta)$ and that the subscript indicating the spline order is dropped for notational convenience in the case of the bivariate basis functions. In combination with a control net $\{\mathbf{X}_i\}_{i=1}^n$ the bivariate spline basis functions (3.21) yield a B-spline surface according to Equation (3.1).

From the vantage point of implementation, a local description of the basis functions is beneficial. To localize the basis functions (3.21) to the elements, two data structures are required: An IEN-array and element extraction operators [89].

The IEN-array provides the connectivity information between the elements and basis functions through $A = \text{IEN}(a, e)$ with element index e and global and local basis function indices, $A = 1, \dots, n$ and $a = 1, \dots, n_e$, respectively. For a bivariate basis function $n_e = (p + 1)^2$ where p is the order of the B-splines (equal in both directions).

Table 3.5 displays the IEN array for the case of $m_\xi = 10$ and $m_\eta = 5$ where m_ξ and m_η are the number of elements in ξ and η directions, respectively. Note that

¹The terms order and degree are used synonymously in this work.

this table only shows the connectivity of the first 10 elements. The pole control points (points 1 and 6 in Figure 3.11 (right)) are made coincident and therefore, only a single point is used. Hence, in Table 3.5, $IEN(a, e) = 1$ when e is from $1, \dots, 10$ and a ranges from $1, \dots, (p + 1)$. Furthermore, attention should be paid to the arrangement of $IEN(a, e)$ when a is 9 and 10 where the IEN array establishes the coincidence of the first two and the last two control points in circumferential direction, which is needed to create a closed or periodic spline in this direction.

Table 3.5 The IEN-array which is constructed for a sphere with 50 elements and 52 control points while $p = 2$, $m_\xi = 10$ and $m_\eta = 5$. This IEN-array maps the local basis function a and the element index e to the corresponding global control point A where $IEN(a, e) = A$. This table shows only the connectivity of the first 10 elements.

a	Element number (e)								
	1	2	3	4	5	6	7	8	9
1	1	1	1	2	3	4	12	13	14
2	1	1	1	3	4	5	13	14	15
3	1	1	1	4	5	6	14	15	16
4	1	1	1	5	6	7	15	16	17
5	1	1	1	6	7	8	16	17	18
6	1	1	1	7	8	9	17	18	19
7	1	1	1	8	9	10	18	19	20
8	1	1	1	9	10	11	19	20	21
9	1	1	1	10	11	2	20	21	12
10	1	1	1	11	2	3	21	12	13

In addition, since the B-spline basis functions are not the same for every element, a linear operator is used to enforce the parent element concept. This mapping reads

$$N^e = C^e B, \quad (3.22)$$

with B the Bernstein polynomials and where C^e is referred to as the Bézier extraction operator. For details on the derivation of the Bézier extraction for multivariate splines, see [89, 90]

Since the normal vector and the mean curvature are present in the stress discontinuity terms of all deformable objects, the BIM requires the evaluation of these two parameters (see Section 2.2). Following the surface parametrization (3.1), the normal vector at any point on the discretized surface is defined as

$$\mathbf{n}^h = \frac{\mathbf{g}_1^h \times \mathbf{g}_2^h}{|\mathbf{g}_1^h \times \mathbf{g}_2^h|}, \quad (3.23)$$

with $\mathbf{g}_1^h = \frac{\partial \mathbf{x}^h}{\partial \xi}$ and $\mathbf{g}_2^h = \frac{\partial \mathbf{x}^h}{\partial \eta}$ the covariant basis vectors. Note that the arguments of the covariant basis vectors and normal vector are omitted for notational convenience.

The mean curvature is defined as the surface divergence of the surface normal and can ultimately be expressed as

$$H^h = \frac{1}{2} \frac{B_{11}A_{22} - 2B_{12}A_{12} + B_{22}A_{11}}{\det(\mathbf{A})}, \quad (3.24)$$

with the components of the first fundamental form \mathbf{A} defined as

$$A_{\alpha\beta} = \mathbf{g}_\alpha^h \cdot \mathbf{g}_\beta^h \quad \alpha, \beta = 1, 2, \quad (3.25)$$

and the components of the second fundamental form \mathbf{B} as

$$B_{\alpha\beta} = \frac{\mathbf{g}_\alpha^h \cdot \mathbf{n}_{,\beta}^h + \mathbf{g}_\beta^h \cdot \mathbf{n}_{,\alpha}^h}{2} \quad \alpha, \beta = 1, 2. \quad (3.26)$$

3.3.2 Initial geometry representation

A spherical initial surface, which, in accordance with Equation (3.1) is parametrized by

$$\mathbf{x}^h(\boldsymbol{\xi}, 0) = \sum_{i=1}^n N_i(\boldsymbol{\xi}) \mathbf{X}_i(0). \quad (3.27)$$

The basis functions $N_i(\boldsymbol{\xi})$ are defined over the bivariate parameter domain $\hat{\Gamma}$ as shown in Figure 3.10. In the ξ -direction, the parameter domain is partitioned by the increasing knot vector $\Xi = [-p_\xi h_\xi, \dots, 0, h_\xi, 2h_\xi, \dots, 1, \dots, 1 + p_\xi h_\xi]$, with $h_\xi = 1/m_\xi$ and m_ξ the number of elements in the ξ -direction. The number of basis functions in ξ -direction is equal to $n_\xi = m_\xi + p_\xi$. In the η -direction, the parameter domain is partitioned by the open knot vector $\mathcal{H} = [0, \dots, 0, h_\eta, 2h_\eta, \dots, 1, \dots, 1]$, where the first and last knot values (0 and 1) are repeated $p_\eta + 1$ times, and with $h_\eta = 1/m_\eta$ and m_η the number of elements in the η -direction. The number of basis function in the η -direction is $n_\eta = m_\eta + p_\eta$, and, consequently, the total number of basis functions is equal to $n = n_\xi n_\eta = (m_\xi + p_\xi)(m_\eta + p_\eta)$ and the total number of elements is equal to $m = m_\xi m_\eta$.

The control net in latitudinal direction is illustrated in Figure 3.11 (left). Note that, in order to create a closed spline, the last p_ξ control points should coincide with the first p_ξ control points. The position of the control points is determined

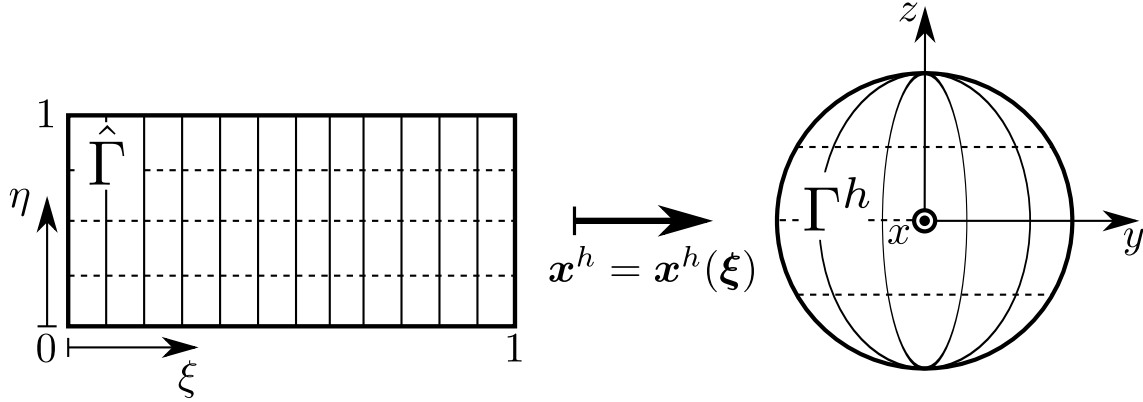


Figure 3.10 The spherical surface Γ is created by mapping the rectangular parameter domain $\hat{\Gamma}$ onto the physical domain.

using an L^2 -projection which optimally (with regard to homogeneity of the radius) approximates an exact circle. Figure 3.11 (right) shows the control net in the longitudinal direction. Note that, in contrast to the B-spline in latitudinal direction, the longitudinal spline is open. The boundaries of this open spline coincide with the poles of the sphere. The initial position of the control points, to be used in combination with the parametric map (3.27), is given by

$$\mathbf{X}_i(0) = \begin{bmatrix} R_j/R & 0 \\ 0 & R_j/R \\ 0 & 0 \end{bmatrix} \mathbf{X}_i^{\xi}(0) + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{X}_j^{\eta}(0), \quad (3.28)$$

with $i = \iota + (j - 1)n_{\xi} = 1, \dots, n (= n_{\xi}n_{\eta})$ and $R_j = [1, 0]\mathbf{X}_j^{\eta}(0)$. To enforce the continuity of the surface at the poles, the corresponding control points are made coincident. It should be mentioned that points 1 and 6 in Figure 3.11 (right) located at the poles of the sphere are special points where the normal vector and the mean curvature computation is numerically impossible. In Section 3.3.3, it is discussed how these points are treated to alleviate the problem.

To gain insight in the accuracy of the isogeometric boundary integral formulation, first, the approximation quality of the spline representation of the deformable object in the initial configuration is discussed here. The normal vector and the mean curvature are the common terms which can be found in the stress terms at the interface of all introduced deformable objects in Section 2.2. Figure 3.12 indicates the mesh convergence behavior of the approximation of the surface area, volume, normal vectors and mean curvatures while the L^2 -error is defined

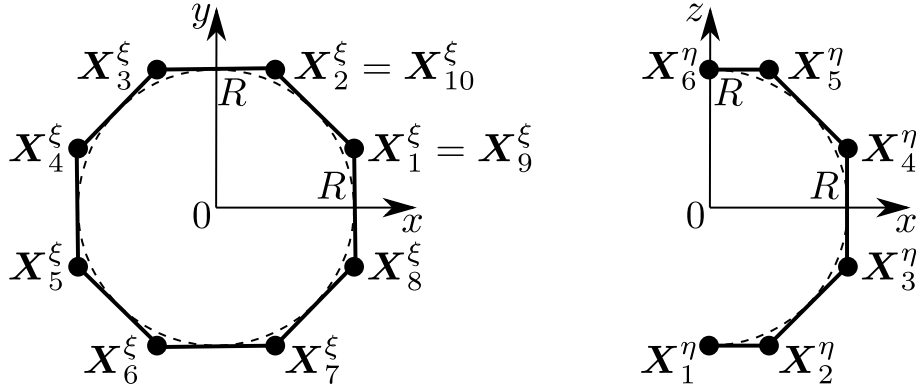


Figure 3.11 Schematic representation of the control net in the latitudinal direction (left) and longitudinal direction (right) for the case that $m_\xi = 8$, $m_\eta = 4$ and $p_\xi = p_\eta = 2$.

as

$$\epsilon_\square = \frac{\|\square_{\text{ana}} - \square^h\|}{\|\square_{\text{ana}}\|} \quad \text{with} \quad \|\square\|_{\hat{\Gamma}} = \sqrt{\int_{\hat{\Gamma}} (\square)^2 d\hat{\Gamma}}. \quad (3.29)$$

In Figures 3.12(a) and 3.12(b), it is observed that the relative errors of the surface area and volume converge asymptotically. In Figures 3.12(c) and 3.12(d), the relative L^2 -errors of the field of normal vectors (3.23) and the mean curvature field (3.24) are plotted versus the number of control points. Both quantities converge asymptotically with the number of control points.

3.3.3 Approximation of the velocity field

In the current analysis, the interface of deformable object is determined by the control net $X_i(t)$ at any moment in time. With the initial surface defined through the initial control net as discussed above, the evolution of the surface is determined by the evolution Equation (2.26). The surface velocity vector \mathbf{u} , which is given by Equation (2.25), is discretized as shown in Equation (3.18). Note that, as for the initial geometry, the velocities of the first and last p_ξ control points in circumferential direction are made coincident to create a closed spline representation in this direction. In contrast to the basis functions used in the traditional BIM, spline basis functions do not possess the Dirac property, and hence the control point velocities cannot be determined by direct point-wise evaluation of (2.25).

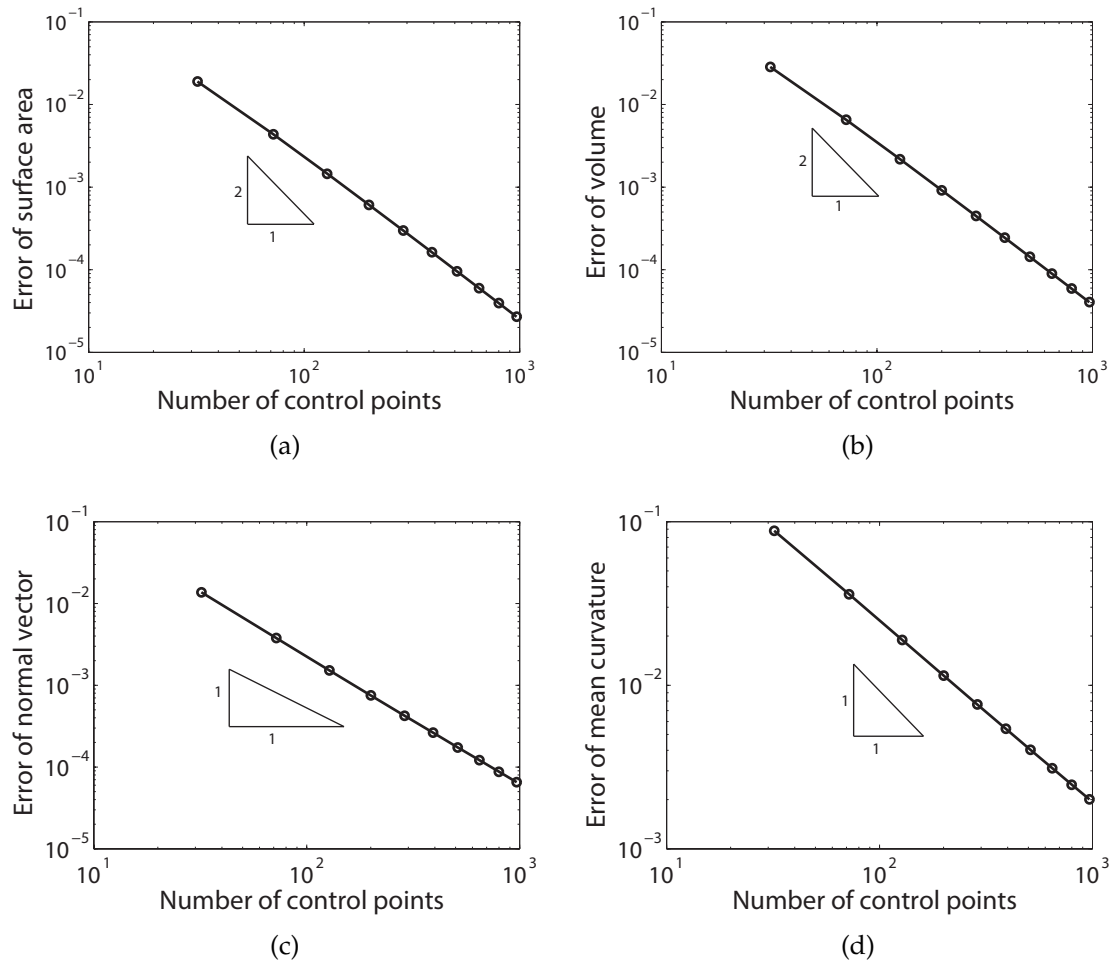


Figure 3.12 The relative error (3.29) of various geometric properties of the spherical initial configuration: a) the surface area; b) the volume; c) the normal vector; d) the mean curvature. All the errors are plotted as a function of the number of control points, n .

Here, the control point velocities are computed by means of an L^2 -projection and a collocation method which are discussed in the following.

L^2 -projection

The control point velocities can be determined by the minimization of the L^2 -norm of the difference between the surface velocity approximation (3.18), \mathbf{u}^h , and the point-wise velocity field computable through Equation (2.25), \mathbf{u} , given by

$$\mathcal{E} = \|\mathbf{u} - \mathbf{u}^h\|_{\hat{\Gamma}}, \quad (3.30)$$

where the L^2 -norm over the parameter domain $\hat{\Gamma}$ is defined in (3.29). To express this minimization problem as a linear system of equations, the discretization (3.18) is reexpressed as

$$\mathbf{u}^h(\boldsymbol{\xi}, t) = \mathbf{N}(\boldsymbol{\xi})\mathbf{a}(t), \quad (3.31)$$

with

$$\mathbf{N} = \begin{bmatrix} N_1 & 0 & 0 & \cdots & N_n & 0 & 0 \\ 0 & N_1 & 0 & \cdots & 0 & N_n & 0 \\ 0 & 0 & N_1 & \cdots & 0 & 0 & N_n \end{bmatrix}, \quad (3.32)$$

and $\mathbf{a}^T = [\mathbf{u}_1^T, \dots, \mathbf{u}_n^T]$. Substitution of Equation (3.31) into Equation (3.30) and performing the minimization results in the linear problem

$$\underbrace{\left[\int_{\hat{\Gamma}} \mathbf{N}^T \mathbf{N} d\hat{\Gamma} \right]}_{\mathbf{A}} \mathbf{a} = \underbrace{\int_{\hat{\Gamma}} \mathbf{N}^T \mathbf{u} d\hat{\Gamma}}_{\mathbf{b}}. \quad (3.33)$$

The evaluation of the integral expressions for the left-hand-side matrix, \mathbf{A} , and right-hand-side vector, \mathbf{b} , is carried out numerically using Gauss quadrature. In all cases, depending on the order of the used B-spline (p), p^2 Gauss points are used. This implies that the convolution integral in Equation (2.25) is to be evaluated in all integration points (at every time step), which results in a nested element loop in the algorithm.

Collocation method

As an alternative to determine the control point velocities, $\mathbf{u}_i(t)$, using an L^2 -projection, the collocation method is employed. Various studies have shown the superior behavior of the isogeometric collocation method over Galerkin method in terms of accuracy-to-computational-time ratio and other aspects [86, 91–93]. The idea of the collocation method is to equate the approximate velocity field (3.18), which is parametrized by $3 \times n$ coefficients, to the velocity field computable through Equation (2.25) in n collocation points, $\{\boldsymbol{\xi}_1^{\text{col}}, \dots, \boldsymbol{\xi}_n^{\text{col}}\}$, which results in the system of $3 \times n$ equations

$$\mathbf{N}(\boldsymbol{\xi}_i^{\text{col}})\mathbf{a} = \mathbf{u}(\boldsymbol{\xi}_i^{\text{col}}) \quad i = 1, \dots, n. \quad (3.34)$$

Like in the case of the L^2 -projection this system can be written in the form

$$Aa = b, \quad (3.35)$$

which can be solved to determine the control point velocities represented by a . Note that the size of this linear system is the same as the L^2 -projection. The number of evaluations of the convolution integral (2.25) is, however, generally different.

Here, the Greville abscissae [94, 95] are used as a starting point for selecting the collocation points. As an example, let us consider the case that a second-order spline ($p_\xi = p_\eta = 2$) with $m_\xi = 8$ elements is available in circumferential direction and $m_\eta = 4$ elements in longitudinal direction. In this case, the knot vectors are given by

$$\Xi = [-2h_\xi, -h_\xi, 0, h_\xi, 2h_\xi, \dots, 1 - h_\xi, 1, 1 + h_\xi, 1 + 2h_\xi]. \quad (3.36)$$

$$\mathcal{H} = [0, 0, 0, h_\eta, 2h_\eta, \dots, 1 - h_\eta, 1, 1, 1]. \quad (3.37)$$

The Greville abscissae in circumferential direction are obtained as

$$\left[-\frac{h_\xi}{2}, \frac{h_\xi}{2}, \frac{3h_\xi}{2}, \dots, \frac{2 - h_\xi}{2}, \frac{2 + h_\xi}{2} \right], \quad (3.38)$$

and those in longitudinal direction as

$$\left[0, \frac{h_\eta}{2}, \frac{3h_\eta}{2}, \dots, \frac{2 - h_\eta}{2}, 1 \right]. \quad (3.39)$$

The Greville abscissae for this case are illustrated in Figure 3.13. Note that for even spline orders the Greville abscissae are face-centered. In the case of odd orders, the abscissae will be vertex centered. Also note that the first and last Greville points in circumferential direction are not inside the parameter domain. In fact, those points are not needed to be used as collocation points since the first two and last two control points in circumferential direction are made coincident in order to preserve the closed spline representation in that direction. Similarly, because the pole control points are made coincident, only a single Greville point is used in circumferential direction as a collocation point at the poles. Because of the fact that the normal vector is not uniquely defined at the poles, a small offset, denoted by ϵ , for those collocation points is employed. In Figure 3.13, the collocation points are also depicted.

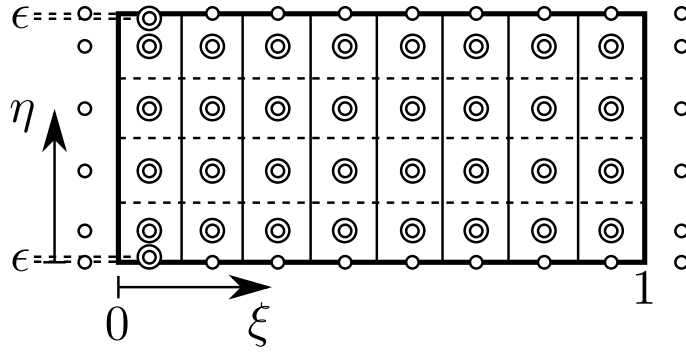


Figure 3.13 Distribution of the Greville abscissae (\circ) and collocation points (\odot) for $m_\xi = 8$, $m_\eta = 4$ and $p_\xi = p_\eta = 2$.

3.4 Conclusions

Since the boundary integral formulation introduced in Chapter 2, only involves the interface between two fluids, the primary concern of the BIM is the discretization of this interface. Traditional BIM, which discretizes the interface using triangular elements surface is found to be not accurate enough in evaluating the mean curvature. This weakness is more pronounced when the higher-order derivatives of the mean curvature are involved in the stress discontinuity at the surface. This is particularly needed when the method is to be applied to *e.g.* a red blood cell or a vesicle.

Isogeometric analysis is introduced to overcome these difficulties by discretizing the interface using a B-spline parametrization. Unlike the traditional BIM, the spline basis functions, used in the isogeometric BIM are generally smooth over the parameter domain and can be used to construct a smooth surface. This advantage of having a smooth surface representation allows for a unique definition of the surface gradients over the complete surface of a deformable object, and therefore facilitates the direct evaluation of the surface normals and the mean curvatures and opens the doors to incorporating higher-order surface derivatives in the formulation.

Isogeometric analysis for drop deformation

4.1 Introduction

The pioneering work on drop behavior was carried out by Taylor [1, 96]. He introduced small deformation theory to express that the mechanism behind drop deformation depends purely on the capillary number and viscosity ratio between the internal and external fluids. Taylor theory is still being used in many studies to find the interfacial tension [97–100] and has been validated experimentally [101, 102]. Small deformation theory is only applicable when the drop remains practically spherical, which is generally the case if the capillary number is much smaller than unity [103]. A few theoretical analyzes limited to small capillary numbers and near-spherical drop shapes are reported in [104, 105]. For larger capillary numbers the drop deviates from the spherical shape and becomes slender, and may eventually break up in multiple drops. Slender-body theories, which can be regarded as extensions of the small deformation theory, have been introduced for such cases [24, 106–109]. Some complex phenomena, such as drop break up and drop-drop interaction, have also been studied both theoretically and experimentally [110].

To study arbitrary deformations of single and multiple drop systems and phenomena such as drop break-up and drop coalescence, numerical analysis is indispensable. In Chapter 2, the principal formulations of the boundary integral method (BIM) have been introduced and discussed. The boundary integral method based on triangulated interfaces has been widely used for the simulation of drops. For instance, excellent agreement between numerical simulations and experimental observations in drop break-up in three-dimensional viscous flows has been reported by Cristini *et al.* [18]. They have also studied the deformation and break-up of drops in isotropic turbulent flow using the boundary integral method (BIM) [111]. The same method has been used to investigate the interaction between deformable drops in Stokes flow with large deformations

[112]. In addition, Janssen and Anderson [24, 113] used the boundary integral method to solve the governing Stokes equations on drops in confined geometries. In this chapter, the isogeometric boundary integral formulation as introduced in Chapter 3 is tested for the case of the drop. Although the traditional BIM has been successfully applied to this case, the use of IGA-BIM is potentially advantageous. The smoothness of the spline basis functions used in IGA allows a unique and unambiguous definition of the surface gradients over the surface of a drop, and consequently, facilitates the straight-forward and efficient evaluation of the normal vectors, mean curvatures, etc.

Following the ideas established in literature, IGA is used to mimic the motion of a drop in shear flow, the deformation of which is characterized by the Taylor deformation

$$D = \frac{L - B}{L + B}, \quad (4.1)$$

where L and B are twice the maximum and minimum distances from the interface to the drop center, respectively. A typical result for the Taylor deformation as a function of time is shown in Figure 4.1. The drop starts deforming and deviating from the initial spherical shape. Then, it elongates as an ellipsoid and orients itself in a fixed direction. Depending on the magnitude of interfacial tension, or the viscosity ratio, the drop might also break up or there might not be a stable drop shape (*e.g.* a tumbling mode).

In the remainder of this chapter, the mathematical formulation of the current model is presented in Section 4.2. Next, in Section 4.3 numerical results are discussed, which contain a verification of the used IGA-BIM method, a spatial discretization study and a comparison between the results obtained by L^2 -projection and the collocation method. Finally, conclusive remarks are presented in Section 4.4.

4.2 Mathematical formulation

As discussed in Section 2.2.1, the mean curvature is the dominant term in the stress discontinuity at the interface of a drop, as expressed in Equation (2.38). In non-dimensional form (for a detailed discussion on the non-dimensionalization procedure, see Appendix 5.A), this force can be rewritten as

$$f(x) = \frac{2}{Ca} H(x) n(x), \quad (4.2)$$

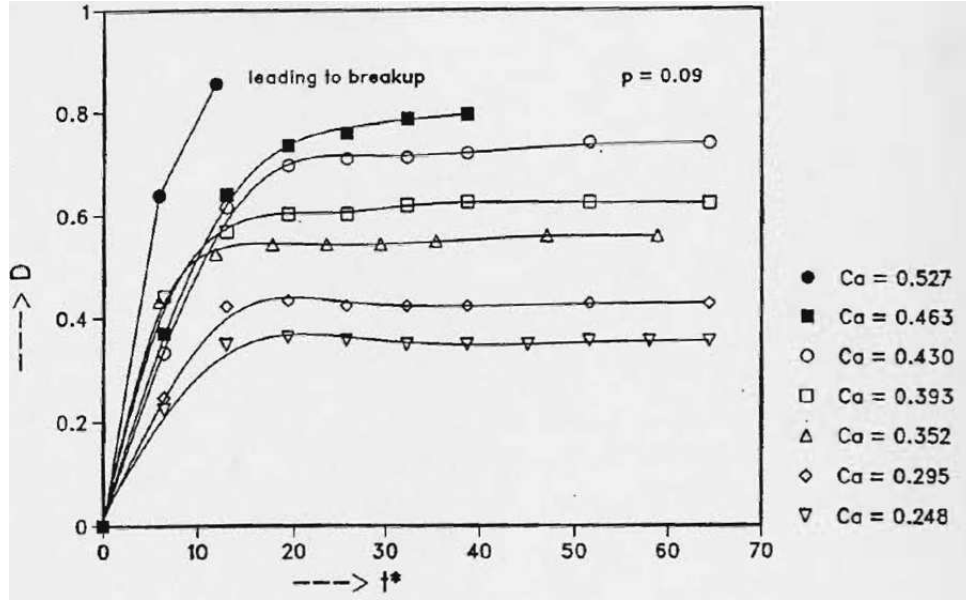


Figure 4.1 A typical experimental result for the Taylor deformation as a function of time for a drop in shear flow. In this particular case, the viscosity ratio (in this figure denoted with p) equals 0.09. The figure is taken from book chapter [114].

where Ca , is capillary number, which is defined as the ratio between viscous and surface tension forces. For a drop with undeformed radius R in shear flow, the capillary number is generally defined as

$$Ca = \frac{R\dot{\gamma}\mu_1}{\sigma}, \quad (4.3)$$

where μ_1 is the matrix fluid viscosity, σ is the interfacial tension between the drop and matrix phases, and $\dot{\gamma}$ is the shear rate which, as shown in Figure 2.1, in simple shear flow is the derivative of the first component of the ambient velocity with respect to the z direction.

With the initial surface defined through the initial control net (see Section 3.3.2), the evolution of the surface is determined by the evolution equation (2.26). Since the interest is in the evolution of the drop shape, and not on the in-plane motion of the interface, in the remainder the tangential components of the surface velocity vector are omitted by using the normal velocity vector

$$\mathbf{u}_n = (\mathbf{u} \cdot \mathbf{n})\mathbf{n}, \quad (4.4)$$

where the full surface velocity vector \mathbf{u} is given by Equation (2.25). This velocity field will be discretized according to Equation (3.18).

With the exception of Section 4.3.1, in the remainder of this section, the numerical analysis of the deformation of an initially spherical drop suspended in a shear flow is considered, which is centered at the origin of the coordinate system with non-dimensional radius equal to one (see Figure 2.1).

Since the forward Euler scheme is conditionally stable, a critical time step size exists. In [83, 115] it is postulated that the critical time step is dictated by the capillary number and spatial discretization according to

$$\Delta t < \mathcal{O}(\min(\Delta x) Ca), \quad (4.5)$$

where Δx is a measure of the minimal element size. Note that in the multivariate case, and particularly in the isogeometric discretization, an unambiguous definition of the element size is missing. Nevertheless, the above expression indicates that a decrease in mesh size will result in a more strict requirement on the allowable time step.

In Table 4.1, the stability of the time integration scheme for various mesh sizes and time step sizes is studied. All simulations in this table have been conducted with $Ca = 0.2$ for a drop in shear flow. The results indicate that indeed, the time step required for a stable time integration decreases as the number of the elements increases.

Table 4.1 Numerical stability of the forward Euler time integration scheme for various mesh sizes (number of control points) and time step sizes. A stable solution is indicated by a checkmark (\checkmark), whereas an unstable solution is marked by a cross (\times).

n	$\Delta t = 0.01$	$\Delta t = 0.005$	$\Delta t = 0.0025$	$\Delta t = 0.0001$
290	\checkmark	\checkmark	\checkmark	\checkmark
394	\times	\checkmark	\checkmark	\checkmark
514	\times	\times	\checkmark	\checkmark
650	\times	\times	\times	\checkmark

4.3 Numerical results

In this section, the numerical results obtained using the isogeometric boundary integral method for the simulation of drop deformation are represented. Initially, a set of verification simulations is performed to test the correctness and accuracy of the IGA boundary integral method. Then, different aspects of drop deformation in shear flow are extensively discussed.

4.3.1 Verification

Two common ways to verify the isogeometric boundary integral method are discussed in this section. The first is drop recovery, for which an initially ellipsoidal drop without background flow is considered. For the simulation, it is expected that the drop will recover to a sphere. The second verification item is a comparison between the current isogeometric BIM and the traditional BIM as reported in previous studies, while the drop is submersed in a shear matrix fluid [24].

Drop recovery

As a first numerical simulation to verify the correctness of isogeometric boundary integral method, the motion of an initially ellipsoidal drop in the absence of a background flow field, $\mathbf{u}_\infty = \mathbf{0}$, is considered. The initial geometry is obtained by applying an affine transformation to the control points of the spherical geometry. This affine transformation is made by stretching the initial control net as follows:

$$\mathbf{X}_i = \underbrace{\begin{pmatrix} s_1 & 0 & 0 \\ 0 & s_2 & 0 \\ 0 & 0 & s_3 \end{pmatrix}}_{\mathbf{S}} \mathbf{P}_i \quad i = 1, \dots, n \quad (4.6)$$

where \mathbf{P} is the control points represented in Section 3.3.2 and \mathbf{S} is the stretching matrix, which in this case is considered with $s_1 = 1.5$ and $s_2 = s_3 = 1$. It is fundamental property of B-splines that an affine transformation of the geometry can be established through the same transformation of the control net.

It is well known that the equilibrium shape of the drop in relaxed condition is a sphere. Figure 4.2 shows the drop deformation in time expressed in terms of the major and minor axes, L and B , for a drop with $\text{Ca} = 0.2$, $\lambda = 1$ and 392 elements (see Section 3.3.2 for a detailed discussion on the definition of the mesh). The initial shape and equilibrium shape are also depicted in the figure. As can be seen, the interfacial force gradually deforms the initially ellipsoidal drop to the spherical equilibrium shape. In this equilibrium state, the major and minor axes are identical. Note that as a consequence of the volume conservation, the stable shape yields $L = B = \sqrt[3]{1.5} \approx 1.145$

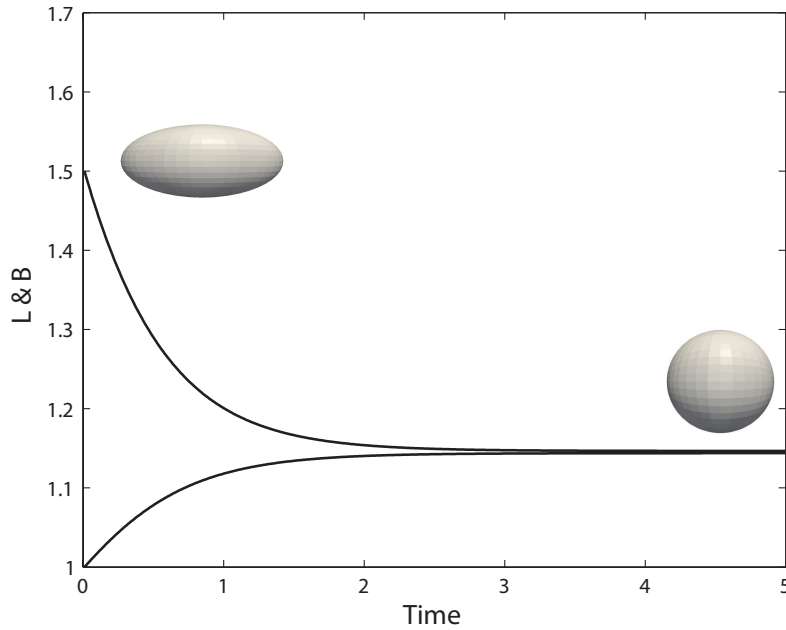


Figure 4.2 Evolution of the major and minor axes, L and B , in time for a drop with an initially spherical shape in the absence of a background flow field for $Ca = 0.2$. Due to the incompressibility condition, the volume of the drop during the deformation, remains constant.

Drop deformation under shear flow

Now, the deformation of an initially spherical drop suspended in a shear flow is considered. In this case, the non-dimensional imposed flow is given by

$$\mathbf{u}_\infty(\mathbf{x}) = \begin{pmatrix} x_3 \\ 0 \\ 0 \end{pmatrix}. \quad (4.7)$$

In Figure 4.3, the drop deformation is shown as a function of time for the case in which the capillary number is equal to 0.2. The drop deformation is measured by means of the Taylor deformation parameter D as defined in Equation (4.1). The presented result is obtained using a mesh with 200 elements. It is observed that initially the Taylor deformation is equal to zero, which corresponds to the spherical initial configuration. The flow gradually elongates and re-orientates the drop. During the deformation of the drop, its centroid remains in the same position, as is expected from symmetry considerations. At approximately $t = 3$, the drop reaches a stationary shape. The obtained stable value of the Taylor

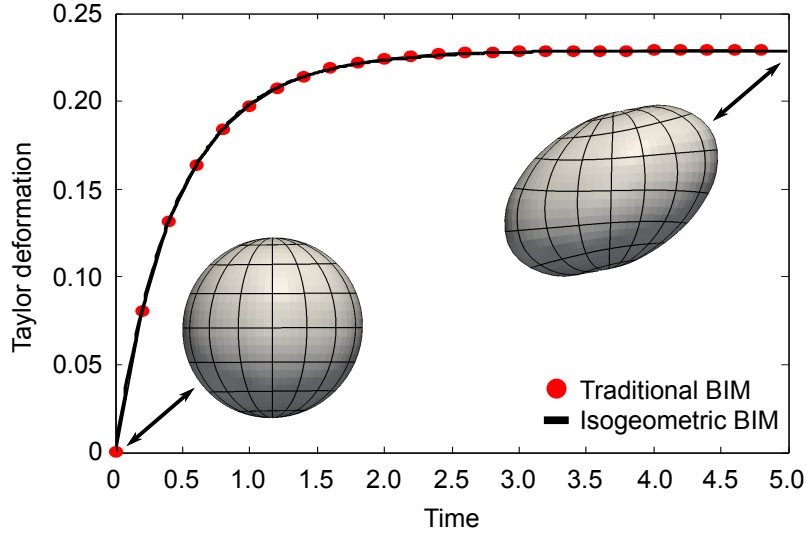


Figure 4.3 Taylor deformation over non-dimensional time for a drop in shear flow for $Ca = 0.2$.

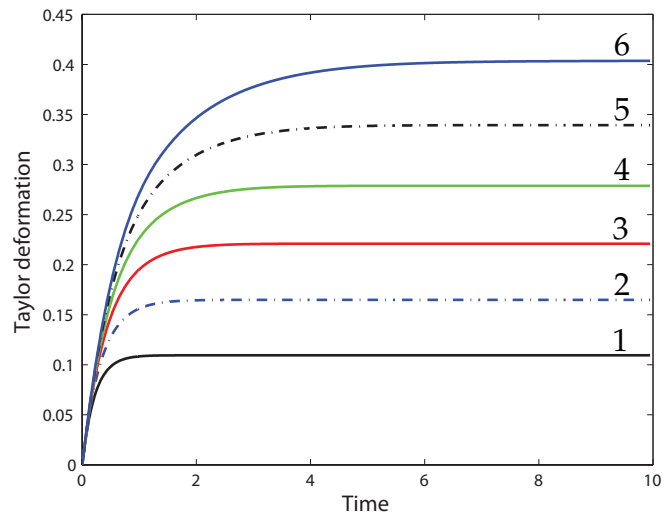
deformation, $D = 0.2235$, closely resembles results reported in literature [24]. For the sake of comparison, in Figure 4.3 also the results obtained using the boundary integral formulations as described in [24] are included. It is observed that the results obtained using the isogeometric approach, closely resemble these results. A more detailed discussion on the performance of the isogeometric approach in terms of computational effort is considered in Section 4.3.3.

In Figure 4.4, the influence of the capillary number on the drop deformation is studied. All results are obtained using 512 quadratic elements (second order B-splines are used, see Section 3.3.1). Figure 4.4(a) shows the Taylor deformation with respect to time for different capillary numbers. The corresponding stable drop shapes are depicted in Figure 4.4(b). With an increase in the capillary number, the viscous forces increase with respect to the interfacial forces. The result is a more severe distortion of the initially spherical drop.

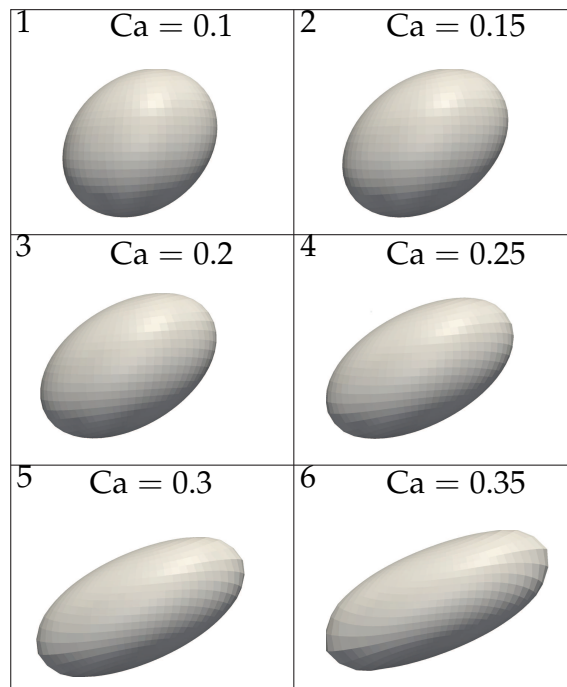
According to [104] the stable Taylor deformation for small capillary numbers is given by

$$D = \frac{5(19\lambda + 16)}{4(\lambda + 1)\sqrt{\left(\frac{20}{Ca}\right)^2 + (19\lambda)^2}}, \quad (4.8)$$

As can be seen from Table 4.2, the isogeometric results closely match the analytically computed values in the case of small capillary numbers, *e.g.* 0.5 and 1.4 percent error for $Ca = 0.1$ and $Ca = 0.15$, respectively. As the capillary



(a)



(b)

Figure 4.4 (a) Taylor deformation in time and (b) stable drop shapes computed with 512 quadratic spline elements for several capillary numbers.

Table 4.2 Comparison of the stable Taylor deformation parameter computed with the isogeometric boundary integral method with 512 quadratic elements and with the analytical expression provided by Equation (4.8).

Ca	Analytical	IGA BIM	Difference%
0.1	0.1089	0.1094	0.459
0.15	0.1624	0.1648	1.477
0.2	0.2149	0.2209	2.792
0.25	0.2660	0.2788	4.812
0.3	0.3156	0.3393	7.5095
0.35	0.3633	0.4036	11.092

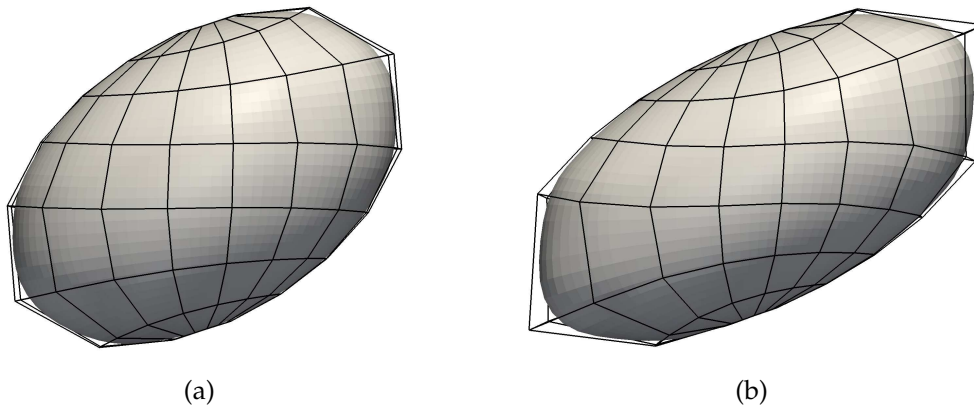


Figure 4.5 Stable drop shape and corresponding control net determined with 130 elements for (a) $Ca = 0.2$ and (b) $Ca = 0.3$.

number increases, the analytical expression (4.8) becomes less accurate, whereas the isogeometric analysis results remain valid. It has to be pointed out, however, that the accuracy of the isogeometric approximating is negatively affected by an increase in capillary number. This effect can be observed from the control net in the stable configuration, as depicted in Figure 4.5 for the cases that $Ca = 0.2$ and $Ca = 0.3$. It is observed that the control net can be coarse at places where the curvature is large, which has a negative effect on the discretization accuracy. Evidently, this effect is intensified when the capillary number (and hence the maximum curvature) is increased. A possible remedy to this effect, is to employ a curvature-based control point redistribution algorithm. Such algorithms have been successfully applied in boundary integral methods [23]. In this thesis, the capillary number is limited to 0.35. In all considered cases, a sufficiently accurate solution is obtained without the use of a control point redistribution algorithm.

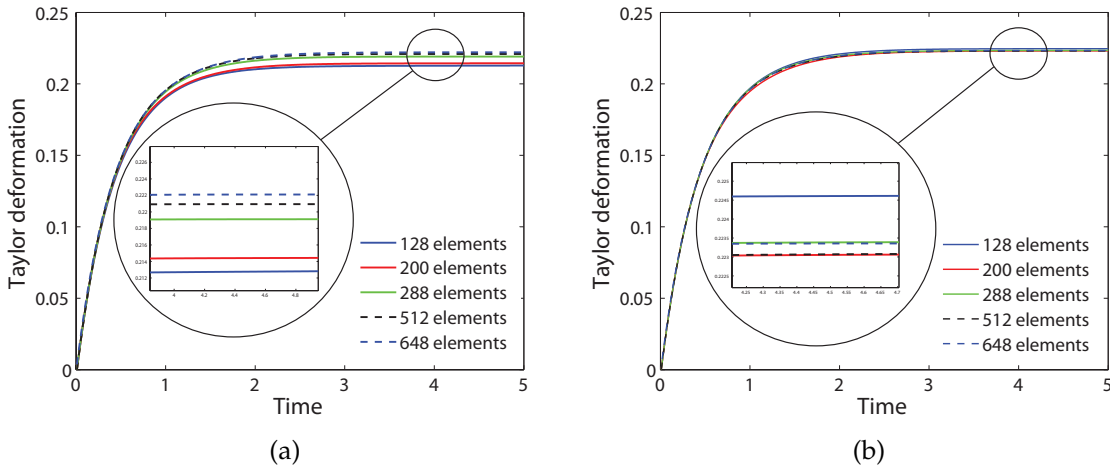


Figure 4.6 Taylor deformation of the drop versus time for various numbers of elements and $Ca = 0.2$. The approximation is based on (a) the collocation method or (b) the L^2 -projection method.

4.3.2 Spatial discretization aspects

In this section, some numerical aspects of the isogeometric spatial discretization are discussed. After consideration of the approximation behavior under mesh refinement, the numerical integration scheme for the evaluation of the convolution integral (2.25) is studied. Also, the effect of the enforcement of the volume constraint by means of the Lagrange multiplier approach is numerically studied.

Mesh convergence

In Figure 4.6, the Taylor deformation as a function of time is compared for various mesh sizes. All results in this figure are obtained using quadratic splines for the case that $Ca = 0.2$. In Table 4.3(left), some mesh characteristics are assembled. Note that the relation between the number of degrees of freedom and the number of elements differs significantly from that in the case of Lagrange elements. For example, the second-order Lagrange element mesh with 200 elements would consist of 762 nodes, which is significantly higher than the 202 control points in the case of the B-spline representation. From Figure 4.6, it is observed that the collocation method converges slower than the L^2 -projection method. A possible reason for this is that the collocation method does not minimize the error in a certain norm; this in contrast to the L^2 -projection. Further comparison between the L^2 -projection and collocation methods is considered in section 4.3.3.

Table 4.3 The relation between the number of control points (n) and number of elements (m) for different orders of splines (p).

m	p	n
128	2	130
200	2	202
288	2	290
512	2	514
648	2	650

m	p	n
200	2	202
200	3	222
200	4	242

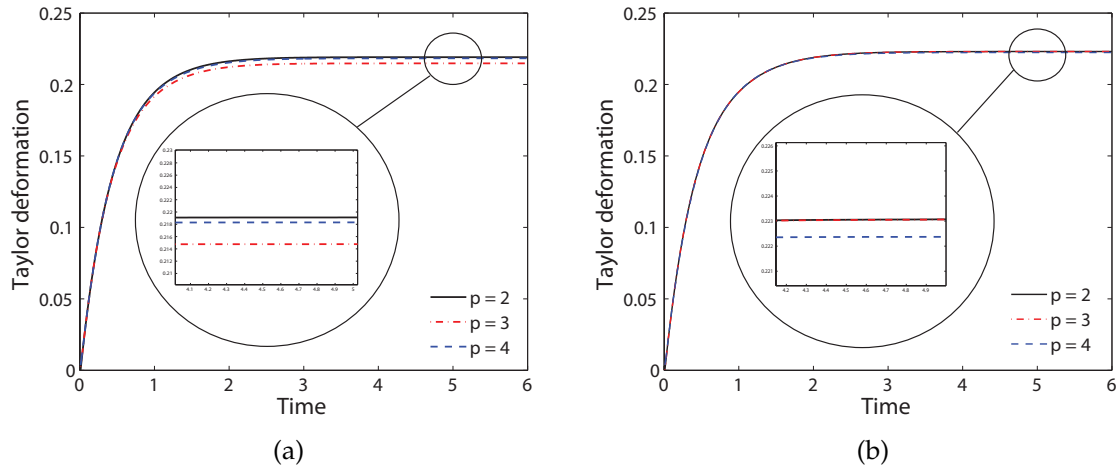


Figure 4.7 Taylor deformation of the drop versus time for various spline orders and $Ca = 0.2$. The approximation is based on (a) the collocation method or (b) the L^2 -projection method.

In Figure 4.7, the response of a drop for various spline orders is studied, while keeping the number of control points constant at 202. In Table 4.3(right), the number of control points for the various meshes is listed. It is noted that the number of control points, and hence the number of degrees of freedom, in the isogeometric analysis setting only marginally increases with an increase in the approximation order; this in contrast to Lagrange elements. It is observed that increasing the spline order only marginally affects the obtained response. However, clear convergence behavior upon increasing the spline order is not observed. The reason for this is that increasing the spline order is, in fact, not a refinement operation in the sense that it (significantly) enriches the approximation space [27].

Numerical integration of the Green's functions

In the current computation of the control point velocities, $\{\mathbf{U}_i\}_{i=1}^n$, the evaluation of the convolution integral (2.25) is required. Since the integrand in (2.25) is singular due to the singular Green's functions, direct evaluation of the integral using Gauss quadrature is inefficient. To alleviate this problem, singularity subtraction [116] is applied. The key idea is illustrated by considering the integral

$$I(\mathbf{x}_0) = \int_{\Gamma} H(\mathbf{x}) \mathbf{G}(\mathbf{x}, \mathbf{x}_0) \mathbf{n}(\mathbf{x}) \, d\Gamma. \quad (4.9)$$

Now, define $\tilde{H}(\mathbf{x}, \mathbf{x}_0) = H(\mathbf{x}) - H(\mathbf{x}_0)$ as the variation of the mean curvature with respect to the curvature at \mathbf{x}_0 , and substitute it in the above integral to obtain

$$I(\mathbf{x}_0) = \int_{\Gamma} \tilde{H}(\mathbf{x}, \mathbf{x}_0) \mathbf{G}(\mathbf{x}, \mathbf{x}_0) \mathbf{n}(\mathbf{x}) \, d\Gamma + H(\mathbf{x}_0) \int_{\Gamma} \mathbf{G}(\mathbf{x}, \mathbf{x}_0) \mathbf{n}(\mathbf{x}) \, d\Gamma. \quad (4.10)$$

Using the identity

$$\int_{\Gamma} \mathbf{G}(\mathbf{x}, \mathbf{x}_0) \mathbf{n}(\mathbf{x}) \, d\Gamma = \mathbf{0}, \quad (4.11)$$

the second term of Equation (4.10) vanishes, and hence

$$I(\mathbf{x}_0) = \int_{\Gamma} \tilde{H}(\mathbf{x}, \mathbf{x}_0) \mathbf{G}(\mathbf{x}, \mathbf{x}_0) \mathbf{n}(\mathbf{x}) \, d\Gamma. \quad (4.12)$$

Compared to integral (4.9), the integrand of (4.12) is non-singular (since \tilde{H} vanishes at the singularity of the Green's function). Consequently, the latter is more suitable for direct numerical integration.

Table 4.4 shows a comparison between the values of the Taylor deformation at $t = 6$ with and without singularity subtraction. The presented values in the first column are obtained when the integrand is singular, *i.e.* equation (4.9). In this case, no clear convergence of the result is observed upon increasing the number of integration points. The second column is produced when the singular points are subtracted, *i.e.* Equation (4.12). In this case, clear convergence of the result is observed upon increasing the number of integration points. Already with only eight integration points in both directions, a result with three digits accuracy is established.

Table 4.4 Numerical integration accuracy with and without singularity subtraction.

Taylor deformation $D(t = 6)$ with 200 elements		
Nr. of integration points	Singular integrand	Non-singular integrand
8^2	0.2198	0.2235
16^2	0.2284	0.2238
32^2	0.2213	0.2238

Volume conservation constraint

It is a fundamental property of the considered model that, due to the incompressibility, the volume of a drop is preserved in time. The discrete approximation of the velocity field however permits volume changes, the magnitude of which depends on the discretization accuracy. In particular, numerical integration errors can lead to significant volume variations. Although such errors can be controlled by adequate integration of the Stokeslet, see Section 4.3.2, it can still be desirable to rigorously enforce the volume conservation property.

The volume constraint can be expressed in terms of the normal velocity field as

$$\int_{\Gamma} \mathbf{u}_n \cdot \mathbf{n} \, d\Gamma = 0, \quad (4.13)$$

where \mathbf{u}_n is defined as in Equation (4.4). Using the discretization (3.31), the discrete form of this constraint is obtained as

$$\underbrace{\int_{\Gamma} \mathbf{n}^T \mathbf{N} \, d\Gamma}_{\mathbf{g}^T} \mathbf{a} = 0. \quad (4.14)$$

Now this constraint can be enforced on the solution obtained using the L^2 -projection, Equation (3.33), by definition of the Lagrange multiplier λ_m and solving the augmented system

$$\begin{bmatrix} \mathbf{A} & \mathbf{g} \\ \mathbf{g}^T & 0 \end{bmatrix} \begin{pmatrix} \mathbf{a} \\ \lambda_m \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ 0 \end{pmatrix}. \quad (4.15)$$

The obtained solution is to be interpreted as the approximation that minimizes (3.30) while satisfying the volume constraint (4.13). Since the collocation method does not follow from a minimization principle, direct application of the constraint by a Lagrange multiplier is not possible. Using the augmented system (4.15) in combination with the original matrix for the the collocation method will

enforce the volume constraint, but the interpretation of the resulting solution is ambiguous.

In Figure 4.8, two different graphs comparing the Taylor deformation in cases with and without volume constraint are presented for a drop with $Ca = 0.2$ discretized using 392 quadratic spline elements. It is observed that in this setting, the volume constraint does not significantly affect the results.

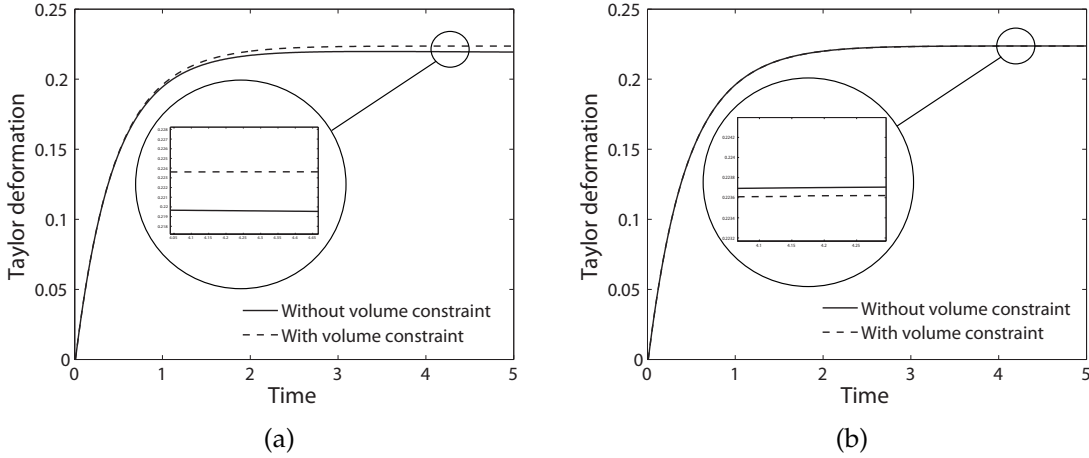


Figure 4.8 Taylor deformation of the drop in shear flow with $Ca = 0.2$ with and without volume constraint with 392 elements using (a) collocation method (b) L^2 -projection method.

4.3.3 Comparison of the L^2 -projection and collocation method

As discussed above, the L^2 -projection method converges faster under mesh refinement than the collocation method. However, it was observed numerically that the collocation method is generally more stable than the L^2 -projection method and allows for the use of larger time steps. This stability issue is illustrated in Figure 4.9. This figure shows the control net simulated by the two methods from the top view for two meshes with a different number of control points. In all simulations, the capillary number is 0.3 and the time step is selected as $\Delta t = 10^{-2}$. From Figure 4.9(a), it is observed that for the case of using the L^2 -projection with 290 control points, an irregular distortion of the control net is encountered upon the deformation of the drop. Ultimately, this behavior will lead to instabilities which result in a diverging solution. Refining the mesh to 514 control points, as shown in Figure 4.9(b), alleviates this problem. However, for larger capillary numbers, similar instabilities are also encountered on this finer

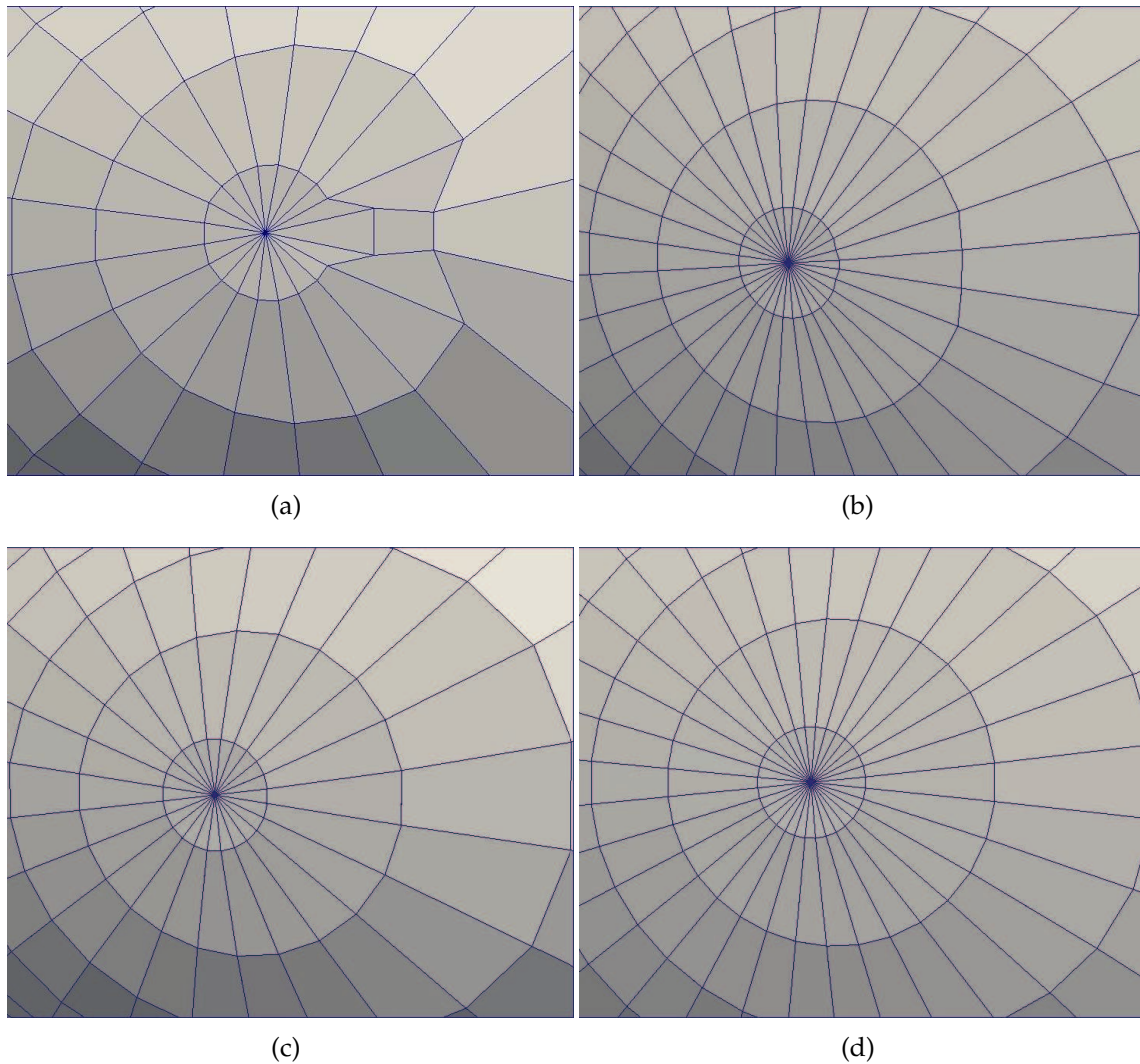


Figure 4.9 Closeup of the top ring of the control net in the stable state using two different schemes: a) L^2 -projection scheme with 288 elements; b) L^2 -projection scheme with 512 elements; c) Collocation with 288 elements; d) Collocation with 288 elements when $Ca = 0.3$.

grid. In Figures 4.9(c) and 4.9(d), the drop evolution is approximated on grids with 290 and 514 control points while making use of the collocation method. The collocation results show that the numerical instabilities encountered using the L^2 -projection have fully disappeared in this case.

In Table 4.5, the computational effort of the collocation method and L^2 -projection method is compared. Various mesh sizes and spline orders are considered. The computational effort is measured by the number of times that the convolution

integral (2.25) is evaluated in every time step and by the time it takes to perform a single time step. To accurately determine the computational time per time step, the average time over ten steps is taken. In contrast to the number of integral evaluations, the computation time is architecture dependent. Table 4.5, however, reveals a direct relation between the number of evaluations and the computation time, which indicates that the number of evaluations is an appropriate measure for the computational effort. For instance, in the case of 394 control points for $p = 2$, a single time step takes 11.08 seconds for the collocation method. Dividing by the number of evaluations yields 0.02812 seconds per evaluation of the convolution integral. For the case with 514 control points for $p = 2$, the computation time per evaluation is 0.03691 seconds. This increase in computation time per evaluation compared to the case with 394 control points is explained by the fact that the finer mesh contains more elements. The computation time for a single evaluation divided by the number of elements is practically constant for the simulations.

From Table 4.5, it is observed that there is a fundamental difference between the collocation method and the L^2 -projection method from the perspective of the number of evaluations of the convolution integral. Whereas in the case of the convolution method, the number of evaluations simply equals the number of the control points, in the case of the L^2 -projection, the number of the evaluations depends on an additional numerical parameter, *viz.* the number of integration points used for the projection. Note that this integration scheme can be selected independent of the integration scheme used for the convolution integral as discussed in Section 4.3.2. Numerical simulations indicate that at least a Gaussian integration scheme of order p is required to accurately determine the projection.

Comparison of the number of evaluations of the convolution integral reveals significant differences between the collocation method and the L^2 -projection. For example, in case of 394 control points and second-order splines, the number of evaluations in the case of the L^2 -projection with an order $p = 2$ integration scheme is approximately 3.5 times larger than in the case of the collocation method. This difference increases significantly if a more accurate integration scheme is considered. Also, it is observed that the ratio between the computational effort of the L^2 -projection and collocation method increases significantly with an increase of the spline order. For example, in the case that 448 fourth order elements and 514 control points are considered with $p + 2$ integration scheme for the L^2 -projection, the collocation method is around 25 times faster than the L^2 -projection.

Table 4.5 A comparison of the computational effort for various simulation settings. Note that there is a direct relation between the number of evaluation of the convolution integral and the CPU time.

m	p	n	Collocation		L^2 projection (number of Gauss points per elements)					
					$k=p^2$		$k=(p+1)^2$		$k=(p+2)^2$	
			Nr. ev.	CPU	Nr. ev.	CPU	Nr. ev.	CPU	Nr. ev.	CPU
392	2	394	394	11.08	1568	38.00	3528	85.35	6272	151.20
364	3	394	394	12.05	3276	87.29	5824	154.72	9100	242.74
336	4	394	394	15.26	5376	160.45	8400	251.76	12096	361.01
512	2	514	514	18.97	2048	65.07	4608	145.54	8192	258.84
480	3	514	514	21.18	4320	154.86	7680	272.33	12000	427.74
448	4	514	514	26.39	7168	285.01	11200	445.59	16128	639.73

4.4 Conclusions

Isogeometric analysis (IGA) was proposed in Chapter 3 as an alternative method to discretize the boundary integral formulation (see Chapter 2) which is capable to dealing with higher order geometric derivatives of an object's interface. As a first example of using IGA-BIM, the velocity field over a drop suspended in an imposed shear flow is computed. IGA permits the usage of an existing singularity subtraction technique, which allows for efficient numerical evaluation of the convolution integrals, which involve singular Green's functions. The motion of the drop is determined by a stepwise computation of the control point velocities using either an L^2 -projection (possibly with enforcement of the volume conservation constraint) or a collocation method.

The results obtained using IGA are in an excellent agreement with results reported in the literature. In addition, the deformation of the drop in time is computed on several meshes. Comparison of the stable drop shapes, obtained using 200 and 512 elements, revealed negligible differences, indicating a reasonable accuracy of the 200 elements solution. The computation of the velocity field over the drop surface using the collocation method was found to be superior in terms of computational effort and stability compared to the computation using the L^2 -projection. The effect of enforcing the volume constraint using a Lagrange multiplier technique was found to be of minor importance.

Successful applying of IGA-BIM on a drop deformation simulation gives confidence to compute the normals and the mean curvatures in the accurate and suitable way. This achievement in the drop deformation simulation opens the doors to incorporating the other deformable objects with more complex stress discontinuity at the interface.

Isogeometric analysis of the inextensible membrane

5.1 Introduction

In the previous chapter, isogeometric analysis was studied in the context of the simulation of drop deformation, and was found to be capable of robustly simulating the motion of drops with relatively coarse meshes. While the simulation of a drop is a perfect first benchmark for testing the capabilities of isogeometric analysis in the context of boundary integral formulations, from a physical point of view it is not directly relevant for the computational modeling of red blood cells. In this chapter, the isogeometric boundary integral formulation as introduced in the previous chapters is extended to the case of a prototypical model of the red blood cell, *viz.* the inextensible membrane.

A red blood cell consists of a nucleus-free capsule which is encircled by a membrane with a composite structure [117]. This membrane is composed of a lipid bilayer and a membrane skeleton. The bilayer is composed of two similar layers of lipid molecules which face away from each other. A well-established property of these layers is their inextensibility, which implies that the surface area of the membrane is locally (and hence also globally) conserved. This conservation of surface area is possible due to the presence of an isotropic tension in the membrane [25]. Although other properties such as *e.g.* viscoelastic effects and bending stiffness have been identified to play a role in the membrane of a red blood cell, the inextensible membrane with isotropic surface tension can be considered as a simplified model of such a cell.

There are numerous computational and analytical studies on the inextensible membrane model of a red blood cell, mostly in shear flow. Kholeif and Weymann [118] conducted a theoretical study on a two-dimensional¹ model of a membrane

¹The term “two-dimensional” here refers to the dimensionality of the physical space. A

with a biconcave shape (which resembles the cross section of an undeformed red blood cell). Depending on the viscosity ratio and the shear rate, the membrane was found to either rotate as a rigid particle or to show tank-treading behavior (*i.e.* the shape-preserving rotation of the membrane around the interior of the cell). The obtained results were found to be in a good agreement with experimental observations reported by Stone [5], Schmid-Schönbein and Wells [119] and Goldsmith [120], confirming that the inextensible membrane is (to some extent) a representative model of a red blood cell. Sugihara and Niimi [121] and Niimi and Sugihara [122] considered an ellipse as the initial shape of a two-dimensional membrane and investigated the flow fields inside and outside of the membrane using the finite element method to discretize the Stokes equations. They computed the tension in the membrane and measured the time until membrane breakup. They also theoretically showed that the membrane undergoes an unsteady cyclic loading as a consequence of its tangential rotation around the interior. Zahalak *et al.* [123] considered a two-dimensional membrane with a circular initial shape and presented a fifth order perturbation series solution of the resulting free boundary-value problem. They studied the tension distribution in the membrane for small and moderate deformations.

The inextensible membrane has also been studied in three dimensions. Keller and Skalak [124] developed an analytical solution by considering a cell with an ellipsoidal shape in a shear flow for which the surface area was conserved only globally. Sutura *et al.* [125] used the same model to estimate the membrane's effective viscous and elastic properties. Zhou and Pozrikidis [25] used the boundary integral method to present both 2D and 3D simulations of an inextensible membrane in shear flow. In 2D, they extended the work by Zahalak *et al.* [123] by the prediction of the membrane behavior for large deformations. They studied the tension distribution in the membrane in the case of a highly elongated initial shape. In addition, they presented a 3D initial ellipsoidal membrane in shear flow and observed a relation between the distribution of the tension and the mean curvature of the surface. They numerically observed that the equilibrium shape of a membrane is independent of the initial inclination angle by simulating three ellipsoids with three different initial orientations.

Though Zhou and Pozrikidis [25] successfully simulated three dimensional inextensible membranes in shear flow for a limited range of initial shapes, they also found that numerical instabilities associated with their boundary integral formulation negatively influence the robustness of the method.

In their simulations, Zhou and Pozrikidis observed a sawtooth instability [126]. In two-dimensional case, this instability manifested itself by minor oscillations in

membrane in such a setting is a closed curve.

the approximated shape of the cell. In three dimensions, however, the numerical results were found to diverge. To eliminate this instability they applied the five-point smoothing formula of Longuet-Higgins and Cokelet [127]. This technique practically eliminated the sawtooth instability problem in two dimensions, but it did not overcome these stability problems in 3D simulations. As an alternative stabilization method they also considered a spectral projection method. To apply this method they expanded the membrane tension in a series of orthonormal basis functions. For the two dimensional case, Fourier series were used and for the three dimensional case either a double Fourier expansion or a Fourier-Legendre expansion was used. In two dimensions, this stabilization method was found to be very effective. In three dimensions, the application of this stabilization strategy permitted the simulation of a limited range of initial shapes.

Based on the results obtained in the previous chapter using the isogeometric boundary integral method, it is anticipated that stable numerical results can be obtained for the inextensible membrane, without the need for artificial stabilization methods as needed for the traditional boundary integral formulation. In this chapter, the formulation is developed for the isogeometric boundary integral modeling of an inextensible membrane. The accuracy and stability of this method are studied using a series of representative numerical simulations.

In Section 5.2, the boundary integral formulation for the inextensible membrane is derived. In Section 5.3, the isogeometric discretization is discussed in the context of the development of a solution algorithm. The developments focus on the enforcement of the inextensibility condition. In Section 5.4, numerical simulations performed with the IGA boundary integral method are presented and discussed, and finally conclusive remarks are presented in Section 5.5.

5.2 Mathematical formulation

The motion of an inextensible membrane suspended in a background Stokes flow is governed by the convolution equation (2.25), which in non-dimensional form reads

$$\mathbf{u}(\mathbf{x}_0) = \mathbf{u}^\infty(\mathbf{x}_0) - \frac{1}{8\pi} \int_{\Gamma} \mathbf{G}(\mathbf{x}_0, \mathbf{x}) \mathbf{f}(\mathbf{x}) d\Gamma. \quad (5.1)$$

Recall that in the derivation of this expression the viscosity of the internal and external fluids was assumed to be equal. As discussed in Section 2.2.2, the stress discontinuity across the inextensible membrane is given by equation (2.39), which

can be written in dimensionless form as

$$\mathbf{f}(\mathbf{x}) = 2\tau(\mathbf{x})H(\mathbf{x})\mathbf{n}(\mathbf{x}) - \nabla_s \tau(\mathbf{x}), \quad (5.2)$$

where $\tau(\mathbf{x})$ is the non-dimensional surface tension, $H(\mathbf{x})$ is the non-dimensional mean curvature, and $\mathbf{n}(\mathbf{x})$ is the unit normal vector to the interface. The reader is referred to Appendix 5.A for a detailed discussion on the non-dimensionalization procedure.

In contrast to the case of the drop as presented in the previous chapter, where the stress jump is explicitly related to the geometry of the interface through equation (4.2), in the case of the inextensible membrane the scalar-valued tension field, $\tau(\mathbf{x})$, is obtained implicitly through the solution of a scalar-valued partial differential equation over the interface. To derive this surface partial differential equation that mimics the inextensibility condition, a parametrized surface $\mathbf{x} = \mathbf{x}(\boldsymbol{\zeta})$ as introduced in Chapter 3 is considered. An infinitesimal surface area element of this surface is represented by

$$dS(\boldsymbol{\zeta}) = |J(\boldsymbol{\zeta})|d\zeta d\eta, \quad (5.3)$$

where the determinant of the metric tensor in this case substantiates to $J(\boldsymbol{\zeta}) = \|\mathbf{g}_3(\boldsymbol{\zeta})\|$, with $\mathbf{g}_3(\boldsymbol{\zeta}) = \mathbf{g}_1(\boldsymbol{\zeta}) \times \mathbf{g}_2(\boldsymbol{\zeta})$. Since the parameter domain, $\hat{\Gamma} \ni \boldsymbol{\zeta}$, does not change over time, and hence conserves its surface area $d\zeta d\eta$ in every point $\boldsymbol{\zeta}$, in order to conserve the surface area $dS(\boldsymbol{\zeta})$ at every point on the surface, it is required that the determinant of the metric tensor, $J(\boldsymbol{\zeta})$, does not change over time:

$$\dot{J}(\boldsymbol{\zeta}) = \frac{d}{dt} \left(J(\boldsymbol{\zeta}) \right) = 0. \quad (5.4)$$

Further elaboration of this equality, which evidently holds for all points on the surface (and hence also for all points on the parametrized domain), then yields

$$\dot{J}(\boldsymbol{\zeta}) = \frac{\mathbf{g}_3(\boldsymbol{\zeta}) \cdot \dot{\mathbf{g}}_3(\boldsymbol{\zeta})}{\|\mathbf{g}_3(\boldsymbol{\zeta})\|} = \mathbf{n}(\boldsymbol{\zeta}) \cdot \dot{\mathbf{g}}_3(\boldsymbol{\zeta}) = 0, \quad (5.5)$$

which upon substitution of the temporal derivative of $\mathbf{g}_3(\boldsymbol{\zeta})$ results in

$$\mathbf{n}(\boldsymbol{\zeta}) \cdot \left(\frac{\partial \mathbf{u}}{\partial \zeta}(\boldsymbol{\zeta}) \times \mathbf{g}_2(\boldsymbol{\zeta}) + \mathbf{g}_1(\boldsymbol{\zeta}) \times \frac{\partial \mathbf{u}}{\partial \eta}(\boldsymbol{\zeta}) \right) = 0, \quad (5.6)$$

where $\mathbf{u}(\boldsymbol{\zeta}) = \dot{\mathbf{x}}(\boldsymbol{\zeta})$ is the vector-valued velocity field defined over the interface. Using the identity $\mathbf{a} \cdot (\mathbf{b} \times \mathbf{c}) = \mathbf{b} \cdot (\mathbf{c} \times \mathbf{a}) = \mathbf{c} \cdot (\mathbf{a} \times \mathbf{b})$, Equation (5.6) can be

rewritten as

$$\mathbf{d}(\boldsymbol{\zeta}) \cdot \frac{\partial \mathbf{u}}{\partial \boldsymbol{\zeta}}(\boldsymbol{\zeta}) - \mathbf{e}(\boldsymbol{\zeta}) \cdot \frac{\partial \mathbf{u}}{\partial \boldsymbol{\eta}}(\boldsymbol{\zeta}) = 0, \quad (5.7)$$

where $\mathbf{d}(\boldsymbol{\zeta}) = \mathbf{g}_2(\boldsymbol{\zeta}) \times \mathbf{n}(\boldsymbol{\zeta})$ and $\mathbf{e}(\boldsymbol{\zeta}) = \mathbf{g}_1(\boldsymbol{\zeta}) \times \mathbf{n}(\boldsymbol{\zeta})$.

Note that the inextensibility condition, Equation (5.7), is a scalar-valued partial differential equation defined over the parameter domain, $\hat{\Gamma} \ni \boldsymbol{\zeta}$. This equation states that for any given interface position, $\mathbf{x}(\boldsymbol{\zeta})$, the interface velocity, $\mathbf{u}(\boldsymbol{\zeta})$, is constrained to the space of surface velocities which do not locally alter the surface area of membrane. When used in combination with the convolution equation (5.1) and the expression for the stress jump over the interface, Equation (5.2), this constraint equation can be solved for the surface tension field $\tau(\mathbf{x})$. In turn, when this surface tension is used in combination with the convolution equation, the motion of the interface can be traced. It is important to note here that no (dimensionless) system parameters are involved in either the computation of the surface tension, or the evaluation of the velocity field through the convolution equation and stress jump definition. This implies that the motion of an inextensible membrane in the non-dimensional setting is solely determined by the background flow field, its initial shape, and possibly its initial velocity (which in this work is always assumed to be equal to zero).

5.3 Solution algorithm

The computational strategy used to determine the motion of the inextensible membrane is presented in Algorithm 2. In essence, this algorithm differs from the algorithm used for the drop deformation only in two ways. First, an additional step is required in each time increment to find the surface tension through the inextensibility condition. Second, the computation of the stress jump over the interface involves a surface gradient term of a discretized scalar-valued field. In the remainder of this section, the focus is on the computation of the surface tension. The reader is referred to Appendix 5.B for a discussion on the determination of the surface gradient of the tension.

5.3.1 Surface tension discretization

The spatial discretization of the interface using isogeometric analysis was discussed in detail in Section 3.3. Given a set of B-spline basis functions, $\{N_i(\boldsymbol{\zeta})\}_{i=1}^n$

Algorithm 2 Forward-Euler time integration procedure for the inextensible membrane

Initialization

$$t = 0$$

$$\mathbf{x}^h = \mathbf{x}_0^h, \dot{\mathbf{x}}^h = \mathbf{0} \quad \triangleright \mathbf{x}_0^h: \text{Initial shape}$$

Time integration

repeat

Computation of $\tau^h(\mathbf{x}, t)$ through inextensibility condition \triangleright See Sec. 5.3.1

Computation of the stress jump $\mathbf{f}(\mathbf{x}, t)$ \triangleright See Eq. 5.2

Computation of the velocity field $\mathbf{u}^h(\mathbf{x}, t)$ \triangleright See Eq. 5.1

Geometry update: $\mathbf{x}^h \leftarrow \mathbf{x}^h + \Delta t \mathbf{u}^h$ $\triangleright \Delta t$: Time step size

Time incrementation: $t \leftarrow t + \Delta t$

until $t \geq t_{\text{end}}$ $\triangleright t_{\text{end}}$: Simulation end time

defined over a parameter domain $\hat{\Gamma}$, the interface is parametrized by

$$\mathbf{x}^h(\boldsymbol{\xi}, t) = \sum_{i=1}^n N_i(\boldsymbol{\xi}) \mathbf{X}_i(t), \quad (5.8)$$

with $\{\mathbf{X}_i(t)\}_{i=1}^n$ the control point positions, and $\{\mathbf{u}_i(t) = \dot{\mathbf{X}}_i(t)\}_{i=1}^n$ the control point velocities. It is opted here to use the same basis functions for the discretization of the surface tension field:

$$\tau^h(\mathbf{x}, t) = \hat{\tau}^h(\boldsymbol{\xi}, t) = \sum_{i=1}^n N_i(\boldsymbol{\xi}) \mathcal{T}_i(t), \quad (5.9)$$

In this expression, $\mathcal{T}(t) = \{\mathcal{T}_i(t)\}_{i=1}^n$ are the control point surface tensions. At every time step in the solution Algorithm 2 the control point surface tensions are found by solving a linear system of equations

$$\mathbf{A}\mathcal{T} = \mathbf{b}, \quad (5.10)$$

where the matrix \mathbf{A} and the right-hand side vector \mathbf{b} evidently follow from the discretization of the inextensibility condition (5.7). The algorithm used to construct this matrix and this right-hand-side vector is presented in Algorithm 3 and will be discussed in detail below.

Before proceeding to this algorithm it is important to note that the partial differential equation representing the inextensibility condition (5.7) does not require a higher-order continuity of the discretized surface tension field (5.9).

However, since it is anticipated that an accurate representation of the tension field, $\tau^h(\boldsymbol{x}, t)$, is essential for the stability of the time integration, the continuous (or even smooth) representation of this field by means of splines is here preferred over a traditional piecewise continuous Lagrange finite element discretization.

Discretization of the surface tension field by means of Lagrange finite elements has been considered in *e.g.* [25]. At first sight an advantage of traditional finite elements compared to spline elements appears to be the interpolatory character of the Lagrange shape functions. In the finite element discretization, the surface tension is associated with the nodes which reside on the surface, this in contrast to the case of splines, where the control points are generally not positioned on the surface. Although this might be intuitive, this discretization will cause the stress jump over the interface to be discontinuous over the element boundaries by virtue of the surface gradient present in Equation (5.2). This is evidently not properly reflecting the physical behavior of the stress discontinuity and can be expected to play a role in the robustness problems with traditional boundary integral methods as reported in literature, *e.g.* [126].

The discretization of the surface tension field by means of the finite element methods has led to the development of penalty methods [26, 128, 129]. In these methods, an artificial membrane stiffness is introduced which in an approximate sense represents the inextensibility condition. In the discrete setting, the penalty methods result in additional contributions to the discrete interface stress jumps. These additional terms are obtained through the insertion of elastic springs between all nodes. Upon increasing the stiffness of these springs, the exact inextensibility condition is approximated more accurately. Penalty methods are capable of explicitly representing the inextensibility condition (*i.e.* without the need for solving a linear system of equation in each time step), but their accuracy has been observed to be limited. This lack of accuracy has driven the development of more robust methods for satisfying the inextensibility condition, also in traditional boundary integral methods [25].

When using isogeometric analysis to find the surface tension field, the use of a penalty method is less intuitive than when traditional finite elements are used. The primary reason for this is that the control net does not represent the surface directly. As a consequence, the physical interpretation of the insertion of springs between all control points is unclear, which makes such an approach less attractive. The derivation of a mathematical framework in which the penalty method can be applied to isogeometric analysis along the lines presented in [128] is possible, but this does not resolve the above-mentioned accuracy problems associated with the penalty methods. Of course, such a penalty method would have the advantage that the surface tension can be approximated in an explicit way. Since the majority of the computational effort in the isogeometric

analysis boundary integral method (as studied for *e.g.* the drop in the previous chapter) is associated with the evaluation of the convolution integrals, the computational costs for solving a linear system of equations in every time step are not problematic. As a consequence, it is here chosen to rigorously resolve the inextensibility condition.

5.3.2 Isogeometric collocation

The matrix A and right-hand-side vector b required to compute the control point surface tensions \mathcal{T} by means of the linear system (5.10) follow by the application of the collocation method to the inextensibility condition (5.7). The collocation points, $\{\xi_i^{\text{col}}\}_{i=1}^n$, are selected in the same way as done for the determination of the control point velocities in Section 3.3.3.

The algorithm used to assemble these arrays is presented in Algorithm 3. In the first step, the interpolation matrix for the interface stress jump is constructed. Subsequently, for every collocation point individually, the parametric derivatives of the Green's functions and background flow field are computed. Per collocation point these derivatives then result in the assembly of a row in the above-mentioned arrays. The individual steps of this algorithm are discussed in detail below.

Algorithm 3 Inextensibility constraint linear system assembly

Construct the interpolation matrix $M(x)$ ▷ See Eq. (5.12)

Collocation point loop

for all $i = 1, \dots, n$ **do**

Compute the velocity field parametric derivatives ▷ See Eq. (5.13)

Compute the right-hand-side vector entry b_i ▷ See Eq. (5.10)

for all elements and integration points **do**

Assemble the i^{th} row of the matrix A ▷ See Eq. (5.21)

end for

end for

Construction of the stress jump interpolation matrix

The stress jump over the interface can be written as a linear combination of the control point surface tensions as

$$f(x) = M(x)\mathcal{T}, \tag{5.11}$$

with $\mathbf{M}(\mathbf{x})$ a $3 \times n$ interpolation matrix. This interpolation matrix is obtained by substitution of the discretization of the surface tension, Equation (5.9), into the stress jump expression, Equation (5.2), to obtain

$$\mathbf{f}(\mathbf{x}) = \sum_{i=1}^n [2N_i(\mathbf{x})H(\mathbf{x})\mathbf{n}(\mathbf{x}) - \nabla_s N_i(\mathbf{x})] \mathcal{T}_i = \mathbf{M}(\mathbf{x})\mathcal{T}. \quad (5.12)$$

Hence, the stress jump interpolation matrix is computed through the evaluation of the spline shape functions and their surface gradients, as well as the normal vector and mean curvature of the interface. Note that the spline basis functions, $N_i(\mathbf{x})$, are here considered to be defined over the the physical surface $\Gamma \ni \mathbf{x}$. The computation of the surface gradient of these basis functions (or any scalar function), is discussed in detail in Appendix 5.B.

Parametric differentiation of the convolution equation

The inextensibility condition (5.7) is related to the tension field (5.9) through the convolution equation (5.1). In order to construct the linear system of equations (5.10) it is required to compute the derivatives of the velocity field with respect to the parametric coordinates. Here, this differentiation process is illustrated for the coordinate ξ , but all derivations hold for the other parametric coordinate, η , as well.

Let $\mathbf{x}_0 = \mathbf{x}_0(\xi)$ be a function of the parametric coordinate ξ . Differentiation of the convolution equation (5.1) with respect to ξ then reads

$$\frac{\partial \hat{\mathbf{u}}}{\partial \xi}(\xi) = \frac{\partial \mathbf{u}^\infty}{\partial \mathbf{x}_0} \frac{\partial \mathbf{x}_0}{\partial \xi}(\xi) - \frac{1}{8\pi} \int_{\Gamma} \frac{\partial \hat{\mathbf{G}}(\xi, \mathbf{x})}{\partial \xi} \mathbf{f}(\mathbf{x}) \, d\Gamma, \quad (5.13)$$

with $\hat{\mathbf{u}}(\xi) = \mathbf{u}(\mathbf{x}_0)$ and $\hat{\mathbf{G}}(\xi, \mathbf{x}) = \mathbf{G}(\mathbf{x}_0, \mathbf{x})$. The derivative of $\mathbf{u}^\infty(\mathbf{x}_0)$ follows directly from the definition of the background flow. For the case of shear flow as introduced in Equation (4.7) this derivative substantiates to

$$\frac{\partial \mathbf{u}^\infty}{\partial \mathbf{x}_0} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \quad (5.14)$$

Note that the coordinate ξ is related to the point \mathbf{x}_0 and not to the integration variable \mathbf{x} . As a consequence, the derivative of the stress jump with respect to the variable ξ is equal to zero. Moreover, in Equation (5.13) the derivative of the

Stokeslet

$$\hat{\mathbf{G}}(\boldsymbol{\zeta}, \mathbf{x}) = \frac{\mathbf{I}}{|\bar{\mathbf{x}}|} + \frac{\bar{\mathbf{x}} \otimes \bar{\mathbf{x}}}{|\bar{\mathbf{x}}|^3}, \quad (5.15)$$

with $\bar{\mathbf{x}}(\boldsymbol{\zeta}, \mathbf{x}) = \mathbf{x} - \mathbf{x}_0(\boldsymbol{\zeta})$ is given by

$$\frac{\partial \hat{\mathbf{G}}}{\partial \boldsymbol{\zeta}} = -\frac{\mathbf{I}}{|\bar{\mathbf{x}}|^2} \frac{\partial |\bar{\mathbf{x}}|}{\partial \boldsymbol{\zeta}} - 3 \frac{\bar{\mathbf{x}} \otimes \bar{\mathbf{x}}}{|\bar{\mathbf{x}}|^4} \frac{\partial |\bar{\mathbf{x}}|}{\partial \boldsymbol{\zeta}} + \frac{\frac{\partial \bar{\mathbf{x}}}{\partial \boldsymbol{\zeta}} \otimes \bar{\mathbf{x}}}{|\bar{\mathbf{x}}|^3} + \frac{\bar{\mathbf{x}} \otimes \frac{\partial \bar{\mathbf{x}}}{\partial \boldsymbol{\zeta}}}{|\bar{\mathbf{x}}|^3}. \quad (5.16)$$

In this expression, the derivative of the length of $\bar{\mathbf{x}}$ is given by

$$\frac{\partial |\bar{\mathbf{x}}|}{\partial \boldsymbol{\zeta}} = \frac{\partial}{\partial \boldsymbol{\zeta}} \left(\sqrt{\bar{\mathbf{x}} \cdot \bar{\mathbf{x}}} \right) = \frac{\bar{\mathbf{x}}}{\sqrt{\bar{\mathbf{x}} \cdot \bar{\mathbf{x}}}} \cdot \frac{\partial \bar{\mathbf{x}}}{\partial \boldsymbol{\zeta}} = \frac{\bar{\mathbf{x}}}{|\bar{\mathbf{x}}|} \cdot \frac{\partial \bar{\mathbf{x}}}{\partial \boldsymbol{\zeta}}. \quad (5.17)$$

where $\frac{\partial \bar{\mathbf{x}}}{\partial \boldsymbol{\zeta}} = -\frac{\partial \mathbf{x}_0}{\partial \boldsymbol{\zeta}}$. It is noted here that the derivatives of the Stokeslet contain singularities. Numerical integration of these functions is a non-trivial aspect and will be studied in Section 5.4.1.

Assembly of the linear system

The linear system of equations (5.10) is constructed by enforcing the inextensibility condition (5.7) in every collocation point $\{\boldsymbol{\zeta}_i^{\text{col}}\}_{i=1}^n$. For every collocation point this results in a single equation, which corresponds to a single row in the system of equations. To derive these equations, the derivatives of the velocity field, $\frac{\partial \mathbf{u}}{\partial \boldsymbol{\zeta}}$ and $\frac{\partial \mathbf{u}}{\partial \boldsymbol{\eta}}$ (see Equation (5.13)), are substituted in the inextensibility constraint (5.7):

$$\begin{aligned} & \mathbf{d}(\boldsymbol{\zeta}_i^{\text{col}})^T \frac{\partial \mathbf{u}^\infty}{\partial \mathbf{x}_0} \frac{\partial \mathbf{x}_0}{\partial \boldsymbol{\zeta}}(\boldsymbol{\zeta}_i^{\text{col}}) - \frac{1}{8\pi} \int_{\Gamma} \mathbf{d}(\boldsymbol{\zeta}_i^{\text{col}})^T \frac{\partial \hat{\mathbf{G}}}{\partial \boldsymbol{\zeta}}(\boldsymbol{\zeta}_i^{\text{col}}, \mathbf{x}) \mathbf{f}(\mathbf{x}) \, d\Gamma + \\ & - \mathbf{e}(\boldsymbol{\zeta}_i^{\text{col}})^T \frac{\partial \mathbf{u}^\infty}{\partial \mathbf{x}_0} \frac{\partial \mathbf{x}_0}{\partial \boldsymbol{\eta}}(\boldsymbol{\zeta}_i^{\text{col}}) + \frac{1}{8\pi} \int_{\Gamma} \mathbf{e}(\boldsymbol{\zeta}_i^{\text{col}})^T \frac{\partial \hat{\mathbf{G}}}{\partial \boldsymbol{\eta}}(\boldsymbol{\zeta}_i^{\text{col}}, \mathbf{x}) \mathbf{f}(\mathbf{x}) \, d\Gamma = 0. \end{aligned} \quad (5.18)$$

After rearranging terms and substitution of the interpolation of the stress discontinuity interpolation (5.11), this expression is rewritten as

$$\left[\frac{1}{8\pi} \int_{\Gamma} \left[\mathbf{d}^T(\boldsymbol{\zeta}_i^{\text{col}}) \frac{\partial \hat{\mathbf{G}}}{\partial \boldsymbol{\zeta}}(\boldsymbol{\zeta}_i^{\text{col}}, \mathbf{x}) - \mathbf{e}^T(\boldsymbol{\zeta}_i^{\text{col}}) \frac{\partial \hat{\mathbf{G}}}{\partial \boldsymbol{\eta}}(\boldsymbol{\zeta}_i^{\text{col}}, \mathbf{x}) \right] \mathbf{M}(\mathbf{x}) \, d\Gamma \right] \boldsymbol{\mathcal{T}} = \mathbf{b}_i, \quad (5.19)$$

where the expression within the square brackets in the left-hand-side corresponds to the i^{th} row in the matrix \mathbf{A} and

$$b_i = \mathbf{d}(\boldsymbol{\xi}_i^{\text{col}})^T \frac{\partial \mathbf{u}^\infty}{\partial \mathbf{x}_0} \frac{\partial \mathbf{x}_0}{\partial \boldsymbol{\xi}}(\boldsymbol{\xi}_i^{\text{col}}) - \mathbf{e}(\boldsymbol{\xi}_i^{\text{col}})^T \frac{\partial \mathbf{u}^\infty}{\partial \mathbf{x}_0} \frac{\partial \mathbf{x}_0}{\partial \boldsymbol{\eta}}(\boldsymbol{\xi}_i^{\text{col}}), \quad (5.20)$$

is the corresponding entry in the right-hand side vector \mathbf{b} .

Evidently, from an implementation point of view, the construction of the linear system (5.10) through the expressions (5.19) results in a triple loop. The outer loop iterates over the collocation points, $\boldsymbol{\xi}_i^{\text{col}}$, where the index $i = 1, \dots, n$ is then associated with the rows in the matrix \mathbf{A} and vector \mathbf{b} . Within the collocation point iteration, two nested loops over the elements and integration points, respectively, are used to assemble the i^{th} row of the matrix:

$$\frac{1}{8\pi} \mathbf{A}_{e=1}^m \int_{\Gamma^e} \left[\mathbf{d}^T(\boldsymbol{\xi}_i^{\text{col}}) \frac{\partial \hat{\mathbf{G}}}{\partial \boldsymbol{\xi}}(\boldsymbol{\xi}_i^{\text{col}}, \mathbf{x}) - \mathbf{e}^T(\boldsymbol{\xi}_i^{\text{col}}) \frac{\partial \hat{\mathbf{G}}}{\partial \boldsymbol{\eta}}(\boldsymbol{\xi}_i^{\text{col}}, \mathbf{x}) \right] \mathbf{M}^e(\mathbf{x}) \, d\Gamma, \quad (5.21)$$

where numerical quadrature is used to evaluate the integral over the elements Γ^e and $\mathbf{M}^e(\mathbf{x})$ is the stress jump interpolation matrix localized to the element e :

$$\mathbf{f}(\mathbf{x})|_{\Gamma^e} = \mathbf{M}^e(\mathbf{x}) \mathbf{T}^e. \quad (5.22)$$

In this expression, \mathbf{T}^e are the control point tensions corresponding to the n_e basis functions with support over the element Γ^e . As discussed in Section 3.3.1, for bivariate basis functions $n_e = (p + 1)^2$ where p is the order of B-splines.

5.4 Numerical simulations

In this section, the performance of the isogeometric boundary integral method for the inextensible membrane is studied. Unless mentioned otherwise, the simulations will consider the deformation of an initially ellipsoidal cell suspended in an iso-viscous shear flow (see Equation (4.7)). These ellipsoids (or, more precisely, oblate spheroids) are centered at the origin of a Cartesian reference frame and have semi-principal axes of length a and $c \leq a$ (two times). The largest semi-principal axis, which is referred to as the semi-major axis, resides in the xz -plane and makes an angle θ_0 with respect to the horizontal x -axis (see Figure 5.1). In the remainder of this chapter, the length of the semi-minor axes, c , is taken equal to one. It is noted that this choice is arbitrary, since it can be scaled with the parameter R_0 used in the non-dimensionalization (see Appendix 5.A). The ratio

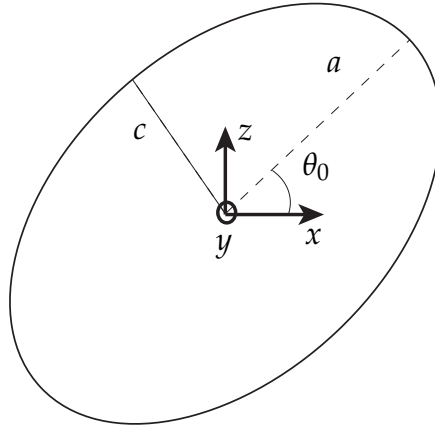


Figure 5.1 Side-view of an initially ellipsoidal cell. The inclination angle θ_0 is defined as the angle between the semi-major axis of length a and the x -axis. The ratio between the lengths of the semi-major axis and two semi-minor axes is denoted by $r = \frac{a}{c}$.

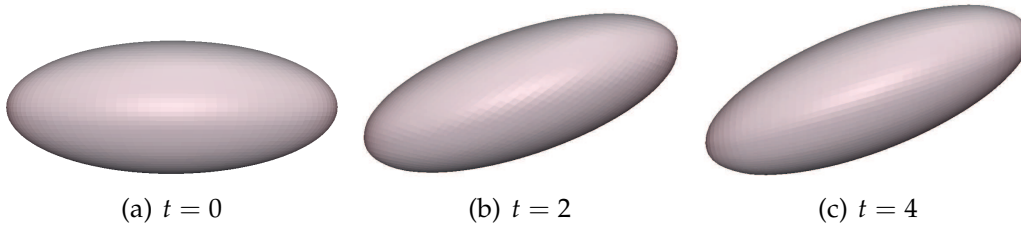


Figure 5.2 The motion of an inextensible membrane is characterized by the deformation of the membrane and the alignment of the cell with a stable inclination angle. For an initially ellipsoidal shape the deformations of membrane are generally small. Although the shape of the cell stabilizes in time, the membrane continues to rotate around the cell's interior (tank-treading motion).

between the lengths of the semi-major axis and semi-minor axes, referred to as the axes ratio, is denoted by $r = \frac{a}{c}$.

Figure 5.2 shows the typical motion of an inextensible membrane in shear flow. A cell with axes ratio $r = 2.5$ and inclination angle $\theta_0 = 0$ deforms as a result of the imposed shear flow. The cell deforms and rotates until reaching a stable shape and taking a fixed direction. Note that in this stable position the membrane is still rotating around its interior (tank-treading behavior). Over time the centroid of the cell remains in the same position, as is expected from symmetry considerations. For an initially ellipsoidal shape the deformations of the membrane are generally minor, *i.e.* the stable shape only slightly differs from an ellipsoid. Deformations will not be minor when considering other initial shapes, as will be discussed later in this chapter.

Although the evolution of the shape of the cell at first sight is similar to that observed for a drop, there is a fundamental difference between the two cases when the deformation of the membrane is considered. In the case of the drop, the rotation of the membrane is of no influence on the evolution of its shape. In the case of the inextensible membrane, the interface does rotate around the cell's interior. This tank-treading behavior is an essential aspect of the motion of an inextensible membrane, and prohibits the deformation of the membrane to be described by the normal component of its velocity only, as was done for the drop in Equation (4.4).

Since the motion of the inextensible membrane is fundamentally different from that of the drop, it is evident that the time step stability criterion, Equation (4.5), is not applicable. Therefore the time step stability criterion is studied numerically in this work. Time integration with time steps ranging from $\Delta t = 0.01$ to $\Delta t = 0.002$ was found to be stable, whereas instabilities occurred when time steps larger than $\Delta t = 0.01$ were used. These observations were made for membranes discretized with various numbers of elements. For the sake of accuracy, all results presented in this chapter are obtained with a dimensionless time step of $\Delta t = 0.002$.

The isogeometric analysis framework for the determination of the surface tension field as discussed in Section 5.3.1 is in essence a collocation method. The choice for this approach was motivated by the numerical observations made for the drop, where the collocation method was found to be robust. It has to be noted, however, that also for the computation of the surface tension alternative approaches, such as *e.g.* an L^2 -projection, could be considered. Apart from this, like for the drop, the computation of the control net velocities, $\{\mathbf{u}_i\}_{i=1}^n$, can be performed by means of both methods considered for the drop, *i.e.* the collocation method or the L^2 -projection method.

The favorable stability of the collocation method compared to the L^2 -projection is also observed for the inextensible membrane. This is illustrated in Figure 5.3, which shows the top view of the control net for a cell with 200 elements and $r = 2.5$. Figure 5.3 (a) shows that in the case of an L^2 -projection an irregular distortion appears in the control net upon the deformation of the membrane. This behavior ultimately results in a diverging solution. Figure 5.3 (b) shows the control net obtained using the collocation method. Clearly, the problematic distortions of the control net observed with the L^2 -projection are no longer present. This result is consistent with the observations for the drop as discussed in the previous chapter.

In the remainder of this section, various computational aspects of the isogeometric boundary integral formulation for the inextensible membrane will be considered. First, in Section 5.4.1, the conservation of the surface area and cell volume is studied. In Section 5.4.2, the B-spline approximation of the surface tension at the membrane and its correlation with the mean curvature is discussed.

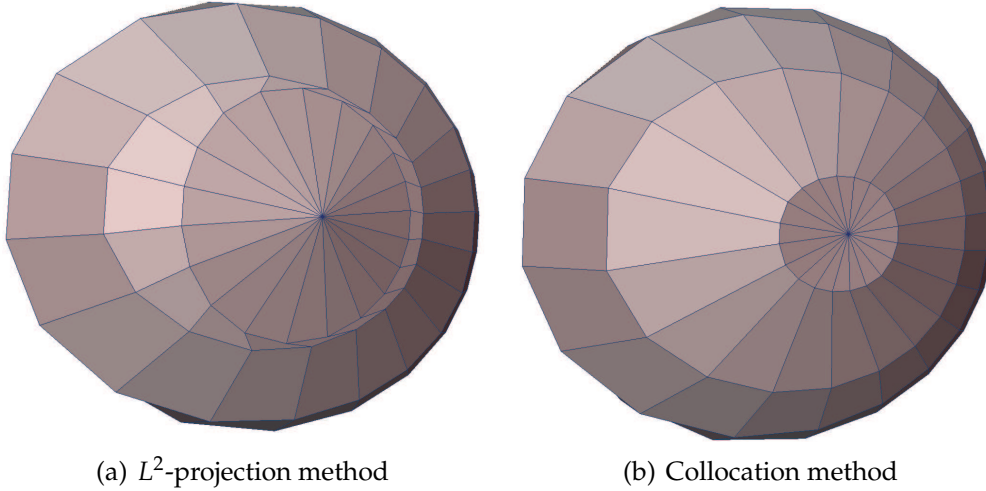


Figure 5.3 Irregular distortions in the control net appear in the case of an L^2 projection, whereas these distortions are not present in the case of a collocation method.

Finally, in Section 5.4.3 the dependence of the cell behavior on its initial shape and inclination angle is considered.

5.4.1 The incompressibility and inextensibility conditions

Two essential characteristics of the inextensible membrane are the global volume conservation of the cell as a result of the incompressibility of the Stokes flow, and the local surface area conservation as a consequence of the inextensibility constraint introduced in this chapter. The conservation of these quantities by the isogeometric boundary integral formulation is studied in this section. Initially ellipsoidal shapes with varying elongations are simulated over a time period of $t = 10$ when all of them have reached a stable shape. The major axis of all these ellipsoids is aligned with the x -axis, *i.e.* $\theta_0 = 0$.

The change in volume over a period of $t = 10$ is reported in Table 5.1 for ellipsoids with axis-ratios running from 2 to 5.5. For all simulations 24×24 Gauss points are used per element to numerically evaluate the surface integrals appearing in the formulation. As observed, the volume variations are small for all considered cases. The variations in volume, which are present despite the use of the volume constraint (see Section 4.3.2), are primarily attributed to integration errors. Unfortunately, singularity subtraction (see Section 4.3.2) cannot be used directly in the case of the inextensible membrane as a consequence of tangential stress jump components in Equation (5.2). Numerical simulations with singularity subtraction only applied to the parts of the integrals that involved

normal components of the stress jump yielded unsatisfactory results, which led to the decision not to use singularity subtraction in the case of the inextensible membrane. It has to be noted that recently Farutin *et al.* [130] presented a novel approach that incorporates the tangential component of the stress jump in a singularity subtraction technique. This novel technique is not used in this work, but it is anticipated to be capable of drastically reducing the computational effort of the current method since accurate integration can be achieved with relatively coarse integration schemes.

Table 5.1 A comparison between the initial and the final volume (at $t = 10$) for ellipsoids with different axes ratios. All results are obtained with 200 elements and 24×24 integration points per element.

Axes ratio (r)	Initial volume (V_0)	Final volume (V_f)	$\frac{V_f - V_0}{V_0}$ %
2	8.38508	8.41214	0.32
2.5	10.48135	10.50622	0.23
3	12.60015	12.57762	0.17
3.5	14.67389	14.69406	0.13
4	16.77017	16.78812	0.10
4.5	18.86644	18.88239	0.08
5	20.96271	20.97686	0.06
5.5	23.05898	23.07147	0.05

Table 5.2 shows the change in surface area over a period of $t = 10$ for ellipsoids with different aspect ratios. All results are obtained using a mesh with 200 elements. Although global surface conservation does not guarantee the pointwise satisfaction of the inextensibility condition, it is here assumed that it is an indicator for the accuracy of the employed method with respect to the satisfaction of the inextensibility constraint. To study the influence of the integration precision on the conservation of the surface area, in Table 5.3 a comparison of various orders of integration schemes is presented. The corresponding evolution of the surface area in time is shown in Figure 5.4. It is observed that the surface area conservation is dramatically influenced by the accuracy of the numerical integration procedures. Using too coarse integration schemes leads to unrealistic variations in the surface area and ultimately to a diverging solution.

Evidently, the precision of the numerical integrations plays an important role in the satisfaction of both the volume and surface area constraint. The accuracy with which the surface tension and velocity field are approximated is also of influence on these constraints. To study this effect the surface area is computed for various meshes, ranging from 128 to 392 elements. The variation of the surface area over time for these meshes is shown in Figure 5.5. It is observed that the variation in

Table 5.2 A comparison between the initial and the final surface area (at $t = 10$) for ellipsoids with different axes ratios. All results are obtained with 200 elements and 24×24 integration points per element.

Axes ratio (r)	Initial surface area (A_0)	Final surface area (A_f)	$\frac{A_f - A_0}{A_0}$ %
2	21.49173	21.23964	1.17
2.5	26.16813	25.92253	0.93
3	30.91305	30.64887	0.85
3.5	35.70263	35.54620	0.43
4	40.52295	40.57564	0.13
4.5	45.36535	45.56840	0.44
5	50.22412	50.60002	0.75
5.5	55.09539	55.56046	0.84

Table 5.3 Surface area at $t = 10$ for a cell with an inextensible membrane discretized with 200 elements and various numbers of Gauss points. The cell is initially ellipsoidal with axes ratio $r = 2$ and initial area $A_0 = 21.49173$.

Surface area at the equilibrium state ($t = 10$)		
Number of integration points	Final surface area	$\frac{A_f - A_0}{A_0}$ %
16^2	20.49149	4.65
20^2	20.99402	2.31
24^2	21.23964	1.17
32^2	21.34242	0.69

the number of elements has a moderate effect on the precision with which the inextensibility condition is satisfied, particularly for meshes with 200 elements or more. Moreover, it must be noted that upon increasing the number of elements the total number of integration points is also increased, since the number of Gauss points per element is kept constant at 24×24 .

Another computational aspect that affects the precision with which the volume and surface area constraints are satisfied is the time integration scheme. For the inextensible membrane the influence of the time step in the forward Euler method was observed to be of minor importance.

5.4.2 The surface tension approximation

The B-spline approximation of the surface tension, Equation (5.9), is studied here for initially horizontally aligned ellipsoids with axes ratios of 2 and 2.5. Figure 5.6

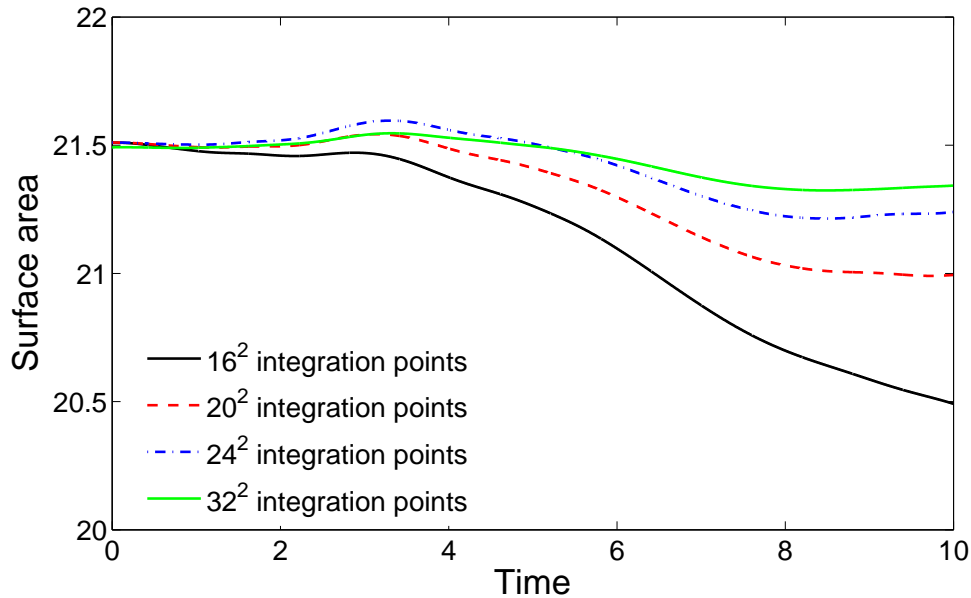


Figure 5.4 Evolution of the surface area in time for an initially ellipsoidal cell with axes ratio $r = 2$. Various numbers of Gauss points are considered, whereas the number of elements is 200 for all cases.

shows the surface tension when the cell has reached a stable shape (while still experiencing tank-treading behavior) for two different meshes with 200 elements (Mesh a) and 512 elements (Mesh b). The observed minor variations between the two meshes are attributed to the approximation accuracy of the B-spline discretization. Overall, it can be concluded that the computed surface tension is objective with respect to the mesh size.

Figure 5.7 shows the surface tension over the equilibrium shape of the cell for an initial ellipsoid with $r = 2$. The surface tension, τ , is minimal where the mean curvatures is highest and vice versa. This feature can be interpreted as the need of higher surface tension in the more stretching area (in the middle of the membrane). This inverse relation between the surface tension and the mean curvature is also reported by Zhou and Pozrikidis [25]. Figure 5.6 shows the surface tension for an axes ratio of $r = 2.5$. The maximum of the surface tension is less compared to the simulation with $r = 2$. The observation that a more elongated ellipsoid results in a lower tension at the surface is in agreement with the results reported by Zhou and Pozrikidis [25].

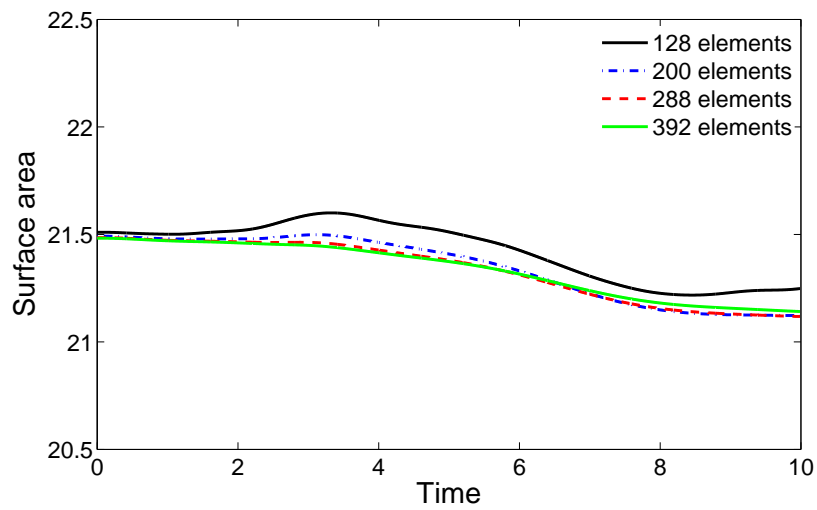
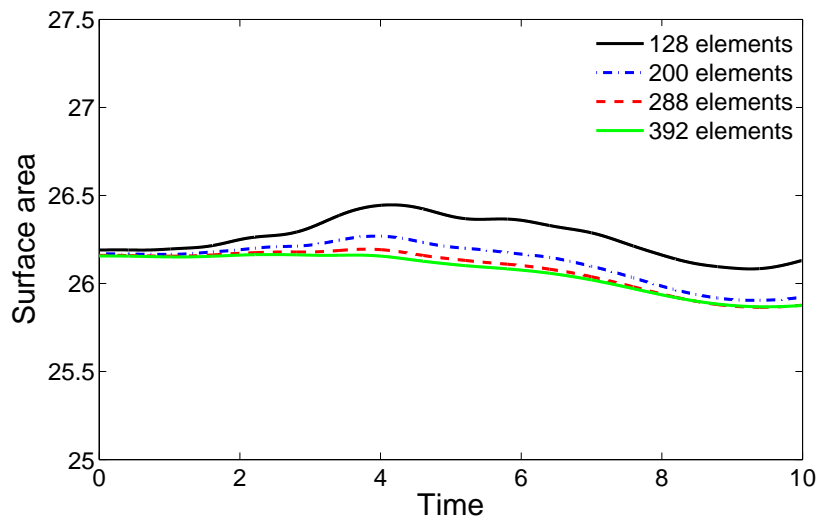
(a) $r = 2$ (b) $r = 2.5$

Figure 5.5 Evolution of the surface area in time for initially ellipsoidal cells with axes ratio (a) $r = 2$ and (b) $r = 2.5$. Various numbers of elements are considered, whereas the number of Gauss points equals 24×24 for all cases.

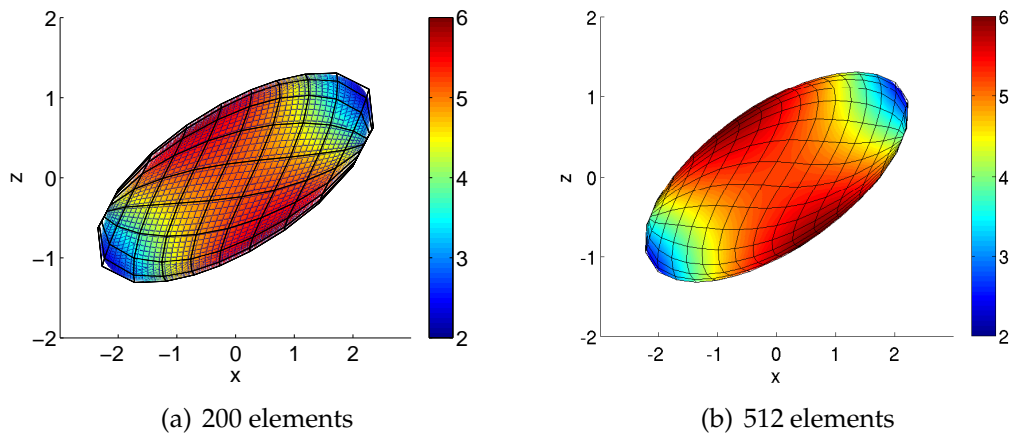


Figure 5.6 Tension distribution over the surface of an initially ellipsoidal cell with axes ratio $r = 2.5$ in its stable shape computed with (a) 200 elements, and (b) 512 elements.

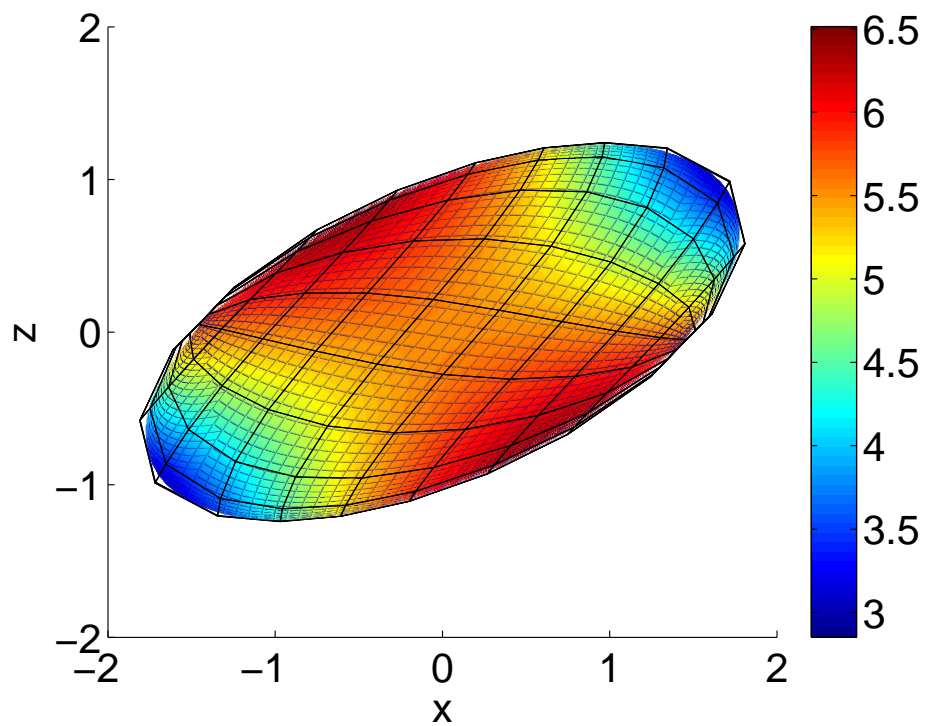


Figure 5.7 Tension distribution over the surface of an initially ellipsoidal cell with axes ratio $r = 2$ in its stable shape. The presented result is obtained using 200 elements.

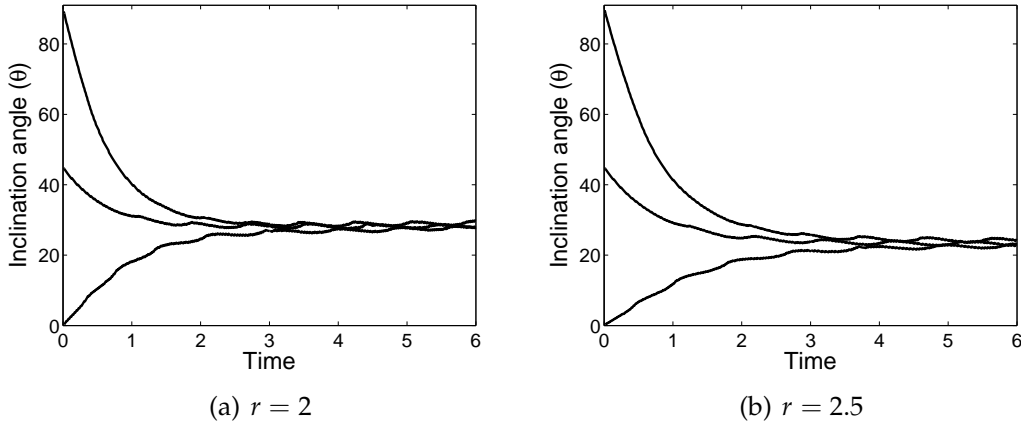


Figure 5.8 The evolution of the cell's inclination angle for initially ellipsoidal shapes with two different axes ratios and various initial inclination angles, θ_0 .

5.4.3 The evolution of the inclination angle

The inclination angle of a cell, $\theta(t)$, changes over time as a consequence of the shear flow in which the cell is suspended. In the initial configuration, the inclination angle equals $\theta(0) = \theta_0$. For $t > 0$, when the cell is generally no longer ellipsoidal, the inclination angle is defined as the angle between the vector pointing from the centroid of the cell to the farthest away point on the membrane and the x -axis. It is noted that this definition is a generalization of the definition for the initial angle, θ_0 .

The evolution of θ over time is shown in Figure 5.8 for ellipsoids with two axes ratios, *viz.* $r = 2$ and $r = 2.5$. Figures 5.9 (a), (c) and (e) show the considered initial angles $\theta_0 = 0^\circ$, $\theta_0 = 45^\circ$ and $\theta_0 = 90^\circ$, respectively. Figures 5.9 (b), (d) and (f) display the corresponding equilibrium shapes of these cells. It is observed that the stable inclination angle of the cell is independent from its initial inclination angle, given that the shape of the cells are identical. This behavior is in agreement with results reported in literature [25].

In Figure 5.10, the stable inclination angle is plotted as a function of the axes ratio, r , of the initially horizontally aligned ellipsoidal cell. The inclination angle typically stabilizes for $t > 3$. It must be noted that due to the tank-treading motion of the membrane minor variations in the inclination angle remain present. For that reason, the stable inclination angles are accurate up to more or less a single significant digit. The obtained three-dimensional results correspond well with similar results reported in literature for two-dimensional simulations [25]. Also, the presented results are in good agreement with the three-dimensional

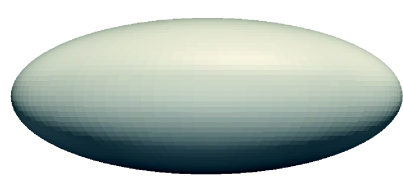
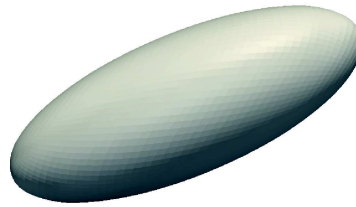
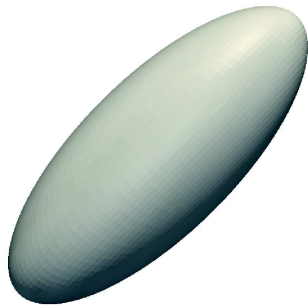
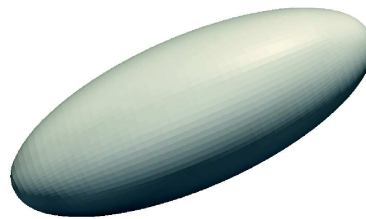
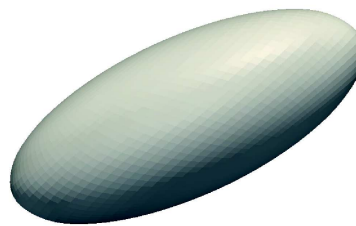
(a) $\theta_0 = 0, t = 0$ (b) $\theta_0 = 0, t = 10$ (c) $\theta_0 = 45^\circ, t = 0$ (d) $\theta_0 = 45^\circ, t = 10$ (e) $\theta_0 = 90^\circ, t = 0$ (f) $\theta_0 = 90^\circ, t = 10$

Figure 5.9 Initial and stable cell shapes for various inclination angles of initially ellipsoidal cells with axes ratio $r = 2$. All results are obtained using 200 elements.

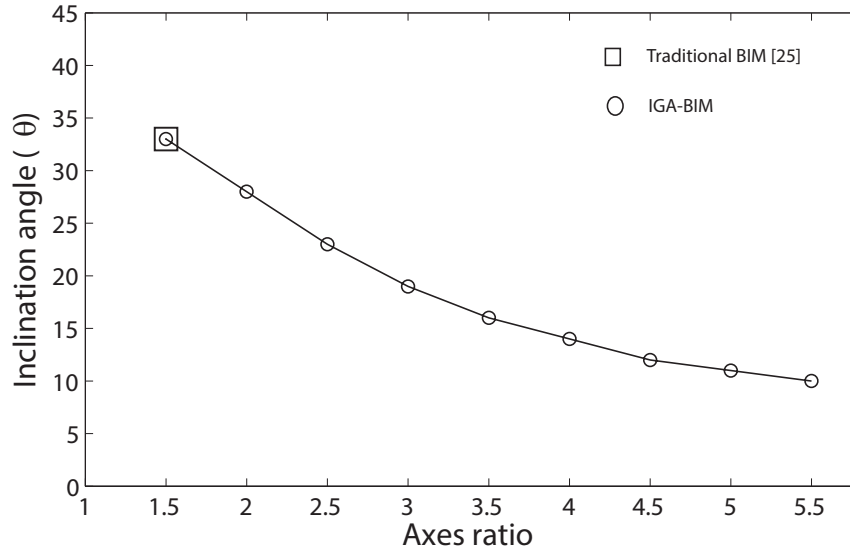


Figure 5.10 The equilibrium inclination angle of cells with varying axes ratios, r , ranging from 1.5 to 5.5. Note that the result for $r = 1.5$ as obtained by Zhou and Pozrikidis [25] using the traditional boundary integral formulation is also depicted for comparison.

result reported by Zhou and Pozrikidis [25] for the axes ratio $r = 1.5$. It is observed that more elongated cells tend to align more closely with the horizontal axis. Figure 5.11 illustrates this for a cell with an inextensible membrane in shear flow with axes ratio $r = 5$, which stabilizes at an inclination angle of only $\theta = 11^\circ$.

Finally the deformation of a biconcave initial shape, which resembles a real red blood cell, is considered. In Figure 5.12, the initial shape and stable shape are shown. It is observed that the biconcave shape deforms dramatically and becomes ellipsoidal, while the membrane shows tank-treading behavior. This observed behavior is in good agreement with results reported by Zhou and Pozrikidis [25] for the two-dimensional case.

To quantify the deformation behavior of the biconcave shape, the non-dimensional deflation parameter (or reduced volume) is defined as

$$s = \frac{V_{cell}}{V_s} = \frac{V_{cell}}{\frac{4\pi}{3} \left(\frac{A}{4\pi}\right)^{\frac{3}{2}}}, \quad (5.23)$$

where A is the surface area of the cell, V_{cell} is the cell volume, and V_s is the volume of a sphere with surface area A . The deflation parameter ranges from 0 to 1, where $s = 1$ corresponds to a sphere. In this case, in the initial state,

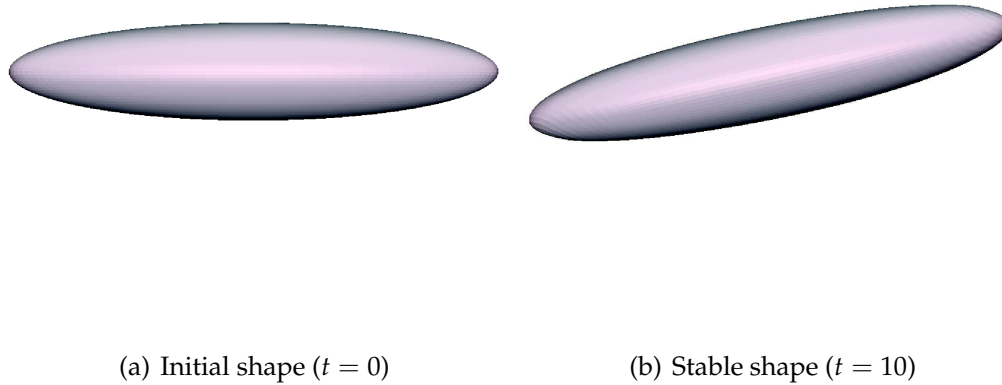


Figure 5.11 The initial and stable shape of a cell with axes ratio $r = 5$ using 200 elements.

$A = 22.48029$, $V = 6.203019$ and therefore $s = 0.6189$. This approximately corresponds to an ellipsoidal cell with axes ratio $r = 5$ (for which $s = 0.6263$). Comparing the equilibrium shape of the biconcave cell with the axes ratio $r = 5$ shown in Figure 5.11 (b) it is observed that these two initial shapes lead to very similar equilibrium shapes. Also note that the final inclination angle for both cases is approximately 11° . Hence, it is concluded that the deflation parameter governs the stable shape of a cell with an inextensible membrane in shear flow. This observation supports results reported in literature [131, 132].

5.5 Conclusions

Extension of the isogeometric boundary integral formulation for the drop, as considered in Chapter 4, to the case of an inextensible membrane is non-trivial due to the presence of the inextensibility condition. This condition results in an implicit definition of the surface tension (which appears in the stress discontinuity definition) through the solution of a surface partial differential equation. An isogeometric boundary integral formulation, based on a collocation method, is proposed to approximate the surface tension.

The isogeometric boundary integral method for the inextensible membrane is found to be robust for a relatively wide range of three-dimensional initial shapes.

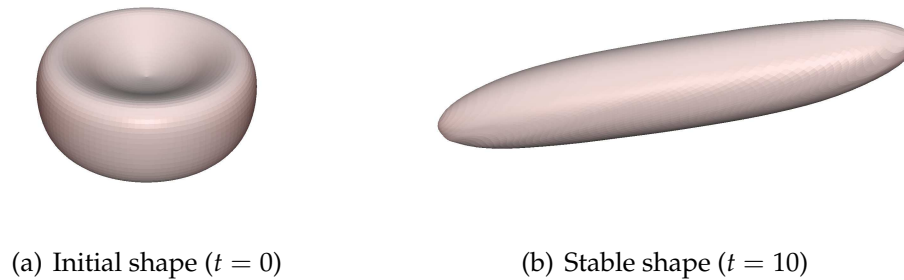


Figure 5.12 The initial and stable shape of a biconcave cell discretized with 288 elements. The initial surface area equals $A_0 = 22.48029$ and the surface area in the stable shape is $A = 22.64561$.

Various results reported in literature are confirmed. Moreover, results only reported for two-dimensional simulations in literature are now generalized to the three-dimensional case. Overall it can be concluded that the isogeometric framework, without the application of stabilization methods, has an improved robustness compared to the traditional boundary integral methods.

The accuracy of the developed isogeometric boundary integral method is found to be affected by the limited precision of the numerical integration schemes. The application of a singularity subtraction technique that incorporates tangential stress jump contributions is anticipated to dramatically improve the accuracy of the current method. This will also provide the possibility for improving the computational performance of the developed method. Another source of inaccuracies is caused by the incapability of the B-spline discretization to accurately represent the tank-treading behavior. Although reasonably accurate results are obtained in every time step, the errors accumulate in time due to the continuous motion of the mesh around the interior of the cell. The solution for this problem should either be found in the usage of adaptive discretization techniques, or in reformulation of the problem to eliminate the tangential motion of the mesh.

Although several improvements can still be made to the isogeometric boundary integral formulation for the inextensible membrane, the developed algorithm has been applied successfully to this prototypical model of a red blood cell. The results obtained for a three-dimensional biconcave initial shape in shear flow demonstrates the capabilities of the isogeometric framework to capture some of the fundamental physics of a red blood cell.

Appendices

5.A Non-dimensionalization of the inextensible membrane formulation

At any given point in the domain, the velocity using the boundary integral method can be computed by

$$\mathbf{u}(\mathbf{x}_0) = \mathbf{u}_\infty(\mathbf{x}_0) - \frac{1}{8\pi\mu} \int_{\Gamma} \mathbf{G}(\mathbf{x}_0, \mathbf{x}) \mathbf{f}(\mathbf{x}) d\Gamma, \quad (5.24)$$

where \mathbf{u}_∞ is a far-field imposed velocity (shear flow in this case), \mathbf{G} is the Green's function and \mathbf{f} is the stress jump across the interface and according to Pozrikidis [15], in the case of an inextensible membrane, it is defined as

$$\mathbf{f} = 2H\tau\mathbf{n} - \nabla_s \tau, \quad (5.25)$$

where H is the mean curvature, \mathbf{n} is the unit outward normal vector and τ is the surface tension. The following dimensionless variables are considered [25]

$$H^* = dH, u^* = \frac{u}{kd}, u_\infty^* = \frac{u_\infty}{kd}, \Delta^* = d^2\Delta, \tau^* = \frac{\tau}{\mu kd}, \mathbf{G}^* = d\mathbf{G}, \quad (5.26)$$

with $1/k$ the characteristic time and kd the characteristic velocity. Substituting the dimensionless variables in (5.26) into the stress discontinuity (5.25), gives

$$\mathbf{f} = \mu k \underbrace{(2\tau^* H^* \mathbf{n} - \nabla_s^* H^*)}_{\mathbf{f}^*}. \quad (5.27)$$

Substituting new stress jump (5.27) into the velocity approximation (5.24), leads to

$$k d \mathbf{u}^*(\mathbf{x}_0) = k d \mathbf{u}_\infty^*(\mathbf{x}_0) - \frac{1}{8\pi\mu} \left[\int_{\Gamma} \mu k d \mathbf{f}^* \mathbf{G}^*(\mathbf{x}_0, \mathbf{x}) d\Gamma^* \right]. \quad (5.28)$$

Thus

$$\mathbf{u}^*(\mathbf{x}_0) = \mathbf{u}_\infty^*(\mathbf{x}_0) - \frac{1}{8\pi} \int_{\Gamma} \mathbf{f}^* \mathbf{G}^*(\mathbf{x}_0, \mathbf{x}) d\Gamma^*. \quad (5.29)$$

For the sake of convenience, the superscript $*$ is omitted for the non-dimensional parameters in the main context of the chapter.

5.B Surface gradient of the scalar-valued field

Let f denotes a scalar-valued parameter and x be the position vector which is related to the parametric domain ξ by

$$f(x) = \hat{f}(\xi). \quad (5.30)$$

Therefore, differentiation of f with respect to x can be written as

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial \xi} \frac{\partial \xi}{\partial x}. \quad (5.31)$$

Multiplication with $\frac{\partial \xi^T}{\partial x}$ gives

$$\frac{\partial f}{\partial x} \frac{\partial \xi^T}{\partial x} = \frac{\partial f}{\partial \xi} \frac{\partial \xi}{\partial x} \frac{\partial \xi^T}{\partial x} = \frac{\partial f}{\partial \xi} A^{-1}, \quad (5.32)$$

with $A_{\alpha\beta} = g_\alpha \cdot g_\beta$ where $\alpha, \beta = 1, 2$ is the first fundamental matrix explained in Section 3.3.1. Taking the transpose of (5.32) and multiplying that with $\frac{\partial x}{\partial \xi}$ and using the symmetry of A gives

$$\frac{\partial f^T}{\partial x} = \frac{\partial x}{\partial \xi} A^{-1} \frac{\partial f^T}{\partial \xi}. \quad (5.33)$$

On the other side,

$$\frac{\partial f}{\partial x} n = \frac{\partial f}{\partial \xi} A^{-1} \frac{\partial x^T}{\partial \xi} n, \quad (5.34)$$

where n is the unit outward normal vector to the position. Since $\frac{\partial x}{\partial \xi}$ and n are orthogonal, $\frac{\partial x^T}{\partial \xi} n = \mathbf{0}$. Therefore, by considering $\nabla = \left(\frac{\partial}{\partial x}\right)^T$, the surface gradient operator is defined as

$$\nabla_s f = \nabla f - (\nabla f \cdot n)n = \frac{\partial f^T}{\partial x} - \underbrace{\left(\frac{\partial f}{\partial x} n\right)}_0 n = \frac{\partial f^T}{\partial x}. \quad (5.35)$$

Therefore, substituting Equation (5.33) into Equation (5.35) gives

$$\nabla_s f = \frac{\partial \mathbf{x}}{\partial \bar{\xi}} \mathbf{A}^{-1} \frac{\partial f}{\partial \bar{\xi}}^T, \quad (5.36)$$

where can be re-expressed as

$$\nabla_s f = \sum_{\alpha=1}^2 \sum_{\beta=1}^2 A_{\alpha\beta}^{-1} \left(\frac{\partial f}{\partial \bar{\xi}_\beta} \mathbf{g}_\alpha \right), \quad (5.37)$$

and using the symmetry of \mathbf{A}^{-1} ,

$$\nabla_s f = \sum_{\alpha=1}^2 \sum_{\beta=1}^2 A_{\alpha\beta}^{-1} \left[\frac{1}{2} \left(\frac{\partial f}{\partial \bar{\xi}_\beta} \mathbf{g}_\alpha + \frac{\partial f}{\partial \bar{\xi}_\alpha} \mathbf{g}_\beta \right) \right]. \quad (5.38)$$

A validation process has been performed to verify the correctness of the formulation (5.38) to take the surface gradient of any scalar parameters in the parametrized surface in Appendix 5.C.

5.C Verification of the surface gradient of a scalar-valued field formulation

To examine the validity of (5.38), an arbitrary surface is produced. Given an arbitrary set of control points $\mathbf{B}_{i,j}$, $i = 1, \dots, 4$, $j = 1, \dots, 4$ and polynomial orders p , a tensor product surface is defined by

$$\mathbf{C}(\bar{\xi}) = \sum_{i=1}^4 \sum_{j=1}^4 N_{i,p}(\bar{\xi}) N_{j,p}(\eta) \mathbf{B}_{i,j}, \quad (5.39)$$

where $N_{i,p}(\bar{\xi})$, $N_{j,p}(\eta)$ are univariate Bernstein polynomials of order p which in current case is 3. $\mathbf{B}_{i,j}$ is given by

$$\mathbf{B} = \begin{pmatrix} (0, 0, 1) & (1, 0, \frac{1}{2}) & (2, 0, \frac{1}{2}) & (3, 0, 1) \\ (0, 2, 2) & (1, 2, \frac{3}{2}) & (2, 2, \frac{3}{2}) & (3, 2, 2) \\ (0, 4, 1) & (1, 4, \frac{1}{2}) & (2, 4, \frac{1}{2}) & (3, 4, 1) \\ (0, 6, 2) & (1, 6, \frac{3}{2}) & (2, 6, \frac{3}{2}) & (3, 6, 2) \end{pmatrix}. \quad (5.40)$$

Setting $C_1(\xi) = x$ and $C_2(\xi) = y$ where C_1 and C_2 are the first and second components of \mathbf{C} , and solving these equations gives $\xi = \begin{pmatrix} y \\ 6x \\ 3 \end{pmatrix}$. A parametrized surface $z = C_3(\xi)$ can then be implicitly defined by the function

$$r(\mathbf{x}) = z - C_3(\xi). \quad (5.41)$$

Considering $\xi_0 = 1/4$ and $\eta_0 = 2/3$ leads to $(x_0, y_0, z_0) = (2, 3/2, 103/72)$. Now, two functions are available that the derivatives can be easily taken from. $r(\mathbf{x})$ can be treated analytically by taking the simple gradient with respect to \mathbf{x} . Hence, the normal vector and the mean curvature can be found analytically as follows

$$\mathbf{n}(\mathbf{x}) = \frac{\nabla r(\mathbf{x})}{\|\nabla r(\mathbf{x})\|}, \quad (5.42)$$

and

$$H = \frac{1}{2} \nabla \cdot \mathbf{n}(\mathbf{x}). \quad (5.43)$$

In addition, further surface derivatives of the mean curvature (e.g. $\nabla_s H$) can be obtained. Also, the surface derivative of the mean curvature ($\nabla_s H$) in the parametrized space can be computed using Equation (5.38). Both cases result in

$$\nabla_s H = \left(-\frac{8120180224}{1320139673\sqrt{1097}}, -\frac{100033626944}{106931313513\sqrt{1097}}, -\frac{36654211840}{106931313513\sqrt{1097}} \right).$$

Therefore, this can verify the correctness of Equation (5.38).

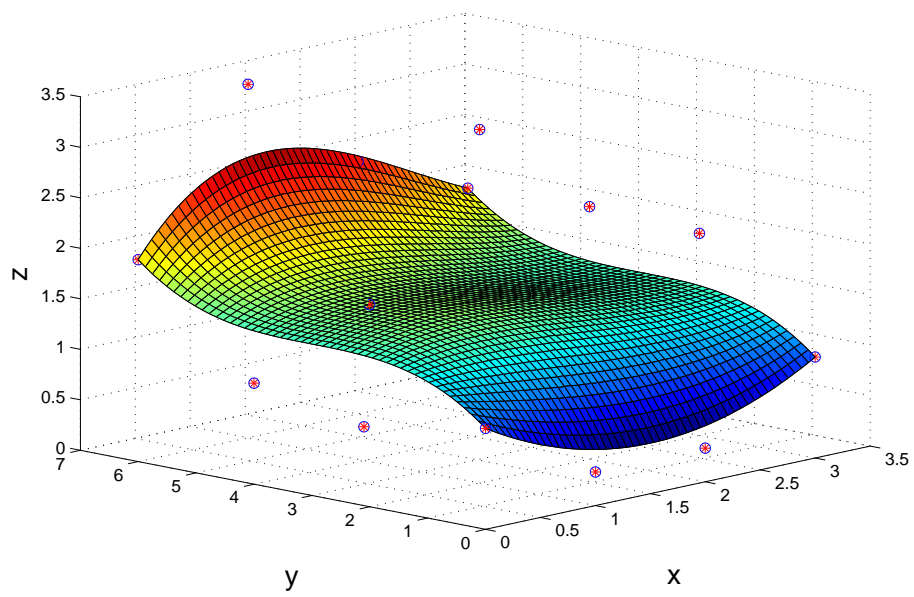


Figure 5.13 An arbitrary analytical function built by a given set of control points (red dots) and cubic basis function.

6.1 Introduction

In the previous chapters, two ways of discretizing the interface have been discussed. Using a linear surface triangulation (Chapter 3) the curvature of the surface and its derivatives – which occur in the stress jump across the interface of *e.g.* a vesicle – cannot be determined with sufficient accuracy. This makes the triangulation approach unsuitable for the simulation of a vesicle. To remedy these problems, in Section 3.3, isogeometric analysis is presented. The key idea about isogeometric analysis is that the higher-order continuity of the employed basis functions allows for the direct evaluation of the curvature and its derivatives. The applicability of isogeometric analysis in combination with the boundary integral formulation (IGA-BIM) has been demonstrated for the cases of a drop (Chapter 4) and an inextensible membrane (Chapter 5).

Despite promising results for the drop and the inextensible membrane, instabilities are encountered when simulating a vesicle in a background flow using IGA-BIM. In Figure 6.1, two steps of this simulation are shown when the time step is taken as $\Delta t = 0.001$. Figure 6.1 (a) displays an object at the initial moment which is an ellipsoid made by quartic B-splines ($p = 4$) with 160 elements. Note that since the second-order derivative of the mean curvature (fourth-order derivative of the interface position) is needed to obtain the stress jump over the interface of a vesicle, quartic B-splines with their C^3 -continuity are anticipated to yield a well-defined parametrization. From the simulation, it is observed that the vesicle starts to deform and orient in agreement with the expectations during the first steps of the simulation. After some time, however, irregular behavior of the surface at the poles is observed. Figure 6.1 (b) shows the geometry one step before the simulation breaks down completely ($t = 0.18$).

The irregular behavior at the poles is explained by the fact that – despite the

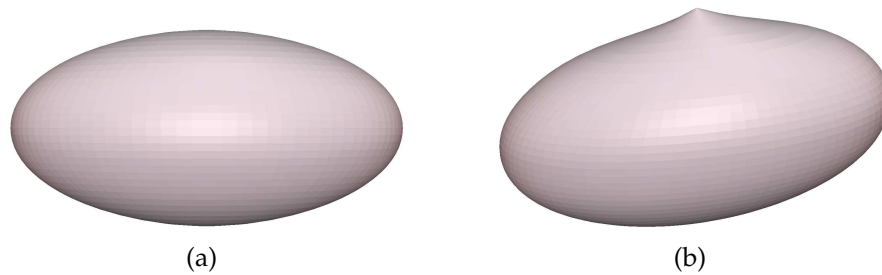


Figure 6.1 A vesicle deformation in shear flow with 160 elements and quartic B-splines ($p = 4$) (a) at the initial moment, and (b) at $t = 0.18$.

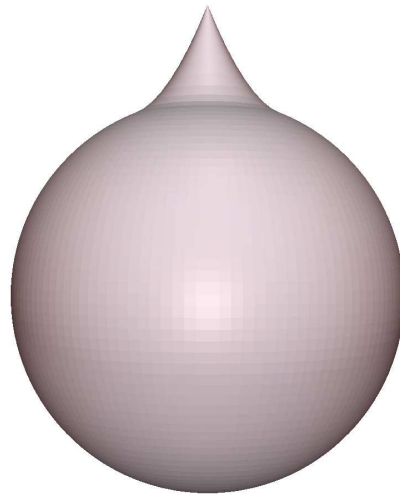


Figure 6.2 A geometry with 128 elements and quadratic splines with the control point at the pole being pinched out. Non-smoothness of the surface at the pole is observed.

higher-order continuity of the surface – the normal, curvature and its derivative cannot be evaluated at the pole. This is a consequence of the degeneration of the elements at that point, *i.e.* the quadrilateral shape is reduced to a triangular shape by making all control points at one of the edges coincident. As a result, at the pole the basis functions (plotted over the physical domain) are C^0 only, this in contrast to the C^{p-1} continuity of the basis over the domain excluding the poles. In Figure 6.2, this reduced continuity at the poles is illustrated by pinching out the control point associated with the pole.

To alleviate the problems associated with evaluating parametric derivatives at the poles, the use of a hybrid geometry representation is proposed. The conceptual idea of this hybrid geometry is to define two parametrizations of the same geometry and to define the hybrid geometry as the weighted average of these two parametrizations. By ensuring that the poles are situated at different positions, it

is then possible to use the other geometry when parametric derivatives are to be evaluated at a pole. In the case of a spherical initial geometry, a natural choice for the alternative geometry is to use a rotation of the original geometry with 90 degrees, such that the pole of the new geometry is situated at the equator of the original geometry.

In Section 6.2, the formulation for the hybrid geometry is introduced. In Section 6.3, the required mathematical frameworks to make the hybrid geometry such as hybrid spline basis functions and weight function definition are presented. The approximation properties of the hybrid geometry such as the normal vector and the mean curvature are studied and results are presented in Section 6.4. Using the hybrid geometry the outcome of a few simulations over the deformation of a drop and an inextensible membrane is shown and the possibilities of using the hybrid geometry for the vesicle simulation is discussed in Section 6.5. Eventually, the conclusions are drawn in Section 6.6.

6.2 The hybrid spline geometry formulation

In this section, the conceptual idea of merging two geometric descriptions as outlined above is formalized. Consider the geometries x_A and x_B , parametrized by the coordinates $\zeta_A \in \hat{\Gamma}$ and $\zeta_B \in \hat{\Gamma}$, respectively. In Figure 6.3, these two practically indistinguishable, geometries are depicted. Although the geometries are very similar, the parametrizations are fundamentally different, which is directly visible from the plotted control nets. In geometry A , the poles are situated on the top and the bottom of the sphere, while in geometry B , the poles are situated at the left and right.

Merging the two geometries x_A and x_B to obtain the hybrid geometry x_w has the potential of alleviating the problems related to the evaluation of the parametric gradients around the poles, since evaluation at the poles is circumvented by making use of the alternative geometry. In the remainder of this section, the hybrid geometry will first be approached from the geometric perspective. After that, the very same concept will be considered from the point of view of the underlying splines basis, which is to be used in conjunction with one of the velocity field approximation methods introduced in the previous chapters.

6.2.1 The weighted-average hybrid geometry

Rather than having a selection criterion for whether to use geometry A or geometry B , the hybrid geometry is defined as the weighted average of these

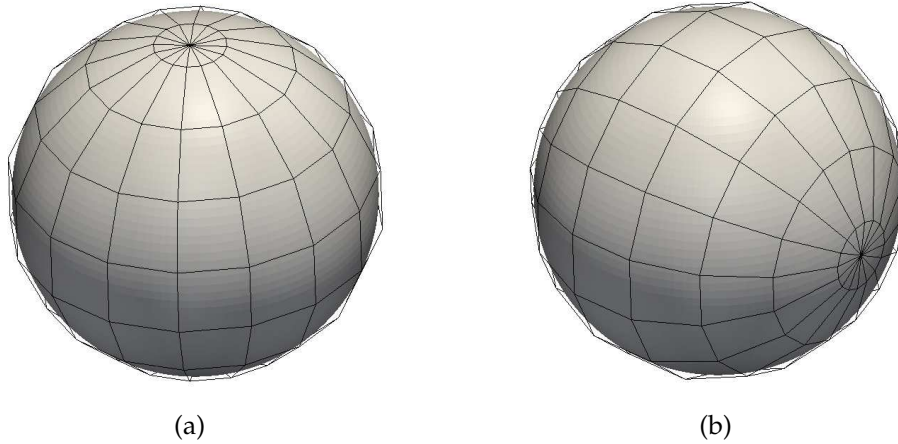


Figure 6.3 (a) Geometry A and (b) Geometry B , with 128 elements and quadratic B-splines.

two geometries,

$$\mathbf{x}_w(\boldsymbol{\zeta}) = w(\boldsymbol{\zeta})\mathbf{x}_A(\boldsymbol{\zeta}_A(\boldsymbol{\zeta})) + (1 - w(\boldsymbol{\zeta}))\mathbf{x}_B(\boldsymbol{\zeta}_B(\boldsymbol{\zeta})), \quad (6.1)$$

where the weight function takes values between 0 and 1. A typical weight function is shown in Figure 6.4. In the case that at a certain point on the geometry $w = 1$, geometry A is used, while geometry B is used when its value equals 0. The above-mentioned selection criterion resembles the case in which w can only take the values 0 and 1. However, upon inspection of Equation (6.1) it is observed that this choice would result in a discontinuous surface in the case that \mathbf{x}_A is not exactly equal to \mathbf{x}_B . Evidently, the smoothness of the hybrid geometry is controlled by the smoothness of the weight function. For example, if the curvature is to be properly defined everywhere on the geometry, the weight function should be continuously differentiable (like the geometry parametrizations A and B). In addition, the weight function w and all its derivatives should vanish at the pole of geometry A , and $1 - w$ and all its derivatives should equate to 0 at the pole of geometry B . If higher-order parametric derivatives of the surface are required, the smoothness of the geometry parametrizations and weight function should be increased further.

In the case that the geometries A and B exactly coincide, *i.e.* for every point on the surface of geometry A a point at exactly the same position on the surface of geometry B can be identified and vice versa, the dependence of the parametric coordinates $\boldsymbol{\zeta}^A$ and $\boldsymbol{\zeta}^B$ on the parametric coordinate of the hybrid geometry, $\boldsymbol{\zeta}$,

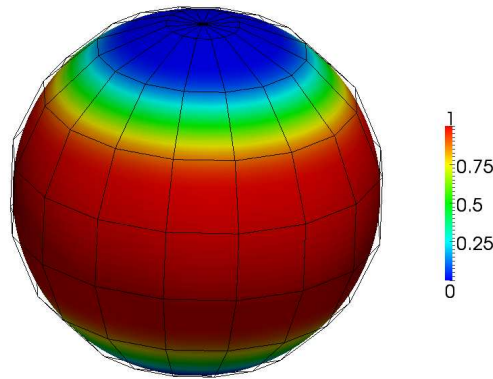


Figure 6.4 A typical weight function w plotted on geometry A .

follows from

$$\mathbf{x}_w(\boldsymbol{\zeta}) = \mathbf{x}_A(\boldsymbol{\zeta}_A) = \mathbf{x}_B(\boldsymbol{\zeta}_B), \quad (6.2)$$

which yields

$$\boldsymbol{\zeta}_A = \mathbf{x}_A^{-1}(\mathbf{x}_w(\boldsymbol{\zeta})) \quad \text{and} \quad \boldsymbol{\zeta}_B = \mathbf{x}_B^{-1}(\mathbf{x}_w(\boldsymbol{\zeta})). \quad (6.3)$$

Note that, in the case of exactly matching geometries, a direct relation between the coordinates $\boldsymbol{\zeta}_A$ and $\boldsymbol{\zeta}_B$ is obtained as

$$\boldsymbol{\zeta}_B(\boldsymbol{\zeta}_A) = \mathbf{x}_B^{-1}(\mathbf{x}_A(\boldsymbol{\zeta}_A)). \quad (6.4)$$

This expression can be generalized for the case in which the geometries A and B do not match exactly, but are approximately the same ($\mathbf{x}_A \approx \mathbf{x}_B$), to yield

$$\boldsymbol{\zeta}_B(\boldsymbol{\zeta}_A) = \arg \min_{\boldsymbol{\zeta} \in \hat{\Gamma}} |\mathbf{x}_A(\boldsymbol{\zeta}_A) - \mathbf{x}_B(\boldsymbol{\zeta})|. \quad (6.5)$$

This expression states that given a parametric coordinate $\boldsymbol{\zeta}_A$ and corresponding physical coordinate $\mathbf{x}_A(\boldsymbol{\zeta}_A)$, the parametric coordinate $\boldsymbol{\zeta}_B$ is defined as the point which corresponds to the position on the surface $\mathbf{x}_B(\boldsymbol{\zeta}_B)$ closest to $\mathbf{x}_A(\boldsymbol{\zeta}_A)$. This mapping is schematically shown in Figure 6.5. In Section 6.2.2, the numerical evaluation of this map from the parametric coordinate $\boldsymbol{\zeta}_A$ to the coordinate $\boldsymbol{\zeta}_B$ is discussed.

In the case of geometries that are approximately the same, the coordinate of the

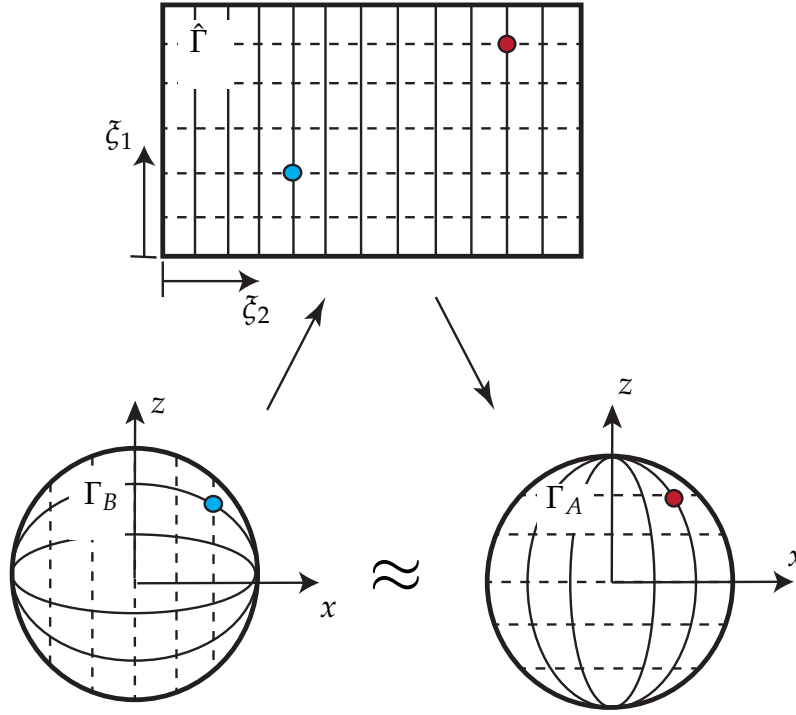


Figure 6.5 Schematic representation of the mapping from parametric coordinate ξ_A to coordinate ξ_B .

hybrid geometry at ξ_A is obtained by

$$\mathbf{x}_w(\xi_A) = w(\xi_A)\mathbf{x}_A(\xi_A) + (1 - w(\xi_A))\mathbf{x}_B(\xi_B(\xi_A)), \quad (6.6)$$

where the choice is made to define the weight function w based on the parametrization of geometry A (see Section 6.3.2 for details). Similarly, one could have defined all functions in terms of ξ_B . Given a particular weight function, the parametrization of the hybrid geometry is fully determined by the parametrizations A and B . In the case of spline discretizations this implies that the hybrid geometry follows from the control points $\{\mathbf{X}_{A,i}\}_{i=1}^n$ and $\{\mathbf{X}_{B,i}\}_{i=1}^n$.

The velocity field of the hybrid geometry, which governs the evolution of the surface in time, follows by differentiation of (6.6) as

$$\mathbf{u}_w(\xi_A) = w(\xi_A)\mathbf{u}_A(\xi_A) + (1 - w(\xi_A))\mathbf{u}_B(\xi_B(\xi_A)), \quad (6.7)$$

which in the case of splines depends on the control point velocities $\{\mathbf{U}_{A,i}\}_{i=1}^n$ and $\{\mathbf{U}_{B,i}\}_{i=1}^n$ only. Note that here it is implicitly assumed that the map $\xi_B = \xi_B(\xi_A)$ is independent of time. In other words, the parametric coordinates ξ_A and ξ_B are couple to each other through the geometries at a fix moment in time,

for which the initial configuration is evidently the natural choice. The choice to make this assumption makes sense when approaching the problem from the basis function perspective, as will be discussed in Section 6.3.1. From the perspective of numerical implementation this choice means that the non-linear map between the two parametric coordinates, which is a computationally expensive operation, is only to be evaluate in the initial configuration.

6.2.2 Numerical evaluation of the parametric coordinate mapping

The mapping between the parametric coordinates ξ_A and ξ_B as mathematically described by Equation (6.5) is evaluated numerically by the minimization of

$$\Pi(\xi) = \frac{1}{2} |x_A(\xi_A) - x_B(\xi)|^2 = \frac{1}{2} (x_A(\xi_A) - x_B(\xi))^T (x_A(\xi_A) - x_B(\xi)), \quad (6.8)$$

which is written as

$$r(\xi_B) = - \left. \frac{\partial \Pi}{\partial \xi} \right|_{\xi=\xi_B} = \left[\left. \frac{\partial x_B}{\partial \xi} \right|_{\xi=\xi_B} \right]^T (x_A(\xi_A) - x_B(\xi_B)) = \mathbf{0}, \quad (6.9)$$

in which the point ξ_A is given and the point ξ_B is to be determined. Note that even if there is not an exact match between the points x_A and x_B , the residual $r(\xi_B)$ is equal to zero.

Newton iterations are used to solve the system of equations (6.9). Given a starting point ξ_B^k , with $k = 0$ being an iteration counter, the updated value of ξ_B is given by

$$\xi_B^{k+1} = \xi_B^k - \left[\left. \frac{\partial r}{\partial \xi} \right|_{\xi=\xi_B^k} \right]^{-1} r(\xi_B^k). \quad (6.10)$$

This expression is evaluated recursively until convergence is reached. The norm of the residual is used as a convergence criterion. The distance between x_A and x_B cannot be used to verify convergence directly, since in general this distance does not go to zero.

An essential aspect of the minimization procedure outline above is the selection of the starting point, ξ_B^0 . A pool of starting points is created by sampling all elements in the parametric domain by a uniform scheme. Based on these sampling points, a sorted list of starting points with increasing distance $|x_A(\xi_A) - x_B(\xi_B^0)|$ is created. The first item of this list is then used to initiate the Newton iterations.

Most often the Newton procedure will converge (at a quadratic rate). However, a few situations exist in which this iterative procedure does not yield the desired result. The most common situation in which this occurs is if the point ζ_B does not reside in the same element as the starting point ζ_B^0 . In this case, the Newton solver can diverge. If this occurs the next item in the sorted list of starting points is used to re-initiate the Newton procedure. This is repeated until the correct element is encountered and convergence is reached. Since the list of starting points is sorted, only a few different starting points are needed. Another complication in the Newton iterations occurs in the very special case in which the point ζ_B is positioned exactly at the pole of geometry A . In this case, the Jacobian matrix in the Newton iterations is not properly defined, which results in a non-converging procedure. In this special case, which was encountered only in the context of post-processing, a very fine sampling of the elements surrounding the pole is used to find an approximation of the point ζ_B without Newton iterations.

In Figure 6.6, the components of ζ_B as evaluated by the above minimization procedure are shown. For this simulation the geometries x_A and x_B are parametrized using quadratic splines with 160 control points each (128 elements). Geometry A is parametrized as discussed in Section 3.3.2. Geometry B is obtained by a clockwise rotation of 90 degrees of this geometry around the y -axis. The control points of geometry B then follow from these of geometry A as

$$\mathbf{X}_{B,i} = \mathbf{R}\mathbf{X}_{A,i} \quad i = 1, \dots, n, \quad (6.11)$$

with n the number of control points and

$$\mathbf{R} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix}. \quad (6.12)$$

For the results presented in Figure 6.6 the parameter domain has been chosen as $\hat{\Gamma} = [-\pi, \pi] \otimes [0, \pi]$. In order to properly visualize the periodic nature of the first component of ζ_B , its absolute value is plotted. It is clearly observed from the second component that the poles of geometry B , which correspond to the points where the $\zeta_{B,2} = 0$ or π , are situated exactly on the equator of geometry A . Evidently, the weight function is to be selected such that the weight at the equator of geometry A is equal to 1 and hence, the pole of geometry B is not to be evaluated.

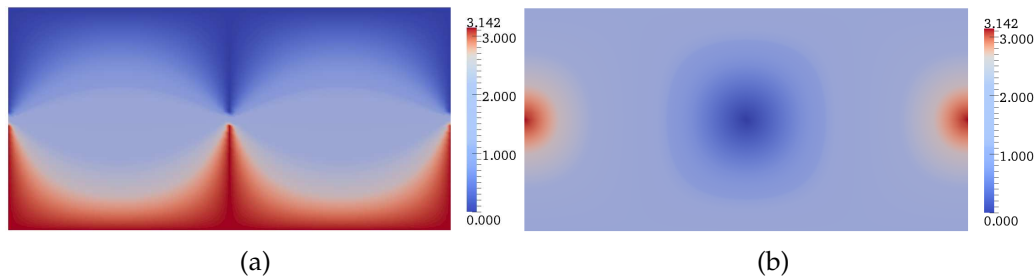


Figure 6.6 The two components, (a) $\zeta_{B,1}$ and (a) $\zeta_{B,2}$, of ζ_B , plotted as a function of ζ_A .

6.2.3 Evaluation of parametric gradients

With the hybrid geometry defined in Section 6.2.1 and the minimization algorithm for the evaluation of the parameter coordinate mapping discussed in Section 6.2.2, the basis for the computation of the normal vector and curvature (and derivatives thereof) of the hybrid geometry is formed. Before proceeding to the derivation of the parametric description of these geometric quantities, first the fundamentally different character of the hybrid geometry in the vicinity of the poles is emphasized. In Figure 6.7, the original geometry, x_A , with the pinched-out pole as discussed earlier in the introduction of this chapter is shown in subfigure (a). Subfigure (b) displays the hybrid geometry with the pole pinched out. Evidently, in contrast to the original geometry, the surface remains smooth in the hybrid case. In fact, upon proper selection of the B-spline order and weight function, the curvature and its derivatives remain well-defined over the complete surface of the hybrid geometry.

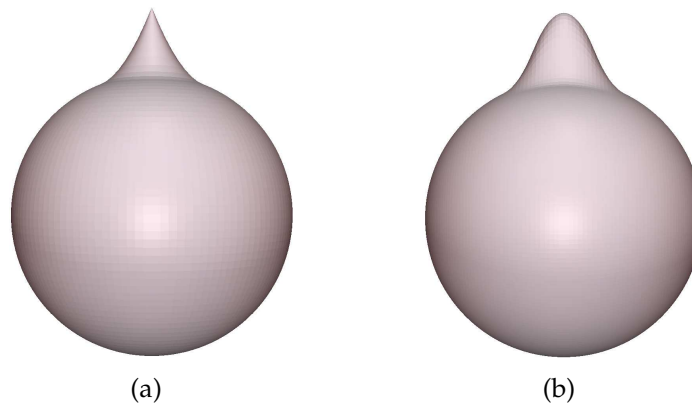


Figure 6.7 Behavior of the surface around a pinched-out pole for (a) Geometry A, and (b) the hybrid geometry.

To illustrate how the parametric derivatives of the hybrid geometry are computed, the definition (6.6) is considered. For the sake of notational simplicity the coordinate ξ_A is replaced by ξ , such that the derivative of the hybrid geometry with respect to this coordinate ξ is given by

$$\frac{\partial \mathbf{x}_w}{\partial \xi_i} = \frac{\partial w}{\partial \xi_i} (\mathbf{x}_A - \mathbf{x}_B) + w \frac{\partial \mathbf{x}_A}{\partial \xi_i} + (1 - w) \frac{\partial \mathbf{x}_B}{\partial \xi_{B,k}} \frac{\partial \xi_{B,k}}{\partial \xi_i}, \quad (6.13)$$

with $i = 1, 2$. Assuming that \mathbf{x}_A is approximately equal to \mathbf{x}_B , simplifies the above expression to

$$\frac{\partial \mathbf{x}_w}{\partial \xi_i} = \mathbf{g}_{w,i} \approx w \mathbf{g}_{A,i} + (1 - w) \mathbf{g}_{B,i}, \quad (6.14)$$

with

$$\mathbf{g}_{A,i} = \frac{\partial \mathbf{x}_A}{\partial \xi_i} \quad \text{and} \quad \mathbf{g}_{B,i} = \frac{\partial \mathbf{x}_B}{\partial \xi_{B,k}} \frac{\partial \xi_{B,k}}{\partial \xi_i}. \quad (6.15)$$

The normal vector for the hybrid geometry can then be computed through the expression (3.23) by using either (6.13) or its approximation (6.14). Higher-order derivatives of the hybrid geometry, as required for the evaluation of *e.g.* the curvature, can be computed through differentiation of (6.13). Motivated by Equation (6.14) an approximation of the curvature of the hybrid geometry is obtained as

$$H = w H_A + (1 - w) H_B, \quad (6.16)$$

with H_A and H_B the mean curvature evaluated at the geometries A and B , respectively. The Laplacian of the curvature can be approximated similarly.

6.3 Algorithmic aspects

In Algorithm 4, the computational procedure to simulate the motion of the interface using the hybrid spline parametrization is schematically shown. In essence, this algorithm does not differ from the algorithm used for a simulation with a standard B-spline geometry. In this section, this similarity will be clarified by considering the hybrid spline formulation from the basis function point of view. Moreover, in this section the definition of the weight function is further clarified.

Algorithm 4 Hybrid geometry time integration procedure*Initialization* $t = 0$ Define \mathbf{X}_A & \mathbf{X}_B ▷ \mathbf{X}_A & \mathbf{X}_B : Initial control netsDetermine \mathbf{x}_A and \mathbf{x}_B Minimizing $|\mathbf{x}_A(\boldsymbol{\xi}_A) - \mathbf{x}_B(\boldsymbol{\xi}_B)|$

▷ See Section 6.2.2

Define the weight function w

▷ See Section 6.3.2

Find $\mathbf{x} = w\mathbf{x}_A + (1 - w)\mathbf{x}_B$ *Time integration***repeat** Computation of the stress jump $\mathbf{f}(\mathbf{x}, t)$ Computation of the velocity field $\mathbf{u}(\mathbf{x}, t)$ Geometry update: $\mathbf{x} \leftarrow \mathbf{x} + \Delta t \mathbf{u}$ ▷ Δt : Time step size Time incrementation: $t \leftarrow t + \Delta t$ **until** $t \geq t_{\text{end}}$ ▷ t_{end} : Simulation end time**6.3.1 Hybrid spline basis functions**

The hybrid spline is defined as described in Equation (6.6). Using the B-spline parametrization of \mathbf{x}_A and \mathbf{x}_B , this expression can be rewritten as

$$\mathbf{x}_w(\boldsymbol{\xi}) = w(\boldsymbol{\xi}) \sum_{i=1}^n N_{A,i}(\boldsymbol{\xi}) \mathbf{X}_{A,i} + (1 - w(\boldsymbol{\xi})) \sum_{i=1}^n \hat{N}_{B,i}(\boldsymbol{\xi}) \mathbf{X}_{B,i}, \quad (6.17)$$

where $\boldsymbol{\xi}_A$ in Equation (6.6) is replaced by $\boldsymbol{\xi}$ and the basis functions N_B are considered as functions of the coordinate $\boldsymbol{\xi}$, which means that the mapping from $\boldsymbol{\xi}_A$ to $\boldsymbol{\xi}_B$ as discussed in Section 6.2.2 is employed. The above expression can be cast into the standard form

$$\mathbf{x}_w(\boldsymbol{\xi}) = \sum_{i=1}^{2n} N_i(\boldsymbol{\xi}) \mathbf{X}_i, \quad (6.18)$$

where $\{N_i(\boldsymbol{\xi})\}_{i=1}^{2n}$ represents the basis functions for the hybrid geometry and $\{\mathbf{X}_i\}_{i=1}^{2n}$ is the collection of control points corresponding to the geometries A and B :

$$\mathbf{N} = \begin{pmatrix} w\mathbf{N}_A \\ (1 - w)\mathbf{N}_B \end{pmatrix}, \quad \mathbf{X} = \begin{pmatrix} \mathbf{X}_A \\ \mathbf{X}_B \end{pmatrix}. \quad (6.19)$$

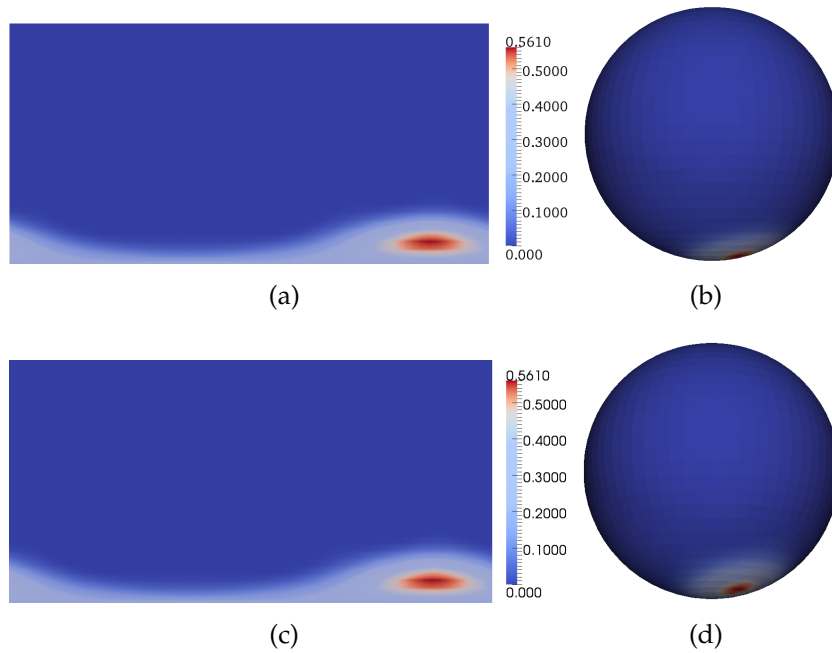


Figure 6.8 A basis function in N_B positioned in the vicinity of a pole of geometry A plotted over the parameter domain (a) and physical domain (b). Subfigures (c) and (d) display the corresponding basis function in N .

The basis functions $N_A(\xi)$ are piece-wise polynomials which are defined over the parameter domain $\hat{\Gamma}$. The pulled-back basis functions $N_B(\xi)$ are not piece-wise polynomial as a consequence of the non-linear map from ξ_A to ξ_B . From the perspective of numerical integration, this is an aspect to be aware of, since standard Gauss quadrature can be inaccurate. It has to be remarked, however, that due to the presence of the singular Green's functions, relatively fine integration schemes are used anyway. For that reason, in the context of the IGA-BIM the non-polynomial character of the pulled-back basis functions is not anticipated to be problematic.

Figure 6.8 (a) and (b) show a basis function $N_{B,i}$ in the neighborhood of the south pole of the geometry. The weight function over the support of this basis function is practically equal to zero. As a consequence, the corresponding basis function in N , which is shown in Figure 6.8 (c) and (d) is very similar. An important observation is that, in contrast to the original basis functions in N_A , this basis function is smooth over the pole of geometry x_A .

From the basis function definition in (6.19) it is observed that the composite basis

functions N form a partition of unity over the parameter domain:

$$\sum_{i=1}^{2n} N_i(\xi) = w \underbrace{\sum_{i=1}^n N_{A,i}(\xi)}_1 + (1-w) \underbrace{\sum_{i=1}^n N_{B,i}(\xi)}_1 = 1 \quad \forall \xi \in \hat{\Gamma}. \quad (6.20)$$

This partition of unity property plays a role when scaling of the geometry by means of affine transformations.

6.3.2 Weight function definition

Selecting the proper weight function to identify the inaccurate zone, and to cover the accurate part of the domain in the best way, is an important aspect in the hybrid geometry formulation. In Section 6.2.1, it was pointed out that the weight function should always be between 0 and 1, while it should vanish at the poles. Knowing these characteristics, two different types of weight functions are presented here.

Weight function using B-splines A natural choice for defining the weight function is to use the B-spline discretization used for the original geometries. The essential idea is to assign a weight, W_i , to each of the control points in order to obtain the weight function

$$w(\xi) = \sum_{i=1}^n N_{A,i}(\xi) W_i. \quad (6.21)$$

Note that this weight is not related to the control point weights used to construct non-uniform rational B-splines (NURBS). A slice of the control net is shown in Figure 6.9. It is remarked that the weight function depends on the parametric coordinate ξ_1 only, and hence is constant in circumferential direction. This figure shows the interpolation of the control point weights by means of a quadratic B-spline. A value of zero is assigned to the control points near the poles, whereas a value of one is used near the equator. By setting the first and last $p+1=3$ weight values to zero, the weight function and its derivatives vanish at the poles. A zone in which the weight is equal to 1 is created around the equator by assigning the value of one to the remaining control points. Evidently, this B-spline weight function has C^1 continuity.

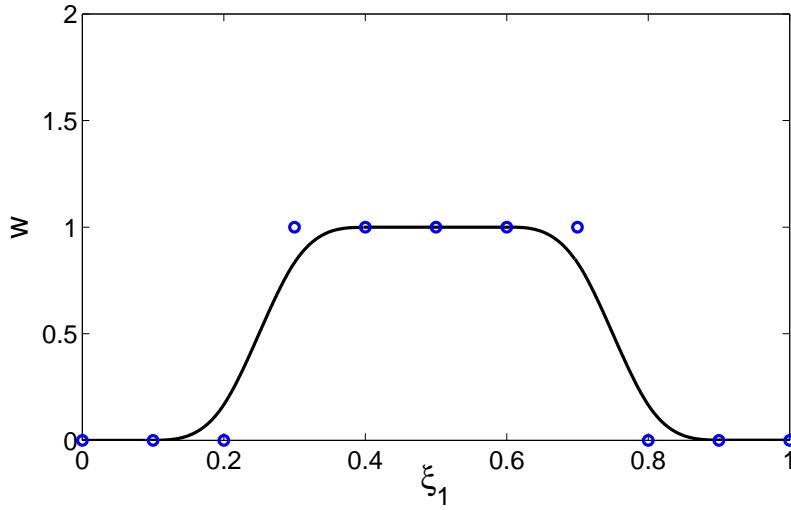


Figure 6.9 A weight function constructed using a quadratic B-spline, plotted in the parametric direction ξ_1 . The blue dots reflect the control point weight values.

Weight function using trigonometric functions Because of their smoothness, trigonometric functions are an attractive choice for the construction of the weight function. Here the following C^1 -continuous weight function is defined:

$$w(\xi) = \begin{cases} 0 & 0 \leq \xi_1 < 0.1 \\ \frac{1}{2} + \frac{1}{2} \sin\left(\frac{10}{3}(\xi_1 - 0.25)\pi\right) & 0.1 \leq \xi_1 < 0.4 \\ 1 & 0.4 \leq \xi_1 < 0.6 \\ \frac{1}{2} + \frac{1}{2} \sin\left(\frac{10}{3}(\xi_1 + 0.75)\pi\right) & 0.6 \leq \xi_1 < 0.9 \\ 0 & 0.9 \leq \xi_1 \leq 1. \end{cases} \quad (6.22)$$

Figure 6.10 displays this trigonometric weight function. As shown in this figure, in the vicinity of the poles ($0 \leq \xi_1 \leq 0.1$ and $0.9 \leq \xi_1 \leq 1.0$), $w(\xi_1) = 0$. In the zone $0.4 \leq \xi_1 \leq 0.6$, which covers the middle zone of the geometry, $w(\xi_1) = 1$. These zones are smoothly connected through a sine function.

In Figure 6.11, the weight function presented in Figure 6.9 is plotted over the parameter domain in terms of ξ_A and over the physical domain. In subfigure (c), the weight function as a function of the parameter coordinate ξ_B is shown.

In the remainder of this chapter, the capability of these two weight functions to reduce the error in the computation of the mean curvature is studied.

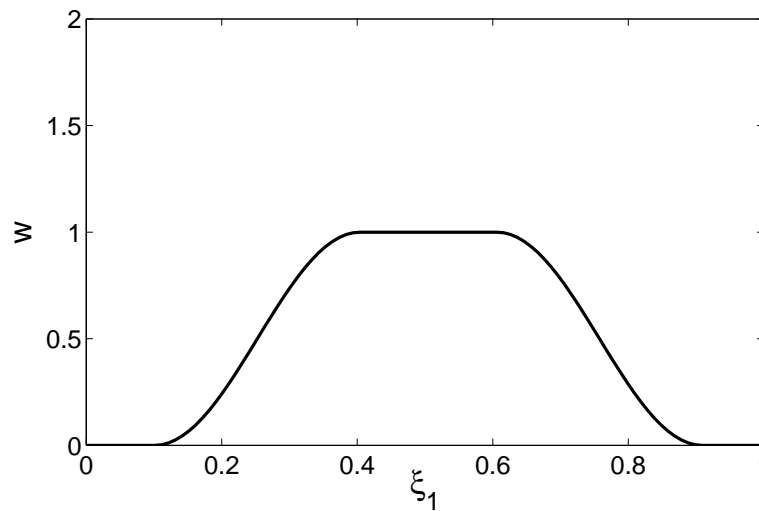


Figure 6.10 The weight function over the parametric domain using the trigonometric function presented in Equation (6.22).

6.4 Approximation properties

In this section, the approximation properties of the hybrid spline geometry are studied. The approximation of the normal vector and mean curvature is considered to gain a better understanding of the geometrical properties of the hybrid spline, and, indirectly, on the accuracy of the stress jump across the membrane of the deformable objects.

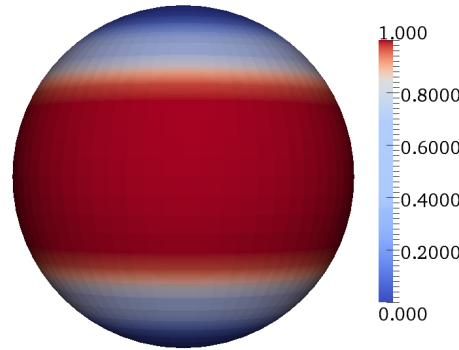
6.4.1 Normal vector

The normal vector in the hybrid geometry, \mathbf{n}_w , is computed as the normalized cross-product of the weighted average of the covariant basis vectors over the geometries A and B as defined in Equation (6.14). Figure 6.12 shows the relative L^2 -error (as defined in (3.29)) of the field of normal vectors, \mathbf{n}_w , versus the number of control points for the geometry A and the hybrid geometry. Note that in this global error measure the error of geometry B is identical to that of geometry A .

In the case of using quadratic B-spline ($p = 2$), it is observed that the relative L^2 -error of the normal vector converges asymptotically. The same rate of convergence is observed for the hybrid geometry as for the original geometries. Overall, the observed error in the hybrid geometry is slightly smaller than that of the original geometries. The advantage of the hybrid geometry is more pronounced when a quartic B-spline ($p = 4$) is used to create geometry A . In



(a)



(b)



(c)

Figure 6.11 The weight function w over (a) the parametric domain in terms of ξ_A , (b) the physical domain, and (c) the parametric domain in terms of ξ_B .

this case, no clear convergence is observed in Figure 6.12 (b), while the error is significantly reduced in the hybrid geometry. Upon closer inspection of the normal vector this behavior can be explained. In Figure 6.13, the first component of the normal vector (n_x) of the geometries A (a and b), B (c and d) and the hybrid geometry (e and f) is shown, both over the parameter and physical domains. All results are obtained using 288 elements and $p = 2$. Note that the other components of the normal vector (n_y and n_z) show similar behavior. Since the normal vector is accurately described over the geometries A and B using this mesh, overall no notable difference is observed between the original geometries

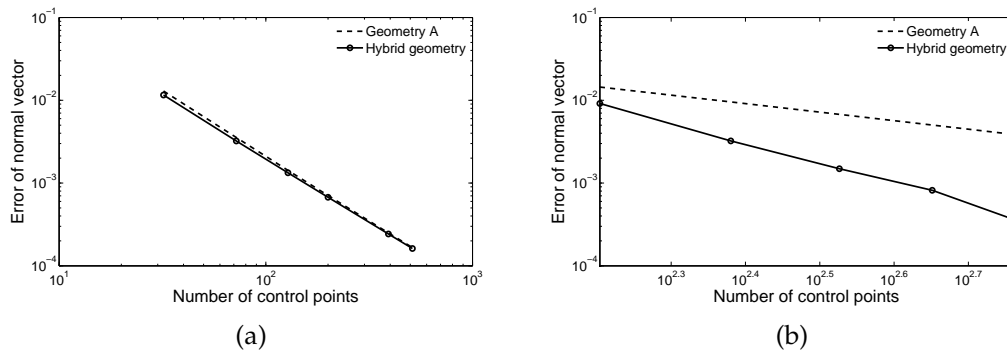


Figure 6.12 The relative L^2 -error of the normal vector of the spherical initial configuration for the geometry A and the hybrid geometry when (a) $p = 2$ (b) $p = 4$.

and the hybrid geometry. An exception to this is the value of n_x at the pole of geometry B , which, as seen is the plot is inaccurate as a consequence of the fact that it cannot be evaluated properly in this point. In the hybrid geometry, the evaluation of this point is circumvented. A refined mesh with a quartic ($p = 4$) hybrid geometry is shown in Figure 6.14. Obviously, similar results are obtained for this case.

6.4.2 Mean curvature

The mean curvature of the hybrid geometry is computed using Equation (6.16). The L^2 -error of the mean curvature as defined in Equation (3.29) is shown in Figure 6.15. Linear convergence behavior is observed for this error measure with $p = 2$. In accordance with the expectations, the same rate of convergence is observed for the hybrid geometry as for the original geometries. However, the error itself is smaller.

In the case of using a quartic B-spline, however, no clear convergence is observed for geometry A , which is in agreement with the behavior of the normal vector, the hybrid geometry shows a drastic error reduction in the computation of the mean curvature. To study this error further, in Figure 6.16 the mean curvature is plotted for the geometry A (a and b), B (c and d) and the hybrid geometry (e and f). All plots are constructed with 128 elements and $p = 2$. In the geometries A and B , significant errors are observed at the poles. In the hybrid geometry, these inaccuracies are eliminated.

The behavior of the curvature of the hybrid spline geometry under mesh refinement is studied in Figure 6.17. For the sake of comparison the same color bar range is used for all plots. The computed ranges of the mean curvature for

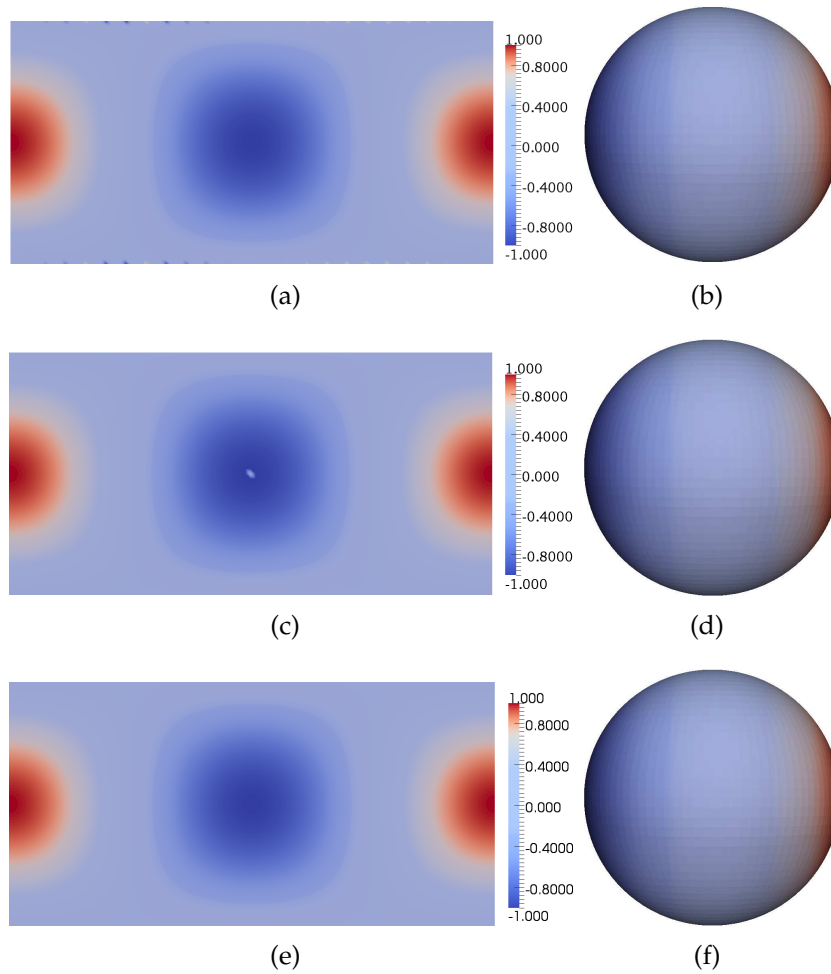


Figure 6.13 The first component of the normal vector (n_x) of geometry *A* (a and b), geometry *B* (c and d), and the hybrid geometry (e and f). All figures are based on 288 elements and $p = 2$.

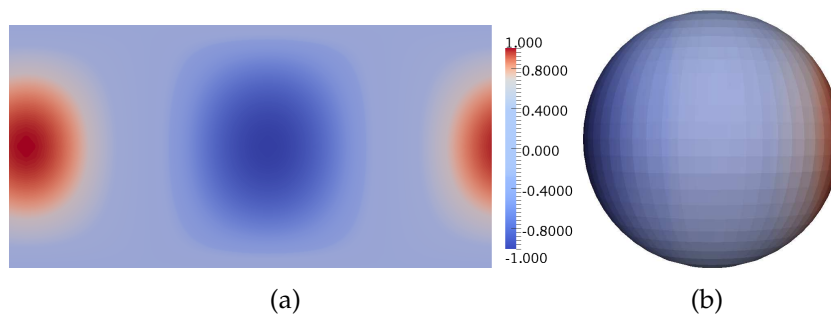


Figure 6.14 The first component of the normal vector (n_x) for the hybrid geometry with 336 elements and $p = 4$.

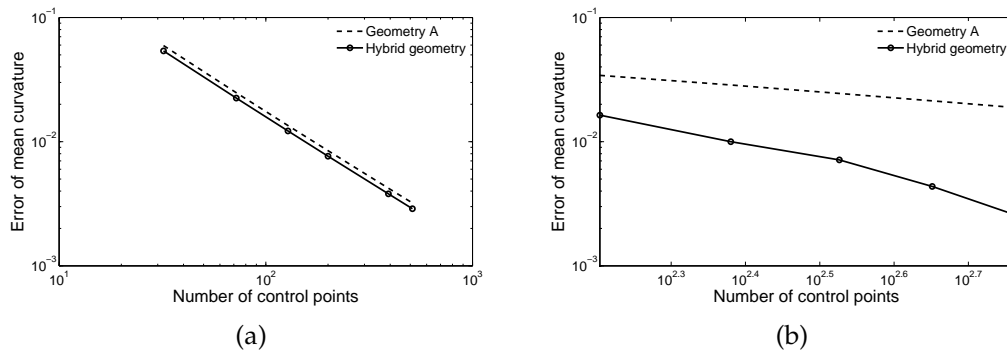


Figure 6.15 The relative L^2 -error of the mean curvature of the spherical initial configuration for geometry A and the hybrid geometry when (a) $p = 2$ (b) $p = 4$.

Table 6.1 Range of the mean curvature H for the hybrid geometry and geometry A (the points located at $\zeta_1 = 0$ and $\zeta_1 = 1$ are excluded) for various numbers of the elements.

Nr. elements	Range of H	Range of H_A
128	1.924 - 2.039	1.881 - 2.027
200	1.951 - 2.024	1.923 - 2.017
288	1.965 - 2.017	1.946 - 2.012

various numbers of elements for geometry A and hybrid geometry are collected in Table 6.1. Note that since the mean curvature is not defined at the poles, to compute this range for geometry A , the points located at $\zeta_1 = 0$ and $\zeta_1 = 1$ are excluded. Clear convergence of the curvature approximation is observed from both this table and the figures. Accurate results are also obtained when quartic B-splines are used, as shown in Figure 6.18.

In Figure 6.19, the curvature approximation using the two different weight functions introduced above is compared. For the simulations in subfigures (a) and (b) 128 elements and quadratic splines ($p = 2$) are used. No notable differences are observed for this case. For a hybrid geometry with 336 elements and quartic splines ($p = 4$), as shown in the subfigures (c) and (d), the accuracy of the mean curvature for the geometry created by the trigonometric weight function is more accurate than that of the B-spline weight functions. This difference results from the interaction between the weight function and the mean curvature inaccuracies in the neighborhood of the poles.

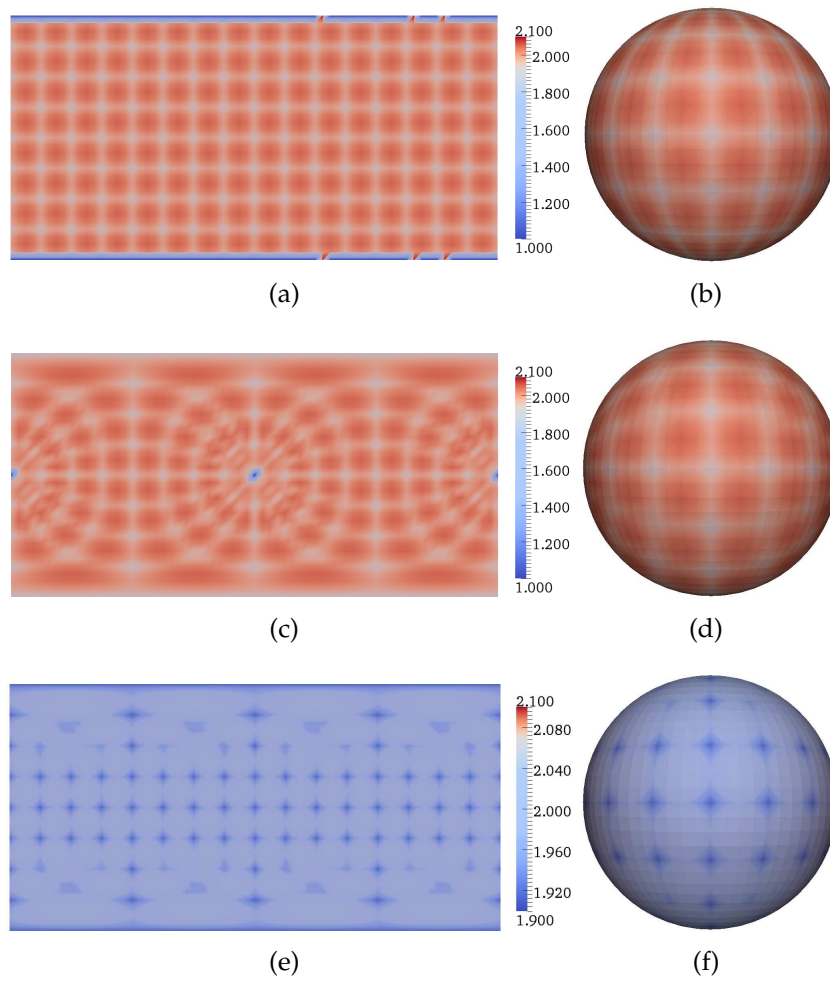


Figure 6.16 The mean curvature of geometry A (a and b), geometry B (c and d), and the hybrid geometry (e and f). All figures are based on 128 elements and $p = 2$.

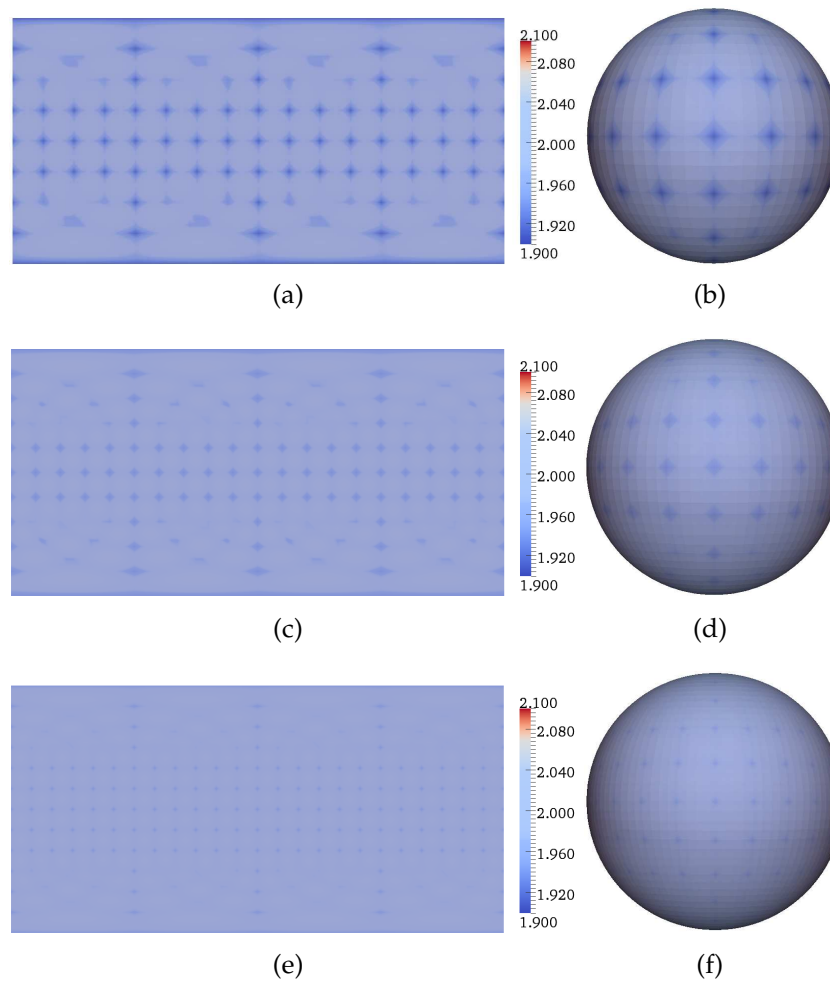


Figure 6.17 The mean curvature of the hybrid geometry with second-order B-splines ($p = 2$) for (a and b) 128 elements, (c and d) 200 elements, and (e and f) 288 elements.

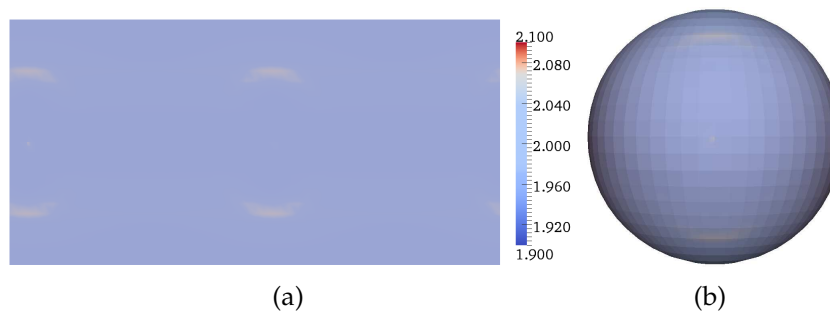


Figure 6.18 The mean curvature of the hybrid geometry with fourth-order B-splines ($p = 4$) and 336 elements.

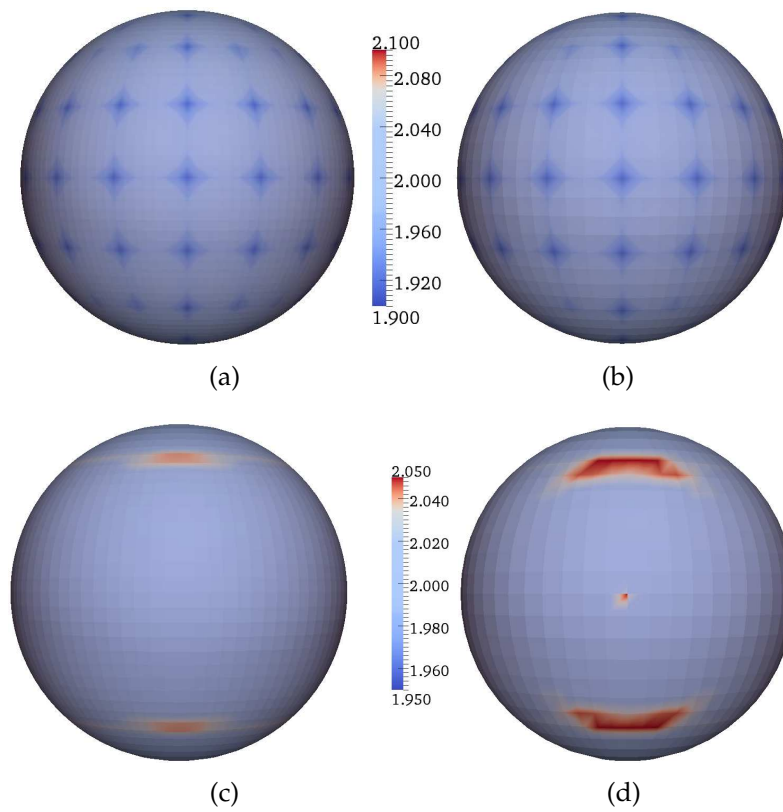


Figure 6.19 The mean curvature of the hybrid geometry with second-order B-splines ($p = 2$) with 128 elements using (a) the trigonometric weight function and (b) the B-spline weight function. Figures (c) and (d) show these weight functions for fourth-order B-splines ($p = 4$) with 336 elements.

6.5 Numerical results

In this section, the performance of the hybrid geometry to simulate the motion of the interface of the deformable objects using the isogeometric boundary integral method is studied. Since the collocation method is found to be advantageous compared to the L^2 -projection method (see Section 3.3.3), this method is used to approximate the velocity of the (hybrid) interface.

The collocation points for each geometry (A and B) should be taken as demonstrated in Section 3.3.3 while the convolution integral in Equation (2.25) is evaluated by means of the application of the hybrid geometry. Thus, the velocity of both geometries A and B are found using the modified curvature, and their control nets move separately.

6.5.1 Drop deformation

In this section, the deformation of an initially spherical drop suspended in shear flow is considered. The drop deformation is measured by means of the Taylor deformation parameter D as defined in Equation (4.1). Figure 6.20 shows a comparison between the Taylor deformation over time for different simulations of a drop using the individual geometries A and B and the hybrid geometry. An initially spherical mesh with 288 elements is considered in background shear flow. As expected, the same behavior as observed in Chapter 4 is visible from the presented graph for the hybrid geometry. The Taylor deformation starts from zero and reaches a stationary plateau. Although a slight oscillation exists in the deformation graph of the hybrid geometry, the final Taylor deformation approaches $D = 0.1091$, which corresponds well with the reported Taylor deformation in Table 4.2 where $D = 0.1094$.

Increasing the capillary number leads to larger deformations and more elongation of the interface. In the case of $Ca = 0.2$, as presented in Figure 6.21 (b), a stable deformed shape is obtained. In comparison with Figure 6.21 (a) with $Ca = 0.1$, this shape is more elongated and reaches $D = 0.2216$, which is slightly off from the reported Taylor deformation in Table 4.2 where $D = 0.2209$. Although this difference is small, see Figure 6.21 (b), further increasing of the capillary number leads to the simulation break-down.

In the case of larger deformations (bigger capillary numbers), the hybrid geometry does not give the same interface movement as the individual geometries A and B , despite the fact that the hybrid geometry decreases the error of the computed mean curvature, see Figure 6.17. The reason can be interpreted as follows. In Figure 6.22, a set of points is shown in the parametric domain and

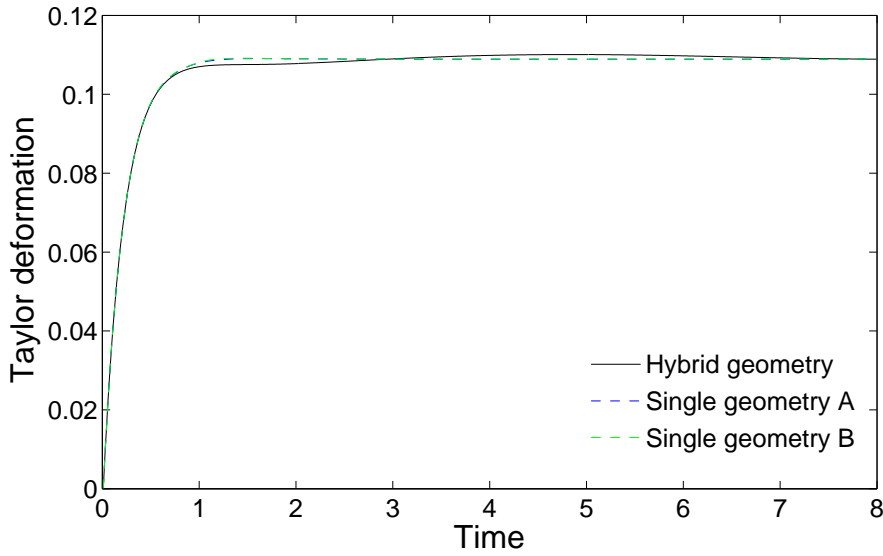


Figure 6.20 Taylor deformation of a drop with 288 elements versus time with $Ca = 0.1$ for the geometries *A* & *B* and the hybrid geometry.



Figure 6.21 Stable drop shape using the hybrid geometry with 288 elements for (a) $Ca = 0.1$ (b) $Ca = 0.2$.

its corresponding physical geometry. In the single geometry, the computed mean curvature for all of these points is the same. Since these points are located at the pole of geometry *B*, the weight function in this zone equals one and therefore in using the hybrid geometry, according to Equation (6.16), the mean curvature is taken from the corresponding points in geometry *A*. The data for the mean curvature of the points in this figure and their corresponding points on the hybrid geometry is listed in Table 6.2.

Although the accuracy of the computed mean curvature in the hybrid geometry is improved in comparison with the single geometry, an inequality between

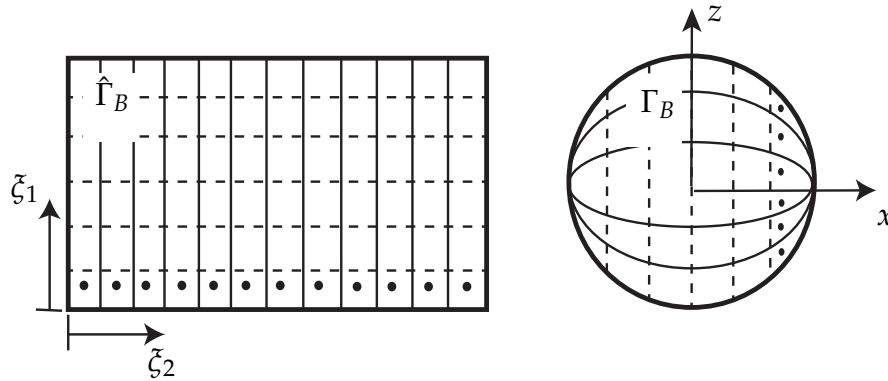


Figure 6.22 A set of points in the parametric domain of geometry B near the pole and their corresponding points at the physical domain.

the computed mean curvature is created. This discrepancy comes from the minimization process since for the identical points in the parametric domain of geometry B , similar (but not identical) corresponding points in the parametric domain of the geometry A exist. This difference results in the perturbation in the computation of the stress discontinuity and hence the velocity of the mentioned points. Subsequently, these disturbed velocities lead to irregular movement of the control points, particularly those close to the pole. This irregular behavior can also be extended to the movement of the other control net (geometry A). This behavior is clearly shown in Figure 6.23 for a cell with 72 elements. The movement of the control net of the single geometry B in shear flow at $t = 1.2$ is depicted in Figure 6.23 (a). The points closed to the pole in this geometry move as expected. But, as is shown in Figure 6.23 (b), in the case of using the hybrid geometry, this movement is no longer appropriate due to the disturbance of the mean curvatures for these points. In the case of small deformations ($Ca < 0.25$), this effect can be controlled by increasing the number of elements.

6.5.2 Inextensible membrane deformation

In this section, the deformation of an initially ellipsoidal cell suspended in shear flow is considered, while the geometrical setup of the cell is the same as used in Chapter 5. Figure 6.24 shows the typical motion of an inextensible membrane in an imposed shear flow where the axes ratio $r = 2$ and initial inclination angle $\theta_0 = 0$. The cell deforms while the membrane rotates around its interior. This observation is in agreement with the results reported in Chapter 5. In addition, the tension distribution over the equilibrium shape of the cell for an initial ellipsoid with $r = 2$ is plotted for geometries A and B in Figure 6.25 (Note that

Table 6.2 Comparison of the computed mean curvature for a set of points near the pole between the hybrid geometry and single geometry B .

Point Id.	The hybrid geometry	The single geometry
1	2.0098	2.0556
2	2.0499	2.0556
3	2.0098	2.0556
4	2.0098	2.0556
5	2.0499	2.0556
6	2.0098	2.0556
7	2.0098	2.0556
8	2.0499	2.0556
9	2.0098	2.0556
10	2.0098	2.0556
11	2.0499	2.0556
12	2.0098	2.0556

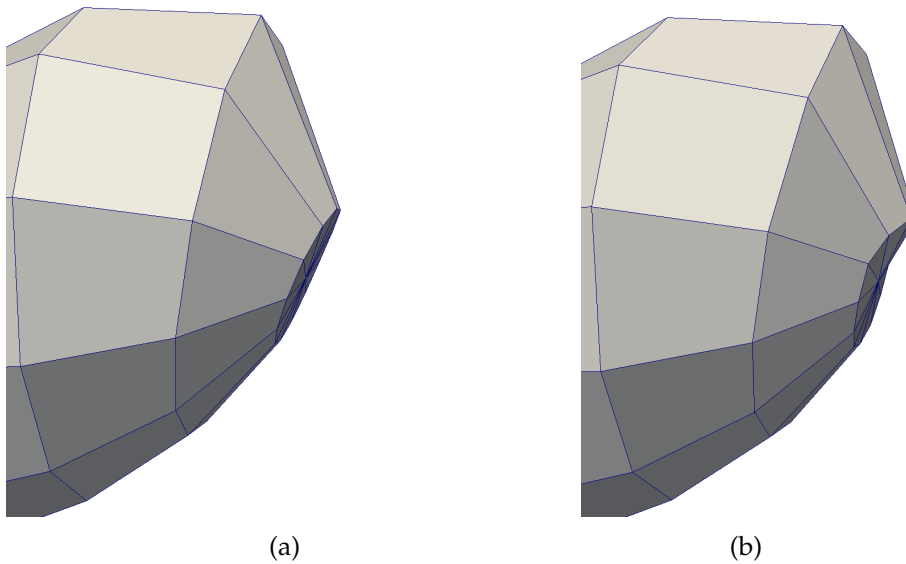


Figure 6.23 A side view snapshot of the control net (with 96 points) near the pole of (a) individual geometry B (b) geometry B used to move the hybrid geometry.

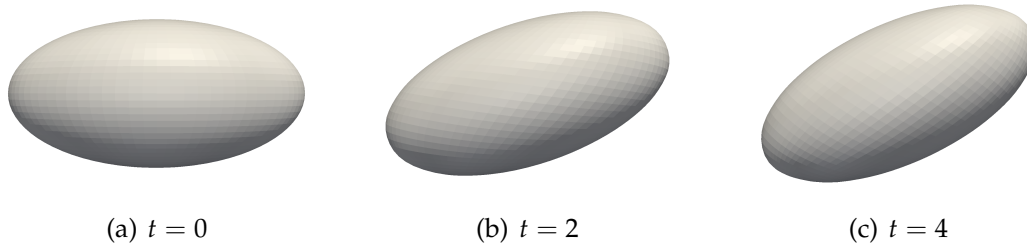


Figure 6.24 The motion of an inextensible membrane using the hybrid geometry at different times for a cell with axes ratio $r = 2$, inclination angle $\theta_0 = 0$ and 128 elements.

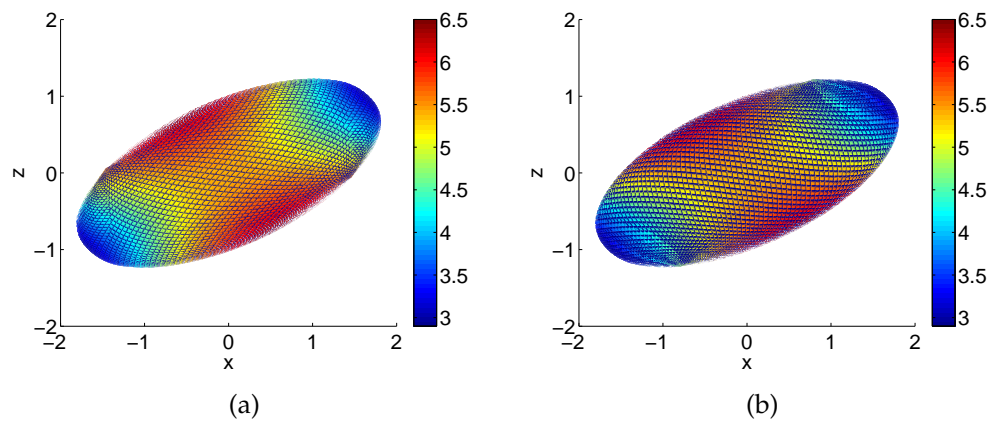


Figure 6.25 Tension distribution over the surface of the cell in its stable shape computed with 128 elements for (a) geometry *A* (b) geometry *B*.

since the tension is implicitly solved over the control points, separated surface tensions for geometries *A* and *B* are needed). It is shown that this distribution is in good agreement ($\tau_{\max} = 6.15$) with the results shown in Figure 5.7 where the maximum tension is $\tau_{\max} = 6.37$.

Although the behavior of the inextensible membrane using the hybrid geometry for a cell with axes ratio $r = 2$ is in agreement with the previously reported observations in Chapter 5, the surface area changes from initially $A_0 = 21.512$ to the final $A = 20.924$ (2.7 % deviation from the initial value). This difference will be intensified when the axes ratio increases and thus the deformation grows. This can be attributed to the same reason causing the drop deformation failure using the hybrid geometry as explained in Section 6.5.1, where despite of the error reduction in computing the mean curvature, the accuracy of the computed surface tension is not improved.

6.5.3 Vesicle deformation

More realistic simulation of a vesicle (and also a red blood cell) requires computation of the bending force across the membrane, which includes the higher-order derivative of the mean curvature. As discussed in Section 6.1, it is expected that the hybrid geometry reduces the error of computing the Laplace-Beltrami of the mean curvature. Unfortunately, the problems encountered with the simulation of the drop and inextensible membrane using the hybrid geometry hindered the application of the hybrid geometry concept to the simulation of a vesicle.

6.6 Conclusions

Computation of higher-order derivatives of geometrical quantities such as the mean curvature is required for a realistic red blood cell model. To alleviate the problems associated with the C^0 -continuity of the poles of the splines considered in the previous chapters, a hybrid geometry is suggested which is capable of accurately computing higher-order geometric derivatives.

The results obtained using the hybrid geometry show a drastic error reduction in computing the mean curvature. Nevertheless, it is also found that the equality between the points of each geometry with identical mean curvature is affected. Although this influence contributes to the reduction of the error of the computed mean curvature, this disturbance restricts the current capabilities of the hybrid geometry in the simulation of the deformable objects. It is shown that the hybrid geometry approach properly works in limited cases for the drop and inextensible membrane. Evidently, to generalize this approach, further improvements that resolve the mentioned problems are essential.

Conclusions and recommendations

7.1 Conclusions

The Boundary Integral Method (BIM) is known to be a powerful method to model the deformation of deformable objects (drops, capsules, vesicles, red blood cells, etc). A major advantage of this method above of the technique is that a discretization is only required at the interface, which is a result of the possibility to recast the full three-dimensional Stokes problem to a two-dimensional surface problem. Triangular elements are often used to discretize the surface of the deformable objects. In such a traditional boundary integral formulation, the surface is parametrized by the definition of piece-wise continuous, but non-smooth, basis functions over the elements. Evidently, the mean curvature – which has an essential contribution in the stress discontinuity at the interface of the deformable objects – is not defined over the element boundaries, and as a consequence different methods have been suggested to reconstruct the mean curvature. This reconstruction can lead to inaccuracies in the computation of the stress jump term, which affects the capability of the method to reliably simulate the motion of the deformable objects. This weakness is more pronounced when higher-order derivatives of the geometry, such as the Laplacian of the mean curvature, are involved in the stress discontinuity at the surface. This is particularly the case when the method is applied to more complex objects such as red blood cells or vesicles.

In this thesis, isogeometric analysis is proposed as a methodology to overcome the accuracy problems of the traditional boundary integral formulation. Using isogeometric analysis, the interface of the deformable object is described using higher-order continuous B-splines. The boundary integral formulation is then employed to compute the motion of the associated control net. The spline basis functions used in this isogeometric BIM are generally smooth over the parameter domain and allows for a unique definition of the surface gradients over the

complete surface of the objects (with the exception of a few special points). This advantage results in the direct evaluation of the surface normals, mean curvatures and higher-order surface derivatives in the formulation.

As a first example of using IGA-BIM, the deformation of a drop suspended in shear flow is computed. Efficient numerical evaluation of the convolution integral, which involves singular Green's functions, is achieved by means of a singularity subtraction technique. The motion of the drop is determined by a stepwise computation of the control points velocity using either an L^2 -projection or a collocation method. The computation of the velocity field over the drop surface using the collocation method was found to be superior in terms of computational effort and stability compared to the computation using the L^2 -projection.

As a second example, IGA-BIM is used for the simulation of the motion of an inextensible membrane, which can be considered as a preliminary model for a red blood cell. This application is a non-trivial extension of the drop case due to the presence of the pointwise surface inextensibility condition. An isogeometric boundary integral formulation based on the collocation method is introduced to implicitly obtain the surface tension needed to keep the surface area of the cell unchanged. Using this computed surface tension, the IGA-BIM is used to determine the motion of the object. IGA-BIM has been successfully applied to the surface velocity computation of a large variety of three-dimensional shapes, without the application of additional stabilization methods. Furthermore, the results obtained from an initially three-dimensional biconcave shape in shear flow demonstrates the capability of the isogeometric framework to capture some of the fundamental physics of a red blood cell.

The most prominent difference between the model for a red blood cell and an inextensible membrane is the existence of a stress jump term related to the Laplacian of the mean curvature in the case of a red blood cell, which can be interpreted as a fourth order derivative of the position at the interface. Upon appropriate selection of the spline order ($p \geq 4$), these derivatives are well defined everywhere, with the exception of the poles. Although the poles are not dramatically affecting the simulation procedure in the cases of the drop and inextensible membrane, they do result in irregular deformations in the case of a red blood cell, which ultimately leads to simulation break-down. Evidently, these problems can be attributed to inaccuracies arising from the computed Laplacian of the mean curvature at the poles, which happen because of the non-smoothness of the basis functions at these points.

Using a hybrid geometry representation, the inaccuracies of the computed mean curvature are shown to be drastically reduced. This geometry is defined as the weighted average of two parametrizations of the same geometry. By

appropriate selection of the weighting function, the negative influence of the poles can be eliminated. Although the errors in the computation of the geometric derivatives are reduced using the hybrid discretization, additional perturbations are introduced which negatively effect the stability of the current formulation. Further developments on the hybrid geometry formulation are therefore needed to exploit its full potential.

7.2 Recommendations

The main objective of this thesis was to simulate the movement of the interface of viscous drops and capsules with different complexities of their membranes. However, due to complications, some assumptions and simplifications have been made. Below, some recommendations for future work are given.

- Obviously, in this thesis the incorporation of the viscosity ratio is neglected and all the simulations are restricted to the iso-viscous case. This assumption simplifies the integration terms and therefore time expense to find the velocity. Although this assumption covers a large range of applications of microfluidic devices, having a contrast between the viscosities of the internal and external fluids generalizes the application of the simulations and confirms the capabilities of the IGA-BIM.
- A singularity subtraction technique is implemented in the case of drop deformation. As shown in Chapter 4, this drastically improved the efficiency of the integration. Unfortunately, due to the presence of a tangential stress jump term, a straightforward application of this subtraction technique is not possible in the case of the inextensible membrane. It is anticipated that the integration inaccuracies due to the singular integration negatively influence the stability of the IGA-BIM. A solution to this problem was recently proposed in [132], with a singularity subtraction technique that incorporates the tangential stress jump. It is highly recommended to incorporate this technique to dramatically improve the accuracy of the current method.
- In the current thesis, forward Euler time stepping is used to move the interface in time. Application of implicit time integration is anticipated to significantly improve the robustness of the simulations.

References

- [1] G. I. Taylor. The formation of emulsions in definable fields of flow. *Proc. R. Soc. London* 136:501–523, **1934**.
- [2] S. Torza, R. G. Cox, and S. J. Mason. Particle motions in sheared suspensions: Xxvii. transient and steady deformation and burst of liquid drops. *Journal of Colloid Science* 38:395–411, **1972**.
- [3] W. Bartok and S. J. Mason. Particle motions in sheared suspensions: Viii. singlets and doublets of fluid spheres. *Journal of Colloid Science* 14:13–26, **1959**.
- [4] B.J. Bentley and L.G. Leal. An experimental investigation of drop deformation and breakup in steady, two-dimensional linear flows. *J. Fluid Mech.* 167:241–283, **1986**.
- [5] H.A. Stone. Dynamics of drop deformation and breakup in viscous fluids. *Annu. Rev. Fluid Mech.* 26:65–102, **1994**.
- [6] R. Lipowsky and E. Sackmann. *Structure and dynamics of membranes, from cells to vesicles*. North-Holland, **1995**.
- [7] C. Pozrikidis. *Modeling and Simulation of Capsules and Biological Cells*. CRC Press, **2003**.
- [8] T. W. Secomb. Mechanics of red blood cells and blood flow in narrow tubes. In C. Pozrikidis, editor, *Modeling and Simulation of Capsules and Biological Cells* 163–196, **2003**.
- [9] P.B. Canham. The minimum energy of bending as a possible explanation of the biconcave shape of the human red blood cell. *J. Theor. Biology* 26:77–81, **1970**.
- [10] E.A. Evans. Bending resistance and chemically induced moments in membrane bilayers. *Biophys. J.* 14:923–931, **1974**.
- [11] T. Fischer, M. Stohr-Lissen, and H. Schmid-Schonbein. The red cell as a fluid droplet: tank tread-like motion of the human erythrocyte membrane in shear flow. *Science* 202:894–896., **1978**.
- [12] R. Skalak and P.I. Branemark. Deformation of red blood cells in capillaries. *Science* 164:717–719, **1969**.
- [13] V. Kantsler and V. Steinberg. Transition to tumbling and two regimes of tumbling motion of a vesicle in shear flow. *Phys. Rev. Lett.* 96:036001, **2006**.
- [14] M. Abkarian, C. Lartigue, and A. Viallat. Tank treading and unbinding of deformable vesicles in shear flow: Determination of the lift force. *Phys. Rev. Lett.* 88:068103, **2002**.
- [15] C. Pozrikidis. *Boundary-Integral and Singularity Methods for Linearized Viscous Flow*. Cambridge U.P., **1992**.

- [16] J. M. Rallison and A. Acrivos. A numerical study of the deformation and burst of a viscous drop in an extensional flow. *J. Fluid Mech.* 89:191–200, **1987**.
- [17] L. Gaul, M. Kogel, and M. Wagner. *Boundary Element Methods for Engineers and Scientists*. Springer, **2002**.
- [18] V. Cristini, J. Blawdziewicz, and M. Loewenberg. Drop breakup in three-dimensional viscous flows. *Phys. Fluids* 10:1781, **1998**.
- [19] V. Cristini, J. Blawdziewicz, and M. Loewenberg. An adaptive mesh algorithm for evolving surfaces: Simulation of drop breakup and coalescence. *J. Comput. Phys.* 168:445, **2001**.
- [20] A. Chesters. The modelling of coalescence of fluid-liquid dispersions: a review of current understanding. *Trans. Inst. Chem. Engrs. Part A* 69:259, **1991**.
- [21] P.J.A. Janssen, P.D. Anderson, G.W.M. Peters, and H.E.H. Meijer. Axisymmetric boundary integral simulations of film drainage between two viscous drops. *J. Fluid Mech.* 567:65, **2006**.
- [22] A.J. Griggs, A.Z. Zinchenko, and R.H. Davis. Low-reynolds-number motion of a deformable drop between two parallel plane walls. *Int. J. Multiphase Flow* 33:182, **2007**.
- [23] I. Bazhlekov, P.D. Anderson, and H.E.H. Meijer. Nonsingular boundary integral method for deformable drops in viscous flows. *Phys. Fluids* 16:1064–1081, **2004**.
- [24] P.J.A. Janssen and P.D. Anderson. Boundary-integral method for drop deformation between parallel plates. *Phys. Fluids* 19:043602, **2007**.
- [25] Hua Zhou and C. Pozrikidis. Deformation of liquid capsules with incompressible interfaces in simple shear flow. *J. Fluid Mech.* 283:175–200, **1995**.
- [26] B. Kaoui, G.H. Ristow, I. Cantat, C. Misbah, and W. Zimmermann. Lateral migration of a two-dimensional vesicle in unbounded poiseuille flow. *Phys. Rev. E* 77:021903, **2008**.
- [27] T.J.R. Hughes, J.A. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry, and mesh refinement. *Comput. Method Appl. M.* 194:4135–4195, **2005**.
- [28] J.A. Cottrell, T.J.R. Hughes, and Y. Bazilevs. *Isogeometric Analysis Toward Integration of CAD and FEA*. Wiley, **2009**.
- [29] C. Politis, A.I. Ginnis, P.D. Kaklis, and C. Feurer K. Belibassakis. An isogeometric beam for exterior potential-flow problems in the plane. In *proceeding of: Proceedings of the 2009 ACM Symposium on Solid and Physical Modeling, San Francisco, California, USA*.
- [30] T. Takahashi and T. Matsumoto. An application of fast multipole method to isogeometric boundary element method for laplace equation in two dimensions. *Eng. Anal. Bound. Elem.* 36:1766–1775, **2012**.
- [31] R.N. Simpson, S.P.A. Bordas, J. Trevelyan, and T. Rabczuk. A two-dimensional isogeometric boundary element method for elastostatic analysis. *Comput. Methods Appl. Mech. Engrg.* 209:87–100, **2012**.
- [32] L. Heltai, M. Arroyo, and A. DeSimone. Nonsingular isogeometric boundary element method for stokes flows in 3d. *Comput. Method. Appl. M.* 268:514–539, **2014**.
- [33] G.W. Stachowiak and A.W. Batchelor. *Engineering Tribology*. Elsevier Butterworth-Heinemann, **1967**.

- [34] S. Succi. *The Lattice Boltzmann Equation for Fluid Dynamic and Beyond*. Oxford University Press, 2001.
- [35] F. Janoschek, F. Toschi, and J. Harting. Simulations of blood flow in plain cylindrical and constricted vessels with single cell resolution. *Macromol. Theor. Simul.* 20:562–570, 2011.
- [36] J. Wu and C.K. Aidun. Simulating 3d deformable particle suspensions using lattice boltzmann method with discrete external boundary force. *Int. J. Numer. Meth. Fl.* 62:765–783, 2010.
- [37] M.M. Dupin, I. Halliday, and C.M. Care L.L. Munn. Lattice boltzmann modelling of blood cell dynamics. *Int. J. Comput. Fluid D.* 22:481–492, 2008.
- [38] K. Tsubota, S. Wada, and T. Yamaguchi. Particle method for computer simulation of red blood cell motion in blood flow. *Comput. Meth. Prog. Bio.* 83:139–146, 2006.
- [39] K. Tsubota, S. Wada, and T. Yamaguchi. Simulation study on effects of hematocrit on blood flow properties using particle method. *J. Biomech. Sci. Eng.* 1:159–170, 2006.
- [40] C.W. Hirt and B.D. Nichols. Volume of fluid (vof) method for the dynamics of free boundaries. *J. Comput. Phys.* 39:201–225, 1981.
- [41] D.J.E. Harvie, M.R. Davidson, J.J. Cooper-White, and M. Rudman. A parametric study of droplet formation through a microfluidic contraction: Low viscosity newtonian droplets. *Chem. Eng. Sci.* 46:150–166, 2006.
- [42] Y. Renardy. The effects of confinement and inertia on the production of droplets. *Rheol. Acta* 46:521–529, 2007.
- [43] M. Rudman. A volume-tracking method for incompressible multifluid flows with large density variations. *Int. J. Numer. Methods Fluids* 28:357–378, 1998.
- [44] S. Osher and J.A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. *J. Comput. Phys.* 79:12–49, 1988.
- [45] M. Sussman, P. Smereka, and S. Osher. A level set approach for computing solutions to incompressible two-phase flow. *J. Comput. Phys.* 114:146–159, 1994.
- [46] M. Sussman and Puckett. E. A coupled level set and volume-of-fluid method for computing 3d and axisymmetric incompressible two-phase flows. *J. Comput. Phys.* 162:301–307, 2000.
- [47] J.U. Brackbill, D.B. Kothe, and C. Zemach. A continuum method for modeling surface tension. *J. Comput. Phys.* 100:335–354, 1992.
- [48] D.M. Anderson, G.B. McFadden, and A.A. Wheeler. Diffuse-interface method in fluid mechanics. *Ann. Rev. Fluid Mech.* 30:139–165, 1998.
- [49] A. Esmaeeli and G. Tryggvason. Direct numerical simulations of bubbly flows. part 1. low reynolds number arrays. *J. Fluid Mech.* 377:313–345, 1998.
- [50] A. Esmaeeli and G. Tryggvason. Direct numerical simulations of bubbly flows. part 2. moderate reynolds number arrays. *J. Fluid Mech.* 385:325–358, 1999.
- [51] S. Unverdi and G. Tryggvason. A front-tracking method for viscous, incompressible multi-fluid flows. *J. Comput. Phys.* 100:25–37, 1992.
- [52] F. H. Harlow and J. E. Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Phys. Fluids* 12:2182, 1965.

- [53] B. J. Daly. Numerical study of two fluid rayleigh-taylor instability. *Phys. Fluids* 10:297, **1967**.
- [54] O. Kabasa, H. K. Celik, A. Ozmerzi, and I. Akinci. Drop test simulation of a sample tomato with finite element method. *J. Sci. Food Agric.* 88:1537–1541, **2008**.
- [55] D Zhou, P. Yue, and J. J. Feng. Viscoelastic effects on drop deformation in a converging pipe flow. *J. Rheol.* 52:469–487, **2008**.
- [56] L. Nabors, A.J Baumgartner, S. Janke, J. Rose, W.J. Wagner, and R. Capen. Red blood cell orientation in pulmonary capillaries and its effect on gas diffusion. *J. Appl. Physiol.* 94:1634–1640, **2003**.
- [57] P.J.A. Janssen and P.D. Anderson. Modeling film drainage and coalescence of drops in a viscous fluid. *Macromol. Mater. Eng.* 296:238–248, **2011**.
- [58] H. Zhaoa, A.H.G. Isfahani, L.N. Olsonc, and J.B. Freunda. A spectral boundary integral method for flowing blood cells. *J. Comput. Phys.* 229:3726–3744, **2010**.
- [59] H. Zhao, E.S.G. Shaqfeh, and V. Narsimhan. Shear-induced particle migration and margination in a cellular suspension. *Phys. Fluids* 24:011902, **2012**.
- [60] C. Coull and C. Pozrikidis. Motion of an array of drops through a cylindrical tube. *J. Fluid Mech.* 358:1–28, **1998**.
- [61] S.K. Veerapaneni, A. Rahimian, G. Biroso, and D. Zorin. A fast algorithm for simulating vesicle flows in three dimensions. *J. Comput. Phys.* 230:5610–5634, **2011**.
- [62] M.E. Staben, K.P. Galvin, and R.H. Davis. Low-reynolds-number motion of a heavy sphere between two parallel plane walls. *Chem. Eng. Sci.* 61:1932–1945, **2006**.
- [63] A.Z. Zinchenko M.E. Staben and R.H. Davis. Dynamic simulation of spheroid motion between two parallel plane walls in low-reynolds-number poiseuille flow. *J. Fluid Mech.* 553:187–226, **2006**.
- [64] M.E. Staben, A.Z. Zinchenko, and R.H. Davis. Motion of a particle between two parallel plane walls in low-reynolds-number poiseuille flow. *Phys. Fluids* 15:1711, **2003**.
- [65] D.F. Zhang and H.A. Stone. Drop formation in viscous flows at a vertical capillary tube. *Phys. Fluids* 9:2234, **1997**.
- [66] A.Z. Zinchenko and R.H. Davis. A boundary-integral study of a drop squeezing through interparticle constrictions. *J. Fluid Mech.* 564:227–266, **2006**.
- [67] I. Stakgold. *Boundary Value Problems of Mathematical Physics*. Macmillan, **1968**.
- [68] O.D. Kellogg. *Foudations of Potential Theory*. Dover, **1954**.
- [69] A.E.H. Love. *A Treatise on the Mathematical Theory of Elasticity*. Dover, **1944**.
- [70] H.A. Lorentz. Ein allgemeiner satz, die bewegung einer reibenden flussigkeit betreffend, nebst einigen anwendungen desselben (a general theorem concerning the motion of a viscous fluid and a few consequences derived from it). *Abhand. Theor. Phys.* 1:23–42, **1907**.
- [71] J.R. Blake. A note on the image system for a stokeslet in a no-slip boundary. *Pcps-P. Camb. Philol. S.* 70:303–310, **1971**.
- [72] N. Liron and S. Mochon. Stokes flow for a stokes-let between 2 parallel flat plates. *J. Eng. Math.* 10:287–303, **1976**.

- [73] R.B. Jones. Spherical particle in poiseuille flow between planar walls. *J. Chem. Phys.* 121:483–500, **2004**.
- [74] M. Bloor and M. Wilson. Method for efficient shape parameterization of fluid membranes and vesicles. *Phys. Rev. E* 61:4218–4229, **2000**.
- [75] W. Helfrich. Elastic properties of lipid bilayers: theory and possible experiments. *Z. Naturforsch. C* 28:693–703, **1973**.
- [76] Z. Ou-Yang. Anchor ring-vesicle membranes. *Phys. Rev. A* 41:4517–4520, **1990**.
- [77] Z. Ou-Yang and W. Helfrich. Bending energy of vesicle membranes: General expressions for the first second and third variation of the shape energy and applications to spheres and cylinders. *Phys. Rev. E* 39:5280–5288, **1989**.
- [78] U. Seifert. Configurations of fluid membranes and vesicles. *Adv. Phys.* 46:13–137, **1997**.
- [79] U. Seifert, K. Berndl, and R. Lipowsky. Shape transformations of vesicles: Phase diagram for spontaneous curvature and bilayer-coupling models. *Phys. Rev. A* 44:1182–1202, **1991**.
- [80] M.P. Do Carmo. *Differential Geometry of Curves and Surfaces*. New Jersey: Prentice Hall, **1976**.
- [81] S. Svetina, D. Kuzman, R.E. Waugh, and B. Zihlerl, P. Zeks. The cooperative role of membrane skeleton and bilayer in the mechanical behaviour of red blood cells. *Bioelectrochemistry* 62:107–113, **2004**.
- [82] M. Lowenberg and E.J. Hinch. Numerical simulation of a concentrated emulsion in shear flow. *J. Fluid Mech.* 321:395–419, **1996**.
- [83] A.Z. Zinchenko, M.A. Rother, and R.H. Davis. A novel boundary-integral algorithm for viscous interaction of deformable drops. *Phys. Fluids* 9:1493–1511, **1997**.
- [84] S. Ramanujan and C. Pozrikidis. Deformation of liquid capsules enclosed by elastic membranes in simple shear flow: Large deformations and the effect of fluid viscosities. *J. Fluid Mech.* 361:117–143, **1998**.
- [85] Y. Sui, Y.T. Chew, P. Roy, and H.T. Low. A hybrid method to study flow-induced deformation of three-dimensional capsules. *J. Comput. Phys.* 227:6351–6371, **2008**.
- [86] H. Lin, Q. Hu, and Y. Xiong. Consistency and convergence properties of the isogeometric collocation method. *Comput. Method. Appl. M.* 267:471–486, **2013**.
- [87] M. G. Cox. The numerical evaluation of b-splines. *J. Inst. Math. Appl.* 10:134–147, **1972**.
- [88] C. De Boor. On calculating with b-splines. *J. Approx. Theory* 6:50–62, **1972**.
- [89] M. J. Borden, M. A. Scott, J. A. Evans, and T. J. R. Hughes. Isogeometric finite element data structures based on bézier extraction of NURBS. *Int. J. Numer. Meth. Eng.* 87:15–47, **2011**.
- [90] M. A. Scott, M. J. Borden, C. V. Verhoosel, T. W. Sederberg, and T. J. R. Hughes. Isogeometric finite element data structures based on bézier extraction of t-splines. *Int. J. Numer. Method. Eng.* 88:126–156, **2011**.
- [91] L. Beirão da Veiga, C. Lovadina, and A. Reali. Avoiding shear locking for the timoshenko beam problem via isogeometric collocation methods. *Comput. Method. Appl. M.* 241-244:38–51, **2012**.
- [92] F. Auricchio, L. Beirão da Veiga, J. Kiendl, C. Lovadina, and A. Reali. Locking-free isogeometric collocation methods for spatial timoshenko rods. *Comput. Method. Appl. M.* 263:113–126, **2013**.

- [93] D. Schillinger, J.A. Evans, A. Reali, M.A. Scott, and T.J.R. Hughes. Isogeometric collocation: Cost comparison with galerkin methods and extension to adaptive hierarchical nurbs discretizations. *Comput. Method. Appl. M.* 267:170–232, **2013**.
- [94] G. Farin. *Curves and Surfaces for Computer Aided Geometric Design*. Academic Press, San Diego, **1997**.
- [95] R.N. Simpson, S.P.A. Bordas, J. Trevelyan, and T. Rabczuk. A two-dimensional isogeometric boundary element method for elastostatic analysis. *Comp. Meth. Appl. Mech. Eng.* 209:87–100, **2012**.
- [96] G. I. Taylor. The viscosity of a fluid containing small drops of another fluid. *Proc. R. Soc. London* 138:41–48, **1932**.
- [97] Y. Hu and A. Lips. Estimating surfactant surface coverage and decomposing its effect on drop deformation. *Phys. Rev. Lett.* 91:044501, **2003**.
- [98] L. G. Leal. Flow induced coalescence of drops in a viscous fluid. *Phys. Fluids* 16:1833–1851, **2004**.
- [99] H. Yang, C. C. Park, Y. T. Hu, and L. G. Leal. The coalescence of two equal-sized drops in a two-dimensional linear flow. *Phys. Fluids* 13:1087–1106, **2001**.
- [100] Y. Yoon, M. Borell, C. C. Park, and L. G. Leal. Viscosity ratio effects on the coalescence of two equal-sized drops in a two-dimensional linear flow. *J. Fluid Mech.* 525:355–379, **2005**.
- [101] W. Bartok and S. G. Mason. Particle motions in sheared suspensions: Viii. singlets and doublets of fluid spheres. *J. Colloid Sci.* 14:13–26, **1959**.
- [102] S. Torza, R. G. Cox, and S. G. Mason. Particle motions in sheared suspensions xxvii. transient and steady deformation and burst of liquid drops. *J. Colloid Interface Sci.* 38:395–411, **1972**.
- [103] C. L. Tucker III and P. Moldenaers. Microstructural evolution in polymer blends. *Annu. Rev. Fluid Mech.* 34:177–210, **2002**.
- [104] R. G. Cox. The deformation of a drop in a general time-dependent fluid flow. *J. Fluid Mech.* 37(3):601–623, **1969**.
- [105] D. BarthegevesBiesel and A. Acrivos. Deformation and burst of a liquid droplet freely suspended in a linear shear field. *J. Fluid Mech.* 61(01):1–22, **1973**.
- [106] G. I. Taylor. Conical free surfaces and fluid interfaces. *Proc. 11 Int. Congr. Appl. Mech. Munich* 790–796, **1964**.
- [107] A. Acrivos and T. S. Lo. Deformation and breakup of a single slender drop in an extensional flow. *J. Fluid Mech.* 86:641–672, **1978**.
- [108] E. J. Hinch and A. Acrivos. Long slender drops in a simple shear flow. *J. Fluid Mech.* 98:305–328, **1980**.
- [109] D. V. Khakhar and J. M. Ottino. Deformation and breakup of slender drops in linear flows. *J. Fluid Mech.* 166:265–285, **1986**.
- [110] J. M. Rallison. The deformation of small viscous drops and bubbles in shear flows. *Ann. Rev. Fluid Mech.* 16:45–66, **1984**.
- [111] V. Cristini, J. Blawdziewicz, M. Loewenberg, and L. F. Collins. Breakup in stochastic stokes flows: sub-kolmogorov drops in isotropic turbulence. *J. Fluid Mech.* 492:231–250, **2003**.

- [112] A. Z. Zinchenko, M. A. Rother, and R. H. Davis. Cusping, capture, and breakup of interacting drops by a curvatureless boundary-integral algorithm. *J. Fluid Mech.* 391:249–292, **1999**.
- [113] P. J. A. Janssen and P. D. Anderson. A boundary-integral model for drop deformation between two parallel plates with non-unit viscosity ratio drops. *J. Comput. Phys.* 227(20):8807–8819, **2008**.
- [114] H.E.H. Meijer, J.M.H. Janssen, and P.D. Anderson. Mixing of immiscible liquids. In: Ica Manas, editor, *Mixing and Compounding of Polymers: Theory and Practice*, 41–177. Hanser Publications, **2009**.
- [115] M. Loewenberg and E. J. Hinch. Collision of two deformable drops in shear flow. *J. Fluid Mech.* 338:299–315, **1997**.
- [116] E. Klaseboer, Q. Sun, and D. Y. C. Chan. Non-singular boundary integral methods for fluid mechanics applications. *J. Fluid Mech.* 696:468–478, **2012**.
- [117] L. Dintenfass. Considerations of the internal viscosity of red cells and its effect on the viscosity of whole blood. *Angiology* 13:333–344, **1962**.
- [118] A. Kholeif and D. Weymann. Motion of a single red blood cell in plane shear flow. *Biorheology* 11:337–348, **1974**.
- [119] H. Schmid-Schönbein and E. Wells. Fluid drop like transition of erythrocytes under shear stress. *Science* 165:288–291, **1969**.
- [120] H. Goldsmith. Deformation of human red cells in tube flow. *Biorheology* 7:235–242, **1971**.
- [121] M. Sugihara and H. Niimi. Numerical approach to the motion of a red blood cell in couette flow. *Biorheology* 21:734–749, **1982**.
- [122] H. Niimi and M. Sugihara. Cyclic loading on the red cell membrane in a shear flow: a possible cause of hemolysis. *J. Biomech. Engng.* 107:91–95, **1985**.
- [123] I. Zahalak, P.R. Rao, and S.P. Sutera. Large deformations of a cylindrical liquid-filled membrane by a viscous shear flow. *J. Fluid Mech.* 179:283–305, **1987**.
- [124] S.R. Keller and R. Skalak. Motion of a tank-treading ellipsoidal particle in a shear flow. *J. Fluid Mech.* 120:27–47, **1982**.
- [125] S.P. Sutera, P.R. Pierre, and G.I. Zahalak. Deduction of intrinsic mechanical properties of the erythrocyte membrane from observations of tank-treading in the rheoscope. *Biorheology* 26:177–197, **1989**.
- [126] C. Pozrikidis. The axisymmetric deformation of a red blood cell in uniaxial straining stokes flow. *J. Fluid Mech.* 216:231–254, **1990**.
- [127] M.S. Longuet-Higgins and E.D. Cokelet. The deformation of steep surface waves on water i. a numerical method of computation. *Proc. R. Soc. Lond. A* 350:1–26, **1976**.
- [128] M. Kraus, W. Wintz, U. Seifert, and R. Lipowsky. Fluid vesicles in shear flow. *Phys. Rev. Lett.* 77:17, **1996**.
- [129] S. Sukumaran and U. Seifert. Influence of shear flow on vesicles near a wall: A numerical study. *Phys. Rev. E* 64:011916, **2001**.
- [130] A. Farutina, T. Bibenb, and C. Misbaha. 3d numerical simulations of vesicle and inextensible capsule dynamics. *J. Fluid Mech.* , **submitted**.

- [131] B. Kaoui, G. Biroso, and C. Misbah. Why do red blood cells have asymmetric shapes even in a symmetric flow? *Phys. Rev. Lett.* 103:188101, **2009**.
- [132] T. Seon T. Biben J. Beaucourt, F. Rioual and C. Misbah. Steady to unsteady dynamics of a vesicle in a flow. *Phys. Rev. E* 69:011906, **2004**.

Summary

Three-dimensional boundary integral modeling of viscous drops and capsules

Deformable objects such as drops, vesicles and red blood cells have received considerable research interest due to their industrial and biomedical applications. For example, the study of the deformation of drops is important to investigate the dynamics of polymer blends intended for fiber-, film- or bulk polymer production or water-in-water biopolymer mixtures. Capsules, vesicles and red blood cells (RBC) also play an essential role in biomedical engineering. To obtain a useful model for the motion of these objects, an accurate and powerful method to approximate the velocity field of the deformable objects is required. The boundary integral method (BIM), which relies on a mathematical reduction from the three-dimensional volume description to a two-dimensional surface representation, is one of the most accurate known methods to approximate this velocity field in the creeping flow regime and is used in this thesis.

Triangular elements are often taken as a method to discretize the surface of the above-mentioned objects. Different approaches have been developed to determine the mean curvature of each element when linear triangular meshes are used. Evidently, the mean curvature is not defined over the element boundaries and as a result many different methods (*e.g.* non-singular contour integration) have been introduced to minimize errors in the velocity calculation. Although these inaccuracies do not drastically influence the modeling of rather simple objects like drops, these errors are intensified for objects with more complex stress discontinuities at the interface such as vesicles and red blood cells. The physical limits where a triangular description can be applied have been investigated in this thesis. Although that for some material conditions results are obtained, the stability of the method was found to be insufficient to yield a versatile simulation tool for vesicles and red blood cells. Therefore an alternative method for the spatial approximation of the interface was sought.

The combination of isogeometric analysis with the boundary integral method

(IGA-BIM) is proposed in this thesis to overcome the mentioned difficulties in computing higher-order derivatives on surfaces. Isogeometric analysis defines the interface using a B-spline discretization and makes it possible to unambiguously evaluate normal vectors, mean curvatures and derivatives thereof. As a first application, the deformation of a viscous drop is considered and its behavior in a shear flow is investigated using the IGA-BIM. To achieve an efficient numerical evaluation of the convolution integrals that appear in the formulation, a singular subtraction technique is implemented to accurately integrate the singularities introduced by the Green's functions. The motion of the droplet is simulated by a stepwise computation of control point velocities using either an L^2 -projection or a collocation method. The numerical simulation of the drop deformation is performed for various (second-, third- and fourth-order) B-spline degrees. The method is shown to render objective results with respect to both the mesh size and the spline order. The importance of the smoothness of splines is more pronounced in the cases where higher-order gradients appear in the surface force term. Such higher-order terms can be found in *e.g.* models for the red blood cell.

Unlike the drop, the physics of a red blood cell is more complicated and does not allow its membrane surface area to change during the deformation, due to the existence of a lipid bilayer at the membrane. As a first approximation of a red blood cell, a drop with an inextensible membrane is considered. The deformation of this cell suspended in a shear flow is solved using the IGA-BIM that has been extensively verified for the deforming viscous drop. A staggered time integration scheme is introduced to enforce the surface area constraint. The resulting algorithm requires the evaluation of the force discontinuity at the membrane, which – in contrast to the formulation for the drop – involves the surface gradient of the tension in the membrane. Therefore, an appropriate formulation for finding the surface gradient of any scalar function using the IGA discretization is presented. The dependency of the equilibrium shape of the cell on the initial configuration and the inclination angle is studied. In addition, the correlation between the tension distribution and the mean curvature across the surface is investigated.

To obtain a more realistic simulation of a RBC, the bending force of the RBC membrane is required to be added to the model. The importance of IGA is in this case more pronounced since a higher-order gradient of the mean curvature is present in the bending force across the membrane. Although IGA allows for the unambiguous evaluation of higher-order gradients, interpolation inaccuracies in the vicinity of the poles were found to cause simulations to be unstable. To resolve these problems a hybrid geometry parametrization is developed which preserves the smoothness of the surface, but avoids the necessity to evaluate

higher-order gradients around the poles. To study the appropriateness of this hybrid geometry parametrization, it is applied for the simulation of a drop and an inextensible membrane and the possibilities of the hybrid method for application to the red blood cell case are discussed.

Samenvatting

Driedimensionale randintegraalmodellering van viskeuze druppels en capsules

Door hun industriële en biomedische toepassingen bestaat er een aanzienlijke onderzoeksinteresse in deformeerbare objecten zoals druppels, vesicles en rode bloedcellen. De studie van de vervorming van druppels is bijvoorbeeld belangrijk voor onderzoek naar de dynamica van polymeermengsels voor fiber-, film- of bulkpolymeerproductie, of water-in-water biopolymeer mengsels. Capsules, vesicles en rode bloedcellen (RBC) spelen een essentiële rol in de biomedische technologie. Om een bruikbaar model voor de beweging van deze objecten te verkrijgen is een nauwkeurige en krachtige approximatie van het snelheidsveld van de deformeerbare objecten vereist. De randintegraalmethode (BIM), welke is gebaseerd op een wiskundige reductie van de driedimensionale volumebeschrijving tot een tweedimensionale oppervlakterepresentatie, is één van de meest nauwkeurige bekende methoden voor de approximatie van dit snelheidsveld in het kruipstromingregime en wordt gebruikt in dit proefschrift.

Driehoekselementen worden vaak gebruikt om het oppervlak van de bovengenoemde objecten te discretiseren. Verschillende benaderingen om de gemiddelde kromming van ieder element te bepalen zijn ontwikkeld voor het geval dat gebruik wordt gemaakt van driehoeksroosters. Het is overduidelijk dat de gemiddelde kromming niet is gedefinieerd over de elementranden, waardoor een groot aantal verschillende methoden (bijv. niet-singuliere contourintegratie) is voorgesteld om de fouten in de snelheidsberekening te minimaliseren. Hoewel deze onnauwkeurigheden de modellering van betrekkelijk eenvoudige objecten zoals druppels niet dramatisch beïnvloeden, worden deze fouten versterkt voor objecten met meer complexe spanningsdiscontinuïteiten over de rand, zoals vesicles en rode bloedcellen. De fysische grenzen van de toepassing van een driehoeksbeschrijving zijn in dit proefschrift onderzocht. Hoewel resultaten zijn verkregen voor sommige materiaalcondities, is de stabiliteit van de methode niet toereikend bevonden om te resulteren in een breed toepasbaar simulatiemodel

voor vesicles en rode bloedcellen. Daarom is er gezocht naar een alternatieve methode voor de ruimtelijke approximatie.

In dit proefschrift is de combinatie van isogeometrische analyse met de rand-integraalmethode (IGA-BIM) voorgesteld om de genoemde problemen met betrekking tot de berekening van hogere-order afgeleiden over oppervlakten te verhelpen. Isogeometrische analyse definieert de rand door middel van een B-spline discretisatie en maakt het mogelijk om de normaalvectoren, gemiddelde kromming en afgeleiden daarvan eenduidig te bepalen. Als een eerste toepassing is de vervorming van een viskeuze druppel beschouwd en diens gedrag in een schuifstroming is bestudeerd door middel van IGA-BIM. Ter verkrijging van een efficiënte numerieke evaluatie van de convolutie-integralen in de formulering, is een singulariteit-subtractie techniek geïmplementeerd om de singulariteiten ten gevolge van de Green's functies nauwkeurig te integreren. De beweging van de druppel is gesimuleerd met een stapsgewijze berekening van de controlepunt snelheden, door gebruik te maken van een L^2 -projectie of een collocatiemethode. De numerieke simulatie van de druppelvervorming is uitgevoerd voor verschillend (tweede-, derde- en vierde-orde) B-spline ordes. Het is aangetoond dat de methode objectieve resultaten genereert ten aanzien van zowel de roostergrootte als de spline orde. Het belang van de gladheid van splines komt sterker naar voren in het geval waar hogere-orde gradiënten in de oppervlaktekracht aanwezig zijn. Zulke hogere-orde bijdragen komen bijvoorbeeld voor in modellen voor de rode bloedcel.

De fysica van een rode bloedcel is ingewikkelder dan die van de een druppel. In tegenstelling tot de druppel verandert het oppervlak niet gedurende de vervorming, door de aanwezigheid van een lipiden bilaag in het membraan. Als een eerste versimpeling van een rode bloedcel is een onuitrekbaar membraan beschouwd. De vervorming van deze cel in een schuifstroming wordt bepaald met de IGA-BIM, welke uitgebreid is geverifieerd voor de vervorming van druppels. Een stapsgewijs tijdsintegratieschema is geïntroduceerd om te voldoen aan de onuitrekbaarheidsrestricte. Het resulterende algoritme vereist de evaluatie van de krachtdiscontinuïteit over het membraan, welke – in tegenstelling tot de formulering voor de druppel – de oppervlaktegradiënt van de spanning in het membraan bevat. Daarom is er een formulering voorgesteld voor de bepaling van een oppervlaktegradiënt voor een willekeurige scalaire functie met de IGA discretisatie. De invloed van de beginconfiguratie en inclinatiehoek op de evenwichtsvorm van de cel is bestudeerd. Daarnaast is de correlatie tussen de spanningsverdeling en de gemiddelde kromming van de rand onderzocht.

Voor een realistischere simulatie van een RBC is het vereist om de buigkracht van het RBC membraan aan het model toe te voegen. Het belang van isogeometrische analyse komt in dit geval sterker naar voren, doordat hogere-

orde gradiënten van de gemiddelde kromming aanwezig zijn in the buigkracht over het membraan. Hoewel IGA de eenduidige evaluatie van hogere-orde gradiënten toestaat, blijken interpolatie onnauwkeurigheden in de nabijheid van de polen de simulaties instabiel te maken. Om deze problemen op te lossen is een hybride geometrieparametrisatie ontwikkeld, welke de gladheid van het oppervlak behoudt, maar de noodzaak om hogere-orde gradiënten te evalueren rondom de polen vermijdt. Om de geschiktheid van deze hybride geometrieparametrisatie te onderzoeken is deze toegepast voor de simulatie van een druppel en een onuitrekbaar membraan en de toepassingsmogelijkheden van de hybride methode voor de modellering van de rode bloedcel zijn besproken.

Acknowledgements

In the long way of passing the four years of my PhD career, several people helped me and this thesis would never have come together without their guidance and support. I would like to express my deep gratitude to all who made this work possible. First of all, I gratefully and sincerely thank prof.dr.ir. Patrick Anderson for all his support, patience, brilliant ideas, advice, recommendations and most importantly, his affability. Dear Patrick, it was an honor in my life to be your student, and to take advantages of improving my scientific skills under your supervision.

In addition, I would also like to dedicate my deep appreciation to dr.ir. Clemens Verhoosel who turned on a light in the darkness of my PhD challenging issues. Dear Clemens, I would never forget September 2011, when your son was born, my project got reborn. Although I should have learned a lot of new scientific stuff to continue this job using IGA, your tremendous help never let me down and motivated me to approach many interesting IGA-related observations. It was a pleasure to collaborate with you and learn a lot of scientific affairs as well as life affairs.

A week after I came into The Netherlands, I experienced my first gathering in "Henri-Chapelle" with the other researchers in the "blood in motion" project, namely, Patrick Anderson, Han Mijer, Frans van de Vosse, Federico Toschi, Arjen Bogaerds, Jens Harting, Florian Janoschek and René van der Burgt. We continued these meetings when Francesca Storti, Thomas van Kempen, Timm Krüger and Badr Kauoi joined us for several times afterward and helped us to have enjoyable and fruitful discussions. Also, during these four years, I always used the great scientific help of my colleagues Zahra (Afrooz), Nick, Yang, Daniel, Arash, Michiel, Iaroslav, Oleksandr, Ye Wang, Carina, Fabio and Anna.

Moreover, I would like to acknowledge specifically Marleen, Yvon and Leo. I saw Marleen at the first time when I arrived at Eindhoven train station. She picked me up and brought me to the home agency, and helped me to bring my suitcases to my home. This was just one of the favors she gave to me and I always used her kind support in these years. I wish to extend my great gratitude to Leo who

helped me whenever I faced a technical software and computer problem.

Also, my officemates in the university who always made a pleasant working environment, including Afrooz, Carina, Oleksandr, Misha, Arash and recently Monica deserve exceptional appreciation.

During this period of time, I enjoyed from being with a few specific groups. Special thanks to my "Dutch practicing" group including Elena, Francesca, Serkan, David, Lucia and also Lale, Ahmet, and Chen Ket who joined us recently. I never forget those enjoyable dinners mostly in the "Zwarte Doos", normally in every Tuesday when we were just allowed to speak in Dutch. Of course my sincere appreciation goes to our lovely Dutch teacher, Elly Arkesteijn, for her effective role in our Dutch knowledge.

My special gratitude to my team mates in "Eindhoven Students Indoor Soccer League" that we successfully won the title of champion in this season, including Hamed, Ali, Kaveh, Hamid Reza, Amir Reza, Adrian and previously, Amin and Saber.

In addition, I would like to express my gratitude to a few friends in particular since they played an important role in my life in Eindhoven to carry on better life here. My special thanks go to Reza who became my best friend here, Ebrahim (Arash) & Farimah, Afrooz & Mohammad, Mojtaba who provided me a lot of interesting books, Hamid Tavassoli, Amir Naderi, Amir Jalali Rad, Javad, Arash & Elnaz, Saman & Sanaz, Hamid & Azar, Samaneh & Saeed, Sara Ghazanfari, Sara Rahimi, Amin & Parisa, Hamid Reza & Marzyeh, Hamed & Negar, Ali & Niloofar, Kamyar & Pegah, Hossein & Samaneh, Faisal, Saeed Zandi, Saeed Masroor, Raheleh, Ameneh, Nilgoon & Hadi, Mahnaz, Timo, Gretjan, Hamed Mortazavi, Behrooz, Luca, Mona, Juan & Sol, Andrés & Lorenza, Mohammad & Flora, Salman & Leila, Hossein Eslami, Eleni, Hamid Hosseini, Chiara and Giulia.

Last but certainly not the least, I heartily acknowledge my lovely parents, Hossein Ali & Monireh. Though I was usually far from you in a lot of years of my educations in past ten years, you never let me feel lonely. That makes me indebted to you with all my deep feeling. Dear dad and mom, no words would be sufficient to express my gratitude and I am always grateful for your encouragement in entire of my life. I sincerely thank my brothers, Iman & Amir and my sister in law, Saemeh for all of your tremendous help and support.

امین احمدی جونیڈی

Amin Ahmadi Joneidi
Eindhoven, January 2014

Curriculum Vitae

Amin Ahmadi Joneidi was born on 4th of September 1985 in Ghaem Shahr, a city in a northern province Mazandaran, Iran. After finishing his high school and a pre-university education in 2003, he started his education in academia and finished his B.Sc. degree in Mechanical Engineering (Solid Mechanics) in four years at Mazandaran university. In 2007, he entered Shahrood University of Technology to continue his education and got his M.Sc. degree on the thesis entitled "*Investigation of viscoelastic flow and heat transfer in a curved pipe*" at the department of Mechanical Engineering (Energy Conversion) under the supervision of prof.dr. M.H. Kayhani and dr. M. Norouzi. In the beginning of 2010, he moved to Eindhoven, a southern city of the Netherlands, where he took a position as a Ph.D. student in Polymer Technology Group, chaired by prof.dr.ir. H.E.H. Meijer, at Eindhoven University of Technology. Two years after, his supervisor prof.dr.ir. P.D. Anderson established his research group named "Structure and Rheology of Complex Fluids" and subsequently Amin joined his group. His Ph.D. research, which concerns "*Three-dimensional boundary integral modeling of viscous drops and capsules*", has been supervised by prof.dr.ir. P.D. Anderson and dr.ir. C.V. Verhoosel.

